



Présentation de SQL Anywhere[®] Studio

Réf. du document : DC03918-01-0901-01
Dernière mise à jour : Janvier 2004

Copyright © 1989–2004 Sybase, Inc. Copyright partiel © 2001–2004 iAnywhere Solutions, Inc. Tous droits réservés.

Tout ou partie de cette publication ne peut être reproduit, transmis ou traduit, sous quelque forme ou par quelque moyen que ce soit (électronique, mécanique, manuel, optique ou autre) sans l'accord écrit préalable d'iAnywhere Solutions, Inc. iAnywhere Solutions, Inc est une filiale de Sybase, Inc.

Sybase, le logo SYBASE, AccelaTrade, ADA Workbench, Adaptable Windowing Environment, Adaptive Component Architecture, Adaptive Server, Adaptive Server Anywhere, Adaptive Server Enterprise, Adaptive Server Enterprise Monitor, Adaptive Server Enterprise Replication, Adaptive Server Everywhere, Adaptive Server IQ, Adaptive Warehouse, AnswerBase, Anywhere Studio, Application Manager, AppModeler, APT Workbench, APT-Build, APT-Edit, APT-Execute, APT-Library, APT-Translator, ASEP, AvantGo, AvantGo Application Alerts, AvantGo Mobile Delivery, AvantGo Mobile Document Viewer, AvantGo Mobile Inspection, AvantGo Mobile Marketing Channel, AvantGo Mobile Pharma, AvantGo Mobile Sales, AvantGo Pylon, AvantGo Pylon Application Server, AvantGo Pylon Conduit, AvantGo Pylon PIM Server, AvantGo Pylon Pro, Backup Server, BayCam, Bit-Wise, BizTracker, Certified PowerBuilder Developer, Certified SYBASE Professional, Certified SYBASE Professional (logo), ClearConnect, Client Services, Client-Library, CodeBank, Column Design, ComponentPack, Connection Manager, Convoy/DM, Copernicus, CSP, Data Pipeline, Data Workbench, DataArchitect, Database Analyzer, DataExpress, DataServer, DataWindow, DB-Library, dbQueue, Developers Workbench, Direct Connect Anywhere, DirectConnect, Distribution Director, Dynamic Mobility Model, Dynamo, e-ADK, E-Anywhere, e-Biz Integrator, E-Whatever, EC Gateway, ECMAP, ECRTP, eFulfillment Accelerator, Electronic Case Management, Embedded SQL, EMS, Enterprise Application Studio, Enterprise Client/Server, Enterprise Connect, Enterprise Data Studio, Enterprise Manager, Enterprise Portal (logo), Enterprise SQL Server Manager, Enterprise Work Architecture, Enterprise Work Designer, Enterprise Work Modeler, eProcurement Accelerator, eremote, Everything Works Better When Everything Works Together, EWA, Financial Fusion, Financial Fusion (and design), Financial Fusion Server, Formula One, Fusion Powered e-Finance, Fusion Powered Financial Destinations, Fusion Powered STP, Gateway Manager, GeoPoint, GlobalFIX, iAnywhere, iAnywhere Solutions, ImpactNow, Industry Warehouse Studio, InfoMaker, Information Anywhere, Information Everywhere, InformationConnect, InstaHelp, Intelligent Self-Care, InternetBuilder, iremote, iScript, Jaguar CTS, jConnect for JDBC, KnowledgeBase, Logical Memory Manager, M-Business Channel, M-Business Network, M-Business Server, Mail Anywhere Studio, MainframeConnect, Maintenance Express, Manage Anywhere Studio, MAP, MDI Access Server, MDI Database Gateway, media.splash, Message Anywhere Server, MetaWorks, MethodSet, ML Query, MobicATS, My AvantGo, My AvantGo Media Channel, My AvantGo Mobile Marketing, MySupport, Net-Gateway, Net-Library, New Era of Networks, Next Generation Learning, Next Generation Learning Studio, O DEVICE, OASiS, OASiS (logo), ObjectConnect, ObjectCycle, OmniConnect, OmniSQL Access Module, OmniSQL Toolkit, Open Biz, Open Business Interchange, Open Client, Open Client/Server, Open Client/Server Interfaces, Open ClientConnect, Open Gateway, Open Server, Open ServerConnect, Open Solutions, Optima++, Orchestration Studio, Partnerships that Work, PB-Gen, PC APT Execute, PC DB-Net, PC Net Library, PhysicalArchitect, Pocket PowerBuilder, PocketBuilder, Power Through Knowledge, Power++, power.stop, PowerAMC, PowerBuilder, PowerBuilder Foundation Class Library, PowerDesigner, PowerDimensions, PowerDynamo, Powering the New Economy, PowerJ, PowerScript, PowerSite, PowerSocket, Powersoft, Powersoft Portfolio, Powersoft Professional, PowerStage, PowerStudio, PowerTips, PowerWare Desktop, PowerWare Enterprise, ProcessAnalyst, QAnywhere, Rapport, Relational Beans, RepConnector, Replication Agent, Replication Driver, Replication Server, Replication Server Manager, Replication Toolkit, Report Workbench, Report-Execute, Resource Manager, RW-DisplayLib, RW-Library, S.W.I.F.T. Message Format Libraries, SAFE, SAFE/PRO, SDF, Secure SQL Server, Secure SQL Toolset, Security Guardian, SKILS, smart.partners, smart.parts, smart.script, SQL Advantage, SQL Anywhere, SQL Anywhere Studio, SQL Code Checker, SQL Debug, SQL Edit, SQL Edit/TPU, SQL Everywhere, SQL Modeler, SQL Remote, SQL Server, SQL Server Manager, SQL Server SNMP SubAgent, SQL Server/CFT, SQL Server/DBM, SQL SMART, SQL Station, SQL Toolset, SQLJ, Stage III Engineering, Startup.Com, STEP, SupportNow, Sybase Central, Sybase Client/Server Interfaces, Sybase Development Framework, Sybase Financial Server, Sybase Gateways, Sybase Learning Connection, Sybase MPP, Sybase SQL Desktop, Sybase SQL Lifecycle, Sybase SQL Workgroup, Sybase Synergy Program, Sybase User Workbench, Sybase Virtual Server Architecture, SybaseWare, Syber Financial, SyberAssist, SybMD, SyBooks, System 10, System 11, System XI (logo), SystemTools, Tabular Data Stream, The Enterprise Client/Server Company, The Extensible Software Platform, The Future Is Wide Open, The Learning Connection, The Model For Client/Server Solutions, The Online Information Center, The Power of One, TotalFix, TradeForce, Transact-SQL, Translation Toolkit, Turning Imagination Into Reality, UltraLite, UltraLite.NET, UNIBOM, Unilib, Uninull, Unisep, Unistring, URK Runtime Kit for UniCode, Versacore, Viewer, VisualWriter, VQL, Warehouse Control Center, Warehouse Studio, Warehouse WORKS, WarehouseArchitect, Watcom, Watcom SQL, Watcom SQL Server, Web Deployment Kit, Web.PB, Web.SQL, WebSights, WebViewer, WorkGroup SQL Server, XA-Library, XA-Server et XP Server sont des marques de Sybase, Inc. ou de ses filiales.

Tous les autres noms de produit, société ou marque appartiennent à leurs propriétaires respectifs.

Table des matières

Préface	ix
Documentation SQL Anywhere Studio	x
Accès des informations en ligne	xiii
Conventions	xvi
La base de données exemple Adaptive Server Anywhere . .	xviii
Base de données exemple CustDB	xix
Observations et sources d'informations complémentaires . .	xx
I Présentation de SQL Anywhere Studio	1
1 Introduction à SQL Anywhere Studio	3
Bienvenue dans SQL Anywhere Studio	4
Installation de SQL Anywhere Studio	6
2 Présentation d'Adaptive Server Anywhere et UltraLite	9
Présentation d'Adaptive Server Anywhere	10
Utilisations prévues	11
Spécificités	12
Serveur de base de données	14
Applications	15
Présentation d'UltraLite	16
Différences entre Adaptive Server Anywhere et UltraLite . . .	18
3 Présentation des technologies de réplication	21
Introduction	22
Bases de données consolidées et bases de données distantes	23
Méthodes de diffusion	26
Technologies de réplication Sybase	29
II Plates-formes supportées	35
4 Plates-formes supportées par SQL Anywhere Studio	37
Introduction	38
Systèmes d'exploitation Windows et NetWare	40
Systèmes d'exploitation UNIX, Linux et Macintosh	46
Bases de données consolidées pour synchronisation MobiLink	50
Pilotes ODBC iAnywhere Solutions - plates-formes supportées	51

Plates-formes cibles UltraLite	52
Versions des systèmes d'exploitation	54
III Utilisation des bases de données	59
5 Architecture des applications de base de données	61
Principes fondamentaux des bases de données relationnelles	62
Éléments d'un système de base de données	67
Interaction entre les différents éléments	70
Architecture informatique multiniveau	73
Utilisation de plusieurs bases de données	74
Interfaces de programmation d'application	76
Dans Adaptive Server Anywhere	81
6 Sélection de données dans des tables	85
SQL et base de données	86
Instruction SELECT	89
Sélection d'une table complète	90
Sélection de colonnes dans une table	92
Tri du résultat d'une requête	95
Sélection de lignes dans une table	99
7 Sélection de données dans plusieurs tables	107
Introduction	108
Jointure de tables par produit cartésien	109
Utilisation de la clause ON pour restreindre une jointure . . .	110
Jointure de tables à l'aide de jointures par clé	112
Jointure de tables à l'aide de jointures naturelles	114
Jointure de tables avec des jointures externes	116
8 Sélection de données à l'aide de sous-requêtes	117
Présentation des sous-requêtes	118
Sous-requêtes monoligne et multiligne	121
Sous-requêtes remplaçant des jointures	123
9 Sélection des données d'agrégat	125
Regroupement des données	126
Présentation des fonctions d'agrégat	127
Application des fonctions d'agrégat aux données groupées .	128
Restriction de groupes	130
10 Mise à jour de la base de données	133
Introduction	134

Ajout de lignes à une table	135
Modification de lignes dans une table	136
Suppression de lignes	137
Regroupement des modifications dans des transactions . . .	138
Contrôle d'intégrité	141
IV Didacticiels de SQL Anywhere Studio	145
11 Base de données exemple	147
A propos de la base de données exemple	148
Leçon 1 : Copie de la base de données exemple	150
Leçon 2 : Démarrage du serveur de base de données Adaptive Server Anywhere	151
Leçon 3 : Affichage de la fenêtre du serveur de base de données	153
Leçon 4 : Fermeture de la base de données	155
Résumé	156
12 Etablissement de la connexion	157
Connexions	158
Sources de données ODBC	159
Leçon 1 : Création d'une source de données ODBC	160
Connexion à l'aide de la source de données ODBC exemple	163
Autres méthodes de connexion	164
Déconnexion de la base de données exemple	165
Résumé	166
13 Interactive SQL	167
A propos d'Interactive SQL	168
Leçon 1 : Démarrage d'Interactive SQL	169
Leçon 2 : Interface Interactive SQL	170
Leçon 3 : Affichage de données avec Interactive SQL	176
Leçon 4 : Utilisation des instructions SQL	183
Résumé	191
14 Gestion des bases de données avec Sybase Central	193
Sybase Central	194
Leçon 1 : Démarrage de Sybase Central et connexion	195
Leçon 2 : Interface Sybase Central	196
Leçon 3 : Affichage de la base de données exemple	198
Leçon 4 : Création et modification de tables	203
Leçon 5 : Gestion des utilisateurs et des groupes	205
Leçon 6 : Visualisation et modification des procédures stockées	207

Leçon 7 : Sauvegarde d'une base de données	210
Restauration de la base de données exemple	211
Résumé	212
15 Conception et création d'une base de données	213
Conception d'une base de données	214
Conception et création d'une base de données simple	215
Leçon 1 : création d'un fichier de base de données	216
Leçon 2 : connexion à la base de données	218
Leçon 3 : conception et création d'une table	220
Leçon 4 :	222
Leçon 5 : définition des propriétés de colonne	224
Leçon 6 : conception et création des relations entre les tables	227
Résumé	230
16 Synchronisation de bases de données avec MobiLink	231
Présentation de MobiLink	232
Introduction	233
Leçon 1 : Création de bases de données	235
Leçon 2 : Préparation des bases de données pour la syn- chronisation	239
Leçon 3 : Démarrage du serveur de synchronisation MobiLink	242
Leçon 4 : Exécution de l'utilitaire client de synchronisation MobiLink	243
Résumé	245
17 Réplication de données avec SQL Remote	247
Présentation de SQL Remote	248
Leçon 1 : Mise en route	249
Leçon 2 : Configuration de la base de données consolidée	251
Leçon 3 : Configuration de la base de données distante	255
Leçon 4 : Réplication des données	258
Leçon 5 : Restitution de la base de données et de ses para- mètres	261
Résumé	263
18 Conception de bases de données avec PowerAMC	265
Présentation de PowerAMC	266
Leçon 1 : Mise en route	268
Leçon 2 : Ajout d'une colonne	273
Leçon 3 : Vérification de votre travail	275
Leçon 4 : Enregistrement des modifications et génération de la base de données	276
Résumé	279

19 Génération d'états avec InfoMaker	281
Présentation d'InfoMaker	282
Leçon 1 : Mise en route	283
Leçon 2 : Génération d'un état élémentaire	284
Leçon 3 : Amélioration de l'état	287
Résumé	291
V Annexe	293
20 Glossaire	295
Index	315



Préface

Présentation

Ce manuel présente Sybase SQL Anywhere Studio, système complet de gestion et de synchronisation de base de données relationnelle pour l'informatique mobile, embarquée et de groupe de travail.

SQL Anywhere Studio est un ensemble de composants logiciels permettant de travailler avec des bases de données relationnelles. Il inclut le système de gestion de base de données relationnel Adaptive Server Anywhere et UltraLite, ainsi que la technologie de réplication et de synchronisation des données. Il comporte également des applications pour la conception et le déploiement des bases de données, et pour la création d'états personnalisés et de formulaires de saisie de données.

A qui s'adresse ce manuel ?

Ce manuel est destiné à tous les développeurs d'application et aux administrateurs de base de données qui utilisent SQL Anywhere Studio.

Avant de commencer

Une connaissance élémentaire des bases de données relationnelles et de SQL est requise pour lire ce manuel.

Documentation SQL Anywhere Studio

Ce manuel fait partie de la documentation SQL Anywhere Studio. Cette section décrit les différents manuels de la documentation et explique comment les utiliser.

Documentation
SQL Anywhere Studio

La documentation SQL Anywhere Studio est disponible sous plusieurs formes : format en ligne regroupant tous les manuels dans un fichier d'aide unique, fichier PDF (un par manuel) et manuels imprimés disponibles à l'achat. La documentation est constituée des manuels suivants :

- ◆ **Introduction à SQL Anywhere Studio** Ce manuel présente les technologies de gestion de base de données et de synchronisation offertes par SQL Anywhere Studio. Il inclut des didacticiels destinés à vous initier aux différentes parties qui composent SQL Anywhere Studio.
- ◆ **Nouvelles fonctionnalités de SQL Anywhere Studio** Ce manuel s'adresse aux utilisateurs des versions antérieures du logiciel. Il répertorie les nouvelles fonctionnalités de cette version et des versions précédentes du produit et décrit les procédures de mise à niveau.
- ◆ **Base de données Adaptive Server Anywhere - Guide d'administration** Ce manuel traite du lancement, de la gestion et de la configuration des bases de données et des serveurs de base de données.
- ◆ **Adaptive Server Anywhere - Guide de l'utilisateur SQL** Ce manuel explique comment concevoir et créer des bases de données, comment importer, exporter et modifier des données, comment récupérer des données et construire des procédures stockées et des triggers.
- ◆ **Adaptive Server Anywhere - Manuel de référence SQL** Ce manuel de référence fournit une description complète du langage SQL utilisé par Adaptive Server Anywhere. Il présente également les tables et procédures système d'Adaptive Server Anywhere.
- ◆ **Adaptive Server Anywhere - Guide de programmation** Ce manuel explique comment construire et déployer les applications de base de données à l'aide des langages de programmation C, C++ et Java. Les utilisateurs de Visual Basic et PowerBuilder peuvent utiliser les interfaces de programmation fournies par ces outils. Ce manuel décrit également le fournisseur de données ADO.NET pour Adaptive Server Anywhere.
- ◆ **Adaptive Server Anywhere - Messages d'erreur** Ce manuel fournit la liste complète des messages d'erreur d'Adaptive Server Anywhere, ainsi que des informations de diagnostic.

-
- ◆ **SQL Anywhere Studio - Guide de sécurité** Ce manuel fournit des informations sur les fonctionnalités de sécurité incluses dans les bases de données Adaptive Server Anywhere. Dans le cadre des critères d'évaluation de la sécurité des systèmes informatiques (TCSEC, Trusted Computer System Evaluation Criteria), le gouvernement américain a décerné la classe C2 à Adaptive Server Anywhere 7.0. Ce manuel s'adresse plus particulièrement à ceux qui souhaitent utiliser la version actuelle d'Adaptive Server Anywhere conformément aux exigences de l'environnement certifié C2.
 - ◆ **Synchronisation MobiLink - Guide de l'utilisateur** Ce manuel décrit l'utilisation du système de synchronisation des données MobiLink pour l'informatique mobile, qui permet le partage des données entre une base Oracle, Sybase, Microsoft ou IBM unique et plusieurs bases de données Adaptive Server Anywhere ou UltraLite.
 - ◆ **Synchronisation MobiLink - Manuel de référence** Ce manuel présente les options MobiLink, les scripts de synchronisation, les instructions SQL, les procédures stockées, les utilitaires, les tables système et les messages d'erreur.
 - ◆ **Synchronisation MobiLink exécutée par un serveur - Guide de l'utilisateur** Ce manuel présente la synchronisation MobiLink exécutée par un serveur, une fonctionnalité MobiLink qui permet d'exécuter la synchronisation à partir d'une base de données consolidée.
 - ◆ **QAnywhere - Guide de l'utilisateur (en anglais)** Ce manuel décrit la plate-forme de messagerie MobiLink QAnywhere qui permet le développement et le déploiement d'applications de messagerie pour les clients mobiles et sans fil, et pour les postes de travail traditionnels et ordinateurs portables.
 - ◆ **iAnywhere Solutions - Pilotes ODBC** Ce manuel décrit comment configurer les pilotes ODBC pour accéder aux bases de données consolidées autres qu'Adaptive Server Anywhere à partir du serveur de synchronisation MobiLink et de l'accès aux données distantes Adaptive Server Anywhere.
 - ◆ **SQL Remote - Guide de l'utilisateur** Ce manuel décrit tous les aspects du système de répllication des données SQL Remote pour l'informatique mobile. Ce système permet le partage des données entre une seule base Adaptive Server Anywhere ou Adaptive Server Enterprise et plusieurs bases de données Adaptive Server Anywhere via un lien indirect tel que la messagerie électronique ou le transfert de fichiers.

-
- ◆ **SQL Anywhere Studio - Aide** Ce manuel inclut l'aide contextuelle de Sybase Central, d'Interactive SQL et d'autres outils graphiques. Il n'est pas fourni avec la documentation imprimée.
 - ◆ **Base de données UltraLite - Guide de l'utilisateur** Ce manuel s'adresse à tous les développeurs UltraLite. Il présente le système de base de données UltraLite et fournit des informations communes à toutes les interfaces de programmation UltraLite.
 - ◆ **Guides des interfaces UltraLite** Un manuel est fourni pour chacune des interfaces de programmation UltraLite. Certaines de ces interfaces sont fournies en tant que composants UltraLite pour le développement d'applications rapide, et d'autres sous forme d'interfaces statiques pour le développement en C, C++ et Java.

PowerAMC et InfoMaker disposent également de leur propre documentation en ligne.

Formats de la documentation

La documentation de SQL Anywhere Studio se présente sous les formats suivants :

- ◆ **Documentation en ligne** La documentation en ligne comprend toute la documentation de SQL Anywhere Studio, y compris les manuels et l'aide contextuelle des outils SQL Anywhere. Elle est mise à jour à chaque version de maintenance du produit et constitue la source de documentation la plus complète et la plus récente.

Pour accéder à la documentation en ligne sous les systèmes d'exploitation Windows, choisissez Démarrer ► Programmes ► SQL Anywhere 9 ► Documentation en ligne. Vous pouvez explorer la documentation en ligne à partir du sommaire, de l'index et à l'aide de la fonction de recherche de l'Aide HTML dans le volet gauche ou au moyen des liens et des menus dans le volet droit.

Pour accéder à la documentation en ligne sous les systèmes d'exploitation UNIX, reportez-vous à la documentation HTML de votre installation SQL Anywhere.

- ◆ **Documentation au format PDF** Les manuels SQL Anywhere sont fournis au format PDF et se lisent avec Adobe Acrobat Reader.

La documentation au format PDF est accessible à partir des manuels en ligne ou depuis le menu Démarrer de Windows.

- ◆ **Manuels imprimés** L'ensemble des manuels est disponible auprès de Sybase ou à la boutique en ligne de Sybase, l'eShop. Pour y accéder, cliquez sur le lien How to Buy ► eShop à l'adresse suivante : <http://www.iAnywhere.com>.

Accès des informations en ligne

La documentation en ligne SQL Anywhere Studio comprend un certain nombre d'outils facilitant la navigation et la recherche d'informations.

Documentation en ligne (PDF)

La documentation en ligne (PDF) est la copie exacte de la documentation imprimée au format PDF, consultable avec Adobe Acrobat. La présence et l'apparence de certaines fonctionnalités parmi celles qui sont citées ci-après dépendent de la version d'Acrobat ou d'Acrobat Reader utilisée.

- ◆ **Onglet Signets** L'onglet Signets affiche la liste de tous les chapitres composant le manuel. Vous pouvez cliquer sur un chapitre ou une page pour l'afficher, développer ou réduire les en-têtes et visualiser la structure du document. Si vous cliquez sur un en-tête, la page correspondante s'affiche dans la partie droite de la fenêtre. L'onglet Signets permet d'obtenir une vue globale de la structure du manuel et d'accéder rapidement à des chapitres ou à des pages spécifiques.
- ◆ **Liens actifs** A l'instar des renvois "reportez-vous à", les liens actifs apparaissent en bleu. Lorsque vous déplacez votre curseur sur le lien, la main se transforme en doigt pointeur. Si vous cliquez sur un lien, vous êtes renvoyé à la rubrique correspondante, soit dans le manuel actif, soit dans d'autres manuels. Les liens entre manuels ne fonctionnent pas correctement si Acrobat est exécuté en tant que module externe de certains navigateurs. Ils fonctionnent bien depuis Internet Explorer ou directement depuis Acrobat.
- ◆ **Boutons Page suivante/Page précédente** Les boutons Page suivante et Page précédente situés en haut à gauche de la fenêtre d'aide vous permettent de retracer les différentes étapes et de revoir les pages que vous avez consultées dans le même ordre. Chaque fois que vous cliquez sur le bouton Page précédente, c'est la page que vous avez consultée avant la rubrique affichée à l'écran qui apparaît. Il faut avoir cliqué au moins une fois sur le bouton Page précédente pour que le bouton Page suivante devienne actif et le reste jusqu'à ce que vous ayez accédé à la page consultée en dernier.
- ◆ **Boutons Première page/Dernière page** Si vous cliquez sur le bouton Première page, la première page du manuel s'affiche tandis que la dernière page apparaît si vous cliquez sur le bouton Dernière page.

Documentation en ligne

La documentation en ligne est la copie exacte de la documentation imprimée au format de l'Aide HTML Microsoft. La présence et l'apparence de certaines fonctionnalités parmi celles qui sont citées ci-après dépendent de la version d'Acrobat ou d'Acrobat Reader utilisée.

-
- ◆ **Onglet Sommaire** L'onglet Sommaire affiche la liste de tous les manuels qui composent la documentation SQL Anywhere Studio. Pour afficher les différents chapitres des manuels, double-cliquez sur ces derniers, pour afficher les pages d'un chapitre, cliquez dessus. L'onglet Sommaire permet d'obtenir une vue globale de la structure de la documentation et d'accéder rapidement à des chapitres ou à des pages spécifiques.
 - ◆ **En-tête de page** L'en-tête de page est situé en haut de chaque page et décrit l'emplacement de la page en cours. Cet emplacement est décrit comme suit : titre du manuel ► titre du chapitre ► titre de la page. Vous pouvez cliquer sur les titres de manuel ou de chapitre pour y accéder. A mesure que vous parcourez les pages dans la partie droite de la fenêtre, l'indicateur de position dans l'onglet Sommaire situé à gauche se déplace également.
 - ◆ **Onglet Index** L'index contient des mots-clés associés aux chapitres et aux pages. Vous pouvez saisir un mot ou une expression dans le champ supérieur de l'onglet Index pour que la rubrique correspondante s'affiche dans le volet de navigation inférieur. Lorsque vous avez trouvé l'entrée souhaitée, il suffit de double-cliquer sur cette dernière pour afficher la page correspondante.
 - ◆ **Onglet Rechercher** Saisissez un mot ou une expression dans le champ supérieur de l'onglet Rechercher, puis cliquez sur Liste des rubriques pour afficher toutes les rubriques contenant ce mot ou cette expression. Si vous souhaitez retrouver un terme/une expression parfaitement identique, placez le/la entre guillemets (""). Pour affiner votre recherche, cochez les cases situées en bas de page ou bien servez-vous des termes de recherche booléens en cliquant sur la flèche en regard de la zone de recherche. L'option Rechercher les résultats précédents permet de limiter la recherche aux rubriques affichées lors d'une précédente recherche. L'option Rechercher les mots similaires permet de rechercher des mots dont l'orthographe est proche de ceux que vous saisissez. L'option Rechercher uniquement dans les titres limite la recherche aux titres. Lorsque vous avez trouvé l'entrée souhaitée, il suffit de double-cliquer sur cette dernière pour afficher la page correspondante.
 - ◆ **Onglet Favoris** Lorsque vous trouvez une page à laquelle vous serez susceptible d'accéder à nouveau, ajoutez-là à votre liste de favoris pour y accéder rapidement. Affichez la page que vous souhaitez enregistrer, cliquez sur l'onglet Favoris, puis sur le bouton Ajouter en bas de la fenêtre. Le fait de cliquer sur Ajouter ajoute la rubrique répertoriée dans la zone Rubrique en cours. Vous pouvez modifier votre liste de favoris en sélectionnant une rubrique puis en cliquant sur Supprimer. Les entrées

des Favoris sont préservées lors des mises à jour de la documentation en ligne.

- ◆ **Boutons Précédent/Suivant** Les boutons Précédent et Suivant situés dans la partie supérieure droite de la fenêtre d'aide s'apparente à l'action de tourner les pages d'un livre. Précédent permet de passer à la rubrique précédente (dans l'ordre chronologique). Suivant permet de passer à la rubrique suivante (dans l'ordre chronologique).
- ◆ **Boutons Page suivante/Page précédente** Les boutons Page suivante et Page précédente situés sur la barre de menus vous permettent de retracer les différentes étapes et de revoir les pages que vous avez consultées dans le même ordre. Chaque fois que vous cliquez sur le bouton Page précédente, c'est la page que vous avez consultée avant la rubrique affichée à l'écran qui apparaît. Il faut avoir cliqué au moins une fois sur le bouton Page précédente pour que le bouton Page suivante devienne actif et le reste jusqu'à ce que vous ayez accédé à la page consultée en dernier.
- ◆ **Bouton Liens Web** Le bouton Liens Web situés dans la partie supérieure de la fenêtre d'aide offre un accès rapide aux ressources iAnywhere Solutions, notamment le site Web d'iAnywhere, le support technique, l'enregistrement, les téléchargements et la communauté de développeurs iAnywhere Solutions.
- ◆ **Bouton Parcourir** Cliquez sur le bouton Parcourir pour accéder rapidement aux didacticiels, à la liste des actions que vous pouvez effectuer et au glossaire.
- ◆ **Bouton Réduire** Certaines pages contiennent des en-têtes qui peuvent être réduits, comme l'indique les signes plus (+) ou moins (-) placés à gauche en regard de l'en-tête. Vous pouvez cliquer sur le bouton Réduire pour fermer tous les en-têtes ou sur le bouton Développer pour les ouvrir.
- ◆ **Bouton PDF** Cliquez sur le bouton PDF pour ouvrir le manuel en cours au format PDF.

Conventions

Cette section décrit les conventions syntaxiques, typographiques et graphiques utilisées dans ce manuel.

Conventions syntaxiques Les exemples de syntaxe SQL s'appuient sur les conventions suivantes :

- ◆ **Mots-clés** Tous les mots-clés SQL sont indiqués en majuscules. Par exemple :

```
ALTER TABLE [ propriétaire.]nom_table
```

- ◆ **Marques de réservation** Les éléments à remplacer par des expressions ou des identificateurs appropriés apparaissent en *italique*. Par exemple :

```
ALTER TABLE [ propriétaire.]nom_table
```

- ◆ **Éléments répétitifs** Dans une liste contenant des éléments qui se répètent, le nom d'un élément de la liste est suivi de points de suspension (...), comme *contrainte_colonne* dans l'exemple ci-dessous :

```
ADD définition_colonne [ contrainte_colonne, ... ]
```

Une liste peut comprendre un ou plusieurs éléments. Dans ce dernier cas, les éléments sont séparés les uns des autres par une virgule.

- ◆ **Éléments facultatifs** Les éléments facultatifs d'une instruction sont placés entre crochets.

```
RELEASE SAVEPOINT [ nom_pointdesauvegarde]
```

Les crochets indiquent que le paramètre *nom_point_de_sauvegarde* est facultatif. Les crochets ne doivent pas être saisis.

- ◆ **Options** Lorsqu'un seul élément de la liste (ou aucun) doit être déclaré, les éléments sont séparés par des barres verticales et la liste est placée entre crochets.

```
[ ASC | DESC ]
```

Par exemple, vous pouvez choisir le paramètre ASC ou DESC, ou aucun des deux. Les crochets ne doivent pas être saisis.

- ◆ **Options obligatoires** Lorsque vous devez obligatoirement sélectionner l'une des options, les choix possibles sont séparés par une barre et placés entre accolades.

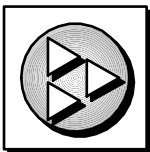
```
[ QUOTES { ON | OFF } ]
```

Si l'option QUOTES est utilisée, ON ou OFF doit également être spécifié. Les crochets et les accolades ne doivent pas être saisis.

Icônes

Les icônes suivantes sont utilisées dans cette documentation :

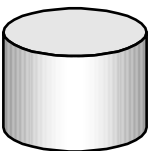
- ◆ Application cliente



- ◆ Serveur de base de données, par exemple Sybase Adaptive Server Anywhere



- ◆ Base de données. Dans certains diagrammes élaborés, l'icône peut servir à représenter à la fois la base de données et le serveur qui la gère.



- ◆ Middleware de réplication ou de synchronisation. Ils participent au partage de données entre les bases. Exemples : serveur de synchronisation MobiLink, agent de message SQL Remote.



- ◆ Interface de programmation



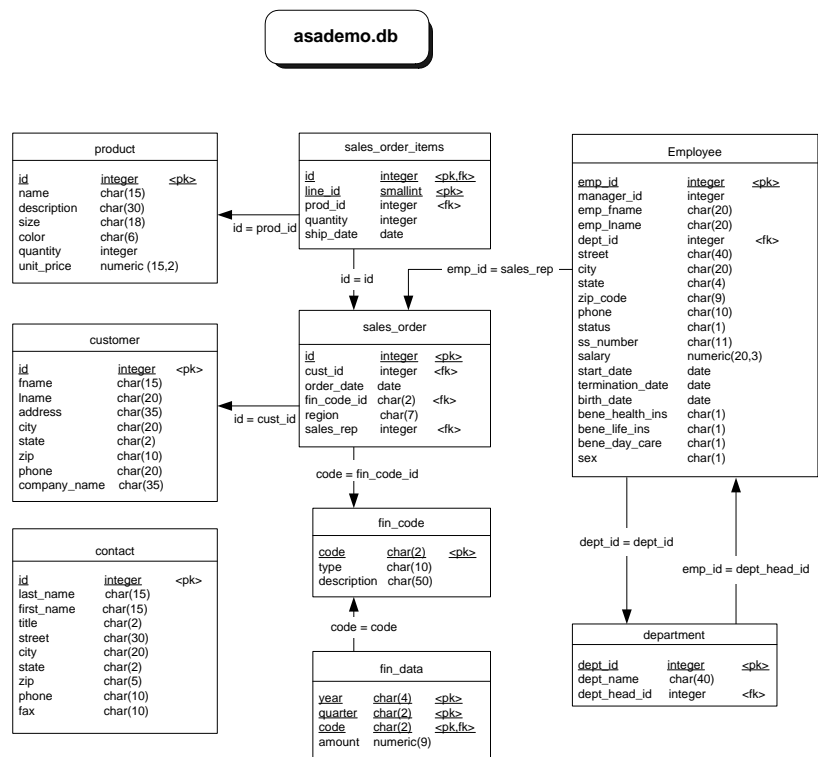
La base de données exemple Adaptive Server Anywhere

De nombreux exemples de la documentation font appel à la base de données exemple Adaptive Server Anywhere.

La base de données exemple est stockée dans un fichier appelé *asademo.db* situé dans votre répertoire SQL Anywhere.

Cette base de données représente une petite société. Elle contient des informations concernant la société (employés, services et finances), les produits et les ventes (commandes, clients et contacts). Toutes les informations contenues dans la base de données relèvent de la pure fiction.

Le schéma suivant illustre les tables de la base de données exemple et leurs relations.



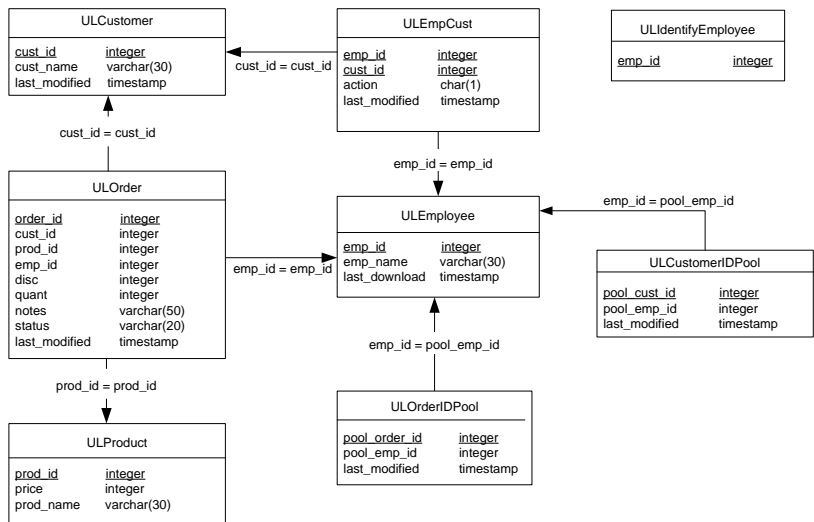
Base de données exemple CustDB

De nombreux exemples cités dans la documentation MobiLink et UltraLite font appel à la base de données exemple UltraLite.

La base de données de référence UltraLite est stockée dans un fichier appelé *custdb.db* et situé dans le sous-répertoire *Samples\UltraLite\CustDB* de votre répertoire SQL Anywhere. Vous trouverez également une application complète créée à partir de cette base de données.

Cette application est un exemple de base de données de gestion des ventes pour un fournisseur de matériel informatique. Elle contient des informations relatives aux clients, aux produits et à l'équipe commerciale du fournisseur.

Le schéma suivant illustre les tables de la base de données CustDB et la manière dont elles sont reliées entre elles.



Observations et sources d'informations complémentaires

Si vous avez des observations sur le contenu de cette documentation et sur le logiciel, nous vous serions reconnaissants de nous en faire part.

Vous pouvez nous les adresser par l'intermédiaire des forums (newsgroups) mis en place pour discuter des technologies de SQL Anywhere. Ces forums sont accessibles sur le serveur *forums.sybase.com*.

Les forums se déclinent comme suit :

- ◆ [sybase.public.sqlanywhere.general](#).
- ◆ [sybase.public.sqlanywhere.linux](#).
- ◆ [sybase.public.sqlanywhere.mobilink](#).
- ◆ [sybase.public.sqlanywhere.product_futures_discussion](#).
- ◆ [sybase.public.sqlanywhere.replication](#).
- ◆ [sybase.public.sqlanywhere.ultralite](#).

Débit de responsabilité pour les forums

La société iAnywhere Solutions n'est nullement tenue de fournir des solutions, des informations ou des avis concernant ses forums. Elle n'est pas dans l'obligation de fournir un service supplémentaire autre que celui de l'opérateur système prévu pour contrôler le service ni de garantir son fonctionnement ainsi que sa disponibilité.

Les conseillers techniques d'iAnywhere Solutions, ainsi que d'autres catégories de personnel participent à l'exploitation du forum lorsque leurs disponibilités le leur permettent. Ils apportent leur aide de manière bénévole et ne peuvent pas être disponibles régulièrement pour fournir des solutions ou des informations. Leur aptitude à fournir une aide dépend de leur charge de travail.

Vous pouvez envoyer vos commentaires et suggestions par email à l'équipe chargée de la documentation SQL Anywhere à l'adresse iasdoc@iAnywhere.com. Même s'il n'est pas prévu que nous répondions aux emails envoyés à cette adresse, soyez assurés que vos suggestions seront lues avec grand intérêt.

PREMIÈRE PARTIE

PRÉSENTATION DE SQL ANYWHERE STUDIO

Cette partie présente SQL Anywhere Studio et les technologies de gestion et de réplication de base de données.

CHAPITRE 1

Introduction à SQL Anywhere Studio

Présentation

Ce chapitre présente les éléments constitutifs de SQL Anywhere Studio.

Sommaire

Sujet :	page
Bienvenue dans SQL Anywhere Studio	4
Installation de SQL Anywhere Studio	6

Bienvenue dans SQL Anywhere Studio

Avec SQL Anywhere Studio, vous pouvez fournir des informations aux systèmes de bases de données mobiles, embarqués et de groupe de travail dans l'ensemble de votre organisation.

SQL Anywhere Studio inclut les composants décrits ci-après.

Système de gestion de base de données relationnelle

◆ **Adaptive Server Anywhere** La base de données relationnelle au coeur du produit correspond à une base de données SQL transactionnelle, conçue pour une utilisation personnelle et en groupe de travail. Adaptive Server Anywhere fonctionne sur un large éventail de systèmes d'exploitation, notamment diverses versions de Windows et UNIX, ainsi que sous Novell NetWare. Il s'exécute sur du matériel allant de serveurs de groupe de travail à plusieurs unités centrales jusqu'aux PC les plus modestes, ainsi que sur des périphériques Windows CE.

◆ **UltraLite** Base de données destinée à la création et au déploiement d'applications de base de données relationnelle sur de petits périphériques, tels que Palm Computing Platform et Windows CE. UltraLite comporte un support intégré pour la synchronisation MobiLink. UltraLite convient particulièrement aux petits périphériques car il permet de créer des bases de données relationnelles avec moins de 50 Ko d'espace disque.

Techniques de synchronisation des données

◆ **MobiLink** Permet la synchronisation bidirectionnelle des données entre une base de données centrale et plusieurs bases de données distantes UltraLite ou Adaptive Server Anywhere. La base de données centrale peut être une base Adaptive Server Anywhere, Adaptive Server Enterprise ou toute autre SGBD du type Oracle, Microsoft SQL Server ou IBM DB2.

◆ **SQL Remote** Permet la répllication bidirectionnelle de données, fondée sur les messages, entre une base de données centrale et plusieurs bases de données distantes. SQL Remote permet de répliquer des données entre des ordinateurs portables et une base de données centrale, à l'aide de la messagerie électronique ou de l'accès téléphonique.

◆ **Agent de répllication** Permet de répliquer des données à partir de bases de données Adaptive Server Anywhere vers d'autres bases de données, via le Replication Server de Sybase.

Outils de développement, de conception et d'administration

◆ **InfoMaker** Permet d'interroger des bases de données et de créer des états personnalisés plus élaborés et percutants. InfoMaker est également un assistant de données personnel qui vous offre diverses méthodes de traitement des données.

- ◆ **PowerAMC** Permet de concevoir, générer, documenter et gérer des bases de données.
- ◆ **Outils de gestion et de développement** SQL Anywhere Studio inclut l'outil de gestion de base de données Sybase Central, l'utilitaire de base de données Interactive SQL, ainsi que l'éditeur de requête et le débogueur de procédures stockées.

SQL Anywhere Studio comprend en outre un module d'activation de l'accessibilité dont l'installation est facultative. Ce composant fournit le module Sun AccessBridge qui est chargé chaque fois que vous utilisez Sybase Central ou Interactive SQL. Des logiciels tiers, tels que des logiciels de lecture d'écran, exploitent ce module pour permettre l'accès aux fonctionnalités du logiciel.

InfoMaker et PowerAMC sont uniquement disponibles pour les systèmes d'exploitation Windows. Il est toutefois possible de les utiliser comme clients d'un serveur s'exécutant sur n'importe quel système d'exploitation pris en charge.

Composants sous
licence séparée

Les composants suivants sont soumis à une licence indépendante et doivent être commandés pour que vous puissiez les installer. Pour commander ces composants, reportez-vous à la carte incluse dans le coffret SQL Anywhere Studio ou consultez notre site Web à l'adresse <http://www.sybase.com/detail?id=1015780>.

- ◆ **Option Java** La machine virtuelle Java et les classes d'exécution qui permettent d'utiliser Java dans la base de données doivent être commandées séparément.
- ◆ **Option de sécurité** Le logiciel de cryptage de données (sécurité de la couche de transport) entre un serveur Adaptive Server Anywhere ou MobiLink et un client doit être commandé séparément. Le cryptage de fichier de base de données AES est inclus dans le logiciel de base.

Installation de SQL Anywhere Studio

Le mode d'installation de SQL Anywhere Studio dépend du système d'exploitation utilisé. Vous devez vous assurer que vous possédez les licences appropriées avant d'installer le logiciel.

❖ Pour installer SQL Anywhere Studio (systèmes d'exploitation Windows)

1. Démarrez l'installation en exécutant le programme *setup.exe* dans le répertoire racine du CD-ROM de SQL Anywhere Studio. Suivez les instructions de l'assistant.

Le programme d'installation vous invite à choisir les composants à installer.

❖ Pour installer SQL Anywhere Studio (Novell NetWare ou Windows CE)

1. Vous devez installer Adaptive Server Anywhere for NetWare à partir d'une machine connectée au serveur NetWare et fonctionnant sous un système d'exploitation Microsoft Windows. Exécutez le programme d'installation SQL Anywhere Studio et choisissez l'installation NetWare ou Windows CE.

❖ Pour installer SQL Anywhere Studio (UNIX)

1. Les instructions d'installation dépendent du système UNIX utilisé. Pour plus d'informations, reportez-vous au document *Read Me First* fourni avec SQL Anywhere Studio pour UNIX.

Groupe de programmes SQL Anywhere

Avec un système d'exploitation Windows, l'installation du logiciel entraîne la création d'un **groupe de programmes** SQL Anywhere. Pour y accéder, cliquez sur le bouton Démarrer et sélectionnez Programmes ► SQL Anywhere 9.

L'installation de SQL Anywhere Studio sous UNIX ne crée pas de groupe de programmes.

Éléments du groupe de programmes

Le groupe de programmes contient tout ou partie des éléments ci-dessous. Les éléments affichés dépendent des choix effectués à l'installation du logiciel.

- ◆ **Adaptive Server Anywhere** Contient les éléments suivants :
 - ◆ **Interactive SQL** Démarre l'utilitaire Interactive SQL pour l'envoi d'instructions SQL à une base de données.

- ◆ **Exemple de serveur réseau** Démarre le serveur réseau et charge la base de données exemple.
- ◆ **Administrateur ODBC** Démarre ce programme pour la configuration et l'édition de sources de données ODBC.
- ◆ **Exemple de serveur personnel** Démarre le serveur personnel et charge la base de données exemple.
- ◆ **Groupe de programmes MobiLink** Permet d'accéder aux programmes et aux exemples de la synchronisation MobiLink.
- ◆ **Groupe de programmes UltraLite** Permet d'accéder aux programmes et aux exemples UltraLite.
- ◆ **Mises à jour disponibles** Permet d'accéder à une page Web contenant des informations sur les dernières mises à jour de votre version de SQL Anywhere Studio.
- ◆ **Ressources Internet iAnywhere** Ouvre une page Web contenant des informations sur iAnywhere Solutions.
- ◆ **Documentation en ligne** Ouvre la documentation en ligne d'Adaptive Server Anywhere.
- ◆ **Sybase Central** Démarre Sybase Central, l'outil de gestion de base de données.

Vous disposez en outre d'éléments relatifs à InfoMaker et à PowerAMC.

CHAPITRE 2

Présentation d'Adaptive Server Anywhere et UltraLite

Présentation Ce chapitre présente le système de base de données relationnelle Adaptive Server Anywhere, ainsi que la base de données Adaptive Server Anywhere à empreinte mémoire réduite, UltraLite. Il décrit leur utilisation, leurs fonctionnalités et leurs différences.

Sommaire	Sujet :	page
	Présentation d'Adaptive Server Anywhere	10
	Utilisations prévues	11
	Spécificités	12
	Serveur de base de données	14
	Applications	15
	Présentation d'UltraLite	16
	Différences entre Adaptive Server Anywhere et UltraLite	18

Présentation d'Adaptive Server Anywhere

Ce chapitre décrit les particularités d'Adaptive Server Anywhere et en explicite le rôle.

Adaptive Server Anywhere fournit une série d'outils pour enregistrer et gérer les données. Vous pouvez utiliser ces outils pour insérer des données dans la base, modifier la structure de la base de données et consulter ou modifier les données.

Le système de gestion de bases de données Adaptive Server Anywhere est au coeur de SQL Anywhere Studio. Adaptive Server Anywhere a été conçu pour des tâches requérant une base de données SQL aux fonctionnalités complètes. Il est conçu pour opérer dans des environnements variés. Il exploite la mémoire disponible et les ressources CPU, fournissant ainsi de bonnes performances dans des environnements à fortes ressources. En outre, il s'adapte parfaitement aux environnements dont les ressources physiques et d'administration de base de données sont limitées, notamment les environnements informatiques mobiles, les bases de données embarquées et les serveurs de base de données destinés aux petites et moyennes entreprises.

Utilisations prévues

Adaptive Server Anywhere remplit parfaitement les rôles suivants :

◆ **Serveur de base de données pour petites et moyennes entreprises**

Adaptive Server Anywhere est conçu pour répondre aux besoins des petites et moyennes entreprises, avec un nombre d'utilisateurs allant de quelques personnes à plusieurs centaines. En effet, le système fournit aux groupes de travail et aux sociétés une base de données hautes performances, adaptée (mais pas limitée) aux environnements dont les ressources matérielles et l'administration sont limitées.

Adaptive Server Anywhere peut utiliser plusieurs processeurs et jusqu'à 64 Go de mémoire. Nos clients possèdent des bases de données Adaptive Server Anywhere avec des dizaines de gigaoctets de données utilisées en production.

◆ **Base de données embarquée** De nombreuses applications requièrent une base de données "en arrière-plan". Par exemple, les questionnaires de données personnelles, les systèmes de gestion de documents, les applications de surveillance réseau—toute application stockant des informations. Adaptive Server Anywhere est conçu pour servir de base de données à ces applications. L'option de déploiement UltraLite est conçue pour les environnements embarqués disposant de ressources très limitées.

L'un des atouts clés des bases de données embarquées est qu'elles peuvent fonctionner à 100 % sans administration. Adaptive Server Anywhere a prouvé l'intérêt d'une telle fonctionnalité dans un grand nombre d'applications commerciales exigeantes.

◆ **Informatique mobile** Les ordinateurs portables et les notebooks sont désormais répandus sur de nombreux lieux de travail. Adaptive Server Anywhere joue parfaitement le rôle de base de données SQL pour ces ordinateurs. Avec la synchronisation MobiLink et la réplication SQL Remote, Adaptive Server Anywhere développe le traitement par transaction au sein de l'entreprise. L'option de déploiement UltraLite et la technologie de synchronisation MobiLink offrent les fonctionnalités de base de données complètes sur des périphériques disposant de ressources limitées, tels les ordinateurs portables.

Spécificités

Adaptive Server Anywhere intègre les spécificités technologiques suivantes :

- ◆ **Système complet de gestion de base de données relationnelle SQL** Adaptive Server Anywhere est un système de gestion de base de données relationnelle à traitement par transaction (SGBDM), doté de fonctionnalités complètes de restauration, de sauvegarde en ligne, d'actions d'intégrité référentielle, ainsi que de procédures stockées, de triggers, d'une fonction de contrôle de concurrence au niveau ligne, de planifications et d'événements, d'un langage SQL riche et de toutes les fonctionnalités que vous attendez d'un SGBD relationnel SQL complet.
- ◆ **Economie de ressources matérielles** Adaptive Server Anywhere requiert moins de mémoire et de ressources disque que les autres systèmes de gestion de base de données.
- ◆ **Facilité d'utilisation** Adaptive Server Anywhere s'optimise automatiquement et est très facile d'utilisation. Adaptive Server Anywhere ne requiert pas de lourdes tâches d'administration de bases de données, contrairement à la plupart des systèmes de gestion de bases de données relationnelles.
- ◆ **Utilisation en mode autonome et en réseau** Adaptive Server Anywhere peut être utilisé en mode autonome comme une base de données embarquée dans une application centrée sur les données ou comme un serveur réseau en environnement client/serveur multi-utilisateur ou à trois niveaux. En tant que système de base de données embarqué, il peut être lancé automatiquement par une application, dès que nécessaire.
- ◆ **Hautes performances** Même si Adaptive Server Anywhere est conçu pour une administration simple et des ressources modérées, il s'agit d'un SGBD évolutif, à hautes performances. Adaptive Server Anywhere peut être exécuté sur plusieurs CPU, est doté d'un optimiseur de requêtes évolué et propose des outils de surveillance et d'optimisation des performances.
- ◆ **Interfaces standard** Adaptive Server Anywhere fournit un pilote ODBC 3.5 natif pour optimiser les performances des applications ODBC et un pilote OLE DB pour les environnements de programmation ActiveX Data Object (ADO). Il inclut un fournisseur de données ADO.NET pour Adaptive Server Anywhere et est livré avec Sybase jConnect pour JDBC, ainsi qu'un pilote iAnywhere JDBC. Il prend en charge les interfaces Embedded SQL et Sybase Open Client.

- ◆ **Une solution multiplate-forme** Adaptive Server Anywhere fonctionne sur plusieurs systèmes d'exploitation, dont Windows, Novell NetWare, Sun Solaris et Linux.

☞ Les éléments disponibles peuvent varier d'une plate-forme à l'autre. Pour plus d'informations, reportez-vous à la section [“Plates-formes supportées par SQL Anywhere Studio”](#) à la page 37.

Conditions système

☞ Pour plus d'informations sur les systèmes d'exploitation supportés pour les composants de SQL Anywhere Studio, reportez-vous à la section [“Plates-formes supportées par SQL Anywhere Studio”](#) à la page 37.

Configuration requise en matière de logiciel réseau

Si vous utilisez un serveur réseau Adaptive Server Anywhere, le programme réseau adéquat doit être installé et opérationnel.

Le serveur réseau Adaptive Server Anywhere est disponible pour Windows, Novell NetWare, Linux et UNIX.

Adaptive Server Anywhere supporte le protocole réseau TCP/IP et le protocole SPX pour Novell NetWare.

Serveur de base de données

Deux versions du serveur de base de données Adaptive Server Anywhere sont incluses dans le produit :

- ◆ **Le serveur de base de données personnel** Ce serveur est destiné à une utilisation mono-utilisateur sur une seule machine : par exemple, en tant que serveur de base de données embarqué. Il est également utile pour les travaux de développement.

Le nom de l'exécutable du serveur personnel est le suivant :

- ◆ Sous UNIX, *dbeng9*.
 - ◆ Sous Windows, à l'exception de Windows CE, *dbeng9.exe*.
- ◆ **Le serveur de base de données réseau** Outre les fonctionnalités du serveur personnel, le serveur réseau supporte les communications client/serveur sur un réseau. Il convient à une exploitation multi-utilisateur.

Le nom de l'exécutable du serveur réseau est le suivant :

- ◆ Sous UNIX, *dbsrv9*.
- ◆ Sous Windows, y compris Windows CE, *dbsrv9.exe*.
- ◆ Sous Novell NetWare, le serveur est le module NLM (NetWare Loadable Module) *dbsrv9.nlm*.

Fonctionnalités SQL
identiques dans chaque
version

Le moteur de traitement des requêtes est identique dans les deux versions du serveur. Elles prennent en charge exactement le même langage SQL et les mêmes fonctionnalités de base de données. Le serveur personnel ne supporte ni les communications réseau, ni plus de dix connexions concurrentes, ni plus d'un processeur. Les applications développées pour un serveur personnel fonctionnent de la même façon sur un serveur réseau.

Applications

Cette section présente quelques-unes des applications de base de données fournies avec Adaptive Server Anywhere dans le produit SQL Anywhere Studio. Ces applications vous aident à concevoir, créer et gérer vos bases de données.

Les composants installés dépendent du système d'exploitation utilisé, des choix effectués lors de l'installation du logiciel et de la façon dont le produit vous a été fourni : seul ou en tant que composant d'un autre produit.

Par exemple, si vous avez reçu Adaptive Server Anywhere en tant que composant d'un autre produit, vous ne disposez peut-être pas des deux versions du serveur de base de données. De même, tous les composants ne sont pas disponibles sur tous les systèmes d'exploitation. Par exemple, il n'existe pas de serveur personnel pour NetWare, mais uniquement un serveur réseau.

Sybase Central

Sybase Central propose une interface utilisateur graphique qui permet de créer et de modifier les bases de données et leurs objets, d'examiner la structure des bases et de gérer le serveur de base de données. Vous pouvez l'utiliser pour créer une base de données, ajouter une table, insérer une colonne dans une table, etc.

Interactive SQL

Interactive SQL est une application permettant de saisir et d'envoyer des instructions SQL à une base de données. Avec Interactive SQL, vous pouvez interroger et modifier des données de votre base ou en modifier la structure. Tout ce qui peut être réalisé dans Sybase Central peut l'être dans Interactive SQL, mais les tâches d'administration sont plus simples dans Sybase Central.

Utilitaires

Le jeu d'utilitaires fourni permet d'exécuter des tâches d'administration telles que la sauvegarde d'une base de données. Ces utilitaires offrent la possibilité d'inclure des fichiers batch pour les tâches répétitives.

☞ Pour plus d'informations sur les utilitaires d'administration, reportez-vous à la section **“Utilitaires d'administration de base de données”** [ASA - Guide d'administration, page 503].

Présentation d'UltraLite

L'arrivée sur le marché de petits périphériques informatiques, tels que les ordinateurs de poche, les pagers et les téléphones portables, a entraîné une demande pour une base de données avec des besoins en mémoire encore plus modestes que ceux d'Adaptive Server Anywhere. Une des possibilités évidentes était de concevoir un moteur de base de données relationnelle allégé, mais notre expérience a prouvé que chaque application et chaque client disposent d'un ensemble de fonctions précis qui sont essentielles pour eux. De plus, une telle approche implique que les clients connaissent deux bases de données différentes, se distinguant d'une manière ou d'une autre.

C'est pourquoi nous avons alors développé UltraLite, une technologie innovante qui utilise une base de données de référence et le code source d'application du client pour créer un moteur de base de données relationnelle ne contenant que les fonctions d'Adaptive Server Anywhere utilisées par cette application. Chaque demande est enregistrée avec un plan d'accès complet pour une exécution rapide ; le code nécessaire à l'exécution des tâches requises est intégré dans le moteur de base de données UltraLite. Chaque moteur de base de données UltraLite est différent, mais la plupart d'entre eux ne correspondent qu'à quelques dizaines de kilo-octets et peuvent être facilement exécutés sur un périphérique aussi réduit qu'un pager.

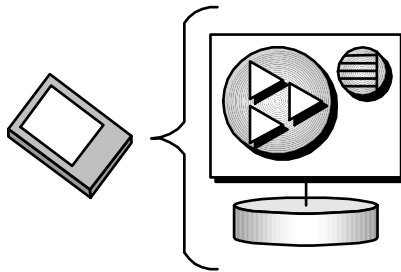
Adaptive Server Anywhere sert de base de données de référence lors de la création de votre application UltraLite, si bien que les instructions SQL, les types de données etc. sont exactement ceux d'Adaptive Server Anywhere. UltraLite est une technologie de déploiement pour Adaptive Server Anywhere et non pas un autre système de base de données. Les tâches que doit exécuter chaque moteur de base de données UltraLite sont exécutées d'une manière entièrement compatible avec Adaptive Server Anywhere.

UltraLite comporte une technologie de synchronisation MobiLink intégrée qui lie votre application au réseau d'informations.

UltraLite prend en charge la plate-forme Palm Computing, Windows CE et les autres systèmes d'exploitation utilisés dans les petits périphériques, tels que Java.

Architecture UltraLite

Pour fournir une application de base de données à un petit périphérique, par exemple un organisateur, vous pouvez avoir recours à la technologie de déploiement UltraLite. Avec UltraLite, le serveur de base de données et l'application font partie du même processus et le serveur est propre à l'application.



Dans ce cas, il est possible que la base de données ne se présente pas comme un fichier sur le disque. La méthode de stockage de la base de données dépend de la plate-forme de déploiement.

Différences entre Adaptive Server Anywhere et UltraLite

Cette section met en évidence les différences entre Adaptive Server Anywhere et UltraLite pour vous aider à choisir la technologie la plus adaptée à vos besoins.

Si vous savez de quelles ressources vous disposez, il pourra vous être facile de choisir entre Adaptive Server Anywhere et UltraLite. Si vous prévoyez un déploiement sur une plate-forme prise en charge par les deux technologies, comme Windows CE, réfléchissez aux points suivants.

Adaptive Server Anywhere

Depuis de nombreuses années, Adaptive Server Anywhere fournit une technologie de base de données relationnelle conçue spécifiquement pour les besoins de l'informatique mobile, embarquée et de groupe de travail. Ce produit a été conçu dès le départ dans cet esprit. Adaptive Server Anywhere présente les avantages suivants :

- ◆ Adaptive Server Anywhere est conçu pour fonctionner de façon efficace avec une mémoire, un niveau de CPU et un espace disque limités. Les fonctions principales du type optimiseur de requêtes et mécanisme de mise en mémoire cache des données sont spécifiquement conçues pour fonctionner sans utilisation excessive des ressources. Adaptive Server Anywhere contient également des fonctions qui permettent de profiter des serveurs de groupe de travail, notamment la prise en charge de nombreux utilisateurs, l'évolutivité vers plusieurs CPU et les fonctions avancées concurrentes. Si vous exécutez la majorité de vos déploiements sur des PC, Adaptive Server Anywhere est le bon choix.
- ◆ Adaptive Server Anywhere est une solution multiplate-forme. La même base de données s'exécute sous Windows (Windows 95 et ultérieurs, Windows NT et ultérieurs et Windows CE), sous UNIX, notamment Linux, et Novell NetWare. Vous pouvez déplacer un fichier de base de données d'un système d'exploitation vers un autre.
- ◆ Adaptive Server Anywhere est conçu pour fonctionner sans administration, ce qui en fait une base de données embarquée idéale. Adaptive Server Anywhere fournit un optimiseur de requête à auto-optimisation, des fonctionnalités de gestion d'événements et de planification intégrées, ainsi que des mécanismes de démarrage et d'arrêt automatique.
- ◆ De nombreuses années de collaboration fructueuse avec les clients ont

permis de mettre au point un large éventail de fonctions testées en conditions réelles. Il s'agit non seulement des fonctionnalités standard des procédures stockées, des triggers, de l'intégrité référentielle déclarative, du traitement complet des transactions et de leur restauration, mais également de toutes les petites fonctions supplémentaires qui peuvent faire le succès ou l'échec d'un projet.

- ◆ Les technologies de synchronisation de SQL Anywhere (SQL Remote et MobiLink) ont été conçues pour vous permettre d'intégrer les bases de données Adaptive Server Anywhere dans l'infrastructure de votre organisation.
- ◆ Adaptive Server Anywhere peut être plus avantageuse si vous préférez une autre interface que Java ou Embedded SQL, comme ODBC ou OLE DB, et que la plate-forme cible est suffisamment puissante pour ne pas nécessiter UltraLite.

UltraLite : la base de données “modèle réduit”

UltraLite est une technologie innovante qui utilise une base de données de référence et le code source d'application du client pour créer un moteur de base de données relationnelle ne contenant que les fonctions d'Adaptive Server Anywhere utilisées par cette application. UltraLite présente les avantages suivants :

- ◆ Les moteurs de base de données UltraLite comportent uniquement le code nécessaire à l'exécution des tâches spécifiées lors de la compilation de l'application, si bien que vous ne pouvez pas utiliser de SQL dynamique pour exécuter des demandes ad hoc sur une base de données UltraLite.
Chaque moteur de base de données et base de données UltraLite peut être utilisé avec une seule application à la fois : si vous souhaitez utiliser plusieurs applications avec une seule base de données, vous devez opter pour Adaptive Server Anywhere.
- ◆ Le modèle de programmation pour UltraLite active un code d'accès à la base de données indépendant de la plate-forme, ce qui permet de “porter” l'interface utilisateur sur de nouvelles plates-formes ou de nouveaux périphériques sans avoir à modifier la couche d'accès aux données sous-jacente.
- ◆ Si vous exécutez un déploiement sur de petits périphériques tels que la plate-forme Palm Computing, UltraLite est le plus adapté.

CHAPITRE 3

Présentation des technologies de réplication

Présentation

La **réplication** consiste à partager des données entre des bases physiquement distinctes. Lorsqu'une application modifie des données partagées sur l'une des bases de données, les modifications sont diffusées vers autres bases incluses dans la configuration de réplication. La diffusion des modifications peut s'effectuer de manières diverses et via une variété de canaux, ce qui préserve l'intégrité des données tout en permettant des configurations de réplication flexibles. La réplication de données est également appelée **synchronisation** de données.

Sybase offre trois technologies de réplication. **MobiLink** et **SQL Remote** sont conçues pour la réplication entre une base de données centrale et un grand nombre de bases de données distantes. **Replication Server** est destiné à la réplication en quasi-temps réel entre un petit nombre de bases de données.

Sommaire

Sujet :	page
Introduction	22
Bases de données consolidées et bases de données distantes	23
Méthodes de diffusion	26
Technologies de réplication Sybase	29

Introduction

Cette section présente les concepts élémentaires de la réplication de données.

Avantages de la réplication de données

Disponibilité des données	L'un des principaux avantages d'un système de réplication est que les données sont disponibles localement et non plus par le biais de connexions potentiellement onéreuses, moins fiables et plus lentes, établies avec une seule base de données centralisée. Les données sont accessibles localement, même en l'absence de toute connexion à un serveur central, de sorte que vous n'êtes pas coupé de vos données en cas de défaillance d'une connexion réseau longue distance.
Temps de réponse	La réplication améliore les temps de réponse des requêtes d'interrogation pour deux raisons. Les requêtes sont traitées sur un serveur local sans accès à un réseau étendu, ce qui accélère le débit. Par ailleurs, le traitement local allège la charge du serveur de bases de données central, ce qui permet de moins solliciter le processeur.

Défis posés aux technologies de réplication

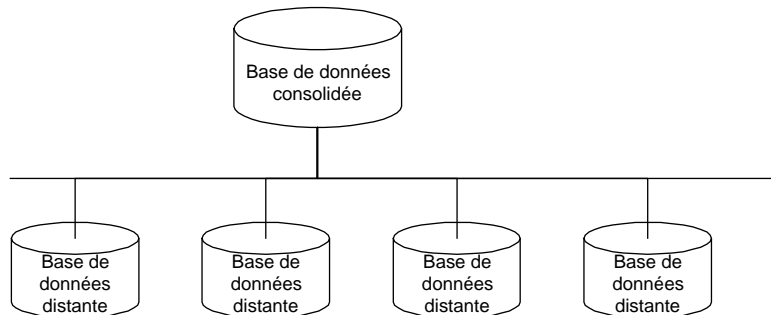
	Toute technologie de réplication se doit de répondre à plusieurs défis qui se posent du fait de la plus grande souplesse conférée par la réplication.
Intégrité des transactions	<p>La préservation constante de l'intégrité des transactions sur chaque base constitue l'un des défis que tout système de réplication doit relever.</p> <p>Replication Server et SQL Remote répliquent des parties du journal de transactions pour le maintien des transactions lors de la réplication : les transactions sont répliquées dans leur intégralité ou ne le sont pas du tout. Cette fonction garantit une intégrité transactionnelle au niveau de chacune des bases de données de la configuration.</p> <p>MobiLink consolide les modifications apportées à plusieurs transactions validées. Ces modifications sont appliquées à une autre base de données lors d'une seule transaction.</p>
Cohérence des données	L'autre défi posé aux systèmes de réplication est la préservation de la cohérence des données dans l'ensemble de la configuration. Les systèmes de réplication assurent une cohérence lâche dans l'ensemble de la configuration : autrement dit, toutes les modifications sont répliquées sur chaque site de façon cohérente sur une période donnée, mais des copies différentes de ces données peuvent se trouver sur différents sites à n'importe quel moment.

Bases de données consolidées et bases de données distantes

MobiLink et SQL Remote permettent de répliquer des données entre une base de données consolidée et un ensemble de bases distantes.

Une **base de données consolidée** est une base qui contient toutes les données à répliquer. Une **base de données distante** est une copie de la base de données consolidée qui peut s'exécuter sur le même site que la base consolidée ou sur un site physiquement distant.

Le schéma ci-dessous représente une petite installation.



Utilisateurs distants

Une installation de réplication inclut de nombreuses bases de données distantes. Chaque base de données distante contient un sous-ensemble de données dans la base de données consolidée. Chaque base distante est une base de données séparée physiquement, généralement située sur un ordinateur distinct. Toutes les bases distantes doivent maintenir une cohérence avec la base de données consolidée.

L'ensemble de la configuration de réplication peut être considérée comme une base de données dispersée, la copie maîtresse de toutes les données étant conservée sur le site de la base consolidée.

Chaque site distant qui soumet des répliqués à la base consolidée est considéré comme un **utilisateur distant** de cette base. Lorsqu'un site distant est un serveur multi-utilisateur, l'ensemble du site est considéré comme un unique utilisateur distant de la base consolidée.

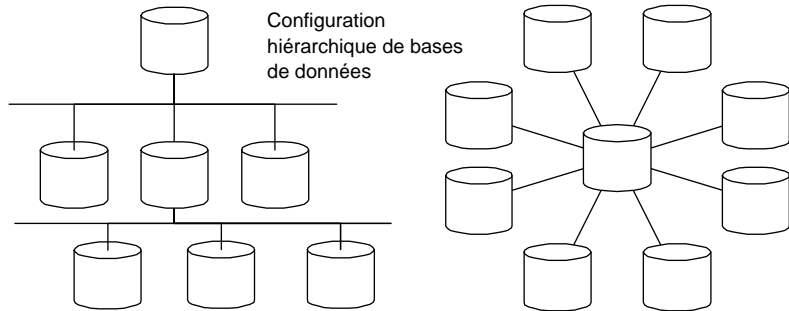
Configurations hiérarchiques des bases de données

Dans une **configuration hiérarchique**, chaque base de données possède une seule base parente, sauf la base de données consolidée, qui n'en a aucune.

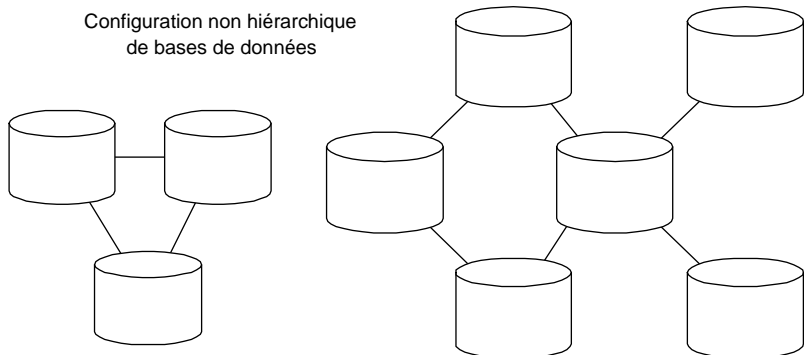
SQL Remote supporte les configurations hiérarchiques de base de données ;

la réplication d'égal à égal ou toute autre configuration non hiérarchique ne sont pas supportées. MobiLink est aussi normalement utilisé dans une configuration hiérarchique, mais peut également être utilisé dans d'autres configurations.

Si l'on prend deux bases qui partagent directement des données dans une configuration hiérarchique, l'une est toujours au-dessus de l'autre dans la hiérarchie.



Dans les configurations non hiérarchiques, les notions “supérieur” et “inférieur” ne sont pas clairement définies.



Dans une installation MobiLink ou SQL Remote, chaque base de données contient l'intégralité ou un sous-ensemble des données répliquées par la base de données du niveau immédiatement supérieur.

Les bases de données distantes peuvent contenir des tables absentes de la base consolidée, à condition que celles-ci ne soient pas impliquées dans la réplication. SQL Remote exige que la table et les noms de colonne des bases de données distantes correspondent à ceux de la base consolidée. En revanche, MobiLink permet le stockage des données dans des colonnes et tables des bases de données distantes qui sont différentes de celles de la base de données consolidée, ce qui assure une plus grande flexibilité.

Réplication bidirectionnelle

Toutes les technologies de réplication Sybase offrent une réplication bidirectionnelle : les modifications apportées à la base consolidée sont diffusées dans les bases distantes et celles effectuées sur les bases distantes sont diffusées dans la base consolidée, puis sur les autres bases distantes. Avec Sybase Replication Server, un ensemble de données ne peut être modifié que sur un site. Avec SQL Remote et MobiLink, les mêmes données peuvent être modifiées simultanément en plusieurs endroits et la résolution des conflits est assurée.

Méthodes de diffusion

Lorsqu'une transaction modifie des données partagées sur l'une des bases de données, la transaction ou les modifications doivent être répliquées sur les autres bases de la configuration de réplication. Cette tâche peut être accomplie de différentes façons.

Réplication par session : MobiLink

Dans un système de réplication par session, la synchronisation se produit en temps réel par le biais d'une liaison de communications directe. Par exemple, la connexion peut s'effectuer par modem, réseau ou modem radio. Les sites distants se connectent à des intervalles de minutes, heures, jours ou semaines.

Un processus de synchronisation orienté session est analogue à une conversation téléphonique dans laquelle toutes les questions en suspens à chaque bout du fil sont résolues. Le processus respecte un format particulier. Un site distant MobiLink commence par ouvrir une connexion sur un serveur de synchronisation MobiLink et par télécharger la liste complète de toutes les modifications apportées à la base distante depuis la synchronisation précédente. A la réception des données, le serveur met à jour la base consolidée, puis renvoie toutes les modifications adéquates. Le site distant intègre tout l'ensemble des modifications, puis renvoie une confirmation et ferme la connexion.

Réplication par message : SQL Remote

SQL Remote échange des données entre les bases par l'intermédiaire de **messages**. Les messages sont en général des fichiers placés dans un répertoire particulier ou des messages électroniques spécialement formatés. Un **agent de message**, rattaché à chaque base de données, envoie des messages relatifs aux modifications concernant ses propres données. Le même agent reçoit également des messages d'une ou de plusieurs autres bases de données, et modifie la base de données en fonction du contenu des messages reçus. Ce système permet de répliquer des bases de données sans connexion directe : une connexion occasionnelle par message, telle que le courrier électronique ou une liaison téléphonique périodique, s'avère suffisante.

Avec les communications par message, chaque message transporte son adresse de destination et d'autres informations de contrôle afin qu'aucune connexion directe ne soit nécessaire entre les applications qui échangent des informations. Par exemple, un message électronique contient l'adresse de destination. Il n'existe aucune connexion directe entre le serveur émetteur et

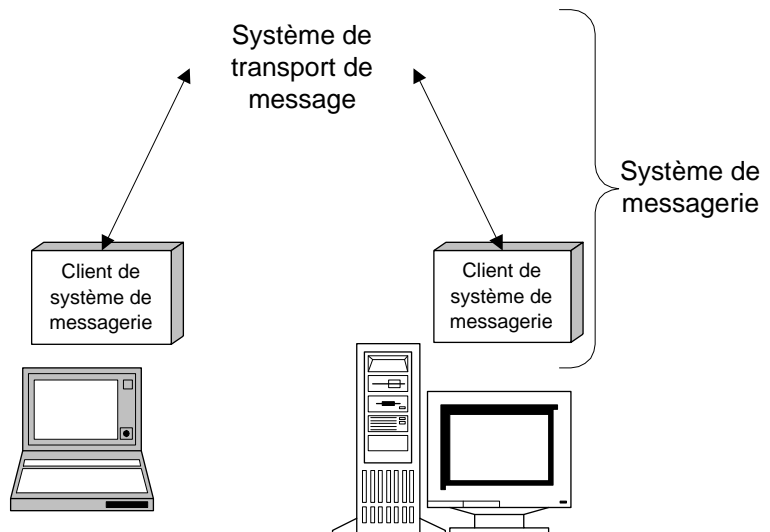
le destinataire.

Services de messagerie
et méthodes de livraison
en différé

De même que les applications client/serveur orientées session s'appuient sur des piles de protocoles de communication en réseau, telles que TCP/IP ou SPX de Novell NetWare, les applications orientées messages reposent sur des services de messagerie, tels que le protocole SMTP (Simple Mail Transfer Protocol) d'Internet, MAPI (API de messagerie de Microsoft), VIM (Vendor Independent Messaging) de Lotus ou sur une simple liaison de partage de fichiers.

Les services de messagerie utilisent une méthode de **livraison en différé** pour l'acheminement de chaque message : par exemple, les systèmes de courrier électronique stockent les messages jusqu'à ce que les destinataires ouvrent leur boîte aux lettres pour les lire.

Dans la mesure où le système de réplication vient s'appuyer sur un système de messagerie, un système de réplication par messages comme SQL Remote n'a pas besoin de mettre en place un système de livraison en différé pour acheminer les messages. De même que les applications client/serveur orientées session n'ont pas à mettre en oeuvre leurs propres piles de protocoles pour transmettre les informations entre le client et le serveur, SQL Remote utilise les systèmes de messagerie existants pour acheminer les messages.



Livraison garantie

Pour un travail en toute fiabilité, un système de réplication par messages doit garantir d'une part que tous les messages arrivent à destination et d'autre part que les messages sont appliqués dans le même ordre que leur ordre

d'envoi. SQL Remote intègre un protocole qui garantit l'application des mises à jour de réplication dans le bon ordre.

Réplication par connexion : Replication Server

Certaines technologies de réplication sont fondées sur l'existence d'une connexion continue, ou du moins presque continue, entre les bases de données. Cette connexion permet aux deux bases de maintenir un dialogue en continu. Ces types de systèmes sont particulièrement performants en matière de réplication rapide des modifications. Effectivement, avec des ressources et une capacité de canal suffisantes, la réplication peut être fiable avec un délai ne dépassant pas quelques secondes.

Replication Server est un système de réplication presque en temps réel conçu à l'origine pour la réplication entre un petit nombre de bases de données. Il est normalement utilisé avec une connexion continue et fiable à vitesse élevée. Il intègre des techniques de livraison en différé qui permettent à la réplication de se poursuivre automatiquement lorsqu'une connexion est perdue puis retrouvée ultérieurement.

L'inconvénient majeur de ce type de système vient du coût élevé dû au maintien d'une connexion continue fiable. Cette restriction rend les technologies par connexion plus adaptées à la réplication entre des bases de données volumineuses et fixes. Dans les environnements où les machines distantes ne sont pas fixes ou ne sont connectées qu'occasionnellement, les technologies par messages ou par session offrent des solutions plus souples.

Technologies de réplication Sybase

Sybase propose trois techniques de réplication :

- ◆ **MobiLink** Technologie par session conçue pour une réplication bidirectionnelle entre une base de données centrale consolidée et un grand nombre de bases de données distantes. Elle supporte une grande diversité de serveurs de base de données consolidée, y compris des bases de données non Sybase. L'administration et les ressources requises sur les sites distants sont minimales, ce qui rend cette technologie parfaitement adaptée à toute une gamme d'applications mobiles. A la fin de chaque session de synchronisation, les bases de données sont cohérentes.
- ◆ **SQL Remote** Technologie de réplication par messages conçue pour la réplication bidirectionnelle des transactions. Elle est conçue pour la réplication bidirectionnelle impliquant un serveur de données consolidées et un grand nombre de bases de données distantes. L'administration et les ressources requises sur les sites distants sont minimales, ce qui rend cette technologie parfaitement adaptée aux bases de données mobiles. Ce système est un système à base de messages. En fonction de la configuration, les délais observés pour l'échange entre les bases de données consolidées et distantes peuvent être de l'ordre de secondes, de minutes ou d'heures.
- ◆ **Replication Server** Technologie de réplication par connexion conçue pour la réplication bidirectionnelle des transactions. Elle est bien adaptée à la réplication entre un petit nombre de bases de données d'entreprise connectées à un réseau haut débit, généralement avec un administrateur sur chaque site. Dans une configuration de ce type, il est possible d'atteindre des délais ne dépassant pas quelques secondes.

Choix d'une technologie de réplication

Chaque technologie de réplication Sybase se prête à des applications particulières. Les descriptions suivantes différencient les technologies pour vous permettre de sélectionner celle qui est la plus adaptée à vos besoins.

Vous devez réfléchir aux éléments importants à prendre en compte dans votre application :

Votre système de base de données consolidée

Dans un environnement de réplication traditionnel, une grande base de données sert de référentiel central pour les informations. Vous pouvez parfois choisir un système de base de données adapté à vos besoins. Il arrive aussi qu'une base de données centralisée existe déjà et que vous deviez adapter le système de réplication en fonction de cette base.

	<p>MobiLink peut fonctionner avec un grand nombre de serveurs de base de données courants, comme Sybase Adaptive Server Anywhere, Sybase Adaptive Server Enterprise, Oracle, Microsoft SQL Server et IBM DB2.</p>
	<p>Dans une installation SQL Remote, la base de données centrale doit être Sybase Adaptive Server Anywhere ou Sybase Adaptive Server Enterprise</p>
<p>Votre système de base de données distant</p>	<p>Les technologies de réplication Sybase diffèrent également au niveau du type de base de données distante qu'elles peuvent supporter. MobiLink accepte une base de données distante de type Adaptive Server Anywhere ou UltraLite.</p>
	<p>SQL Remote ne supporte que des bases distantes Adaptive Server Anywhere.</p>
<p>Caractéristiques du réseau</p>	<p>MobiLink et SQL Remote sont deux technologies bien adaptées aux environnements dans lesquels les connexions sont occasionnelles, les sites distants doivent fonctionner de façon isolée pendant des heures ou des jours, même si la fréquence des synchronisations peut être accélérée dès qu'une connexion réseau est disponible. En revanche, Replication Server est destiné à une connexion permanente pour permettre la réplication rapide de gros volumes de données.</p>
	<p>MobiLink est une technologie basée sur une session. Une connexion en temps réel est nécessaire pendant la synchronisation. Si cette connexion est interrompue avant la fin de la synchronisation, le processus ne pourra se poursuivre qu'à la synchronisation suivante. En revanche, SQL Remote transmet les informations via des messages qui peuvent être envoyés ou reçus en mode asynchrone. Ces messages peuvent se présenter sous la forme de fichiers sur le disque dur ou de messages électroniques. Ils peuvent être traités dès leur réception, pour une réplication incrémentielle.</p>
<p>Temps de latence</p>	<p>Il arrive quelquefois que vos informations doivent être impérativement répliquées de façon immédiate. Dans d'autres circonstances, une réplication une ou deux fois par jour peut se révéler suffisante. En fait, une réplication plus fréquente risque d'être impossible en l'absence de connexion réseau disponible.</p> <p>MobiLink et SQL Remote sont tous les deux destinés principalement aux cas de réplication non régulière, par exemple quelques fois par jour ou par semaine. MobiLink et SQL Remote sont capables de gérer une synchronisation à un rythme plus fréquent, mais les exigences en ressources et en réseau sont plus grandes. Cependant, si les ressources fournies sont suffisantes, des synchronisations MobiLink peuvent se produire à des intervalles de quelques minutes. SQL Remote, lorsqu'il est exécuté en mode continu, permet des réplications à des intervalles de quelques secondes.</p>

Replication Server est conçu pour les configurations nécessitant une réplication quasiment en temps réel.

Nombre de sites distants En présence d'un très grand nombre d'utilisateurs distants, les meilleures options sont MobiLink ou SQL Remote. La conception basée sur des messages de SQL Remote permet à une installation classique de gérer des milliers d'utilisateurs distants. L'évolutivité de MobiLink n'est limitée que par celle du système de gestion de base de données consolidée. Replication Server est conçu pour quelques sites seulement.

Ces chiffres ne sont fournis qu'à titre indicatif, mais il n'existe pas de limite précise au nombre maximal de sites distants exploitables avec ces systèmes. Les chiffres réels dépendent de la quantité des informations répliquées, de la fréquence de synchronisation et de la conception de votre application.

Ordre des transactions SQL Remote réplique les données en parcourant le journal de transactions pour identifier les opérations qu'il doit transférer et prépare un ou plusieurs messages. Il classe ensuite ces messages dans l'ordre et les envoie au site distant ou consolidé. Pour le traitement des messages reçus, SQL Remote procède toujours dans l'ordre dans lequel ils ont été appliqués à l'autre base de données. Lorsque cela s'avère nécessaire, il reporte automatiquement le traitement d'un message jusqu'à ce que tous les messages antérieurs aient été appliqués.

Contrairement à SQL Remote, MobiLink travaille en regroupant les résultats de plusieurs transactions sur la base de données distante en un ensemble de modifications à appliquer à la base de données consolidée. La synchronisation se produisant toujours au niveau d'un délimiteur de transaction, l'intégrité référentielle est préservée. L'ordre des modifications individuelles apportées lors des transactions au niveau du composant n'est pas maintenu. Toutefois, comme les données non validées ne sont jamais synchronisées, l'intégrité des données est préservée.

Cohérence des données à un moment précis Les données des deux bases sont cohérentes dès que la session de synchronisation MobiLink est terminée. La capacité à garantir la cohérence des données sur un site distant à un moment précis est un point fort de la réplication par session. Par exemple, s'il est important que les données d'un site distant correspondent exactement à celle de la base consolidée à une heure précise, 10 heures du matin par exemple, il est possible d'exécuter la synchronisation juste avant 10 heures. Du moment que la synchronisation est réussie, l'actualité des données sur le site distant est assurée.

Lorsque des modifications de données sont répliquées via un échange de messages, il est difficile de garantir une totale cohérence des données d'un site distant en particulier avec celles du site consolidé à un moment précis dans le temps. Par exemple, il peut arriver qu'un message se perde en cours

de route. SQL Remote détecte automatiquement l'erreur et envoie à nouveau le message, mais des interruptions de ce type peuvent entraîner des délais inattendus.

Récapitulatif des caractéristiques de la technologie de réplication

Le tableau suivant récapitule les caractéristiques de chaque technologie de réplication. Les sections suivantes développent les différentes entrées de ce tableau.

Technologie de réplication	Nombre de bases de données	Connexion	Temps de latence	Volume	Types de base de données
MobiLink	Volumineuse	Occasionnelle	Medium	Medium	Hétérogène
SQL Remote	Volumineuse	Occasionnelle	Réduite	Medium	Homogène
Replication Server	Réduite	Permanente	Réduite	Elevée	Hétérogène

Caractéristiques MobiLink

MobiLink a été conçu pour les installations de réplication qui répondent aux conditions ci-dessous :

- ◆ **Grands nombres de bases de données** MobiLink est conçu pour supporter un nombre élevé de bases de données distantes. Une seule installation peut en supporter plusieurs milliers.
- ◆ **Connexions occasionnelles** MobiLink supporte les bases de données qui sont connectées occasionnellement ou indirectement au réseau sur lequel le serveur est actif. L'évolutivité de MobiLink n'est limitée que par celle du système de gestion de base de données consolidée.
- ◆ **Temps de latence moyen à élevé** Le temps de latence correspond au décalage temporel observé entre l'entrée des données dans une base et leur réplication sur chacune des bases de l'installation. En général, les applications se connectent et se synchronisent à des intervalles de minutes, heures ou jours.
- ◆ **Volume faible à moyen** Les informations de transfert pour les sites distants sont préparées pour un seul site distant à la fois. De grandes quantités de données dans un système MobiLink peuvent entraîner de longues durées de connexion puisque le site distant ne peut pas se déconnecter avant la fin de la synchronisation.

- ◆ **Bases de données hétérogènes** MobiLink supporte la plupart des systèmes de base de données relationnelle les plus courants pour les utiliser comme base consolidée. Le schéma des sites distants peut différer de celui de la base de données consolidée parce que vous contrôlez le processus de synchronisation à l'aide de scripts.

Caractéristiques de SQL Remote

SQL Remote a été conçu pour les installations de réplication qui répondent aux conditions ci-dessous :

- ◆ **Grands nombres de bases de données** SQL Remote est conçu pour supporter un nombre élevé de bases de données distantes. Il peut supporter des milliers de bases distantes sur une seule installation car il est possible de préparer simultanément les messages pour de nombreux sites distants.
- ◆ **Connexions occasionnelles** SQL Remote supporte les bases de données qui sont connectées occasionnellement ou indirectement au réseau sur lequel le serveur est actif.
- ◆ **Temps de latence faible à élevé** Un temps de latence élevé implique un décalage temporel important entre l'entrée des données dans une base et leur réplication sur chacune des bases de l'installation. Avec SQL Remote, les messages de réplication sont généralement envoyés à des intervalles exprimés en secondes, minutes, heures ou jours.
- ◆ **Volume faible à modéré** Comme les messages de réplication sont livrés occasionnellement, un volume élevé de transactions sur chaque site distant peut conduire à un volume de messages très important. SQL Remote est par conséquent mieux adapté aux systèmes présentant un volume relativement faible de données répliquées par base de données distante. Sur le site consolidé, SQL Remote s'avère performant puisqu'il peut préparer les messages pour plusieurs sites simultanément.
- ◆ **Bases de données homogènes** SQL Remote supporte les bases de données Adaptive Server Enterprise et Adaptive Server Anywhere. Chacune des bases de données du système doit présenter sensiblement le même schéma.

Caractéristiques de Replication Server

Replication Server a été conçu pour les installations de réplication qui répondent aux conditions ci-dessous :

-
- ◆ **Petits nombres de bases de données** Replication Server a été conçu pour supporter la réplication entre serveurs, sur des installations qui comprennent généralement moins de cent serveurs.
 - ◆ **Connexions permanentes** Les connexions entre les sites primaires et répliqués peuvent être établies sur un réseau étendu, mais Replication Server est conçu pour les situations dans lesquelles le chemin de connexion est quasi permanent pour permettre l'échange de données entre les serveurs de l'installation.
 - ◆ **Faible temps de latence** Un faible temps de latence implique un court décalage temporel entre l'entrée des données dans une base de données et leur réplication sur chaque autre base de l'installation. Avec Replication Server, les messages de réplication sont généralement envoyés quelques secondes après avoir été entrés au niveau du site primaire.
 - ◆ **Volumes élevés** Avec des connexions quasi permanentes et des performances élevées, Replication Server est adapté aux volumes élevés de messages de réplication.
 - ◆ **Bases de données hétérogènes** Replication Server supporte plusieurs grands SGBD et autorise le mappage des noms d'objet au cours de la réplication, de manière à supporter des bases de données hétérogènes.

DEUXIÈME PARTIE

PLATES-FORMES SUPPORTÉES

Cette partie présente les plates-formes supportées par SQL Anywhere Studio, ainsi que les informations spécifiques de ces plates-formes.

CHAPITRE 4

Plates-formes supportées par SQL Anywhere Studio

Présentation

Ce chapitre répertorie les plates-formes supportées pour les différents composants de SQL Anywhere Studio.

Des informations relatives aux plates-formes prises en charge sont également disponibles sur les pages Web

http://www.iAnywhere.com/products/supported_platforms_9.html et
http://www.iAnywhere.com/products/supported_platforms.html.

Disponibilité

Le fait de mentionner une plate-forme supportée n'indique pas nécessairement la disponibilité immédiate de cette dernière.

Sommaire

Sujet :	page
Introduction	38
Systèmes d'exploitation Windows et NetWare	40
Systèmes d'exploitation UNIX, Linux et Macintosh	46
Bases de données consolidées pour synchronisation MobiLink	50
Pilotes ODBC iAnywhere Solutions - plates-formes supportées	51
Plates-formes cibles UltraLite	52
Versions des systèmes d'exploitation	54

Introduction

SQL Anywhere Studio fournit des technologies de synchronisation et de gestion pour un large éventail de systèmes d'exploitation. Il inclut les éléments décrits ci-après :

- ◆ **Adaptive Server Anywhere** Système de base de données relationnelle destiné aussi bien aux serveurs de base de données des PME qu'aux applications de base de données bureautiques et aux périphériques Windows CE.
- ◆ **UltraLite** Base de données relationnelle spécialement conçue pour les petits périphériques, exécutée sur les périphériques Palm OS et Windows CE.
- ◆ **MobiLink** Système de synchronisation fournissant une technologie de synchronisation bidirectionnelle entre plusieurs bases de données Adaptive Server Anywhere ou UltraLite et un serveur de bases de données central.
- ◆ **SQL Remote** Système de synchronisation pour les bases de données Adaptive Server Anywhere et Sybase Adaptive Server Enterprise.
- ◆ **Outils d'administration** Série d'outils permet de gérer tous les éléments de SQL Anywhere Studio.

Cette annexe répertorie les composants de SQL Anywhere Studio en fonction des systèmes d'exploitation sur lesquels ils peuvent être exécutés. Les informations sont organisées par composant au sein de SQL Anywhere Studio.

☞ Les informations contenues dans ce chapitre sont à jour au moment de sa publication. Pour obtenir des informations récentes, rendez-vous à l'adresse suivante : http://www.ianywhere.com/products/supported_platforms.html.

Présentation

Le tableau suivant contient une synthèse des plates-formes supportées pour les principaux éléments : Il ne contient pas d'informations détaillées sur les différences entre les plates-formes supportées citées ou les versions de UNIX supportées, par exemple. Pour des informations détaillées, reportez-vous aux tableaux dans le reste du document.

Il existe deux types de plates-formes UNIX supportées :

- ◆ **Versión intégrale** Les versions intégrales de SQL Anywhere Studio sont disponibles sur Sun Solaris et Linux.
- ◆ **Versión de déploiement** La version de déploiement contient les principaux éléments d'Adaptive Server Anywhere. Certaines versions de

déploiement contiennent le serveur de synchronisation MobiLink. Les plates-formes de la version de déploiement incluent les systèmes d'exploitation HP-UX, IBM AIX, Compaq Tru64 et Linux 64 bits.

☞ Les versions supportées des systèmes d'exploitation cités sont décrites à la section [“Versions des systèmes d'exploitation”](#) à la page 54.

Composant	Win-dows (32 bits)	Win-dows (64 bits)	Mac OS X	UNIX (version intégrale)	UNIX (version de déploiement)	Net-Ware	Win-dows CE	Palm OS
Serveur de base de données Adaptive Server Anywhere	✓	✓	✓	✓	✓	✓	✓	
Clients Adaptive Server Anywhere	✓	✓	✓	✓	✓	✓ ¹	✓	
Bases de données UltraLite	✓ ²						✓	✓
Développement UltraLite	✓							
Serveur de synchronisation MobiLink	✓	✓	✓	✓	✓ ³			
SQL Remote	✓	✓	✓	✓		✓	✓	
Agent de réplication	✓	Logiciel 32 bits		✓				
Outils d'administration	✓	JRE 32 bits	✓	✓		✓ ⁴		

¹Embedded SQL uniquement

²Réservé aux tests et au développement

³AIX uniquement

⁴Série limitée

Systèmes d'exploitation Windows et NetWare

Le tableau suivant répertorie les systèmes d'exploitation supportés pour les différents composants et les différentes fonctionnalités de SQL Anywhere Studio.

Sauf indication contraire, les versions supportées des systèmes d'exploitation cités sont comme décrites à la section "[Versions des systèmes d'exploitation](#)" à la page 54.

Adaptive Server
Anywhere

Les conditions système varient selon la taille des bases de données, la charge de travail et les performances requises. Les configurations suivantes ne sont que des exigences minimales.

Adaptive Server Anywhere (le serveur de base de données personnel et le serveur de base de données réseau) peut fonctionner avec seulement 4 Mo de mémoire et 8 Ko supplémentaires par connexion client. Une plus grande quantité de mémoire améliore considérablement les performances. Si vous utilisez Java dans la base de données, Adaptive Server Anywhere requiert 5 Mo de mémoire supplémentaires. Votre ordinateur doit disposer de cette capacité de mémoire en plus de la capacité requise pour le système d'exploitation.

Composant ou fonctionnalité	Windows (32 bits)	Windows (64 bits)	Windows CE	NetWare
Serveur de base de données personnel	✓	✓		
Serveur de base de données réseau	✓	✓	✓	✓
Clients ODBC	✓	✓	✓ ⁵	
Clients OLE DB	✓	✓	✓	
Clients Embedded SQL	✓	✓	✓	✓
Clients Open Client	✓	✓		
Clients JDBC (jConnect)	✓	Logiciel 32 bits		
clients JDBC (pilote JDBC iAnywhere)	✓	✓		
Préprocesseur SQL	✓	Logiciel 32 bits		

⁵Il est possible que le gestionnaire de pilote ODBC ne soit pas disponible. Une liaison directe au pilote ODBC est peut-être nécessaire.

Composant ou fonctionnalité	Windows (32 bits)	Windows (64 bits)	Windows CE	NetWare
Java dans la base de données	✓			✓
Accès aux données distantes ⁶	✓	✓		✓ ⁷
Procédures stockées externes	✓	✓ ⁸		✓
Redimensionnement dynamique du cache	✓	✓		
Cryptage fort	✓	✓		✓
Instructions CREATE DATABASE, RESTORE DATABASE, DROP DATABASE	✓	✓		✓
Protocole SPX	✓			✓

Plates-formes de développement UltraLite

Composant	Windows NT/2000/XP (32 bits)	Windows (64 bits)	Windows 98 SE	Windows CE	NetWare
Préprocesseur SQL (Embedded SQL uniquement)	✓		✓		
Générateur UltraLite (API C++ statique, API Java statique)	✓		✓		
UltraLite pour AppForge MobileVB	✓				
UltraLite ActiveX	✓				
Native UltraLite pour Java	✓				
UltraLite.NET	✓ ⁹				
UltraLite pour C++	✓				

⁶Nécessite un pilote ODBC iAnywhere Solutions sous Windows. Pour obtenir une liste des plates-formes supportées, reportez-vous à la section “Pilotes ODBC iAnywhere Solutions - plates-formes supportées” à la page 51.

⁷Nécessite Java dans la base de données et le pilote JDBC.

⁸L'ancienne interface des procédures stockées externes est uniquement disponible dans la version 32 bits.

⁹Nécessite Visual Studio.NET avec la version 1.0.5000 ou supérieure de .NET Compact Framework.

☞ Pour plus d'informations sur les plates-formes de déploiement, reportez-vous à la section [“Plates-formes cibles UltraLite”](#) à la page 52.

Environnements de développement

Vous pouvez utiliser l'un des deux environnements de développement suivants pour créer des applications UltraLite Palm :

- ◆ Metrowerks CodeWarrior version 8 ou 9

CodeWarrior comprend une version du Palm SDK. Selon les périphériques spécifiques ciblés, vous pouvez mettre à niveau votre Palm SDK avec une version plus récente que celle fournie dans l'outil de développement.

- ◆ AppForge MobileVB, utilisant le composant UltraLite MobileVB.

Serveur de synchronisation MobiLink

Composant	Windows (32 bits)	Windows (64 bits)	Windows CE	NetWare
Serveur de synchronisation MobiLink	✓	✓		
Synchronisation TCP/IP	✓	✓		
Synchronisation HTTPS	✓	✓		
Synchronisation HTTP	✓	✓		
Sécurité de la couche de transport	✓			
Logique de synchronisation Java	✓	Logiciel 32 bits		
Logique de synchronisation .NET	✓	✓		

MobiLink - clients

Adaptive Server

Composant	Windows (32 bits)	Windows (64 bits)	Windows CE	NetWare
Clients ASA MobiLink (<i>dbmlsync</i>)	✓	✓	✓	
Synchronisation TCP/IP	✓	✓	✓	
Synchronisation HTTP	✓	✓	✓	
Synchronisation HTTPS	✓	✓	✓	
Sécurité de la couche de transport	✓	✓	✓	

Utilitaires MobiLink

Composant	Windows (32 bits)	Windows (64 bits)	Windows CE	NetWare
Redirecteur	✓ ¹⁰			
Utilitaire d'extraction MobiLink (<i>mlxtract</i>)	✓	✓		

SQL Remote

Les composants SQL Remote exécutés sur les bases de données Adaptive Server Enterprise nécessitent les bibliothèques Sybase Open Client et/ou Open Server.

Composant	Windows (32 bits)	Windows (64 bits)	Windows CE	NetWare
Agent de message ASA (<i>dbremote</i>)	✓	✓	✓	✓
Agent de message ASE (<i>ssremote</i>)	✓	Logiciel 32 bits		
File d'attente permanente ASE (<i>ss-queue</i>)	✓	Logiciel 32 bits		
Utilitaire d'extraction ASA (<i>dbxtract</i>)	✓	✓		
Utilitaire d'extraction ASE (<i>ssxtract</i>)	✓	Logiciel 32 bits		
Type de message File	✓	✓	✓	✓
Type de message FTP	✓	✓	✓	✓
Type de message MAPI	✓	✓		
Type de message SMTP	✓	✓	✓	✓

Agent de réplication

L'Agent de réplication nécessite des bibliothèques Sybase Open Client et Open Server.

¹⁰Windows NT/2000/XP seulement.

Composant	Windows (32 bits)	Windows (64 bits)	Windows CE	NetWare
Agent de réplication	✓	Logiciel 32 bits		

Outils d'administration

Composant	Windows (32 bits)	Windows (64 bits)	Windows CE	NetWare
Utilitaires d'administration et de ligne de commande	✓	✓		✓
Sybase Central	✓	JRE 32 bits		
Interactive SQL	✓	JRE 32 bits		✓ ¹¹
DBConsole	✓	JRE 32 bits		✓ ¹²

A l'exception des utilitaires d'administration et de ligne de commande et des outils fournis sur NetWare, les outils d'administration utilisent un environnement d'exécution Java 2, version 1.4.1.

Les paragraphes qui suivent sont extraits de la documentation du SDK Java 2, édition standard version 1.4.1, et s'appliquent aux outils d'administration.

Le kit SDK Java™ 2 est conçu pour les systèmes d'exploitation Windows 95/98 (1ère ou 2ème édition), Windows Me, Windows NT 4.0 avec Service Pack , Windows 2000 Professionnel, Windows 000 Server, Windows 2000 Advanced Server ou Windows P exécutés sur un processeur Intel.

Un processeur Pentium à 166 MHz (ou plus) avec au moins 32 Mo de RAM est requis pour exécuter les applications graphiques. Pour les applets exécutées dans un explorateur via le module de connexion Java, 48 Mo de mémoire vive sont conseillés. Une capacité de mémoire insuffisante peut entraîner des transferts sur disque et une diminution considérable des performances. Pour des performances optimales, il se peut que davantage de mémoire soit nécessaire pour les programmes très puissants.

Assurez-vous que vous disposez de 70 Mo d'espace disque disponible avant d'installer le logiciel Java 2 SDK. Si vous souhaitez installer les fichiers de

¹¹Version plus limitée que celle installée sur les autres systèmes d'exploitation

¹²Version plus limitée que celle installée sur les autres systèmes d'exploitation

téléchargement de la documentation, 120 Mo d'espace disque disponible supplémentaires sont nécessaires.

Dans les installations autres que celles en langue anglaise, seule l'édition Professionnel de Windows 2000 est supportée.

Systèmes d'exploitation UNIX, Linux et Macintosh

Sauf indication contraire, les version supportées des systèmes d'exploitation cités sont comme décrites à la section “[Versions des systèmes d'exploitation](#)” à la page 54.

Adaptive Server
Anywhere

Composant	Solaris	Linux	Mac OS X	HP-UX	AIX	Tru64
Serveur de base de données personnel	✓	✓	✓	✓	✓	✓
Serveur de base de données réseau	✓	✓	✓	✓	✓	✓
Clients ODBC ¹³	✓	✓	✓	✓	✓	✓
Clients Embedded SQL	✓	✓	✓	✓	✓	✓
Préprocesseur SQL	✓	✓	✓	✓	✓	✓
Clients Open Client	✓	✓	✓	✓	✓	✓
Clients JDBC (pilote jConnect)	✓	✓	✓	✓	✓	✓
Java dans la base de données	✓	✓	✓	✓	✓	✓
Accès aux données distantes ¹⁴	✓	✓	✓			
Procédures stockées externes	✓	✓	✓	✓	✓	✓
pilote JDBC iAnywhere ¹⁵	✓	✓	✓	✓	✓	✓

Linux Note pour Adaptive Server Anywhere Des pannes se sont produites dans le noyau SMP 2.4.2 lors de l'utilisation de processeurs multiples. Dans certaines conditions, le système d'exploitation peut

¹³Il est possible que le gestionnaire de pilote ODBC ne soit pas disponible. Une liaison directe au pilote ODBC est peut-être nécessaire.

¹⁴Les sources de données distantes JDBC nécessitent Java dans la base de données et un pilote JDBC. Les sources de données distantes ODBC nécessitent le pilote ODBC iAnywhere Solutions. Pour obtenir une liste des plates-formes supportées, reportez-vous à la section “[Pilotes ODBC iAnywhere Solutions - plates-formes supportées](#)” à la page 51.

¹⁵Nécessite JRE 1.4 (ou supérieur) sur des processeurs Itanium. Nécessite JRE 1,3 (ou supérieur) sur d'autres processeurs.

corrompre la mémoire, ce qui provoque ensuite des blocages d'applications. Les versions de noyau supérieures à 2.4 ne posent pas ce problème.

Plates-formes hôtes

UltraLite

Composant	Solaris	Linux	Mac OS X	HP-UX	AIX	Tru64
Préprocesseur SQL (Embedded SQL uniquement)	✓	✓				
Générateur UltraLite (C++ API, Pure Java)						

Serveur de synchronisation MobiLink

Composant	Solaris	Linux	Mac OS X	HP-UX	AIX (4.3.3 ou supérieur)	Tru64
Serveur de synchronisation MobiLink	✓	✓	✓		✓	
Synchronisation TCP/IP	✓	✓	✓		✓	
Synchronisation HTTPS	✓	✓	✓		✓	
Synchronisation HTTP	✓	✓	✓		✓	
Sécurité de la couche de transport	✓	✓	✓		✓	
Logique de synchronisation Java	✓	✓	✓		✓	
Logique de synchronisation .NET						

MobiLink - client

Adaptive Server

Composant	Solaris	Linux	Mac OS X	HP-UX	AIX	Tru64
Clients ASA MobiLink (dbmlsync)	✓	✓	✓			
Synchronisation TCP/IP (client)	✓	✓	✓			

Composant	Solaris	Linux	Mac OS X	HP-UX	AIX	Tru64
Synchronisation HTTPS (client)	✓	✓	✓			
Synchronisation HTTP (client)	✓	✓	✓			
Utilitaire d'extraction MobiLink (<i>mlxtract</i>)	✓	✓	✓			

SQL Remote

Les composants SQL Remote exécutés sur les bases de données Adaptive Server Enterprise nécessitent les bibliothèques Sybase Open Client et/ou Open Server. Ces bibliothèques ne sont pas disponibles sur Compaq Tru64.

Composant	Solaris	Linux	Mac OS X	HP-UX	AIX	Tru64
Agent de message ASA (<i>dbremote</i>)	✓	✓	✓			
Agent de message ASE (<i>ssremote</i>)	✓	✓				
File d'attente permanente ASE (<i>ssqueue</i>)	✓	✓				
Utilitaire d'extraction ASA (<i>dbxtract</i>)	✓	✓	✓			
Utilitaire d'extraction ASE (<i>ssxtract</i>)	✓	✓				
Type de message File	✓	✓	✓			
Type de message FTP	✓	✓	✓			
Type de message MAPI						
Type de message SMTP	✓	✓	✓			

Agent de réplication

L'Agent de réplication nécessite des bibliothèques Sybase Open Client et Open Server.

Composant	Solaris	Linux	Mac OS X	HP-UX	AIX	Tru64
Agent de réplication	✓	✓				

Outils d'administration

Composant	Solaris	Linux	Mac OS X	HP-UX	AIX	Tru64
Utilitaires d'administration et de ligne de commande	✓	✓	✓	✓	✓	✓
Sybase Central	✓	✓	✓			
Interactive SQL	✓	✓	✓			

A l'exception des utilitaires d'administration et de ligne de commande, les outils d'administration utilisent un environnement d'exécution Java 2, version 1.3.1.

Les paragraphes qui suivent sont extraits des informations relatives à la configuration système du SDK Java2 pour Solaris, édition standard version 1.3.1, et s'appliquent aux outils d'administration.

L'édition standard version 1.3.1 du SDK Java 2 (J2SDK 1.3.1) est destinée aux systèmes d'exploitation Solaris 2.6, Solaris 7, Solaris 8 et Solaris 9.

Avant d'installer le SDK Java 2, assurez-vous que vous avez installé tous les patches nécessaires pour que cette version soit supportée. Pour vous procurer ces patches, allez sur le site Web SunSolve. Vous y trouverez des patches correspondant à chaque plate-forme d'environnement d'exploitation Solaris. Chaque série de patches s'applique à toutes les versions de l'édition standard Java 2 (J2SE) supportées sur la plate-forme en question.

Consultez aussi le document Solaris Font Package Requirements pour plus d'informations sur les polices à utiliser sur votre système.

Bases de données consolidées pour synchronisation MobiLink

MobiLink supporte les bases de données consolidées suivantes :

Serveur de base de données	Version
Adaptive Server Anywhere	courante
Sybase Adaptive Server Enterprise	11.5 et version supérieure
Oracle	8i et 9i
Microsoft SQL Server	7 et 2000
IBM DB2	7.1 UDB

Références

Un pilote ODBC est requis pour connecter le serveur de synchronisation MobiLink à une base de données consolidée. iAnywhere Solutions propose une sélection de pilotes ODBC décrits à la section “[Pilotes ODBC iAnywhere Solutions - plates-formes supportées](#)” à la page 51. Certains fournisseurs de base de données possèdent leurs propres pilotes ODBC.

Pour obtenir une liste des pilotes ODBC et leur état, consultez le lien *Recommended ODBC Drivers for MobiLink* à l’adresse : http://www.iAnywhere.com/developer/technotes/odbc_mobilink.html.

Pilotes ODBC iAnywhere Solutions - plates-formes supportées

Les pilotes ODBC iAnywhere Solutions fournissent un accès ODBC à des bases de données tierces pour le serveur de synchronisation MobiLink et l'accès aux données distantes Adaptive Server Anywhere.

Le pilote ODBC Adaptive Server Anywhere n'est pas répertorié dans cette section. Pour plus d'informations sur le pilote ODBC Adaptive Server Anywhere, reportez-vous à la section "[Systèmes d'exploitation Windows et NetWare](#)" à la page 40.

Les pilotes ODBC iAnywhere Solutions sont disponibles pour les associations de systèmes d'opération et de bases de données suivantes :

Système d'exploitation	Base de données
Windows (32 bits)	Sybase Adaptive Server Enterprise version 11.5 ou supérieure Oracle 8i et 9i IBM DB2 UDB7.1
Solaris 7.0, 8.0 et 9.0	Sybase Adaptive Server Enterprise version 11.5 ou supérieure Oracle 8i et 9i
Solaris 2.6	Sybase Adaptive Server Enterprise version 11.5 ou supérieure
IBM AIX	Sybase Adaptive Server Enterprise version 11.5 ou supérieure Oracle 8i et 9i IBM DB2 UDB7.1
Linux ¹⁶	Sybase Adaptive Server Enterprise version 11.5 ou supérieure Oracle 8i et 9i

¹⁶Distributions Red Hat, SuSE et Caldera uniquement.

Plates-formes cibles UltraLite

Le tableau suivant répertorie les systèmes d'exploitation supportés pour le déploiement UltraLite, y compris les flux de synchronisation.

☞ Pour obtenir la liste des plates-formes supportées, reportez-vous à la section “[Plates-formes de développement UltraLite](#)” à la page 41.

☞ Pour plus d'informations sur les titres de colonne, reportez-vous à la section “[Versions des systèmes d'exploitation](#)” à la page 54.

Composant	Win-dows CE	Palm	Windows XP
API C/C++ statique UltraLite	✓	✓	✓
Embedded SQL UltraLite	✓	✓	✓
API Java statique UltraLite ¹⁷	✓		✓
UltraLite pour AppForge MobileVB	✓	✓	✓
UltraLite ActiveX	✓		✓
UltraLite .NET	✓ ¹⁸		✓ ¹⁹
Native UltraLite pour Java	✓ ²⁰		✓
Composant C++ UltraLite	✓	✓	✓
UltraLite pour M-Business Anywhere	✓ ²¹	✓ ²²	
Synchronisation TCP/IP	✓	✓	✓
Synchronisation HTTP	✓	✓	✓
Synchronisation HTTPS	✓	✓	✓
Synchronisation HotSync		✓	

¹⁷Nécessite Personal Java JDK 1.2.

¹⁸Nécessite .NET Compact Framework version 1.0.3705 ou supérieure.

¹⁹Nécessite .NET Compact Framework version 1.0.5000 ou supérieure.

²⁰Le déploiement sur un périphérique CE/ARM nécessite une machine virtuelle Java sur le device. Native UltraLite pour Java devrait fonctionner avec toutes les machines virtuelles PersonalJava 1.2 (ou supérieures). Toutefois, seules les machines virtuelles Jeode PersonalJava et CrEme (version 3.24 ou supérieure) ont été testées.

²¹Nécessite Pocket PC sur un processeur ARM

²²Nécessite Palm OS 5

Composant	Win- dows CE	Palm	Windows XP
Synchronisation ActiveSync (3.5 et 3.6)	✓ ^{23, 24}		
Sécurité de la couche de transport sur HTTP ou TCP/IP ²⁵	✓ ²⁶	✓ ²⁷	✓

Remarque

La configuration système d'UltraLite en matière de mémoire dynamique signifie qu'il est possible que les périphériques équipés de Palm OS 3.5 ou dotés de moins de 4 Mo de mémoire ne puissent fonctionner qu'avec de très petits schémas de base de données.

²³Non supporté par les composants UltraLite à l'exception de Native UltraLite pour Java et UltraLite.NET

²⁴Non supporté sur Smartphone 2002

²⁵Non supporté par les composants UltraLite L'API Java statique requiert JDK 1.2.2 ou supérieur. JDK 1.4 n'est pas supporté.

²⁶Pocket PC requis

²⁷Version 3.5 ou supérieure.

Versions des systèmes d'exploitation

Sauf indication contraire, les noms de système d'exploitation correspondent aux versions suivantes :

Système d'exploitation	9.0.0	9.0.1
Compaq Tru64	4.0D ou 5.1 sur des processeurs Alpha	Inchangé depuis la 9.0.0

Système d'exploitation	9.0.0	9.0.1
Linux	<p>Les distributions Linux sont toutes basées sur des versions définies du noyau, la bibliothèque C (glibc) et la bibliothèque ncurses. Les versions suivantes sont supportées :</p> <ul style="list-style-type: none"> ◆ noyau 2.2.12 sur SMP et UP, 2.4.18 sur SMP ou 2.4.2 sur UP ◆ glibc 2.1.2, 2.1.3, ou 2.2 ◆ ncurses 4.2 <p>Les distributions Linux suivantes font partie de celles qui remplissent toutes les conditions précédemment citées.</p> <ul style="list-style-type: none"> ◆ Red Hat 6.1, 6.2, 7.0, 7.1, 7.2, 7.3 ou Advanced Server 2.1. ◆ SuSE 6.3, 6.4, 7.0, 7.2, 7.3, ou 8.0 (mais pas 7.1) ◆ Caldera 2.4 ou 3.e ◆ TurboLinux 6.1, 6.5 ou 7.0 ◆ Mandrake 7.2, 8.0, ou 8.1 <p>Les distributions Linux sont supportées sur les processeurs à architecture x86.</p> <p>La version de déploiement 64 bits est disponible pour Red Hat Advanced Server sur des processeurs Itanium 2.</p>	Inchangé depuis la 9.0.0
Hewlett Packard HP-UX	11.0 ou 11i, version 32 bits sur des processeurs PA-RISC HP-UX 11i est aussi supporté dans une version 64 bits sur des processeurs Itanium 2.	Inchangé depuis la 9.0.0

Système d'exploitation	9.0.0	9.0.1
IBM AIX	4.3.1 ²⁸ , 4.3.2, 4.3.3, 5.1 ou 5.2 sur des processeurs Power PC	Inchangé depuis la 9.0.0
Mac OS X	Mac OS X version 10.2.n, où n est supérieur ou égal à 2	Inchangé depuis la 9.0.0
Microsoft Windows (32 bits)	<p>Les versions suivantes sont supportées :</p> <ul style="list-style-type: none"> ◆ Windows NT 4.0 ou version supérieure ◆ Windows 2000 ◆ Windows XP, y compris Windows XP Embedded et Windows XP Tablet PC sur des processeurs à architecture x86. ◆ Vous trouverez la liste des composants requis pour Windows XP Embedded à l'adresse Internet : http://www.sybase.com/detail?id=1017429. ◆ Windows Server 2003, versions 32 bits ◆ Windows 95/98/Me 	Inchangé depuis la 9.0.0
Microsoft Windows (64 bits)	Windows Server 2003, versions 64 bits	Inchangé depuis la 9.0.0

²⁸Le patch 1X81739 est recommandé.

Système d'exploitation	9.0.0	9.0.1
Microsoft Windows CE	<p>Windows CE 3.0 exécuté sur tout processeur suivant :</p> <ul style="list-style-type: none"> ◆ Processeur MIPS ◆ Processeur ARM, y compris la série XScale ◆ Processeur x86 UltraLite <p>UltraLite supporte le processeur x86 pour l'émulateur Pocket PC 2002; Adaptive Server Anywhere supporte les périphériques et l'émulation x86.</p> <p>L'émulateur Windows CE est également supporté à des fins de développement.</p> <p>La version 3.0 de Windows CE supporte Pocket PC, notamment Pocket PC 2002, ainsi que Handheld PC.</p> <p>Windows CE 4.1 sur le processeur XScale (Intel PXA255) en mode ARMV4.</p> <p>Pocket PC 2003 et autres versions de Windows CE 4.2 sur le processeur XScale (Intel PXA255) en mode ARMV4.</p>	<p>Windows CE 4.1 qui s'exécute en mode "Thumb" sur le processeur ARM (ARMV4T) est supporté à partir de la version 9.0.1.</p> <p>UltraLite supporte la plate-forme Smartphone 2002 outre les autres plates-formes Windows CE.</p> <p>L'émulateur de Pocket PC 2000 n'est plus supporté.</p>
Novell NetWare	5.1 avec service pack 6 et 6.0 avec service pack 3 et 6.5.	Inchangé depuis la 9.0.0
Palm OS	Plates-formes Palm Computing exécutant Palm OS version 3.5 ou supérieure.	Inchangé depuis la 9.0.0
Sun Solaris	7.0 ²⁹ systèmes d'exploitation 8.0 ou 9.0, 32 ou 64 bits exécutés sur des processeurs SPARC	Inchangé depuis la 9.0.0

²⁹Les patches 106541-14 et 106980-13 sont recommandés pour résoudre les problèmes SIGIO notés.



TROISIÈME PARTIE

UTILISATION DES BASES DE DONNÉES

Cette partie présente le modèle de base de données relationnel et les principes élémentaires relatifs à la manipulation des données

CHAPITRE 5

Architecture des applications de base de données

Présentation Cette section présente certains termes et concepts importants dans le domaine des bases de données relationnelles.

Sommaire	Sujet :	page
	Principes fondamentaux des bases de données relationnelles	62
	Éléments d'un système de base de données	67
	Interaction entre les différents éléments	70
	Architecture informatique multiniveau	73
	Utilisation de plusieurs bases de données	74
	Interfaces de programmation d'application	76
	Dans Adaptive Server Anywhere	81

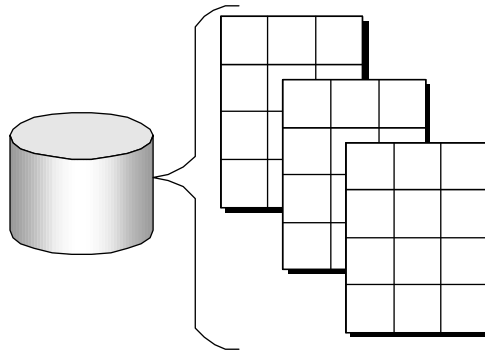
Principes fondamentaux des bases de données relationnelles

Un **système de gestion de base de données relationnelle** (SGBDR) est un système permettant de stocker et d'extraire des données, lesquelles sont structurées en tables liées entre elles.

SQL Anywhere Studio fournit deux systèmes de base de données relationnels. **Adaptive Server Anywhere** est le premier système de base de données relationnel complet offrant une multitude d'applications, depuis un serveur de base de données réseau hébergeant de nombreux clients jusqu'à une base de données compacte intégrée. UltraLite est une base de données relationnelle à empreinte mémoire réduite. La technologie de déploiement **UltraLite** permet d'utiliser les fonctionnalités d'Adaptive Server Anywhere même sur les périphériques les plus compacts.

Tables de base de données

Dans une base de données relationnelle, toutes les données sont contenues dans des **tables**, elles-mêmes constituées de **lignes** et de **colonnes**.



Chaque table contient une ou plusieurs colonnes et chaque colonne est associée à un type de données, par exemple entier, caractères (pour du texte) ou date. Chaque ligne de la table possède dans chaque colonne une valeur unique.

Par exemple, une table contenant des informations sur des employés se présente comme suit :

emp_ID	emp_lname	emp_fname	emp_phone
10057	Huong	Zhang	1096
10693	Donaldson	Anne	7821

Caractéristiques des tables relationnelles

Voici quelques caractéristiques importantes des tables d’une base de données relationnelle :

- ◆ L’ordre des colonnes et des lignes n’a pas d’importance.
- ◆ Chaque ligne contient une et une seule valeur par colonne ou bien NULL, indiquant que la colonne n’inclut aucune valeur.
- ◆ Toutes les valeurs d’une colonne sont du même type de données.

Le tableau suivant présente certains termes formels et informels spécifiques des bases relationnelles pour décrire les tables et leur contenu, ainsi que les termes équivalents dans les bases non relationnelles. Ce manuel utilise les termes informels.

Terme relationnel informel	Terme relationnel formel	Terme non relationnel
Table	Relation	Fichier
Colonne	Attribut	Champ
Ligne	Nuplet	Enregistrement

Contenu des tables

Chaque table de la base doit contenir des informations sur un thème donné, tels que les employés, les produits ou les clients.

En procédant ainsi, vous pouvez construire une structure qui élimine les redondances et les possibles incohérences générées par ces dernières. Par exemple, le service des ventes et le service comptable peuvent tous deux entrer et consulter des données relatives aux clients. Dans une base de données relationnelle, les informations sur les clients ne sont entrées qu’une fois, dans une table accessible par les deux services.

☞ Pour plus d’informations sur la conception d’une base de données, reportez-vous à la section “[Conception d’une base de données](#)” [*ASA - Guide de l’utilisateur SQL*, page 3].

Relations entre les tables

Pour décrire les relations entre les tables, vous disposez des clés primaires et étrangères. Des **clés primaires** identifient chaque ligne de la table de manière unique et des **clés étrangères** définissent les relations entre les lignes des différentes tables.

Les clés primaires et étrangères des bases relationnelles vous permettent de stocker des informations efficacement, notamment sans redondance.

Les tables sont dotées d'une clé primaire

Chaque table d'une base relationnelle doit contenir une **clé primaire**. Il s'agit d'une colonne ou d'un ensemble de colonnes qui identifie chaque ligne de façon unique. Deux lignes d'une table ne peuvent pas avoir la même valeur de clé primaire.

Exemples

Dans la base de données exemple, la table `employee` contient des informations sur les employés. Elle dispose d'une clé primaire nommée `emp_id` qui contient un numéro d'identification (ID) unique attribué à chaque employé. La définition d'une colonne contenant un ID unique est la manière la plus courante d'attribuer des clés primaires. En outre, elle présente l'avantage d'éviter toute ambiguïté, contrairement aux noms et autres identificateurs qui risquent de ne pas être uniques.

Vous pouvez voir une clé primaire plus complexe dans la table `sales_order_items` de la base de données exemple. La table contient des informations sur les articles individuels des commandes de l'entreprise et elle inclut les colonnes suivantes :

- ◆ **id** Numéro identifiant la commande à laquelle appartient l'article.
- ◆ **line_id** Numéro de ligne identifiant chaque ligne (ou article) d'une commande.
- ◆ **prod_id** ID de produit identifiant le produit commandé.
- ◆ **quantity** Quantité, indiquant le nombre d'articles commandés.
- ◆ **ship_date** Date de livraison, indiquant la date à laquelle l'article a été livré.

Un article de commande spécifique est identifié par la commande dans laquelle il figure et par son numéro de ligne dans cette commande. Ces deux numéros sont stockés dans les colonnes `id` et `line_id`. Les articles peuvent avoir une valeur d'`id` commune (correspondant à une commande de plusieurs articles) ou une valeur de `line_id` commune (tous les premiers articles de commandes différentes ont un `line_id` égal à 1). Mais il n'y a pas deux articles qui aient en commun les deux valeurs, de sorte que la clé primaire est formée de ces deux colonnes.

Tables liées par des clés étrangères

Les relations entre les informations de deux tables sont établies à l'aide de **clés étrangères**.

Exemple

La base de données exemple intègre une table contenant les données relatives aux employés et une autre contenant des informations sur chaque service. La table `department` contient les colonnes suivantes :

- ◆ **dept_id** Numéro d'identification du service. Il s'agit de la clé primaire de la table.
- ◆ **dept_name** Nom du département
- ◆ **dept_head_id** ID d'employé du chef de service

Pour savoir à quel service appartient un employé, il n'est pas nécessaire d'entrer le nom du service dans la table `employee`. La table `employee` contient en fait une colonne avec un numéro qui correspond à l'une des valeurs de `dept_id` de la colonne `department`.

La colonne `dept_id` de la table `employee` est appelée clé étrangère par rapport à la table `department`. Une clé étrangère référence une ligne particulière dans une table contenant la clé primaire correspondante.

Dans cet exemple, la table `employee` (qui contient la clé étrangère de la relation) est appelée **table étrangère** ou **table qui référence**. La table `department` (qui contient la clé primaire référencée) est appelée **table primaire** ou **table référencée**.

Autres objets de base de données

Une base de données relationnelle ne se limite pas à un ensemble de tables liées. Elle comporte également les objets suivants :

- ◆ **Index** Les index permettent de retrouver rapidement des informations. Fondamentalement, un index de base de données est conçu sur le même modèle qu'un index de livre. Dans un livre, l'index associe chaque terme référencé à la ou aux pages auxquelles le mot apparaît. Dans une base de données, l'index associe la valeur de chaque colonne indexée à l'emplacement physique où se trouve la ligne de données contenant la valeur indexée.

Les index sont essentiels à la conception d'une base, pour des raisons de performances. Vous devez généralement créer les index vous-même, mais les index concernant les clés primaires et étrangères sont automatiquement générés. Une fois les index créés, leur utilisation est très simple.

- ◆ **Vues** Les vues sont des tables calculées ou virtuelles. Vis-à-vis des applications clientes, elles apparaissent comme des tables, mais ne contiennent pas de données. Dès qu'elles font l'objet d'un accès, les

informations qu'elles contiennent sont calculées à partir des tables sous-jacentes.

Les tables qui contiennent réellement les informations sont parfois appelées **tables sous-jacentes** pour les distinguer des vues. Une vue est définie avec une requête SQL sur des tables sous-jacentes ou d'autres vues.

- ◆ **Procédures stockées et triggers** Il s'agit de programmes inclus dans la base de données, qui permettent d'en manipuler les informations.

Vous pouvez créer et nommer vos propres procédures stockées pour exécuter des requêtes spécifiques sur la base de données ou d'autres tâches. Les procédures stockées acceptent des paramètres. Vous pouvez, par exemple, créer une procédure stockée qui renvoie le nom de tous les clients dont les commandes dépassent un certain montant, montant spécifié comme paramètre dans l'appel de procédure.

Un trigger est une procédure stockée particulière qui, selon son paramétrage, se déclenche dès qu'un utilisateur met à jour, supprime ou insère des données. Vous pouvez associer un trigger à une table ou aux colonnes d'une table. Les triggers sont particulièrement utiles pour maintenir automatiquement les règles de travail dans une base de données.

- ◆ **Utilisateurs et groupes** Chaque utilisateur d'une base de données est affecté d'un ID utilisateur et d'un mot de passe. Vous pouvez octroyer des droits à chaque utilisateur, pour préserver la confidentialité de certaines informations et prévenir toute modification intempestive des données. Il est possible d'affecter des utilisateurs à des groupes afin de faciliter la gestion des autorisations.

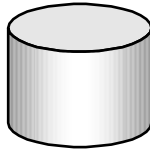
- ◆ **Objets Java** Vous pouvez installer des classes Java dans la base de données. Les classes Java constituent un puissant outil pour concevoir la logique dans la base de données et une classe particulière de types de données définis par l'utilisateur pour conserver les informations.

L'ensemble de ces objets constitue un **système de gestion de base de données relationnelle** (SGBDR). Il s'agit d'un système permettant de stocker et d'extraire des données, lesquelles sont structurées en tables liées entre elles.

Eléments d'un système de base de données

Les systèmes de gestion de bases de données relationnelle comportent les éléments suivants :

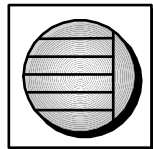
Une base de données Les données sont stockées dans une base de données. Dans les schémas de cette documentation, une base de données est représentée par un cylindre :



Une base de données Adaptive Server Anywhere est un fichier, portant généralement l'extension *.db*. Adaptive Server Anywhere propose une base de données exemple sur laquelle vous pouvez travailler : il s'agit du fichier *asademo.db* qui réside dans le répertoire d'installation d'Adaptive Server Anywhere.

Un serveur de base de données Le serveur de base de données gère la base de données. Aucune application ne s'adresse directement au fichier de la base ; elles passent toutes par le serveur.

Dans la documentation, un serveur de base de données est représenté comme suit :



Adaptive Server Anywhere propose deux versions de son serveur de base de données : le **serveur de base de données personnel** et le **serveur de base de données réseau**. Outre les fonctionnalités du serveur personnel, le serveur réseau supporte les communications client/serveur sur un réseau ; le serveur personnel n'accepte que les connexions provenant d'applications qui s'exécutent sur la même machine. Le moteur de traitement des requêtes est identique sur les deux serveurs.

Une interface de programmation Les applications communiquent avec les serveurs de base de données via une interface de programmation. Vous pouvez utiliser ODBC, JDBC, OLE DB, Sybase Open Client ou Embedded SQL.

Bon nombre d'outils de développement d'applications proposent leur

environnement de programmation qui masque les détails de l'interface sous-jacente. Par exemple, si vous développez une application avec Sybase PowerBuilder, vous n'avez jamais besoin d'appeler des fonctions ODBC. Néanmoins, chacun de ces outils fait appel à une des interfaces de programmation.

L'interface de programmation fournit une bibliothèque d'appels de fonctions permettant de communiquer avec la base. Pour ODBC et JDBC, la bibliothèque est généralement appelée **pilote**. Sous UNIX, elle se présente en général sous la forme d'une bibliothèque partagée ou, sur des PC, sous la forme d'une DLL (bibliothèque de liaison dynamique). L'interface JDBC exploite le pilote Sybase jConnect, lequel est un fichier zip de classes Java compilées.

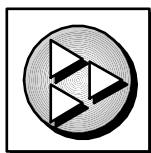
Dans les schémas de cette documentation, une interface de programmation est représentée comme suit :



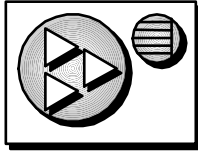
Une application cliente Les applications clientes utilisent l'une des interfaces de programmation pour communiquer avec le serveur de base de données.

Si vous développez une application au moyen d'un outil de développement rapide (RAD), tel que Sybase PowerBuilder, vous constaterez que l'outil dispose de ses propres méthodes pour communiquer avec les serveurs de base de données et que tous les détails de l'interface de langue sont masqués. Dans tous les cas néanmoins, toutes les applications exploitent l'une des interfaces prises en charge.

Dans les schémas de cette documentation, une application cliente est représentée comme suit :



Les serveurs de base de données UltraLite sont personnalisés pour chaque application UltraLite et font partie de l'application elle-même. Une application UltraLite et son serveur de base de données sont représentés comme suit :



Interaction entre les différents éléments

Les applications de base de données peuvent se connecter à un serveur situé sur la même machine ou, dans le cas d'un serveur réseau, sur une machine différente. De plus, avec Adaptive Server Anywhere, vous pouvez créer des bases distribuées, c'est-à-dire des bases physiquement distinctes sur des machines différentes qui partagent les données.

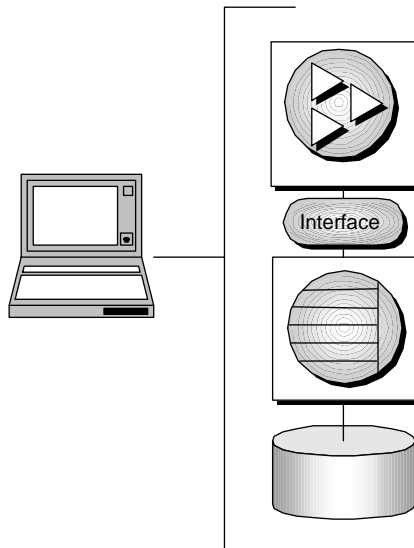
Applications personnelles et bases de données embarquées

Vous pouvez utiliser Adaptive Server Anywhere pour intégrer une application et une base de données complètes sur un seul ordinateur. Dans sa forme la plus simple, il s'agit d'une **application autonome** ou **application personnelle** : elle est indépendante, sans connexion aux autres bases de données. Dans ce cas, le serveur et la base de données peuvent être démarrés par l'application cliente et il est courant de parler de **base de données embarquée**. Du point de vue de l'utilisateur final, la base de données fait partie de l'application. Lorsqu'un serveur de base de données est utilisé comme base embarquée, il est quelquefois appelé **moteur de base de données**.

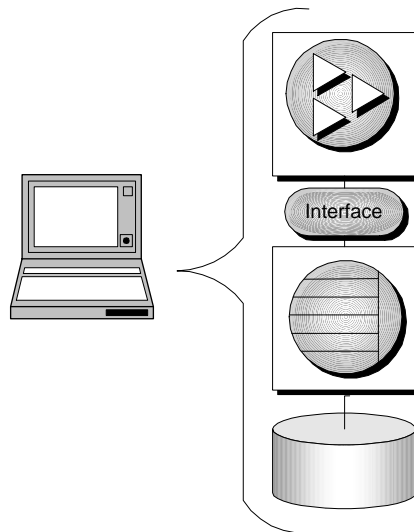
De nombreux systèmes de gestion de base de données relationnelle doivent être administrés par un personnel qualifié. Les bases de données embarquées possèdent la particularité de pouvoir être entièrement exécutées sans aucune administration spécifique.

Le serveur de base de données personnel Adaptive Server Anywhere est généralement utilisé pour les applications autonomes. Une application cliente se connecte via une interface de programmation à un serveur de base de données s'exécutant sur la même machine :

Les applications autonomes ont l'architecture suivante, avec une application cliente qui se connecte via une interface de programmation à un serveur de base de données qui est exécuté sur la même machine.



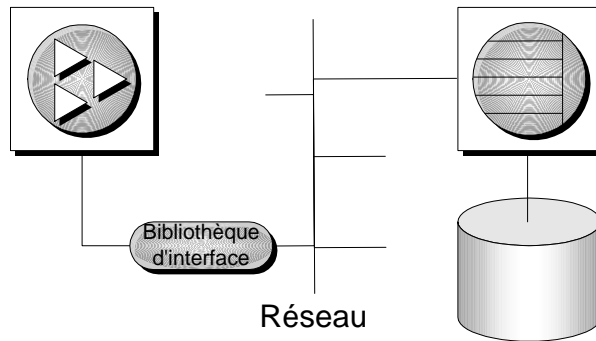
Le serveur de base de données personnel Adaptive Server Anywhere est généralement utilisé avec les applications autonomes, bien qu'il soit également possible d'utiliser des applications sur la même machine que le serveur réseau.



Applications client/serveur et bases de données multi-utilisateur

Avec Adaptive Server Anywhere, vous pouvez mettre en place une installation avec plusieurs applications exécutées sur des machines

différentes et connectées en réseau à un serveur de base de données résidant sur une machine distincte. Il s'agit d'un environnement **client/serveur** ou **base de données multi-utilisateur** avec l'architecture présentée ci-dessous. La bibliothèque d'interface se trouve sur chaque machine cliente.



Dans ce cas, le serveur de base de données est le serveur de base de données réseau Adaptive Server Anywhere, lequel prend en charge les communications réseau.

Pour qu'une application cliente fonctionne dans un environnement client/serveur, vous devez uniquement identifier le serveur auquel elle doit se connecter.

Pour qu'une application cliente fonctionne dans un environnement client/serveur, vous devez uniquement identifier le serveur auquel elle doit se connecter.

Architecture informatique multiniveau

Dans l'informatique multiniveau, la logique de l'application est conservée sur un serveur d'application, tel que Sybase EAServer, qui fait le lien entre le serveur de base de données et les applications clientes. Dans de nombreuses situations, un seul serveur d'application peut accéder à plusieurs bases de données en plus des enregistrements de données non relationnelles. Dans le cas d'Internet, les applications clientes sont basées sur un navigateur et le serveur applicatif est généralement une extension du serveur Web.

Sybase EAServer stocke la logique de l'application sous forme de composants, qu'elle met à la disposition des applications clientes. Il peut s'agir de composants PowerBuilder, de beans Java ou de composants COM.

Les serveurs d'application peuvent eux aussi fournir la logique de transaction à leurs applications clientes, ce qui garantit que des ensembles d'opérations sont exécutés de façon atomique sur plusieurs bases de données. Adaptive Server Anywhere est tout à fait adapté à l'informatique multiniveau et peut participer aux transactions distribuées coordonnées par Microsoft Distributed Transaction Coordinator. Ce dernier est utilisé aussi bien par Sybase Enterprise Application Server que par Microsoft Transaction Server pour fournir des services de transaction aux applications clientes.

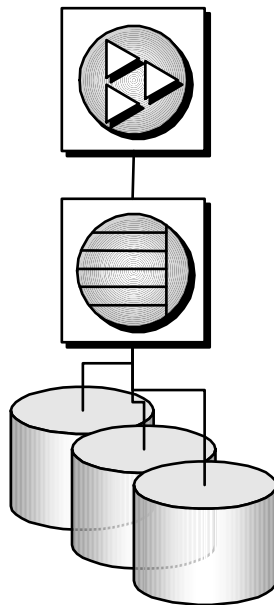
☞ Pour plus d'informations, reportez-vous à la section "[Transactions distribuées et architecture à trois niveaux](#)" [*ASA - Guide de programmation*, page 509].

Utilisation de plusieurs bases de données

Cette section décrit les extensions à l'architecture Adaptive Server Anywhere expliquée ci-dessus. Vous en aurez besoin si vous souhaitez utiliser plusieurs bases de données.

Exécution de plusieurs bases de données sur un serveur de base de données unique

Le serveur personnel Adaptive Server Anywhere et le serveur réseau peuvent gérer plusieurs bases de données simultanément. Chaque connexion doit être effectuée d'une application vers une base de données. En revanche, une application peut utiliser différentes connexions à différentes bases de données. Un ensemble d'applications peut également fonctionner sur différentes bases de données, via le même serveur de base de données.

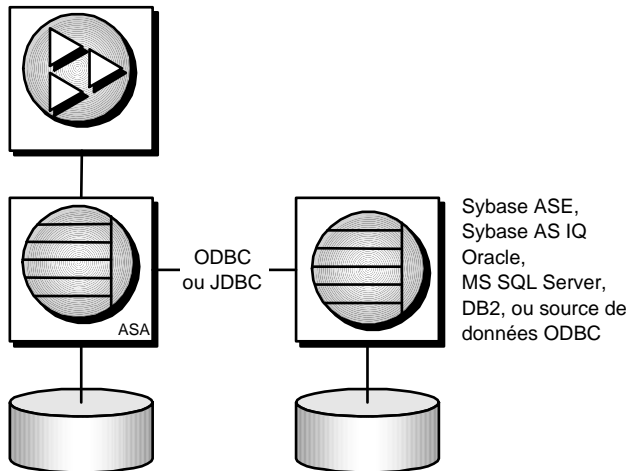


Les bases de données peuvent être lancées en même temps que le serveur de base de données ou lors de la connexion à une base de données à l'aide du paramètre de connexion DatabaseFile.

☞ Pour plus d'informations, reportez-vous aux scripts [“Le serveur de base de données”](#) [ASA - Guide d'administration, page 136] et [“Paramètre de connexion DatabaseFile \[DBF\]”](#) [ASA - Guide d'administration, page 206].

Accès aux données d'autres bases de données

Vous pouvez accéder à des bases de données se trouvant sur plusieurs serveurs, ou même sur un seul serveur, à l'aide des fonctions Adaptive Server Anywhere Remote Data Access. L'application est toujours connectée à une seule base de données comme indiqué dans les diagrammes d'architecture ci-dessus mais, en définissant des serveurs distants, vous pouvez utiliser des tables proxy qui se trouvent sur la base distante comme si elles se trouvaient dans la base à laquelle vous êtes connecté.



➡ Pour plus d'informations, reportez-vous à la section [“Accès aux données distantes”](#) [ASA - Guide de l'utilisateur SQL, page 637].

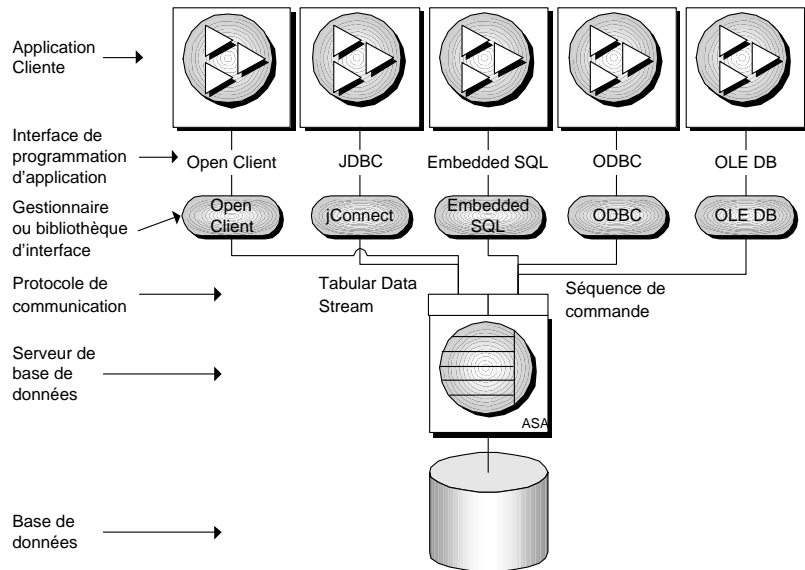
Interfaces de programmation d'application

Cette section décrit plus précisément cette architecture. L'architecture d'une application de base de données est présentée à la section "[Interaction entre les différents éléments](#)" à la page 70.

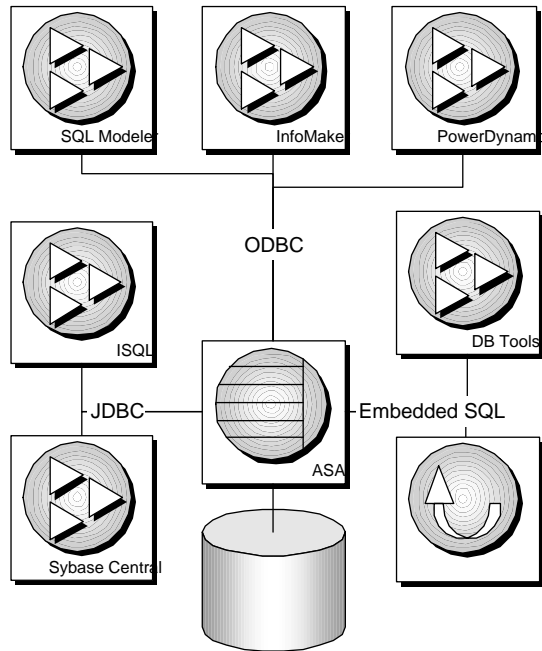
Adaptive Server Anywhere supporte un large éventail d'interfaces de programmation qui offre une grande souplesse de choix tant au niveau des types d'applications que des environnements de développement.

Interfaces et protocoles supportés

Le diagramme suivant indique les interfaces supportées, ainsi que les bibliothèques d'interface utilisées. La plupart du temps, la bibliothèque d'interface porte le même nom que l'interface elle-même.



Les applications fournies avec SQL Anywhere Studio utilisent plusieurs de ces interfaces :



Protocoles de communication

Chaque bibliothèque d'interface communique avec le serveur de base de données via un **protocole de communication**. Adaptive Server Anywhere supporte deux protocoles de communication, **Tabular Data Stream (TDS)** et **Command Sequence**. Ces protocoles sont internes et, le plus souvent, vous utiliserez indifféremment l'un ou l'autre. Le choix de l'environnement de développement est dicté par les outils dont vous disposez.

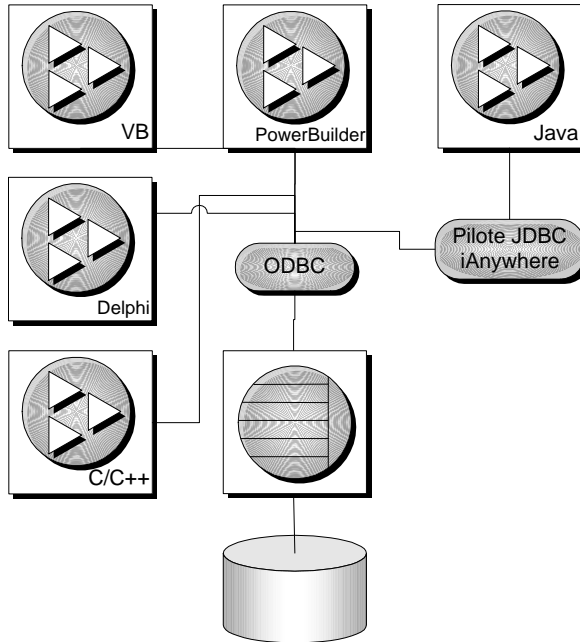
Les principales différences perceptibles concernent la connexion à la base de données. Les applications Command Sequence et TDS n'utilisent pas les mêmes méthodes pour identifier une base de données et un serveur de base de données ; les boîtes de dialogue de connexion sont donc différentes.

TDS Ce protocole est utilisé par Sybase Adaptive Server Enterprise, les applications Open Client et les applications Java qui se servent du pilote JDBC jConnect.

Command Sequence Ce protocole est utilisé par Adaptive Server Anywhere, Adaptive Server IQ et par les applications Embedded SQL, ODBC et OLE DB.

Applications ODBC

Vous pouvez développer des applications ODBC à l'aide de nombreux outils de développement et de langages de programmation, comme le montre la figure ci-dessous.

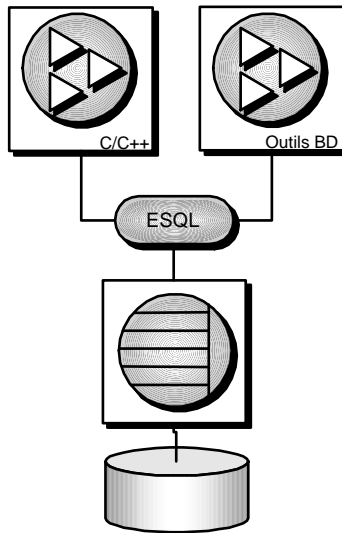


Par exemple, parmi les applications fournies avec SQL Anywhere Studio, InfoMaker et SQL Modeler utilisent ODBC pour se connecter à la base de données.

Applications Embedded SQL

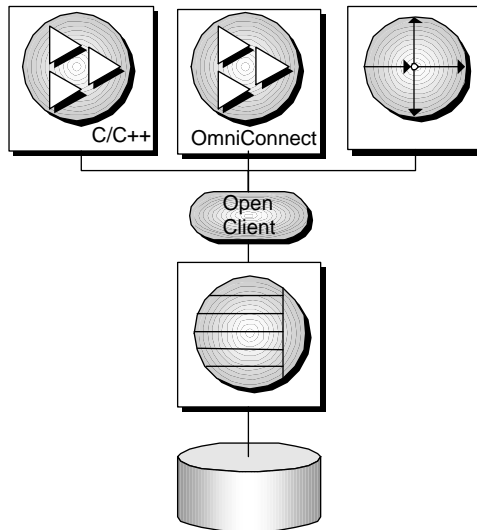
Vous pouvez développer des applications C ou C++ via l'interface de programmation Embedded SQL. Les outils de la base de données de la ligne de commande sont des exemples d'applications développées de cette manière.

Embedded SQL constitue également une interface de programmation pour les applications UltraLite.



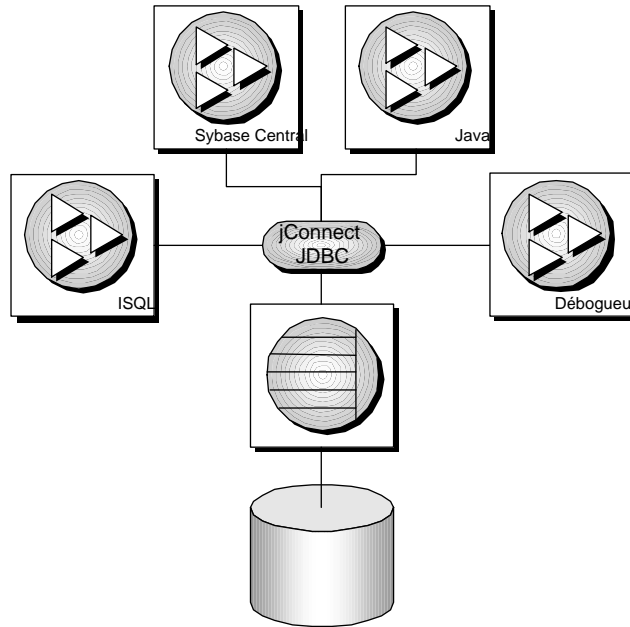
Applications Open Client

Open Client est une interface également supportée par Sybase Adaptive Server Enterprise. Vous pouvez développer des applications Open Client en C ou C++. D'autres applications Sybase, telles qu'OmniConnect ou Replication Server, utilisent Open Client.



Applications JDBC

Vous pouvez développer des applications qui utilisent JDBC pour vous connecter à Adaptive Server Anywhere via Java. Plusieurs des applications fournies avec SQL Anywhere Studio utilisent JDBC : le débogueur de procédure stockée, Sybase Central et Interactive SQL.



Java et JDBC sont également des langages de programmation importants pour développer des applications UltraLite.

Applications OLE DB

Adaptive Server Anywhere intègre un pilote OLE DB. Vous pouvez développer des applications directement via l'interface OLE DB de Microsoft ou via l'interface ActiveX Data Objects (ADO). L'interface ADO est intégrée à Visual Basic et à d'autres outils de programmation Microsoft. SQL Anywhere Studio inclut également un fournisseur de données ADO.NET pour Adaptive Server Anywhere.

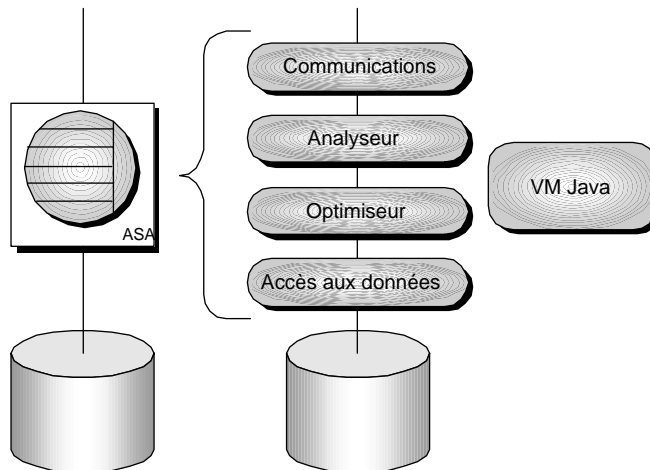
Dans Adaptive Server Anywhere

Vous n'aurez jamais à vous occuper des éléments internes au serveur de base de données, mais un coup d'oeil dans les coulisses ne peut que vous aider à mieux appréhender les processus.

Dans le serveur de base de données

La structure interne du serveur de base de données Adaptive Server Anywhere lui permet de gérer efficacement de nombreuses demandes.

- ◆ Une couche de communication gère les échanges de données avec les applications clientes. Cette couche reçoit les demandes des applications clientes et leur renvoie les résultats. La planification de ces actions est déterminée par une négociation entre le client et le serveur, pour garantir à la fois le maintien du trafic à un niveau minimal et la mise à disposition des données la plus rapide possible côté client.
- ◆ L'analyseur contrôle chaque instruction SQL envoyée au serveur de base de données et la convertit dans un format interne pour qu'elle puisse être traitée.
- ◆ Si la demande est une requête, ou une instruction de mise à jour ou de suppression, il peut y avoir différentes façons d'accéder aux données, de durées variables. La tâche de l'optimiseur est de sélectionner la route optimale pour l'acheminement rapide des données requises.
- ◆ Une machine virtuelle (VM) Java est créée dans le serveur de base de données et toute opération Java envoyée par les applications clientes ou appelée en interne par le serveur de base de données est gérée par la machine VM de Java.
- ◆ Le niveau inférieur du serveur de base de données gère la lecture et l'écriture des données à partir du disque, la mise en mémoire cache des données pour éviter tout accès inutile au disque et l'équilibrage des demandes des différents utilisateurs.



Dans la base de données

Toutes les informations d'une base de données Adaptive Server Anywhere sont enregistrées dans un seul fichier de base de données, lequel peut être copié d'un ordinateur vers un autre. Il est possible de créer des bases de données de plusieurs fichiers, mais cela n'est généralement nécessaire que pour des bases très volumineuses.

Outre le fichier de base de données, Adaptive Server Anywhere exploite deux autres fichiers lorsqu'il exécute une base de données. Il s'agit du journal de transactions et du fichier temporaire.

- ◆ **Le fichier de base de données** En interne, le fichier de base de données est composé de pages : des zones de taille fixe sur le disque. La couche d'accès aux données lit et écrit des données page par page. De nombreuses pages contiennent les données des tables de la base, d'autres contiennent des informations d'index, des informations sur la répartition des données au sein de la base etc.
- ◆ **Le journal de transactions** Le journal de transactions est un fichier distinct qui enregistre toutes les opérations effectuées sur la base de données. Il porte généralement le même nom que le fichier de base de données, à ceci près que son suffixe est *.log* et non pas *.db*. Il assure trois fonctions essentielles.
 - ◆ **Enregistrement des opérations sur les données pour permettre leur restauration** Vous pouvez recréer votre base de données à partir d'une sauvegarde et du journal de transactions si le fichier de base de données venait à être endommagé.
 - ◆ **Amélioration des performances** En consignnant des informations

dans le journal de transactions, le serveur de base de données peut traiter vos instructions en toute sécurité, sans écrire aussi souvent dans le fichier de base de données.

- ◆ **Réplication de la base de données** SQL Remote et l'utilitaire client MobiLink utilisent ce fichier pour répliquer les modifications apportées à votre base de données sur les ordinateurs portables occasionnellement connectés au réseau.

- ◆ **Le fichier temporaire** Le fichier temporaire est ouvert au démarrage du serveur et fermé lorsque celui-ci s'arrête. Comme son nom l'indique, le fichier temporaire sert à stocker des informations temporaires pendant que le serveur est actif. Il ne contient aucune information qui doit être maintenue d'une session à l'autre.

Le fichier temporaire réside dans votre répertoire temporaire. L'emplacement de ce répertoire est généralement indiqué dans la variable d'environnement TEMP.

CHAPITRE 6

Sélection de données dans des tables

Présentation

Ce chapitre présente le langage d'interrogation structuré SQL (Structured Query Language) et les requêtes, qui permettent de récupérer les données d'une base. Il décrit comment récupérer les données d'une seule table.

Pour aborder d'autres parties de la documentation, vous devez connaître les concepts présentés ici.

Sommaire

Sujet :	page
SQL et base de données	86
Instruction SELECT	89
Sélection d'une table complète	90
Sélection de colonnes dans une table	92
Tri du résultat d'une requête	95
Sélection de lignes dans une table	99

SQL et base de données

Pour qu'une application cliente effectue une tâche sur une base de données, par exemple extraire des informations à l'aide d'une requête ou insérer des lignes dans une table, elle doit utiliser des instructions **SQL (Structured Query Language)**. SQL est un langage de base de données relationnelle, normalisé ANSI et ISO.

Selon la manière dont vous développez une application cliente, vous pouvez fournir les instructions SQL dans des appels de fonction à partir du langage de programmation. Certains outils de développement offrent des interfaces utilisateur qui permettent de créer et de générer des instructions SQL.

L'interface de programmation transmet l'instruction SQL au serveur de base de données. Le serveur reçoit l'instruction et l'exécute, renvoyant les informations demandées (résultats des requêtes) à l'application.

Les protocoles de communication client/serveur transfèrent les données entre l'application cliente et le serveur de base de données, les interfaces de programmation, quant à elles, définissent la manière dont une application envoie les informations. Quels que soient l'interface et le protocole utilisés, des instructions SQL sont envoyées à un serveur et les résultats de ces instructions sont renvoyés à l'application cliente.

Exemple

L'instruction SQL suivante extrait le nom de tous les employés de la table `employee` de la base `exemple` :

```
SELECT emp_lname
FROM employee
```

Vous pouvez envoyer ce type de requête au serveur de base de données en utilisant Interactive SQL ou créer la requête dans votre propre application.

Exemple

L'instruction SQL suivante crée une table appelée `food` qui répertorie les types d'aliments et la quantité en stock dans la cafétéria des employés :

```
CREATE TABLE food (
    food_id integer primary key,
    food_type char(20),
    quantity integer
)
```

Requêtes

Dans SQL, "Q" signifie **requête** (query). Pour **extraire** les données d'une base, vous utilisez l'instruction `SELECT`. Dans un système relationnel, les opérations élémentaires effectuées via des requêtes sont les projections, les

restrictions et les jointures. L'instruction SELECT les met toutes en oeuvre.

Projections et restrictions Une **projection** est un sous-ensemble de colonnes d'une table. Une **restriction** (également appelée **sélection**) est un sous-ensemble de lignes d'une table, répondant à certaines conditions.

Par exemple, l'instruction SELECT suivante recherche le nom et le prix de tous les produits coûtant plus de 15 euros :

```
SELECT name, unit_price
FROM product
WHERE unit_price > 15
```

Cette requête utilise une projection (SELECT name, unit_price) et une restriction (WHERE unit_price > 15).

Jointures

Une jointure lie les lignes de deux ou plusieurs tables, en comparant les valeurs des colonnes de chaque table. Par exemple, pour sélectionner les numéros d'identification des bons de commande et les noms des produits de tous les bons comportant plus de 12 articles :

```
SELECT sales_order_items.id, product.name
FROM product JOIN sales_order_items
WHERE sales_order_items.quantity > 12
```

La table product et la table sales_order_items sont jointes sur la base d'une relation de clé étrangère.

Autres instructions SQL

SQL ne se limite pas à l'émission de requêtes. SQL contient des instructions permettant de créer des tables, des vues et d'autres objets de base de données. Il intègre également des instructions permettant de modifier les tables (instructions insert, update et delete) et d'exécuter nombre d'autres tâches de base de données, décrites dans ce manuel.

Tables système

Chaque base de données Adaptive Server Anywhere contient un ensemble de tables système. Ce sont des tables spéciales que le système utilise pour gérer les données et le système. Ces tables sont parfois appelées **référentiel de données** ou **catalogue du système**.

Les tables système contiennent des informations sur la base de données. Vous ne pouvez pas modifier directement les tables système comme vous le faites avec les autres tables. Les tables système contiennent des informations sur les tables d'une base de données, les utilisateurs de la base, les colonnes de chaque table etc. Il s'agit de données sur des données, c'est-à-dire de

métadonnées.

☞ Pour plus d'informations sur les tables système Adaptive Server Anywhere, reportez-vous à la section [“Tables système” \[ASA - Manuel de référence SQL, page 687\]](#).

Instruction SELECT

Toutes les interactions entre les applications (clients) et les serveurs de base de données sont effectuées via l'envoi des instructions SQL au serveur de base de données, qui renvoie les informations au client.

L'instruction SELECT extrait les informations d'une base de données afin qu'elles soient utilisées par l'application cliente. Les instructions SELECT sont également nommées **requêtes**. Les informations sont fournies au client sous forme d'un jeu de résultats. L'application cliente traite ensuite le jeu de résultats. Par exemple, Interactive SQL affiche le jeu de résultats dans le volet Résultats. Les jeux de résultats sont constitués d'un ensemble de lignes, tout comme les tables dans la base de données.

Les instructions SELECT peuvent devenir extrêmement complexes dans des applications recherchant des informations très spécifiques dans un grand nombre de bases de données. Ce chapitre n'utilise que des instructions SELECT simples : des requêtes plus évoluées sont décrites ultérieurement dans d'autres didacticiels.

☞ Pour plus d'informations sur la syntaxe, reportez-vous à la section "Instruction SELECT" [ASA - *Manuel de référence SQL*, page 609].

Remarques

L'idéal serait d'exécuter le logiciel Adaptive Server Anywhere au fur et à mesure que vous parcourez les exemples décrits dans ce chapitre.

Chaque exemple vous invite à saisir des commandes dans Interactive SQL et décrit les éléments qui s'affichent à l'écran. Si vous n'exécutez pas le logiciel parallèlement à l'étude des didacticiels, vous pourrez néanmoins étudier SQL, mais sans en faire l'expérience directe.

Pour suivre les exemples, vous devez avoir lancé Interactive SQL et être connecté à la base de données exemple.

Sélection d'une table complète

L'instruction `SELECT` la plus simple extrait toutes les données d'une base unique. La syntaxe de cette instruction `SELECT` est la suivante :

```
SELECT * FROM nom_table
```

où *nom_table* doit être remplacé par le nom de la table que vous interrogez. L'astérisque (*) est la forme abrégée d'une liste de toutes les colonnes.

❖ Pour répertorier tous les produits vendus par la société

1. Dans Interactive SQL, saisissez ce qui suit dans le volet Instructions SQL, puis appuyez sur F5 pour exécuter l'instruction.

```
SELECT * FROM Product
```

L'instruction `SELECT` extrait toutes les lignes et les colonnes de la table `product` et les affiche sur l'onglet Résultats dans le volet Résultats :

id	name	description	size	color	quantity	unit_price
300	Tee Shirt	Tank Top	Small	White	28	9
301	Tee Shirt	V-neck	Medium	Orange	54	14
302	Tee Shirt	Crew Neck	One size fits all	Black	75	14
400	Baseball cap	Cotton Cap	One size fits all	Black	112	9
...

La table `Product` contient sept colonnes. Chaque colonne a un nom, tel que `color` ou `id`. Il y a une ligne pour chaque produit vendu par la société et chaque ligne a une valeur unique dans chaque colonne. Par exemple, le produit avec l'ID 301 est un Tee Shirt. C'est un style col en V de taille moyenne et de couleur orange.

Remarques

◆ Pas de distinction majuscules/minuscules dans les noms de table

Le nom de table `Product` commence par un P majuscule, bien que le nom réel de la table soit en minuscules. Au moment de la création des bases Adaptive Server Anywhere, vous pouvez choisir d'appliquer ou pas la distinction majuscules/minuscules pour les comparaisons de chaînes, mais en ce qui concerne les identificateurs tels que les noms de table et de colonne, cette distinction n'est pas appliquée.

☞ Pour plus d'informations sur la création de bases de données, reportez-vous à la section [“Création d'une base de données”](#) [ASA - Guide

de l'utilisateur SQL, page 33] ou “Utilitaire de création d’une base de données” [ASA - Guide d’administration, page 537].

- ◆ **Pas de distinction majuscules/minuscules dans les mots-clés SQL**
Vous pouvez entrer indifféremment **select** ou **Select** au lieu de **SELECT**. Dans la documentation, les mots-clés SQL sont généralement en majuscules.
- ◆ **Les sauts de ligne n’ont pas d’importance** Vous pouvez saisir les instructions sur une seule ligne ou les scinder en appuyant sur ENTREE à la fin de chaque ligne. Certaines instructions SQL, telles que **SELECT**, sont composées de plusieurs éléments, appelés **clauses**. Dans de nombreux exemples, chaque clause est généralement placée sur une ligne distincte pour faciliter la lecture ; ainsi, l’instruction se présente comme suit dans la documentation :

```
SELECT *  
FROM product
```

- ◆ **L’ordre des lignes dans le jeu de résultats n’a aucune importance**
Les lignes renvoyées depuis la base de données se présentent dans n’importe quel ordre : cela ne signifie rien de particulier. Si vous souhaitez extraire les lignes dans un ordre donné, vous devez l’indiquer dans la requête.

Exercice

Interrogez d’autres tables dans la base de données exemple, telles que les tables `employee`, `customer`, `contact` ou `sales_order`.

Sélection de colonnes dans une table

Vous pouvez restreindre une instruction `SELECT` en indiquant les colonnes souhaitées immédiatement après le mot-clé `SELECT`. La syntaxe de cette instruction `SELECT` est la suivante :

```
SELECT nom_colonne_1, nom_colonne_2,...  
FROM nom_table
```

où vous devez remplacer *nom_colonne_1* et *nom_colonne_2* par le nom des colonnes de votre choix et *nom_table* par le nom de la table que vous interrogez.

La liste des colonnes du jeu de résultats se nomme la **liste de sélection**. Les noms de colonne sont séparés par des virgules. Il n'y a pas de virgule après la dernière colonne de la liste et si la liste ne contient qu'une colonne. Cette restriction ainsi appliquée aux colonnes est parfois nommée **projection**.

❖ Pour répertorier le nom, la description et la couleur de chaque produit

1. Dans Interactive SQL, saisissez ce qui suit dans le volet Instructions SQL, puis appuyez sur F5 pour exécuter l'instruction.

```
SELECT name, description, color  
FROM product
```

name	description	color
Tee Shirt	Tank Top	White
Tee Shirt	V-neck	Orange
Tee Shirt	Crew Neck	Black
Baseball cap	Cotton Cap	Black
...

Réagencement des colonnes

Les colonnes apparaissent dans l'ordre de leur saisie dans l'instruction `SELECT`. Pour les réorganiser, il suffit de changer l'ordre des noms de colonne dans l'instruction. Par exemple, pour obtenir la colonne **description** à gauche, utilisez l'instruction suivante :

```
SELECT description, name, color  
FROM product
```

Utilisation des colonnes calculées

Il arrive que les colonnes du jeu de résultats ne soient pas simplement les

colonnes d'une table. Il peut s'agir d'expressions calculées à partir des données sous-jacentes. Vous pouvez intégrer plusieurs colonnes de table dans une colonne unique du jeu de résultats et utiliser un grand nombre de fonctions et d'opérateurs pour contrôler les résultats obtenus.

❖ **Pour répertorier la valeur en stock de chaque produit**

1. Dans Interactive SQL, saisissez ce qui suit dans le volet Instructions SQL, puis appuyez sur F5 pour exécuter l'instruction.

```
SELECT id, ( unit_price * quantity ) AS value
FROM product
```

id	value
300	252
301	756
302	1050
400	1008
...	...

Remarques

- ◆ **Les colonnes peuvent être dotées d'un alias** Le nom d'une colonne est par défaut l'expression figurant dans la liste de sélection, mais pour les colonnes calculées, ce nom est trop long et peu explicite. Ici, le nom de la colonne calculée dans la liste de sélection devient value. value est l'**alias** de la colonne.
- ◆ **D'autres opérateurs sont disponibles** Dans l'exemple ci-dessus, l'opérateur de multiplication est utilisé pour combiner les colonnes. Vous pouvez utiliser d'autres opérateurs, notamment les opérateurs arithmétiques standard tels que les opérateurs logiques et les opérateurs de chaînes de caractères.

La requête suivante affiche le nom complet de tous les clients :

```
SELECT id, ( fname || ' ' || lname ) AS "Full name"
FROM customer
```

L'opérateur || concatène les chaînes. Dans cette requête, l'alias de la colonne contient des espaces et doit figurer entre guillemets. Cette règle ne s'applique pas uniquement aux alias de colonne, mais aussi aux noms de table et aux autres identificateurs de la base de données.

☞ Pour obtenir la liste complète des opérateurs, reportez-vous à la section "Opérateurs" [ASA - Manuel de référence SQL, page 12].

-
- ◆ **Des fonctions peuvent être utilisées** Outre la combinaison des colonnes, vous pouvez utiliser toute une gamme de fonctions intégrées afin de produire les résultats que vous souhaitez.

La requête suivante répertorie par exemple les noms de produits en majuscules :

```
SELECT id, UCASE( name )  
FROM product
```

id	UCASE(product.name)
300	TEE SHIRT
301	TEE SHIRT
302	TEE SHIRT
400	BASEBALL CAP
...	...

☞ Pour obtenir la liste complète des fonctions, reportez-vous à la section “Liste alphabétique des fonctions” [[ASA - Manuel de référence SQL](#), page 106].

Tri du résultat d'une requête

Sauf indication particulière, le serveur de base de données renvoie les lignes d'une table dans un ordre non significatif. Il est souvent plus pertinent de consulter les lignes d'une table selon une séquence plus significative. Par exemple, vous pouvez trier les produits par ordre alphabétique.

Pour trier les lignes dans un jeu de résultats, insérez une clause `ORDER BY` à la fin de l'instruction `SELECT`. La syntaxe de cette instruction `SELECT` est la suivante :

```
SELECT nom_colonne_1, nom_colonne_2,...
FROM nom_table
ORDER BY trier_par_nom_colonne
```

où *nom_colonne_1*, *nom_colonne_2* et *nom_table* doivent être remplacés par le nom des colonnes de votre choix et de la table que vous interrogez ; *trier_par_nom_colonne* est une colonne de la table. Vous pouvez utiliser l'astérisque comme forme abrégée désignant toutes les colonnes de la table.

❖ Pour obtenir la liste des produits par ordre alphabétique

1. Dans Interactive SQL, saisissez ce qui suit dans le volet Instructions SQL :

```
SELECT id, name, description
FROM product
ORDER BY name
```

id	name	description
400	Baseball cap	Cotton Cap
401	Baseball cap	Wool cap
700	Shorts	Cotton Shorts
600	Sweatshirt	Hooded Sweat-shirt
...

Remarques

- ◆ **L'ordre des clauses est important** La clause `ORDER BY` doit suivre la clause `FROM` et la clause `SELECT`.
- ◆ **Vous pouvez choisir un ordre croissant ou décroissant** L'ordre par défaut est croissant. Vous pouvez opter pour un ordre décroissant en ajoutant le mot-clé `DESC` à la fin de la clause, comme dans la requête qui suit :

```
SELECT id, quantity
FROM product
ORDER BY quantity DESC
```

id	quantity
400	112
700	80
302	75
301	54
600	39
...	...

- ◆ **Vous pouvez classer par plusieurs colonnes** La requête suivante trie d'abord par taille (alphabétiquement), puis par nom :

```
SELECT id, name, size
FROM product
ORDER BY size, name
```

id	name	size
600	Sweatshirt	Large
601	Sweatshirt	Large
700	Shorts	Medium
301	Tee Shirt	Medium
...

- ◆ **La colonne ORDER BY n'est pas nécessairement incluse dans la liste de sélection** La requête suivante trie les produits par prix unitaire, même si le prix n'est pas inclus dans le jeu de résultats.

```
SELECT id, name, size
FROM product
ORDER BY unit_price
```

id	name	size
500	Visor	One size fits all
501	Visor	One size fits all
300	Tee Shirt	Small
400	Baseball cap	One size fits all
...

- ◆ **Si vous n'utilisez pas une clause ORDER BY et que vous exécutez**

une requête plusieurs fois, vous risquez d'obtenir des résultats différents En effet, il est possible qu'Adaptive Server Anywhere renvoie le même jeu de résultats dans un ordre différent. En l'absence de clause ORDER BY, Adaptive Server Anywhere renvoie les lignes dans l'ordre qui est le plus efficace. Cela signifie que l'aspect des jeux de résultats peut varier en fonction du moment auquel vous avez accédé pour la dernière fois à la ligne et aux autres facteurs. La seule façon de vous assurer que les lignes sont renvoyées dans un ordre particulier consiste à utiliser ORDER BY.

Utilisation des index pour améliorer les performances de ORDER BY

Parfois, le serveur de base de données Adaptive Server Anywhere a plusieurs possibilités pour exécuter une requête avec une clause ORDER BY. Vous pouvez utiliser des index pour permettre au serveur d'explorer les tables de façon plus efficace.

Requêtes comportant des clauses WHERE et ORDER BY

Une requête contenant à la fois une clause WHERE et une clause ORDER BY est un exemple de requête exécutable de plusieurs manières :

```
SELECT *  
FROM customer  
WHERE id > 300  
ORDER BY company_name
```

Ici, le serveur a le choix entre deux stratégies :

1. Parcourir l'ensemble de la table customer dans l'ordre des noms de société, en vérifiant à chaque ligne si l'ID de client est supérieur à 300.
2. Utiliser la clé sur la colonne id pour n'afficher que les noms de société dont l'ID est supérieur à 300. Le résultat obtenu doit ensuite être trié par nom de société.

Si le nombre de valeurs d'ID supérieures à 300 est limité, la seconde stratégie est recommandée ; en effet, seul un petit nombre de lignes est balayé et donc rapidement trié. Si la plupart des valeurs d'ID sont supérieures à 300, la première stratégie est mieux adaptée car aucun tri n'est nécessaire.

Résolution du problème

Le problème ci-dessus peut être résolu en créant un index à deux colonnes sur id et company_name. Adaptive Server Anywhere peut alors utiliser cet index pour sélectionner des lignes dans la table dans l'ordre correct. Toutefois, n'oubliez pas que les index prennent de la place dans le fichier de la base de données et que leur mise à jour génère de l'overhead. Evitez donc de créer des index sans raison valable.

☞ Pour plus d'informations, reportez-vous à la section [“Utilisation d'index”](#) [ASA - *Guide de l'utilisateur SQL*, page 184].

Sélection de lignes dans une table

Vous pouvez limiter les lignes qu’une instruction `SELECT` récupère en ajoutant une clause `WHERE` à votre requête. On parle parfois d’appliquer une **restriction** au jeu de résultats. La clause `WHERE` inclut une **condition de recherche** définissant les lignes qui doivent être renvoyées. La syntaxe de cette instruction `SELECT` est la suivante :

```
SELECT nom_colonne_1, nom_colonne_2, ...
FROM nom_table
WHERE condition_recherche
```

où vous devez remplacer `nom_colonne_1` et `nom_colonne_2` par le nom des colonnes de votre choix et `nom_table` par le nom de la table que vous interrogez. La `condition_recherche` est décrite ci-après. Si vous utilisez une clause `ORDER BY`, elle doit être placée après la clause `WHERE`.

❖ Pour répertorier tous les produits de couleur noire

1. Dans Interactive SQL, saisissez ce qui suit dans le volet Instructions SQL :

```
SELECT *
FROM product
WHERE color = 'black'
```

id	name	description	size	color	quantity	unit_price
302	Tee Shirt	Crew Neck	One size fits all	Black	75	14
400	Baseball cap	Cotton Cap	One size fits all	Black	112	9
501	Visor	Plastic Visor	One size fits all	Black	28	7
...

Remarques

- ◆ La clause `WHERE` inclut une condition de recherche pour sélectionner les lignes. Dans ce cas, la condition de recherche est `color = 'black'`. D’autres conditions de recherche sont décrites dans les sections suivantes.
 - ☞ Pour plus d’informations sur les conditions de recherche, reportez-vous à la section “Conditions de recherche” [ASA - Manuel de référence SQL, page 24].
- ◆ Les guillemets simples autour de **black** sont requis. Ils indiquent que **black** est une chaîne de caractères. Les guillemets doubles ont une autre signification. Ils permettent de transformer des chaînes incorrectes en

chaînes correctes pour les utiliser comme noms de colonnes ou autres identificateurs.

☞ Pour plus d'informations sur les chaînes, reportez-vous à la section "Chaînes" [ASA - Manuel de référence SQL, page 10].

- ◆ La base de données exemple ne faisant pas la distinction majuscules/minuscules, vous obtenez le même résultat que vous recherchez **BLACK**, **black** ou **Black**.
- ◆ L'ordre des clauses est important. La liste SELECT est suivie de la clause FROM, suivie de WHERE, puis de ORDER BY. La saisie des clauses dans un ordre différent constitue une erreur de syntaxe.

Exercice

Créez des requêtes associant tout ce que vous avez appris dans ce chapitre. Voici une requête qui répertorie le nom et la date de naissance de tous les employés nommés John.

```
SELECT ( emp_fname || ' ' || emp_lname) AS Name,
       birth_date
FROM employee
WHERE emp_fname = 'John'
ORDER BY birth_date
```

Name	birth_date
John Letiecq	4/27/1939
John Sheffield	9/25/1955

Comparaison des dates dans les conditions de recherche

Vous pouvez utiliser des opérateurs autres que le signe égal pour sélectionner un ensemble de lignes qui répondent à la condition de recherche. Les opérateurs de non-égalité (<et >) permettent de comparer des nombres, des dates et même des chaînes de caractères.

❖ **Pour répertorier tous les employés nés avant le 13 mars 1964**

1. Dans Interactive SQL, saisissez ce qui suit dans le volet

Instructions SQL :

```
SELECT emp_lname, birth_date
FROM employee
WHERE birth_date < 'March 13, 1964'
ORDER BY birth_date DESC
```

emp_lname	birth_date
Ahmed	12/12/1963
Dill	7/19/1963
Rebeiro	4/12/1963
Garcia	1/23/1963
Pastor	7/14/1962
...	...

Remarques

- ◆ **Conversion automatique en dates** Le serveur de base de données Adaptive Server Anywhere sait que la colonne birth_date contient des dates et convertit automatiquement la chaîne '13 mars 1964' en date.
- ◆ **Formats de dates** Il existe plusieurs façons d'indiquer des dates. Les formats suivants sont tous acceptés par Adaptive Server Anywhere :

```
'March 13, 1964'
'1964/03/13'
'1964-03-13'
```

Vous pouvez définir l'interprétation des dates dans les requêtes en utilisant une option de base de données. Les dates écrites au format *aaaa/mm/jj* ou *aaaa-mm-jj* sont toujours reconnues sans équivoque comme des dates, quelle que soit la valeur de l'option DATE_ORDER.

☞ Pour obtenir des informations sur le contrôle des formats de date reconnus dans les requêtes, reportez-vous aux sections “[Option DATE_ORDER \[compatibilité\]](#)” [ASA - Guide d'administration, page 660] et “[Paramétrage](#)” [ASA - Guide d'administration, page 626].

- ◆ **Autres opérateurs de comparaison** Pour obtenir la liste complète des opérateurs de comparaison disponibles, reportez-vous à la section “[Opérateurs de comparaison](#)” [ASA - Manuel de référence SQL, page 12].

Filtrage dans des conditions de recherche

Le filtrage est une méthode très polyvalente pour identifier les données de type caractère. Dans SQL, les recherches de ce type sont effectuées via le

mot-clé LIKE. Le filtrage utilise des caractères de remplacement pour faire correspondre différentes combinaisons de caractères.

❖ **Pour répertorier tous les employés dont le nom commence par BR**

1. Dans Interactive SQL, saisissez ce qui suit dans le volet Instructions SQL :

```
SELECT emp_lname, emp_fname
FROM employee
WHERE emp_lname LIKE 'br%'
```

emp_lname	emp_fname
Breault	Robert
Braun	Jane

Le signe % dans la condition de recherche indique qu'un nombre quelconque de caractères peut suivre les lettres BR.

❖ **Pour répertorier tous les employés dont le nom commence par BR, suivi ou non d'autres lettres, puis par un T suivi à son tour d'autres lettres ou non**

1. Dans Interactive SQL, saisissez ce qui suit dans le volet Instructions SQL :

```
SELECT emp_lname, emp_fname
FROM employee
WHERE emp_lname LIKE 'BR%T%'
```

emp_lname	emp_fname
Breault	Robert

Le premier signe % correspond à la chaîne **ea**ul, tandis que le second correspond à la chaîne vide (aucun caractère).

L'autre caractère spécial utilisable avec LIKE est le caractère de soulignement (), qui correspond exactement à un caractère. Le modèle 'BR %' permet d'obtenir tous les noms commençant par BR et qui ont un en quatrième position. Dans Braun, le trait de soulignement correspond à A et % à N.

👉 Pour plus d'informations, reportez-vous à la section "[Condition LIKE](#)" [[ASA - Manuel de référence SQL, page 26](#)].

Recherche de lignes par ressemblance phonique

La fonction SOUNDEX permet d'établir une correspondance phonique aux

lignes. Supposons par exemple que quelqu'un laisse un message téléphonique pour une personne dont le nom ressemble vaguement à "Brown". Quels sont les employés de la société dont le nom s'en rapproche ?

❖ **Pour répertorier les employés dont le nom sonne comme Brown**

1. Dans Interactive SQL, saisissez ce qui suit dans le volet Instructions SQL :

```
SELECT emp_lname, emp_fname
FROM employee
WHERE SOUNDEX( emp_lname ) = SOUNDEX( 'Brown' )
```

emp_lname	emp_fname
Braun	Jane

L'algorithme utilisé par SOUNDEX est principalement utile pour les bases de données en langue anglaise.

☞ Pour plus d'informations, reportez-vous à la section "[SOUNDEX \(fonction de chaîne de caractères\)](#)" [*ASA - Manuel de référence SQL*, page 227].

Utilisation de conditions de recherche composées

Les conditions de recherche peuvent être combinées au moyen des opérateurs logiques AND et OR pour créer des conditions de recherche composées. Chaque partie distincte de la condition de recherche est parfois nommée **prédicat**.

❖ **Pour répertorier tous les employés nés avant le 13 mars 1964, à l'exception de l'employé nommé Whitney**

1. Dans Interactive SQL, saisissez ce qui suit dans le volet Instructions SQL :

```
SELECT emp_lname, birth_date
FROM employee
WHERE birth_date < '1964-3-13'
AND emp_lname <> 'whitney'
```

emp_lname	birth_date
Cobb	12/4/1960
Jordan	12/13/1951
Breault	5/13/1947
Espinoza	12/14/1939
Dill	7/19/1963
Francis	9/12/1954

Raccourcis pour les conditions de recherche composées

Forme abrégée
BETWEEN

SQL propose deux formes abrégées pour les conditions de recherche composées. La première, BETWEEN, vise à rechercher un intervalle de valeurs. Par exemple, les deux requêtes suivantes sont équivalentes :

```
SELECT emp_lname, birth_date
FROM employee
WHERE birth_date BETWEEN '1963-1-1' AND '1965-3-31'
```

et

```
SELECT emp_lname, birth_date
FROM employee
WHERE birth_date >= '1963-1-1'
AND birth_date <= '1965-3-31'
```

Forme abrégée IN

La seconde forme abrégée, IN, sert à rechercher une valeur parmi plusieurs. Les deux instructions suivantes sont équivalentes :

```
SELECT emp_lname, emp_id
FROM employee
WHERE emp_lname IN ('yeung', 'bucceri', 'charlton')
```

et

```
SELECT emp_lname, emp_id
FROM employee
WHERE emp_lname = 'yeung'
OR emp_lname = 'bucceri'
OR emp_lname = 'charlton'
```

CHAPITRE 7

Sélection de données dans plusieurs tables

Présentation Ce chapitre décrit les requêtes de base de données qui examinent des informations dans plusieurs tables. SQL fournit à cet effet l'opérateur **JOIN**. Il existe différentes manières de joindre des tables dans des requêtes ; ce chapitre en décrit quelques-unes parmi les plus importantes.

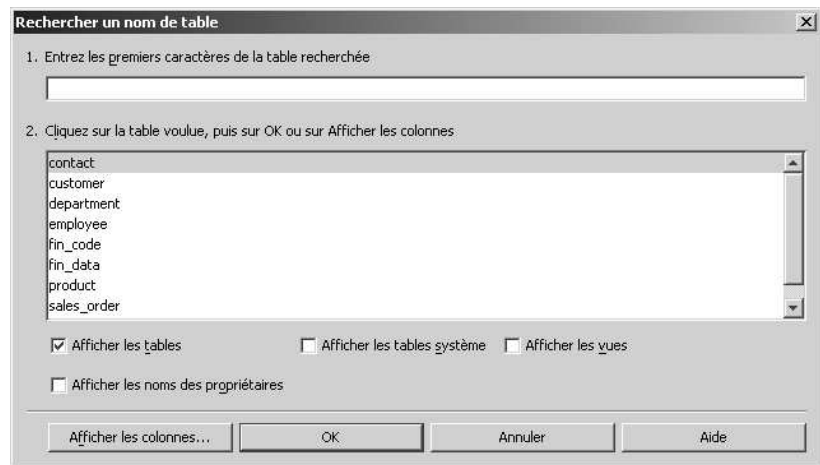
Sommaire	Sujet :	page
	Introduction	108
	Jointure de tables par produit cartésien	109
	Utilisation de la clause ON pour restreindre une jointure	110
	Jointure de tables à l'aide de jointures par clé	112
	Jointure de tables à l'aide de jointures naturelles	114
	Jointure de tables avec des jointures externes	116

Introduction

Il est parfois nécessaire de visualiser des données à partir de plusieurs tables reliées entre elles. Ce chapitre vous explique comment utiliser une **jointure** pour visualiser les données de manière utile et cohérente.

Affichage d'une liste de tables

Pour afficher la liste des tables dans Interactive SQL, appuyez sur la touche F7.



Sélectionnez une table et cliquez sur Afficher colonnes afin de visualiser les colonnes de la table. Appuyez sur la touche ECHAP pour revenir à la liste des tables ; une seconde pression vous ramène dans le volet Instructions SQL. Appuyez sur ENTRÉE et non pas sur ECHAP pour copier dans le volet Instructions SQL le nom de la table ou de la colonne sélectionnée, à la position courante du curseur.

Appuyez sur ECHAP pour sortir de la liste.

☞ Pour visualiser un schéma des tables de la base de données exemple, reportez-vous à la section [“A propos de la base de données exemple”](#) à la page 148.

Jointure de tables par produit cartésien

L'une des tables de la base de données exemple, `sales_order`, répertorie les commandes passées auprès de l'entreprise. Chaque commande comporte une colonne `sales_rep`, qui contient l'ID employé du commercial qui a pris la commande. La table `sales_order` contient 648 lignes et la table `employee` 75 lignes.

Une jointure de produit cartésien constitue un point de départ simple pour comprendre les jointures, mais ne présente pas un grand intérêt pratique.

❖ Pour répertorier toutes les données dans les tables `employee` et `sales_order`

1. Dans Interactive SQL, saisissez ce qui suit dans le volet Instructions SQL, puis appuyez sur F5 pour exécuter l'instruction.

```
SELECT *  
FROM sales_order CROSS JOIN employee
```

Le résultat de cette requête, qui apparaît sur l'onglet Résultats dans le volet Résultats d'Interactive SQL, fait correspondre chaque ligne de la table `employee` avec chaque ligne de la table `sales_order`. Comme `employee` contient 75 lignes et `sales_order` 648, le résultat de la jointure comporte $75 \times 648 = 48\,600$ lignes. Chaque ligne est constituée de toutes les colonnes de la table `sales_order`, suivies de toutes celles de la table `employee`. Ce type de jointure s'appelle un **produit cartésien**.

Les sections qui suivent expliquent comment construire des jointures plus sélectives. Les jointures plus sélectives indiquent qu'une restriction est appliquée à la table du produit cartésien.

☞ Pour plus d'informations, reportez-vous à la section "[Jointures croisées](#)" [[ASA - Guide de l'utilisateur SQL](#), page 300].

Utilisation de la clause ON pour restreindre une jointure

La clause ON applique une restriction aux lignes dans une jointure, de la même façon que la clause WHERE applique des restrictions aux lignes d'une requête.

La clause ON permet de construire des jointures plus utiles que CROSS JOIN ne le permet. Vous pouvez, par exemple, appliquer la clause ON à une jointure des tables sales_order et employee, afin d'extraire uniquement les lignes dans lesquelles le sales_rep de la table sales_order est le même que celui de la table employee dans chaque ligne du résultat. Ainsi, chaque ligne contient des informations sur une commande ainsi que sur le commercial qui en a la charge.

❖ Pour répertorier toutes les commandes avec leur date et l'employé qui en est responsable

1. Dans Interactive SQL, saisissez ce qui suit dans le volet Instructions SQL, puis appuyez sur F5 pour exécuter l'instruction.

```
SELECT employee.emp_lname,  
       sales_order.id,  
       sales_order.order_date  
FROM sales_order JOIN employee  
     ON sales_order.sales_rep = employee.emp_id
```

emp_lname	id	order_date
Chin	2008	4/2/2001
Chin	2020	3/4/2001
Chin	2032	7/5/2001
Chin	2044	7/15/2000
Chin	2056	4/15/2001
...

Remarques

Le nom de la table est indiqué sous la forme d'un préfixe identifiant les colonnes. L'utilisation de ce préfixe clarifie l'instruction et s'impose lorsque deux tables ont une colonne du même nom.

Le résultat de la requête ne contient plus que 648 lignes (une pour chaque ligne de la table sales_order). Sur les 48 600 lignes du produit cartésien, seules 648 contiennent le même numéro d'employé dans les deux tables.

L'agencement des résultats n'a aucune signification. Si vous souhaitez donner un ordre particulier à la requête, ajoutez une clause ORDER BY.

La clause ON inclut des colonnes qui ne sont pas comprises dans le jeu de résultats final.

☞ Pour plus d'informations, reportez-vous à la section [“Conditions de jointure explicites \(expression ON\)”](#) [ASA - Guide de l'utilisateur SQL, page 296].

Jointure de tables à l'aide de jointures par clé

Il existe de nombreux types de jointures entre deux tables liées par une clé étrangère. La jointure la plus courante impose aux valeurs de clé étrangère d'être égales aux valeurs de clé primaire.

L'opérateur KEY JOIN joint deux tables en fonction d'une relation par clé étrangère. En d'autres termes, Adaptive Server Anywhere génère une clause ON qui établit une égalité entre la colonne clé primaire d'une table et la colonne clé étrangère d'une autre.

L'exemple de la section précédente imposait que les valeurs de clé étrangère de la table sales_order soient égales à celles de clé primaire de la table employee.

```
SELECT employee.emp_lname,
       sales_order.id,
       sales_order.order_date
FROM sales_order JOIN employee
   ON sales_order.sales_rep = employee.emp_id
```

Il est possible d'exprimer la requête plus simplement en utilisant une jointure par clé (KEY JOIN) :

```
SELECT employee.emp_lname,
       sales_order.id,
       sales_order.order_date
FROM sales_order KEY JOIN employee
```

L'opérateur de jointure KEY JOIN ne constitue qu'un raccourci de la clause ON ; les deux requêtes sont identiques. La jointure par clé est appliquée par défaut lorsque vous précisez JOIN mais pas CROSS, NATURAL, KEY ou que vous utilisez la clause ON.

Dans le schéma de la base de données employee, les lignes entre les tables représentent les clés étrangères. Vous pouvez utiliser l'opérateur KEY JOIN chaque fois que deux tables du schéma sont jointes par une ligne.

☞ Pour visualiser le schéma de la base de données exemple, reportez-vous à la section [“A propos de la base de données exemple”](#) à la page 148.

Jointures de plusieurs tables

Les opérateurs de jointure permettent de joindre plus de deux tables. La requête suivante sollicite quatre tables pour déterminer la valeur totale des commandes passées par chaque client. Elle met en relation les quatre tables customer, sales_order, sales_order_items et product par le biais de la relation par clé étrangère unique entre chaque paire de tables.

❖ **Pour répertorier les sociétés et le montant total de leurs commandes**

1. Dans Interactive SQL, saisissez ce qui suit dans le volet Instructions SQL, puis appuyez sur F5 pour exécuter l’instruction.

```
SELECT company_name,
       SUM( sales_order_items.quantity *
           product.unit_price) AS value
FROM ( ( customer KEY JOIN sales_order )
      KEY JOIN sales_order_items )
      KEY JOIN product
GROUP BY company_name
```

company_name	value
Bensoul’s Boutique	1332
Bush Pro Shop	2940
Sterling & Co.	6804
Ocean Sports	3744
...	...

Remarques

Il se peut que votre jeu de résultats s’affiche dans un ordre différent. L’ordre des lignes dans le jeu de résultats n’a pas d’importance.

L’exemple utilise l’opérateur SUM, qui est une fonction d’agrégat. Les fonctions d’agrégat utilisent les clauses GROUP BY pour renvoyer des valeurs pour chaque groupe de lignes. Dans cet exemple, la somme de `sales_order_items.quantity * product.unit_price` (c’est-à-dire la somme totale payée par type de produit) est calculée pour chaque `company_name`, de façon à obtenir les ventes de chaque société.

Les parenthèses dans la clause FROM permettent de clarifier l’ordre dans lequel les jointures sont faites.

☞ Pour plus d’informations sur les jointures par clé complexes, reportez-vous à la section “Jointures par clé” [[ASA - Guide de l’utilisateur SQL, page 322](#)].

Jointure de tables à l'aide de jointures naturelles

L'opérateur NATURAL JOIN joint deux tables en fonction des noms de colonne communs. En d'autres termes, Adaptive Server Anywhere génère une clause ON qui établit une égalité entre les colonnes communes de chaque table.

❖ Pour répertorier tous les employés et leurs services

1. Dans Interactive SQL, saisissez ce qui suit dans le volet Instructions SQL, puis appuyez sur F5 pour exécuter l'instruction.

```
SELECT emp_lname, dept_name
FROM employee NATURAL JOIN department
```

emp_lname	dept_name
Whitney	R & D
Cobb	R & D
Breault	R & D
Shishov	R & D
Driscoll	R & D
...	...

Adaptive Server Anywhere examine les deux tables et détermine que le seul nom de colonne qu'elles ont en commun est dept_id. La clause ON suivante est générée en interne et utilisée pour effectuer la jointure :

```
FROM employee JOIN department
ON employee.dept_id = department.dept_id
```

NATURAL JOIN ne constitue qu'un raccourci de la clause ON ; les deux requêtes sont identiques.

Erreurs sur NATURAL JOIN

Cet opérateur de jointure peut générer des erreurs s'il établit une relation d'égalité entre des colonnes que vous ne souhaitez pas nécessairement associer. Par exemple, la requête suivante produit des résultats inattendus :

```
SELECT *
FROM sales_order NATURAL JOIN customer
```

Le résultat de cette requête ne contient aucune ligne. Adaptive Server Anywhere génère en interne la clause ON suivante :

```
FROM sales_order JOIN customer
ON sales_order.id = customer.id
```

La colonne id de la table sales_order est un numéro d'ID pour la *commande*. La colonne id de la table customer est un numéro d'ID pour le *client*. Il n'existe aucune correspondance. Et bien entendu, s'il y avait eu correspondance, celle-ci n'aurait aucun sens.

☞ Pour plus d'informations, reportez-vous à la section “[Jointures naturelles](#)” [ASA - *Guide de l'utilisateur SQL*, page 318].

Jointure de tables avec des jointures externes

Dans les exemples précédents, vous avez créé des jointures qui ne renvoyaient des lignes que si elles satisfaisaient aux critères de jointure. Ces jointures sont nommées **jointures internes** et s'appliquent par défaut. Parfois, vous voudrez peut-être conserver toutes les lignes d'une table. Dans ce cas, utilisez une **jointure externe**.

Vous pouvez choisir une **jointure externe droite** qui préserve toutes les lignes de la table de droite ; une **jointure externe gauche** qui préserve la table de gauche ou une **jointure externe complète** qui préserve toutes les lignes dans les deux tables.

❖ Pour répertorier tous les clients et la date de toutes leurs commandes

1. Dans Interactive SQL, saisissez ce qui suit dans le volet Instructions SQL, puis appuyez sur F5 pour exécuter l'instruction.

```
SELECT lname, order_date, city
FROM customer KEY LEFT OUTER JOIN sales_order
WHERE customer.state = 'NY'
ORDER BY order_date
```

lname	order_date	city
Thompson	(NULL)	Bancroft
Reiser	1993-01-22	Rockwood
Clarke	1993-01-27	Rockwood
Mentary	1993-01-30	Rockland
...

L'instruction contient tous les clients, qu'ils aient ou non passé commande. Si un client n'a passé aucune commande, chaque colonne correspondant aux informations sur les commandes contient la valeur NULL.

☞ Pour plus d'informations, reportez-vous à la section "[Jointures externes](#)" [[ASA - Guide de l'utilisateur SQL, page 302](#)].

CHAPITRE 8

Sélection de données à l'aide de sous-requêtes

Présentation Ce chapitre indique comment intégrer le résultat d'une requête dans une nouvelle instruction `SELECT`. Les sous-requêtes constituent un outil précieux pour construire des requêtes complexes et détaillées.

Sommaire	Sujet :	page
	Présentation des sous-requêtes	118
	Sous-requêtes monoligne et multiligne	121
	Sous-requêtes remplaçant des jointures	123

Présentation des sous-requêtes

Dans une base de données relationnelle, vous pouvez stocker des données reliées entre elles dans plusieurs tables. Vous pouvez extraire des données de tables liées à l'aide de **sous-requêtes**—requêtes qui figurent dans la clause WHERE ou HAVING d'une autre requête. Les sous-requêtes effectuent des requêtes plus faciles à écrire que les jointures et certaines requêtes ne peuvent pas être écrites sans utiliser de sous-requêtes.

Les sous-requêtes utilisent les résultats d'une requête comme partie d'une autre requête. Cette section illustre une situation où les sous-requêtes peuvent être utilisées, en construisant une requête répertoriant les articles de commande pour les produits dont le stock est réduit.

La constitution de cette liste nécessite deux requêtes. Cette section les décrit d'abord séparément, puis présente la requête unique qui produit le même résultat.

❖ Pour répertorier tous les produits dont la quantité en stock est inférieure à 20

1. Dans Interactive SQL, saisissez ce qui suit dans le volet Instructions SQL, puis appuyez sur F5 pour exécuter l'instruction.

```
SELECT id, description, quantity
FROM product
WHERE quantity < 20
```

id	description	quantity
401	Wool cap	12

Cette requête montre que seul le stock des bonnets de laine (wool caps) est réduit.

❖ Pour répertorier les articles de commande relatifs aux bonnets de laine (wool caps)

1. Dans Interactive SQL, saisissez ce qui suit dans le volet Instructions SQL, puis appuyez sur F5 pour exécuter l'instruction.

```
SELECT *
FROM sales_order_items
WHERE prod_id = 401
ORDER BY ship_date DESC
```

id	line_id	prod_id	quantity	ship_date
2082	1	401	48	7/9/2001

id	line_id	prod_id	quantity	ship_date
2053	1	401	60	6/30/2001
2125	2	401	36	6/28/2001
2027	1	401	12	6/17/2001
...

Les deux étapes de ce processus d'identification des articles dont le stock est réduit et des commandes correspondantes peuvent être combinées en une seule requête faisant appel à des sous-requêtes.

❖ **Pour répertorier tous les articles de commande de produits dont le stock est réduit**

1. Dans Interactive SQL, saisissez ce qui suit dans le volet Instructions SQL, puis appuyez sur F5 pour exécuter l'instruction.

```
SELECT *
FROM sales_order_items
WHERE prod_id IN
    ( SELECT id
      FROM product
      WHERE quantity < 20 )
ORDER BY ship_date DESC
```

id	line_id	prod_id	quantity	ship_date
2082	1	401	48	7/9/2001
2053	1	401	60	6/30/2001
2125	2	401	36	6/28/2001
2027	1	401	12	6/17/2001
...

Dans l'instruction, la sous-requête est la phrase entre parenthèses :

```
( SELECT id
  FROM product
  WHERE quantity < 20 )
```

Cette sous-requête crée une liste contenant la valeur de la colonne id pour toutes les lignes de la table product qui satisfont la condition de recherche définie dans la clause WHERE.

La sous-requête renvoie un ensemble de lignes, mais une seule colonne. Le mot-clé IN traite chaque valeur comme appartenant à un ensemble et teste si chaque ligne dans la requête principale fait partie de cet ensemble.

Sous-requêtes monoligne et multiligne

Il existe des contraintes relatives au nombre de lignes et de colonnes qu'une sous-requête peut renvoyer. Si vous utilisez IN, ANY ou ALL, la sous-requête peut renvoyer plusieurs lignes, mais une seule colonne. Si vous utilisez d'autres opérateurs, la sous-requête doit renvoyer une valeur unique.

Sous-requête multiligne

Deux tables de la base de données exemple gèrent des résultats financiers. La table `fin_code` est une petite table contenant les différents codes propres aux données financières et leur signification :

Pour afficher les lignes de recette de la table `fin_data`, saisissez ce qui suit :

```
SELECT *
FROM fin_data
WHERE fin_data.code IN
  ( SELECT fin_code.code
    FROM fin_code
    WHERE type = 'revenue' )
```

year	quarter	code	amount
1999	Q1	r1	1023
1999	Q2	r1	2033
1999	Q3	r1	2998
1999	Q4	r1	3014
2000	Q1	r1	3114

Cet exemple emploie des qualificatifs pour identifier clairement la table à laquelle chaque colonne code fait référence. Dans le cas présent, les qualificatifs auraient pu être omis.

Deux autres mots-clés peuvent être utilisés comme qualificatifs pour que les opérateurs puissent travailler avec plusieurs lignes : ANY et ALL.

L'exemple suivant a le même effet que la requête correcte utilisée précédemment :

```
SELECT *
FROM fin_data
WHERE fin_data.code = ANY
  ( SELECT fin_code.code
    FROM fin_code
    WHERE type = 'revenue' )
```

Une condition `=ANY` est identique à la condition IN, mais ANY est également utilisable avec des inégalités telles que `<` ou `>`, ce qui permet un

usage plus flexible des sous-requêtes.

Le mot-clé ALL est similaire à ANY. La requête suivante, par exemple, affiche les données financières qui ne sont pas des revenus :

```
SELECT *
FROM fin_data
WHERE fin_data.code <> ALL
  ( SELECT fin_code.code
    FROM fin_code
    WHERE type = 'revenue' )
```

Elle est équivalente à la commande suivante, qui utilise NOT IN :

```
SELECT *
FROM fin_data
WHERE fin_data.code NOT IN
  ( SELECT fin_code.code
    FROM fin_code
    WHERE type = 'revenue' )
```

Erreur fréquente
d'utilisation des
sous-requêtes

En général, les jeux de résultats des sous-requêtes sont restreints à une seule colonne. L'exemple suivant n'a aucun sens, car Adaptive Server Anywhere n'a aucun moyen de savoir quelle colonne de la table fin_code il doit comparer à la colonne fin_data.code.

```
-- this query is incorrect
SELECT *
FROM fin_data
WHERE fin_data.code IN
  ( SELECT fin_code.code, fin_code.type
    FROM fin_code
    WHERE type = 'revenue' )
```

Sous-requêtes
monoligne

Une sous-requête utilisée avec un opérateur de comparaison ne doit renvoyer qu'une seule ligne, alors que les sous-requêtes appelées par une condition IN peuvent retourner un ensemble de lignes. La commande suivante, par exemple, génère une erreur car la sous-requête renvoie deux lignes :

```
-- this query is incorrect
SELECT *
FROM fin_data
WHERE fin_data.code =
  ( SELECT fin_code.code
    FROM fin_code
    WHERE type = 'revenue' )
```

Sous-requêtes remplaçant des jointures

Supposons que vous ayez besoin d'une liste chronologique des commandes et des entreprises qui les ont passées, mais que vous préférez afficher le nom des entreprises et non pas leur ID client. Pour ce faire, utilisez une jointure :

Jointure

Pour afficher les ID de commande, les dates et les noms des entreprises pour chaque commande passée depuis le 1er janvier 2001 :

```
SELECT    sales_order.id,
          sales_order.order_date,
          customer.company_name
FROM sales_order
  KEY JOIN customer
WHERE order_date > '2001/01/01'
ORDER BY order_date
```

id	order_date	company_name
2473	1/4/2001	Peachtree Active Wear
2474	1/4/2001	Sampson & Sons
2106	1/5/2001	Salt & Pepper's
2475	1/5/2001	Cinnamon Rainbow's
2036	1/5/2001	Hermanns

Sous-requête

L'instruction suivante génère, par le biais d'une sous-requête, le même résultat que la jointure :

```
SELECT sales_order.id,
       sales_order.order_date,
       ( SELECT company_name FROM customer
         WHERE customer.id = sales_order.cust_id )
FROM sales_order
WHERE order_date > '2001/01/01'
ORDER BY order_date
```

La sous-requête fait référence à la colonne `cust_id` de la table `sales_order` même si celle-ci ne figure pas dans la sous-requête. En effet, la colonne `sales_order.cust_id` fait référence à la table `sales_order` citée dans le corps principal de l'instruction. On parle dans ce cas de **référence externe**. Toute sous-requête comportant une telle référence est appelée **sous-requête liée**.

Vous pouvez utiliser une sous-requête au lieu d'une jointure chaque fois que seule une colonne de l'autre table est requise. (N'oubliez pas que les sous-requêtes ne peuvent renvoyer qu'une colonne.) Dans l'exemple précédent, vous n'aviez besoin que de la colonne `company_name`, de sorte que la jointure a pu être remplacée par une sous-requête.

Si la sous-requête ne peut pas générer de résultat, cette méthode est appelée jointure externe. La jointure étudiée dans les sections précédentes du didacticiel est désignée sous le nom de jointure interne.

Jointure externe

Pour obtenir la liste de tous les clients de l'état de Washington avec leur dernier ID de commande, saisissez ce qui suit :

```
SELECT    company_name, state,
          ( SELECT MAX( id )
            FROM sales_order
            WHERE sales_order.cust_id = customer.id )
FROM customer
WHERE state = 'WA'
```

company_name	state	MAX(sales_order.id)
Custom Designs	WA	2547
It's a Hit !	WA	(NULL)

L'entreprise It's a Hit ! n'ayant passé aucune commande, la sous-requête renvoie la valeur NULL pour ce client. Lorsqu'une jointure interne est utilisée, les entreprises qui n'ont pas passé de commande ne sont pas dans la liste.

Vous pouvez aussi spécifier explicitement une jointure externe. Dans ce cas, une clause GROUP BY est également requise.

```
SELECT company_name, state,
       MAX( sales_order.id )
FROM customer
   KEY LEFT OUTER JOIN sales_order
WHERE state = 'WA'
GROUP BY company_name, state
```

CHAPITRE 9

Sélection des données d'agrégat

Présentation Ce chapitre décrit comment construire des requêtes qui renvoient des informations agrégées. Voici quelques exemples d'informations de ce type :

- ◆ total des valeurs d'une colonne,
- ◆ nombre d'entrées distinctes dans une colonne,
- ◆ valeur moyenne des entrées d'une colonne.

Sommaire

Sujet :	page
Regroupement des données	126
Présentation des fonctions d'agrégat	127
Application des fonctions d'agrégat aux données groupées	128
Restriction de groupes	130

Regroupement des données

Certaines requêtes examinent les aspects des données de votre table qui reflètent les propriétés des groupes de lignes et non des lignes individuelles. Par exemple, vous pouvez chercher à évaluer la somme moyenne qu'un client dépense pour une commande ou déterminer le nombre d'employés qui travaillent dans chaque département. Pour ce type de tâches, vous utilisez les fonctions d'**agrégat** et la clause **GROUP BY**.

Présentation des fonctions d'agrégat

Les fonctions d'agrégat renvoient une valeur unique pour un jeu de lignes. S'il n'y a pas la clause GROUP BY, une fonction d'agrégat renvoie une valeur unique pour toutes les lignes qui correspondent à d'autres aspects de la requête.

❖ Pour répertorier le nombre d'employés dans l'entreprise

1. Dans Interactive SQL, saisissez ce qui suit dans le volet Instructions SQL, puis appuyez sur F5 pour exécuter l'instruction.

```
SELECT COUNT( * )
FROM employee
```

COUNT(*)

75

Le jeu de résultats ne contient qu'une colonne, intitulée **COUNT(*)**, et une ligne, incluant le nombre total d'employés.

❖ Pour répertorier le nombre d'employés dans l'entreprise et la date de naissance du plus jeune et du plus âgé

1. Dans Interactive SQL, saisissez ce qui suit dans le volet Instructions SQL, puis appuyez sur F5 pour exécuter l'instruction.

```
SELECT COUNT(*), MIN(birth_date), MAX(birth_date)
FROM employee
```

COUNT(*)	MIN(employee.birth_date)	MAX(employee.birth_date)
75	1/2/1936	1/18/1973

COUNT, MIN et MAX sont appelées **fonctions d'agrégat**. Il s'agit de fonctions de regroupement de données. Les fonctions statistiques telles que AVG, STDDEV et VARIANCE sont d'autres exemples de fonctions d'agrégat. Toutes, sauf COUNT, utilisent le nom d'une colonne en tant que paramètre.

☞ Pour plus d'informations, reportez-vous à la section "Fonctions d'agrégat" [ASA - Manuel de référence SQL, page 92].

Application des fonctions d'agrégat aux données groupées

Outre le fait de fournir des informations sur une table entière, les fonctions d'agrégat peuvent s'appliquer à des groupes de lignes. La clause GROUP BY classe les lignes par groupe et les fonctions d'agrégat renvoient une valeur unique pour chaque groupe de lignes.

Exemple

❖ Pour répertorier les représentants commerciaux et le nombre de commandes prises par chacun d'entre eux

1. Dans Interactive SQL, saisissez ce qui suit dans le volet Instructions SQL, puis appuyez sur F5 pour exécuter l'instruction.

```
SELECT sales_rep, count( * )
FROM sales_order
GROUP BY sales_rep
ORDER BY sales_rep
```

sales_rep	count(*)
129	57
195	50
299	114
467	56
...	...

Une clause GROUP BY demande à Adaptive Server Anywhere de partitionner le jeu de toutes les lignes qui devaient être renvoyées. Dans chaque partition ou groupe, toutes les lignes ont les mêmes valeurs dans la ou les colonnes nommées. Il n'existe qu'un groupe pour chaque valeur ou groupe de valeurs unique. Dans l'exemple, toutes les lignes de chaque groupe ont la même valeur sales_rep.

Les fonctions d'agrégat telles que COUNT sont appliquées aux lignes de chaque groupe. De cette façon, le jeu de résultats affiche le nombre total de lignes dans chaque groupe. Les résultats de la requête se composent d'une ligne par numéro d'identification de chaque représentant commercial. Chaque ligne contient l'ID du représentant commercial et le nombre total de ses commandes.

Si la clause GROUP BY est spécifiée, la table générée comporte une ligne pour chaque colonne (ou jeu de colonnes) nommée dans la clause

GROUP BY.

☞ Pour plus d'informations, reportez-vous à la section [“Clause GROUP BY : regroupement des résultats d'une requête” \[ASA - Guide de l'utilisateur SQL, page 268\]](#).

Erreur fréquente avec GROUP BY

Une erreur courante avec GROUP BY est de chercher à obtenir des informations qui ne peuvent pas être correctement regroupées. Par exemple :

```
-- This query is incorrect
SELECT sales_rep, emp_lname, COUNT( * )
FROM sales_order KEY JOIN employee
GROUP BY sales_rep
```

génère le message d'erreur :

La référence à 'emp_lname' de la fonction ou de la colonne dans la liste de sélection doit également figurer dans GROUP BY

Une erreur est signalée car Adaptive Server Anywhere ne peut pas garantir que le même nom de famille figure sur chaque ligne de résultats pour un employé avec un ID donné.

Pour réparer cette erreur, ajoutez la colonne à la clause GROUP BY.

```
SELECT sales_rep, emp_lname, COUNT( * )
FROM sales_order KEY JOIN employee
GROUP BY sales_rep, emp_lname
ORDER BY sales_rep
```

Si cela ne convient pas, vous pouvez utiliser une fonction d'agrégat pour sélectionner uniquement une valeur, comme suit :

```
SELECT sales_rep, MAX( emp_lname ), COUNT( * )
FROM sales_order KEY JOIN employee
GROUP BY sales_rep
ORDER BY sales_rep
```

La fonction MAX choisit pour chaque groupe le nom de famille “maximal” (le dernier dans l'ordre alphabétique) dans les lignes de la table. Cette instruction est correcte car il ne peut y avoir qu'une valeur maximale distincte. Dans ce cas, le même nom de famille apparaît sur chaque ligne “détail” à l'intérieur d'un groupe.

Restriction de groupes

Vous avez déjà vu comment limiter les lignes dans le résultat d'une requête à l'aide de la clause WHERE. Vous pouvez également restreindre les lignes dans les groupes à l'aide de la clause HAVING.

❖ Pour obtenir la liste de tous les commerciaux ayant pris plus de 55 commandes

1. Dans Interactive SQL, saisissez ce qui suit dans le volet Instructions SQL, puis appuyez sur F5 pour exécuter l'instruction.

```
SELECT sales_rep, count( * ) AS orders
FROM sales_order KEY JOIN employee
GROUP BY sales_rep
HAVING count( * ) > 55
ORDER BY orders DESC
```

sales_rep	orders
299	114
129	57
1142	57
467	56

Ordre des clauses

GROUP BY doit toujours figurer avant HAVING. Si les deux clauses sont présentes, la clause WHERE doit apparaître avant la clause GROUP BY.

Les clauses HAVING et WHERE peuvent être utilisées dans une seule et même requête. Lorsque la clause HAVING comporte des conditions, les lignes du résultat ne sont logiquement limitées qu'après la génération des groupes. Les critères de la clause WHERE sont évalués avant que les groupes ne soient générés, ce qui permet de gagner du temps.

☞ Pour plus d'informations, reportez-vous à la section [“Clause HAVING : Sélection de groupes de données”](#) [ASA - Guide de l'utilisateur SQL, page 273].

Combinaison des clauses WHERE et HAVING

Vous pouvez parfois définir le même jeu de lignes à l'aide de la clause WHERE ou de la clause HAVING. Dans ce cas, aucune des méthodes n'est plus efficace que l'autre. L'optimiseur analyse toujours automatiquement chaque instruction saisie et sélectionne une méthode efficace d'exécution. Le mieux consiste à utiliser la syntaxe qui décrit le plus clairement le résultat

souhaité. En général, cela signifie éliminer les lignes indésirables dans les clauses qui précèdent.

Exemple

Pour obtenir la liste de tous les commerciaux ayant pris plus de 55 commandes et dont l'ID est supérieur à 1000 :

```
SELECT sales_rep, count( * )
FROM sales_order KEY JOIN employee
WHERE sales_rep > 1000
GROUP BY sales_rep
HAVING count( * ) > 55
ORDER BY sales_rep
```

L'instruction ci-après génère le même résultat.

```
SELECT sales_rep, count( * )
FROM sales_order KEY JOIN employee
GROUP BY sales_rep
HAVING count( * ) > 55 AND sales_rep > 1000
ORDER BY sales_rep
```

Adaptive Server Anywhere détecte que les deux instructions décrivent le même jeu de résultats et, par conséquent, exécute chacune d'entre elles efficacement.

CHAPITRE 10

Mise à jour de la base de données

Présentation Ce chapitre explique comment modifier le contenu de votre base de données. Il décrit notamment les procédures permettant d'ajouter des lignes, de les modifier ou de les supprimer, ainsi que la façon de rendre les modifications permanentes ou de corriger celles que vous avez effectuées.

Sommaire	Sujet :	page
	Introduction	134
	Ajout de lignes à une table	135
	Modification de lignes dans une table	136
	Suppression de lignes	137
	Regroupement des modifications dans des transactions	138
	Contrôle d'intégrité	141

Introduction

Ce chapitre explique comment modifier le contenu de votre base de données. Il existe trois opérations élémentaires :

- ◆ **Insertion** Vous pouvez ajouter des lignes aux tables pour intégrer de nouvelles données.
- ◆ **Suppression** Vous pouvez supprimer des lignes dans les tables pour effacer des données.
- ◆ **Mise à jour** Vous pouvez modifier les lignes existantes dans les tables.

Chaque opération s'effectue grâce à l'exécution d'une instruction SQL.

Ajout de lignes à une table

Supposons que l'entreprise décide de vendre un nouveau produit, par exemple une veste marron en acrylique. Pour ce faire, vous devez ajouter des données à la table product de la base exemple.

❖ Pour ajouter une veste en acrylique marron à la table product

1. Dans Interactive SQL, saisissez ce qui suit dans le volet Instructions SQL, puis appuyez sur F5 pour exécuter l'instruction.

```
INSERT
INTO product ( id, name, description, size, color,
               quantity, unit_price )
VALUES ( 800, 'Vest', 'Acrylic Vest', 'Extra Large',
        'Brown', 42, 20.00 )
```

Si vous faites une erreur et oubliez de définir l'une des colonnes, Adaptive Server Anywhere affiche un message d'erreur.

Vous pouvez également ajouter de nouvelles lignes aux tables de base de données à partir du jeu de résultats dans Interactive SQL.

☞ Pour plus d'informations, reportez-vous à la section [“Modification des valeurs de table dans Interactive SQL”](#) à la page 178.

Valeurs NULL dans les colonnes

La valeur NULL est une valeur spéciale servant à indiquer que quelque chose est inconnu ou inapplicable. Certaines colonnes peuvent contenir la valeur NULL et d'autres pas.

Modification de lignes dans une table

Dans la plupart des bases de données, vous devez mettre à jour les enregistrements qui s’y trouvent. Par exemple, l’ID d’un responsable doit changer lorsque certaines personnes sont transférées d’un service à un autre, ainsi que l’ID de département associé à ces personnes.

❖ Pour transférer l’employé #195 au département 400 à partir d’Interactive SQL

1. Dans Interactive SQL, saisissez ce qui suit dans le volet Instructions SQL, puis appuyez sur F5 pour exécuter l’instruction.

```
UPDATE employee
SET dept_id = 400, manager_id = 1576
WHERE emp_id = 195
```

L’instruction suivante effectue les deux mises à jour en même temps pour l’employé Marc Dill (ID 195).

Il arrive que plusieurs lignes soient mises à jour simultanément car Adaptive Server Anywhere met à jour plusieurs lignes répondant aux conditions de la clause WHERE. Par exemple, si un groupe d’employés du service commercial est muté au marketing et que leur colonne dept_id est mise à jour, l’instruction suivante affecte le numéro 1576 au manager_id de tous les employés du service marketing.

```
UPDATE employee
SET manager_id = 1576
WHERE dept_id = 400
```

Aucune modification n’est apportée aux employés appartenant déjà au service marketing.

Vous pouvez également modifier les lignes à partir du jeu de résultats dans Interactive SQL.

☞ Pour plus d’informations, reportez-vous à la section [“Modification des valeurs de table dans Interactive SQL”](#) à la page 178.

Suppression de lignes

Vous aurez sans doute à supprimer des lignes dans une table. Supposons que Rodrigo Guevara (ID employé 249) quitte l'entreprise. L'instruction suivante le supprime de la table employee.

```
DELETE
FROM employee
WHERE emp_id = 249
```

Avec UPDATE et DELETE, les conditions de recherche peuvent être aussi complexes que vos besoins l'exigent. Par exemple, si la table employee est en cours de réorganisation, l'instruction suivante y supprime tous les employés de sexe masculin embauchés entre le 3 mars 1988 et le 03 mars 1989.

```
DELETE
FROM employee
WHERE sex = 'm'
      AND start_date BETWEEN '1988-03-03'
      AND '1989-03-03'
```

Vous pouvez également supprimer des lignes de tables de base de données à partir du jeu de résultats dans Interactive SQL.

☞ Pour plus d'informations, reportez-vous à la section [“Modification des valeurs de table dans Interactive SQL”](#) à la page 178.

Puisque vous avez apporté des modifications que vous ne souhaitez pas conserver, annulez-les comme suit :

```
ROLLBACK
```

Regroupement des modifications dans des transactions

Dans Adaptive Server Anywhere, vous devez regrouper les commandes dans des transactions. Vous validez une transaction pour que les modifications deviennent permanentes. Lorsque vous modifiez les données, les modifications ne deviennent pas immédiatement permanentes. Elles sont d'abord stockées dans votre **journal de transactions** et ne deviennent permanentes que lorsque vous entrez la commande COMMIT.

Le fait de bien connaître les commandes ou les actions qui démarrent ou terminent une transaction permet d'exploiter pleinement les transactions.

Lancement des transactions

Une transaction est lancée par l'un des événements suivants :

- ◆ la première instruction qui suit une connexion à une base de données ;
- ◆ la première instruction qui suit la fin d'une transaction.

Fin des transactions

Les transactions sont terminées par l'un des événements suivants :

- ◆ Une instruction COMMIT rend permanentes les modifications effectuées dans la base de données.
- ◆ Une instruction ROLLBACK annule toutes les modifications effectuées par la transaction.
- ◆ Une instruction ayant comme effet secondaire une validation automatique est exécutée. Les commandes de définition d'une base de données, telles que ALTER, CREATE, COMMENT et DROP, génèrent une validation automatique ;
- ◆ Une déconnexion d'une base de données exécute une annulation implicite.

Options d'Interactive SQL

Interactive SQL met à votre disposition deux options qui permettent de contrôler quand et comment se termine une transaction :

- ◆ Si vous paramétrez l'option AUTO_COMMIT à ON, Interactive SQL effectue un COMMIT automatiquement pour les résultats après chaque instruction réussie et exécute un ROLLBACK après chaque instruction ayant échoué.
- ◆ L'option COMMIT_ON_EXIT contrôle ce qui se passe au niveau des modifications non validées lorsque vous quittez Interactive SQL. Si cette option a la valeur ON (valeur par défaut), Interactive SQL effectue une validation ; sinon, les modifications non validées sont annulées via une instruction ROLLBACK.

Si vous utilisez une source de données

Par défaut, ODBC opère en mode autocommit. Même si vous avez défini l'option AUTO_COMMIT à OFF dans Interactive SQL, les paramètres d'ODBC l'emportent sur ceux d'Interactive SQL. Vous pouvez modifier le paramètre ODBC en utilisant l'attribut de connexion SQL_ATTR_AUTOCOMMIT. L'autocommit d'ODBC est indépendant de l'option CHAINED.

Adaptive Server Anywhere prend également en charge les commandes Transact-SQL, telles que BEGIN TRANSACTION, pour maintenir la compatibilité avec Sybase Adaptive Server Enterprise.

☞ Pour plus d'informations, reportez-vous à la section [“Présentation du support de Transact-SQL” \[ASA - Guide de l'utilisateur SQL, page 510\]](#).

Validation des modifications

L'instruction COMMIT valide toutes les modifications.

Vous devez utiliser l'instruction COMMIT après des groupes d'instructions dont la juxtaposition est sensée. Par exemple, si vous voulez transférer de l'argent d'un compte client sur un autre, vous devez ajouter l'argent sur un compte et le retirer de l'autre, puis valider l'opération puisque, dans ce cas, vous ne pouvez pas laisser la base de données avec plus ou moins d'argent qu'au moment de son démarrage.

Adaptive Server Anywhere peut valider vos modifications automatiquement en paramétrant l'option AUTO_COMMIT sur ON. C'est une option d'Interactive SQL. Lorsque l'option AUTO_COMMIT est activée (ON), Adaptive Server Anywhere doit mettre à jour la base après chaque insertion, mise à jour et suppression. Cela peut ralentir considérablement les performances. C'est pourquoi il est parfois préférable de désactiver l'option AUTO_COMMIT (OFF).

Utilisez COMMIT avec modération

Lorsque vous testez les exemples de ce didacticiel, prenez soin de ne pas valider toutes les modifications tant que vous n'êtes pas certain de vouloir modifier définitivement la base de données.

☞ Pour plus d'informations sur les options de configuration Interactive SQL, reportez-vous à la section [“Options Interactive SQL” \[ASA - Guide d'administration, page 642\]](#).

Annulation de modifications

Il est possible d'annuler des modifications encore non validées. SQL propose à cet effet l'instruction ROLLBACK.

Cette instruction annule les modifications que vous avez effectuées depuis la dernière validation.

☞ Pour plus d'informations sur les options de configuration Interactive SQL, reportez-vous à la section “Options Interactive SQL” [*ASA - Guide d'administration*, page 642].

Transactions et restauration des données

Supposons qu'une défaillance du système ou une coupure de courant mette brusquement hors service le serveur de base de données. Adaptive Server Anywhere est conçu pour protéger l'intégrité de votre base de données dans ce type de situation. Il met à votre disposition différents moyens indépendants pour restaurer votre base de données. Par exemple, vous disposez d'un **journal de transactions** que vous pouvez stocker sur une unité distincte de sorte qu'en cas de panne sur une unité, vous avez toujours un moyen de restaurer les données. De plus, lorsque vous utilisez un journal de transactions, Adaptive Server Anywhere n'a pas besoin de mettre à jour votre base de données aussi fréquemment, ce qui améliore les performances de la base.

Le traitement par transactions permet au serveur d'identifier les états dans lesquels se trouvent vos données. Le traitement par transaction garantit qu'une transaction est intégralement annulée si, pour une raison quelconque, elle ne se termine pas correctement. Ainsi, la base de données n'est affectée en rien par des transactions ayant échoué.

Avec le traitement par transaction d'Adaptive Server Anywhere, vous avez l'assurance que le contenu des transactions fait l'objet d'un traitement sécurisé, même en cas de panne système inopinée.

☞ Pour plus d'informations sur les méthodes de restauration des données, reportez-vous à la section “Sauvegarde et reprise” [*ASA - Guide d'administration*, page 375].

Contrôle d'intégrité

Adaptive Server Anywhere vérifie automatiquement que vos données sont exemptes de certaines erreurs courantes.

Insertion de données en double

Supposons que vous tentiez de créer un service en spécifiant une valeur dept_id déjà utilisée :

Pour ce faire, entrez la commande :

```
INSERT
INTO department ( dept_id, dept_name, dept_head_id )
VALUES ( 200, 'Eastern Sales', 902 )
```

L'instruction INSERT est rejetée car son exécution aurait violé la règle d'unicité de la clé primaire. Le champ dept_id étant une clé primaire, les valeurs en double ne sont pas autorisées.

Insertion de valeurs qui ne respectent pas les relations

L'instruction suivante insère une nouvelle ligne dans la table sales_order, mais indique par erreur un ID de commercial qui n'existe pas dans la table employee.

```
INSERT
INTO sales_order ( id, cust_id, order_date,
sales_rep)
VALUES ( 2700, 186, '1995-10-19', 284 )
```

Il existe une relation 1-n entre les tables employee et sales_order, avec une jointure entre le champ sales_rep de la table sales_order et le champ emp_id de la table employee. Ce n'est qu'après l'entrée d'un enregistrement de la table contenant la clé primaire pour la jointure (table employee) qu'il est possible d'insérer un enregistrement correspondant contenant la clé étrangère (table sales_order).

Clés étrangères

La clé primaire de la table employee est le numéro d'ID d'employé. Le numéro d'ID de commercial de la table sales_rep est une **clé étrangère** pour la table employee. Autrement dit, chaque numéro de commercial de la table sales_order doit correspondre à un seul numéro d'ID d'employé dans la table employee.

Si vous tentez d'ajouter une commande au commercial 284, vous recevrez le message d'erreur suivant :

Aucune valeur n'a été spécifiée pour la clé primaire de

La clé étrangère 'ky_so_employee_id' dans la table 'sales_order'

La table employee ne contient pas d'employé portant ce numéro d'ID. Cela vous empêche d'insérer des commandes sans ID de commercial correct. Ce type de contrôle de validité est appelé contrôle de l'**intégrité référentielle** car il maintient l'intégrité des références dans les tables de la base de données.

☞ Pour plus d'informations sur les clés primaires et étrangères, reportez-vous à la section "[Relations entre les tables](#)" à la page 63.

Erreurs lors de SUPPRESSIONS ou de MISES A JOUR

Des erreurs de clé étrangère peuvent également se produire lors de mises à jour ou de suppressions. Supposons par exemple que vous tentiez de supprimer le service R&D de la table department. Le champ dept_id, en tant que clé primaire de la table department, constitue le côté "1" de la relation 1-n (le champ dept_id de la table employee est la clé étrangère correspondante et forme donc le côté N). Un enregistrement du côté *un* d'une relation ne peut pas être supprimé tant que tous les enregistrements correspondants du côté "N" ne sont pas supprimés.

```
DELETE
FROM department
WHERE dept_id = 100
```

Exemple : erreurs à la SUPPRESSION

Un message d'erreur s'affiche, indiquant que la base de données contient d'autres enregistrements qui font référence au service R&D et que l'opération de suppression ne sera pas exécutée.

La clé primaire de la ligne de la table 'department' est référencée dans une autre table

Pour supprimer le service R&D, vous devrez au préalable supprimer tous les employés de ce service :

```
DELETE
FROM employee
WHERE dept_id = 100
```

Vous pourrez ensuite supprimer le service R&D.

Entrez maintenant l'instruction ROLLBACK pour annuler les modifications que vous venez d'apporter (en vue d'exercices ultérieurs) :

```
ROLLBACK WORK
```

Toutes les modifications effectuées depuis le dernier COMMIT WORK réussi sont annulées. Si vous n'avez pas encore exécuté de COMMIT, toutes

les modifications effectuées depuis le lancement d'Interactive SQL sont annulées.

Exemple : erreurs à la
MISE A JOUR

Le même message d'erreur est généré si vous effectuez une mise à jour qui rend la base de données incohérente.

Par exemple, l'instruction UPDATE suivante entraîne une erreur d'intégrité :

```
UPDATE department
SET dept_id = 600
WHERE dept_id = 100
```

Dans tous les exemples ci-dessus, l'intégrité de la base de données a été vérifiée avant l'exécution de chaque commande. Aucune opération susceptible de provoquer des incohérences dans la base de données n'est exécutée.

Exemple : contrôle
d'intégrité à l'issue d'un
COMMIT WORK

Il est possible de configurer la base de données de sorte qu'aucun contrôle d'intégrité ne soit effectué avant l'exécution d'un COMMIT WORK. Cela est important si vous souhaitez modifier la valeur d'une clé primaire référencée : par exemple, changer l'ID du service R&D de 100 à 600 dans les tables department et employee. Pour ce faire, la base de données passe, entre ces deux modifications, par un état incohérent. Dans ce cas, vous devez configurer la base pour que la cohérence ne soit vérifiée qu'à la validation.

☞ Pour plus d'informations, reportez-vous à la section "[Option WAIT_FOR_COMMIT \[base de données\]](#)" [*ASA - Guide d'administration*, page 714].

Vous pouvez aussi définir des clés étrangères de sorte qu'elles soient automatiquement tenues à jour. Dans l'exemple ci-dessus, si la clé étrangère de employee vers department est définie avec ON UPDATE CASCADE, la mise à jour de l'ID de service (dept_id) actualise automatiquement la table employee.

Dans les cas ci-dessus, il est impossible de valider une base de données incohérente. Adaptive Server Anywhere effectue d'autres actions si des modifications sont susceptibles d'entraîner des incohérences.

☞ Pour plus d'informations, reportez-vous au chapitre "[Assurer la cohérence des données](#)" [*ASA - Guide de l'utilisateur SQL*, page 87].



QUATRIÈME PARTIE

DIDACTICIELS DE SQL ANYWHERE STUDIO

Cette partie propose des didacticiels destinés à vous aider à démarrer avec tous les composants de SQL Anywhere Studio. Chaque chapitre porte sur un sujet précis.

CHAPITRE 11

Base de données exemple

Présentation Ce chapitre fournit des informations élémentaires sur les serveurs de base de données et la base de données exemple. Il vous explique comment réaliser une copie de la base de données exemple, démarrer le serveur qui exécute la base de données exemple, afficher la fenêtre du serveur de base de données et fermer le serveur.

Sommaire	Sujet :	page
	A propos de la base de données exemple	148
	Leçon 1 : Copie de la base de données exemple	150
	Leçon 2 : Démarrage du serveur de base de données Adaptive Server Anywhere	151
	Leçon 3 : Affichage de la fenêtre du serveur de base de données	153
	Leçon 4 : Fermeture de la base de données	155
	Résumé	156

A propos de la base de données exemple

Le serveur de base de données Adaptive Server Anywhere est le logiciel qui gère la base de données. Le serveur donne accès, par les applications clientes, aux bases de données et traite les commandes de façon efficace et sûre. Une base de données ne peut être gérée que par un seul serveur à la fois.

Tous les accès à votre base de données doivent passer par le serveur.

Deux types de serveur de base de données

Il existe deux versions du serveur de base de données Adaptive Server Anywhere. Le **serveur personnel** peut uniquement accepter les connexions demandées par des applications ou des utilisateurs exploitant la même machine. Il est destiné à une utilisation mono-utilisateur sur une seule machine.

En revanche, le **serveur réseau** supporte les communications client/serveur sur un réseau et convient à une exploitation multi-utilisateur.

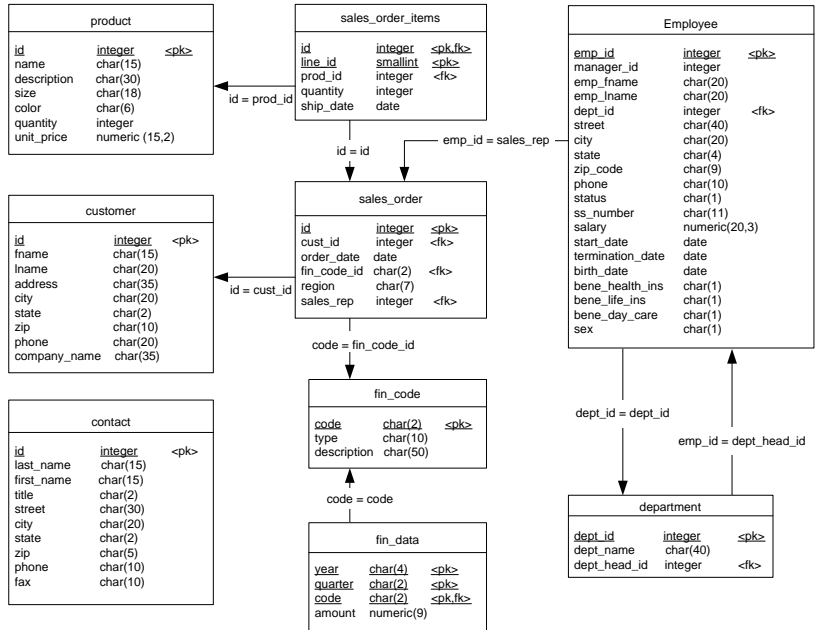
Les deux serveurs fonctionnent à l'identique au niveau du traitement des requêtes et d'autres opérations internes : le serveur de base de données personnel et le serveur de base de données réseau sont équivalents.

Base de données exemple

Ce didacticiel s'articule autour de la base de données exemple. Cette base fournit l'exemple d'une petite entreprise qui fabrique une gamme limitée d'articles de sport. Elle contient des informations concernant la société (employés, services et données financières), les produits et les ventes (commandes, clients et contacts). Toutes les informations contenues dans la base de données exemple relèvent de la pure fiction.

La figure suivante montre les tables de la base de données exemple et leurs relations. Les boîtes représentent les tables et les flèches représentent les relations de clé étrangère entre les tables. Les colonnes clé primaire sont soulignées.

asademo.db



Leçon 1 : Copie de la base de données exemple

Avant de commencer, faites une copie de la base de données exemple pour que vous puissiez restaurer cette dernière une fois les modifications apportées.

❖ Pour copier la base de données exemple

1. Allez dans le répertoire contenant le fichier de base de données exemple, *asademo.db*.

Le répertoire d'installation par défaut est *Program Files\Sybase\Adaptive Server Anywhere 9*.

2. Créez un sous-répertoire dans lequel placer la copie de *asademo.db*.
Par exemple, donnez-lui le nom *demoback*.

3. Créez une copie de *asademo.db* dans le répertoire de sauvegarde.
Travaillez maintenant avec la version originale. Do not change the backup copy.

Leçon 2 : Démarrage du serveur de base de données Adaptive Server Anywhere

Dans cette section, vous allez démarrer le serveur de base de données Adaptive Server Anywhere qui exécute la base de données exemple. La base de données exemple est stockée dans un fichier nommé *asademo.db* situé dans votre répertoire d'installation de SQL Anywhere.

En fonction du système d'exploitation utilisé, vous pourrez choisir la méthode de démarrage du serveur de base de données exécutant la base de données exemple.

❖ Pour démarrer le serveur personnel qui exécute la base de données exemple (Windows)

1. Dans le menu Démarrer, choisissez Programmes ► SQL Anywhere 9 ► Adaptive Server Anywhere ► Exemple de serveur personnel.

Le serveur personnel exécutant la base de données exemple est lancé. La fenêtre du serveur de base de données apparaît mais disparaît rapidement. Le serveur s'affiche alors sous forme d'icône dans la barre d'état système, à l'opposé du bouton Démarrer dans la barre des tâches.

❖ Pour démarrer le serveur qui exécute la base de données exemple (invite de commandes)

1. Ouvrez une fenêtre de commande.
2. Positionnez-vous dans le répertoire SQL Anywhere.
Sous les systèmes d'exploitation Windows, le répertoire d'installation par défaut est *C : \Program Files \Sybase \SQL Anywhere 9*.
3. Démarrez le serveur de base de données qui exécute la base exemple.

Le mode de démarrage du serveur de base de données dépend de votre système d'exploitation et varie selon que vous souhaitez ou pas vous connecter à la base à partir d'autres machines sur le réseau.

- ◆ Si vous voulez uniquement vous connecter à partir de la même machine sous Windows ou sous UNIX, entrez la commande suivante pour démarrer le serveur personnel :

```
dbeng9 -n asademo9 asademo.db
```

- ◆ Si vous voulez vous connecter au serveur de base de données à partir d'autres machines du réseau sous Windows ou sous UNIX, entrez la commande suivante pour démarrer le serveur réseau :

```
dbsrv9 -n asademo9 asademo.db
```

-
- ◆ Sous NetWare, seul le serveur réseau est proposé. Entrez la commande suivante :

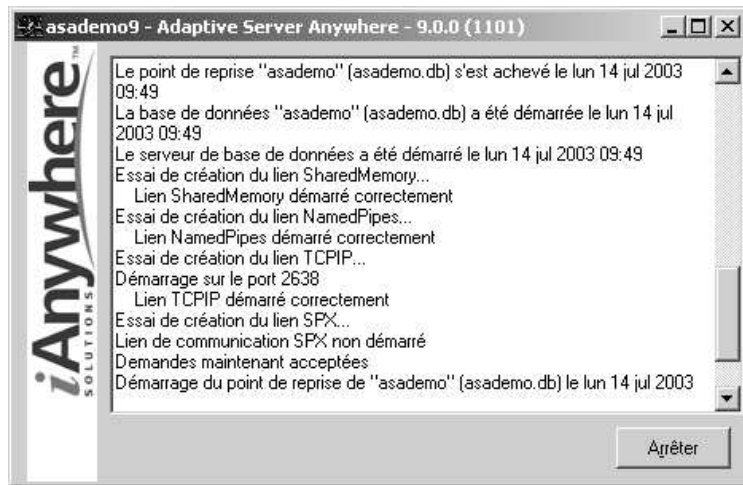
```
load dbsrv9.nlm -n asademo9 asademo.db
```

Le serveur est affiché sous forme d'icône dans la barre d'état système, à l'opposé du bouton Démarrer dans la barre des tâches.

Leçon 3 : Affichage de la fenêtre du serveur de base de données

Vous avez démarré le serveur qui exécute la base de données exemple. Cependant, vous ne pouvez pas encore visualiser, ni manipuler les données de la base.

L'icône Adaptive Server Anywhere est la seule indication du démarrage du serveur de base de données. Pour afficher maintenant (ou ultérieurement) la fenêtre du serveur de base de données dans Windows, double-cliquez sur l'icône Adaptive Server Anywhere dans la barre d'état système.



La fenêtre du serveur contient des informations utiles :

- ◆ **Nom du serveur** Le nom affiché dans la barre de titre (ici **asademo9**) est le **nom du serveur**. Un nom de serveur est affecté à chaque démarrage du serveur de base de données. Ce nom peut être utilisé par les applications lors de leur connexion à une base de données.
- ◆ **Informations de démarrage** Au démarrage, un serveur de base de données réserve de la mémoire pour le traitement des requêtes adressées à la base de données. Cette mémoire est appelée **mémoire cache**. La taille de la mémoire cache est affichée dans la fenêtre. Le cache est structuré en **pages** de taille fixe : cette taille est également affichée.
- ◆ **Informations sur la base de données** Les noms du fichier de base de données et du journal de transactions associé apparaissent dans la fenêtre.

Dans ce cas, la taille du cache de démarrage et la taille de page sont les valeurs par défaut. Dans la plupart des cas, y compris ceux présentés dans ce

didacticiel, les options de démarrage par défaut conviennent.

Leçon 4 : Fermeture de la base de données

Vous pouvez maintenant fermer le serveur de base de données que vous venez de démarrer.

❖ **Pour arrêter le serveur de base de données exécutant la base de données exemple (Windows)**

1. Double-cliquez sur l'icône Adaptive Server Anywhere dans la barre des tâches de Windows.

La fenêtre du serveur de base de données s'affiche.

2. Cliquez sur Arrêter.

Résumé

Dans ce didacticiel, vous avez appris à réaliser une copie de la base de données exemple, à démarrer le serveur exécutant la base de données exemple et à afficher le contenu de la fenêtre du serveur de base de données. Enfin, vous avez appris comment fermer le serveur de base de données.

CHAPITRE 12

Etablissement de la connexion

Présentation Ce chapitre décrit comment créer une source de données ODBC et comment l'utiliser pour établir une connexion entre une base de données et une application. Dans ce cas, l'application utilisée est Sybase Central.

Sommaire	Sujet :	page
	Connexions	158
	Sources de données ODBC	159
	Leçon 1 : Création d'une source de données ODBC	160
	Connexion à l'aide de la source de données ODBC exemple	163
	Autres méthodes de connexion	164
	Déconnexion de la base de données exemple	165
	Résumé	166

Connexions

Toute application cliente qui utilise une base de données doit s'y connecter avant de pouvoir entreprendre quoi que ce soit. Une fois établie, la **connexion** constitue une voie par laquelle transitent toutes les opérations que vous effectuez depuis l'application cliente. Par exemple, vos autorisations à exécuter diverses actions sur la base de données sont déterminées par votre ID utilisateur. Le serveur de base de données connaît cet ID car il fait partie de la demande d'établissement de connexion.

Sources de données ODBC

L'interface **ODBC (Open Database Connectivity)**, définie par Microsoft Corporation, est une interface standard permettant à des applications clientes d'accéder à des systèmes de gestion de bases de données dans les environnements Windows. Pour établir une connexion, vous devez en spécifier les paramètres. Il est souvent pratique de regrouper un jeu de paramètres de connexion et de le stocker dans une **source de données ODBC**. Les sources de données ODBC constituent une méthode pratique pour enregistrer les paramètres de connexion fréquemment utilisés.

Une fois que vous avez une source de données, il suffit de la nommer dans votre chaîne de connexion :

```
DSN=my data source
```

Vous pouvez utiliser des sources de données ODBC pour vous connecter à des bases Adaptive Server Anywhere à partir des applications suivantes :

- ◆ Sybase Central et Interactive SQL,
- ◆ tous les utilitaires Adaptive Server Anywhere,
- ◆ PowerAMC et InfoMaker,
- ◆ tout environnement de développement d'applications qui supporte ODBC, tel que Microsoft Visual Basic, Sybase PowerBuilder et Borland Delphi.
- ◆ toutes les applications clientes Adaptive Server Anywhere sous UNIX. Sous UNIX, la source de données est stockée sous la forme d'un fichier.

Leçon 1 : Création d'une source de données ODBC

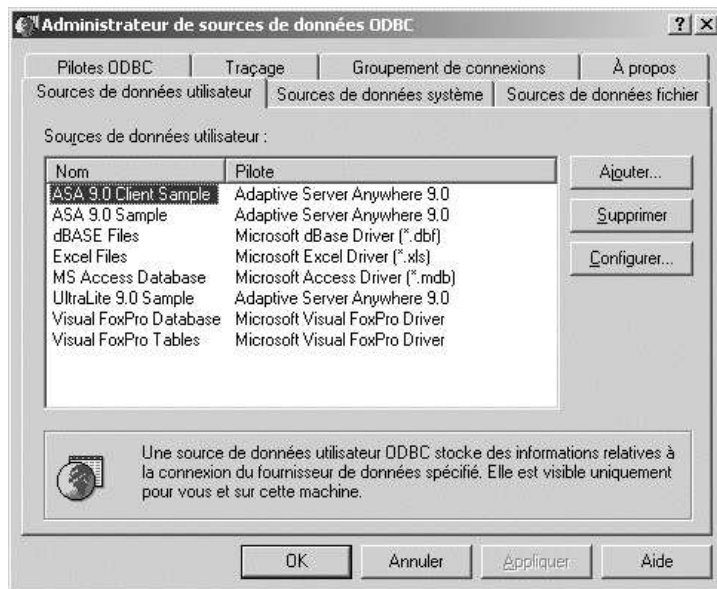
Cette section explique comment créer une source de données ODBC simple sous Windows.

❖ Pour créer une source de données ODBC simple

1. Démarrez l'administrateur ODBC.

- ◆ Dans le menu Démarrer, choisissez Programmes ► SQL Anywhere 9 ► Adaptive Server Anywhere ► Administrateur ODBC.

L'Administrateur de source de données ODBC apparaît, avec une liste des sources de données actuellement installées sur votre ordinateur. Par exemple :

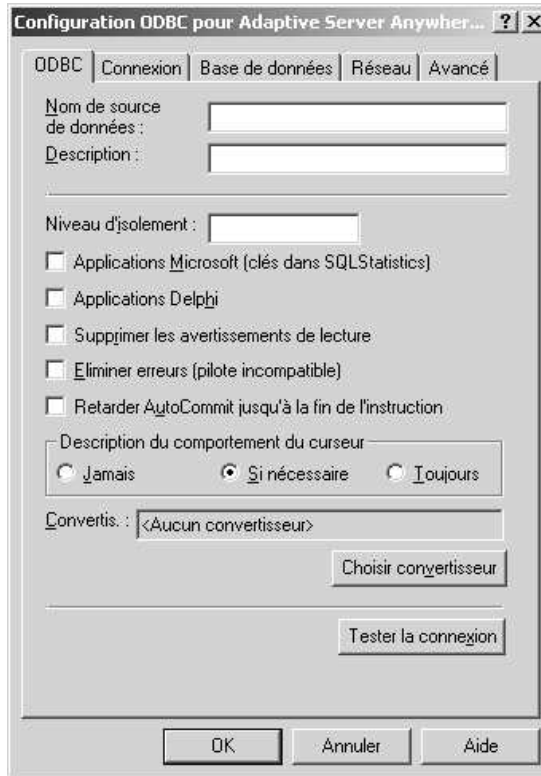


2. Dans l'onglet Sources de données utilisateur, cliquez sur Ajouter.

L'assistant de création d'un groupe apparaît.

3. Sélectionnez **Adaptive Server Anywhere 9.0** dans la liste des pilotes et cliquez sur Terminer.

La fenêtre Configuration ODBC pour Adaptive Server Anywhere apparaît.



Plusieurs champs de cette fenêtre sont facultatifs. Cliquez sur le bouton Aide au bas de chaque onglet pour plus d'informations sur les champs qu'il contient. Dans la plupart des cas, vous n'avez besoin d'utiliser que quelques paramètres. Les paramètres de source de données suivants suffisent pour la base de données exemple Adaptive Server Anywhere :

- ◆ **Nom source données (onglet ODBC)** Entrez **Sample** comme nom de source de données. Il s'agit du nom qui apparaîtra dans la boîte de dialogue Connexion. Il peut comporter des espaces, mais doit être court.
- ◆ **ID utilisateur (onglet Connexion)** Entrez l'ID utilisateur par défaut **DBA**. Il s'agit de l'ID utilisateur de base de données utilisé pour la connexion.
- ◆ **Mot de passe (onglet Connexion)** Entrez **SQL**. Il s'agit du mot de passe par défaut de l'ID utilisateur DBA.
- ◆ **Fichier de base de données (onglet Base de données)** Ouvrez le répertoire d'installation SQL Anywhere, puis sélectionnez le fichier *asademo.db*. Il s'agit de la base de données exemple Adaptive Server Anywhere.

-
4. Cliquez sur OK pour créer la source de données et fermer la boîte de dialogue.

Connexion à l'aide de la source de données ODBC exemple

Dans cet exemple, vous allez connecter la base de données exemple à partir de Sybase Central, l'outil graphique d'Adaptive Server Anywhere pour gérer les bases de données.

❖ Pour vous connecter à la base de données exemple depuis Sybase Central

1. Pour démarrer Sybase Central, sélectionnez Démarrer ► SQL Anywhere 9 ► Sybase Central.

2. Choisissez Outils ► Connecter pour ouvrir la boîte de dialogue Nouvelle connexion.

Si une boîte de dialogue vous invite à choisir un module de connexion, choisissez Adaptive Server Anywhere 9, puis cliquez sur OK.

La boîte de dialogue Connexion apparaît.

3. Sélectionnez l'option Nom de la source de données ODBC.

4. Dans la zone Nom de source de données ODBC, tapez **Sample** et cliquez sur OK.

Vous pouvez aussi cliquer sur le bouton Parcourir et sélectionner ASA 9.0 Sample dans la boîte de dialogue Noms des sources de données.

5. Cliquez sur OK pour vous connecter à la base de données exemple.

Les boîtes de dialogue se ferment et la base de données s'ouvre dans la fenêtre Sybase Central.

Autres méthodes de connexion

Vous pouvez utiliser des sources de données ODBC pour vous connecter à Adaptive Server Anywhere, et ce, non seulement à partir d'applications ODBC, mais également à partir d'applications Embedded SQL, comme des utilitaires, mais également à partir d'Interactive SQL et de Sybase Central, et ce, même si ces applications n'utilisent pas ODBC. La fonctionnalité de traitement des sources de données ODBC est intégrée à la bibliothèque de clients Embedded SQL ainsi qu'à Sybase Central et Interactive SQL.

☞ Adaptive Server Anywhere supporte plusieurs interfaces de programmation en plus de l'interface ODBC. Pour plus d'informations, reportez-vous à la section [“Présentation des connexions” \[ASA - Guide d'administration, page 42\]](#).

La boîte de dialogue Connexion dans Sybase Central et dans Interactive SQL contient des champs permettant d'indiquer un nom de source ODBC ou un fichier source de données ODBC.

Déconnexion de la base de données exemple

Pour terminer ce didacticiel, fermez le serveur de base de données et Sybase Central.

❖ Pour fermer le serveur de base de données

1. Choisissez Outils ► Déconnecter.

Le serveur de base de données se déconnecte de la base de données exemple et se ferme.

❖ Pour fermer Sybase Central

1. Choisissez Fichier ► Quitter.

Sybase Central se ferme.

Résumé

Ce didacticiel vous a appris à créer une source de données ODBC et à l'utiliser pour établir une connexion entre une base de données et Sybase Central.

CHAPITRE 13

Interactive SQL

Présentation

Ce chapitre aborde le démarrage et l'exploitation d'Interactive SQL. Interactive SQL est un utilitaire fourni avec Adaptive Server Anywhere qui permet d'exécuter des instructions SQL, de créer des scripts et d'afficher les données de la base.

Sommaire

Sujet :	page
A propos d'Interactive SQL	168
Leçon 1 : Démarrage d'Interactive SQL	169
Leçon 2 : Interface Interactive SQL	170
Leçon 3 : Affichage de données avec Interactive SQL	176
Leçon 4 : Utilisation des instructions SQL	183
Résumé	191

A propos d'Interactive SQL

Voici les tâches qu'Interactive SQL est susceptible d'accomplir :

- ◆ envoi d'instructions SQL au serveur de base de données ;
- ◆ consultation d'informations dans une base de données ;
- ◆ test des instructions SQL que vous prévoyez d'intégrer à une application ;
- ◆ chargement des données dans une base et exécution de tâches d'administration.

De plus, Interactive SQL peut exécuter des **fichiers de commandes** ou des **fichiers script**. Par exemple, vous pouvez construire des scripts exécutables à répétition dans une base de données, puis les exécuter en mode batch via Interactive SQL.

Leçon 1 : Démarrage d'Interactive SQL

Dans cette section, vous allez démarrer l'utilitaire Interactive SQL, vous connecter à la base de données exemple et entrer une commande. Nous vérifions également que la connexion fonctionne.

❖ Pour démarrer Interactive SQL et vous connecter à la base de données exemple (Windows)

1. Dans le menu Démarrer, choisissez Programmes ► SQL Anywhere ► 9 ► Adaptive Server Anywhere ► Interactive SQL.

La boîte de dialogue Connecter apparaît.

2. Cliquez sur l'option Nom de la source de données ODBC.
3. Cliquez sur Parcourir, puis sélectionnez **ASA 9.0 Sample** dans la liste.
4. Cliquez sur OK.
5. Cliquez sur OK pour vous connecter à la base de données exemple.

❖ Pour vous connecter à la base de données exemple et démarrer Interactive SQL (invite de commandes)

1. Si vous utilisez un système d'exploitation Windows ou UNIX, entrez la commande suivante pour démarrer le serveur de base de données personnel :

```
dbeng9 -n asademo9 asademo.db
```

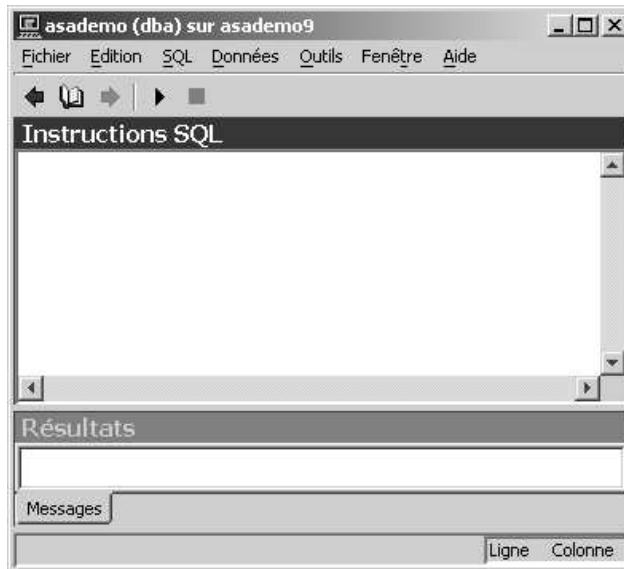
2. Pour démarrer Interactive SQL et vous connecter à la base de données exemple, entrez la commande ci-dessous à l'invite de commandes :

```
dbisql -c "dsn=ASA 9.0 Sample"
```

Leçon 2 : Interface Interactive SQL

Description de la fenêtre principale Interactive SQL

Une fois la connexion établie, la fenêtre Interactive SQL s’affiche.



Le nom de la base de données, l’ID utilisateur et le nom de serveur apparaissent dans la barre de titre, confirmant que la connexion à la base de données a été établie.

Volets Interactive SQL

Interactive SQL inclut les volets suivants :

- ◆ **Instructions SQL** Dans cette zone, vous pouvez saisir des instructions SQL pour accéder à vos données et les modifier. La barre de titre au-dessus du volet Instructions SQL affiche des informations sur la connexion en cours.
- ◆ **Résultats** Ce volet contient les résultats des commandes exécutées. Par exemple, si vous recherchez des données à l’aide d’instructions SQL, les lignes et les colonnes répondant aux critères de recherche s’affichent dans l’onglet Résultats de ce volet. Si les informations ne tiennent pas toutes dans le volet, des barres de défilement apparaissent automatiquement. Vous pouvez modifier le jeu de résultats sur l’onglet Résultats.
☞ Pour plus d’informations sur la modification du jeu de résultats, reportez-vous à la section “[Modification des valeurs de table dans Interactive SQL](#)” à la page 178.

L'onglet Messages affiche des messages du serveur de base de données relatifs aux instructions SQL que vous exécutez dans Interactive SQL.

Dans le volet Résultats, les onglets Plan et Plan UltraLite affichent le plan d'exécution de l'optimiseur de requête pour une instruction SQL.

Vous pouvez définir des options pour les onglets et les volets dans la fenêtre principale Interactive SQL à partir de la boîte de dialogue Options du menu Outils.

La barre de titre en haut de la fenêtre affiche les informations de connexion suivantes :

```
database-name ( userid ) on server-name
```

Par exemple, si vous vous connectez à la base de données exemple à l'aide de la source de données ODBC exemple ASA 9.0, la barre de titre se présente comme suit :

```
asademo ( dba ) on asademo9
```

Utilisation de la barre d'outils Interactive SQL

La barre d'outils Interactive SQL s'affiche en haut de la fenêtre Interactive SQL. Elle contient des boutons permettant l'exécution des commandes les plus courantes. A partir de cette barre d'outils, vous pouvez :

- ◆ rappeler l'instruction SQL exécutée immédiatement avant votre position dans la liste historique ;
- ◆ afficher la liste des 50 dernières instructions SQL exécutées ;
- ◆ rappeler l'instruction SQL exécutée immédiatement après votre position dans la liste historique ;
- ◆ exécuter l'instruction SQL actuellement affichée dans le volet Instructions SQL ;
- ◆ interrompre l'exécution de l'instruction SQL en cours.

Lorsque vous placez le curseur au-dessus de ces boutons, vous obtenez une brève description des fonctions qu'ils exécutent.

Ouverture de plusieurs fenêtres

Dans Interactive SQL, vous pouvez ouvrir plusieurs fenêtres. Chacune correspond à une connexion à une base de données. Vous pouvez ainsi vous connecter simultanément à plusieurs bases sur des serveurs différents ou établir des connexions simultanées à une même base de données.

❖ Pour ouvrir une nouvelle fenêtre Interactive SQL

1. Choisissez Fenêtre ► Nouvelle fenêtre.

La boîte de dialogue Connecter apparaît.

2. Dans la boîte de dialogue Connexion, entrez les options de connexion puis cliquez sur OK pour établir la connexion.

Les informations de connexion (notamment le nom de la base de données, votre ID utilisateur et le nom du serveur de base de données) apparaissent dans la barre de titre, au-dessus de la fenêtre Instructions SQL.

Vous pouvez aussi vous connecter ou vous déconnecter d'une base avec les commandes Connecter et Déconnecter du menu SQL ou en exécutant une instruction CONNECT ou DISCONNECT.

Informations complémentaires sur Interactive SQL

Interactive SQL est doté d'un ensemble de fenêtres qui permettent de configurer son apparence et son comportement. Pour plus d'informations, reportez-vous à la section "[Aide d'Interactive SQL](#)" [*SQL Anywhere Studio - Aide*, page 157].


A partir d'Interactive SQL, vous pouvez accéder à l'assistant de paramétrage d'index, également disponible depuis Sybase Central, conçu pour vous aider à améliorer les performances des requêtes. Pour plus d'informations, reportez-vous à la section "[Présentation de l'Assistant de paramétrage d'index](#)" [*ASA - Guide de l'utilisateur SQL*, page 73].

Depuis Interactive SQL, vous pouvez également accéder à l'éditeur de requête qui permet de concevoir, analyser et tester tout type de requête. Pour plus d'informations, reportez-vous à la section "[Aide de l'Editeur de requête](#)" [*SQL Anywhere Studio - Aide*, page 237].

Raccourcis clavier Interactive SQL

Interactive SQL offre les raccourcis clavier suivants :

Touche de fonction	Description
ALT+F4	Ferme Interactive SQL.
ALT+GAUCHE	Affiche l'instruction SQL précédente dans la liste historique.
ALT+DROITE	Affiche l'instruction SQL suivante dans la liste historique.

Touche de fonction	Description
CTRL+ATTN	Interrompt l'instruction SQL qui est en cours d'exécution.
CTRL+C	Dans le volet Résultats, copie la ou les ligne(s) et en-têtes de colonne sélectionnés dans le Presse-papiers. Dans le volet Instructions SQL, copie le texte sélectionné dans le Presse-papiers.
CTRL+FIN	Place le curseur à la fin des données du volet.
CTRL+H	Affiche l'historique des instructions SQL exécutées.
CTRL+DÉBUT	Place le curseur en haut du volet actif.
CTRL+N	Efface le contenu de la fenêtre Interactive SQL.
CTRL+P	Imprime le contenu du volet Instructions SQL. Vous pouvez configurer l'affichage du texte imprimé dans la boîte de dialogue Options d'Interactive SQL.  Pour plus d'informations sur la définition des options d'impression, reportez-vous à la section "Onglet Impression" [<i>SQL Anywhere Studio - Aide</i> , page 171].
CTRL+Q	Affiche l'éditeur de requête. L'éditeur de requête permet de construire des requêtes SQL. Lorsque vous avez fini de construire votre requête, cliquez sur OK pour l'exporter dans le volet Instructions SQL.
CTRL+S	Enregistre le contenu du volet Instructions SQL.
ECHAP	Efface le volet Instructions SQL.
F1	Ouvre l'aide.
F2	Modifie la valeur sélectionnée dans le jeu de résultats. A l'aide de la touche Tab, vous pouvez vous déplacer d'une colonne à l'autre à l'intérieur de la ligne.
F5	Exécute tout le texte figurant dans le volet Instructions SQL. Vous pouvez aussi cliquer sur le bouton Exécuter l'instruction SQL de la barre d'outils.

Touche de fonction	Description
F7	Affiche la boîte de dialogue Consulter le nom de table. Dans cette boîte de dialogue, vous pouvez rechercher et sélectionner une table, puis appuyer sur ENTRÉE pour insérer le nom de la table dans le volet Instructions SQL au niveau du curseur. Ou encore, après avoir sélectionné une table de la liste, appuyez à nouveau sur F7 pour afficher les colonnes de cette table. Vous pouvez ensuite sélectionner une colonne et appuyer sur ENTRÉE pour insérer le nom de la colonne dans le volet Instructions SQL au niveau du curseur.
F8	Affiche la boîte de dialogue Consulter le nom de procédure. Dans cette boîte de dialogue, vous pouvez rechercher et sélectionner une procédure, puis appuyer sur la touche ENTRÉE pour insérer le nom de la procédure dans le volet Instructions SQL au niveau du curseur.
F9	Exécute le texte sélectionné dans le volet Instructions SQL. Si aucun texte n'est sélectionné, toutes les instructions sont exécutées.
PG. SUIV	Descend d'une page dans les données du volet.
PG. PRÉC	Remonte d'une page dans les données du volet.
MAJ+F5	Affiche le plan de l'instruction dans le volet Instructions SQL sans exécuter l'instruction.

Les raccourcis clavier suivants sont disponibles lorsque le volet Instructions SQL est actif :

Touche de fonction	Description
CTRL+]]	Déplace le curseur jusqu'à l'accolade correspondante. La recherche d'accolade correspondante peut être utilisée pour les parenthèses, les accolades, les crochets et les chevrons.
CTRL+RET. ARR	Supprime le mot situé à gauche du curseur.

Touche de fonction	Description
CTRL+SUPPR	Supprime le mot situé à droite du curseur.
CTRL+R	Ouvre la boîte de dialogue Atteindre dans laquelle vous pouvez spécifier la ligne que vous voulez atteindre.
CTRL+L	Supprime la ligne courante dans le volet Instructions SQL et la place dans le Presse-papiers.
CTRL+MAJ+]]	Etend la sélection jusqu'à l'accolade correspondante. La recherche d'accolade correspondante peut être utilisée pour les parenthèses, les accolades, les crochets et les chevrons.
CTRL+MAJ+L	Supprime la ligne courante.
CTRL+MAJ+U	Remplace les caractères sélectionnés par des majuscules.
CTRL+U	Remplace les caractères sélectionnés par des minuscules.
F3	Recherche l'occurrence suivante du texte sélectionné.
DÉBUT	Place le curseur au début de la ligne courante ou sur le premier mot de la ligne courante.
MAJ+F3	Recherche l'occurrence précédente du texte sélectionné.
MAJ+DÉBUT	Etend la sélection au début du texte de la ligne courante.

Leçon 3 : Affichage de données avec Interactive SQL

Une des principales utilisations d'Interactive SQL consiste à parcourir les données de table. Cette section explique comment consulter des informations dans la base exemple. Les requêtes se présentent sous la forme d'instructions SQL.

Vous pouvez afficher les informations d'une base en utilisant l'instruction `SELECT` dans Interactive SQL. L'exemple suivant montre comment saisir des données dans le volet Instructions. Une fois que vous avez tapé la commande, vous devez cliquer sur le bouton Exécuter l'instruction SQL de la barre d'outils afin de l'exécuter.

Lorsque l'instruction est exécutée, les données qu'elle renvoie (ou jeu de résultats) s'affichent dans le volet Résultats. Utilisez les barres de défilement pour visualiser les zones de la table qui ne sont pas visibles dans le volet. Par défaut, les numéros de ligne apparaissent à gauche du jeu de résultats.

❖ Pour afficher toutes les lignes et colonnes de la table `employee`

1. Dans le volet Instructions SQL, tapez l'instruction SQL suivante :

```
SELECT * FROM employee
```

2. Appuyez sur F5 pour exécuter l'instruction.

La requête affiche toutes les colonnes de la table `employee`. Cette table contient des informations sur les employés d'une entreprise fictive. Vous voyez alors les éléments suivants :



Interactive SQL recherche les informations en envoyant une requête au serveur de base de données. Le serveur de base de données, à son tour, consulte les informations de la table employee et les renvoie à Interactive SQL.

A ce stade, vous pouvez consulter les données d'autres tables de la base de données exemple telles que les tables product, department et sales_order.

emp_id	manager id	emp_lname	emp_fname	...
102	501	Fran	Whitney	...
105	501	Matthew	Cobb	...
129	902	Philip	Chin	...
148	1293	Julie	Jordan	...
...				

Vous pouvez ajouter, supprimer et mettre à jour des lignes dans le jeu de résultats.

☞ Pour plus d'informations sur la modification du jeu de résultats,

reportez-vous à la section [“Modification des valeurs de table dans Interactive SQL”](#) à la page 178.

Modification des valeurs de table dans Interactive SQL

Lorsque vous avez exécuté une requête dans Interactive SQL, vous pouvez modifier le jeu de résultats pour modifier la base de données. Vous pouvez également sélectionner des lignes du jeu de résultats et les copier pour les utiliser dans d’autres applications. Interactive SQL prend en charge la modification, l’insertion et la suppression des lignes. Ces actions s’exécutent également à l’aide des instructions UPDATE, INSERT et DELETE.

Pour modifier une ligne ou une valeur dans le jeu de résultats, vous devez avoir les autorisations adéquates pour la table ou la colonne dont vous voulez modifier des valeurs. Par exemple, si vous voulez supprimer une ligne, vous devez détenir l’autorisation DELETE pour la table à laquelle la ligne appartient.

La modification du jeu de résultats échoue dans les cas suivants :

- ◆ Vous tentez de modifier une ligne ou une colonne pour laquelle vous n’avez pas d’autorisation.
- ◆ Vous sélectionnez les colonnes d’une table avec une clé primaire, mais vous ne sélectionnez pas toutes les colonnes clé primaire.
- ◆ Vous tentez de modifier le jeu de résultats d’une jointure (par exemple, le jeu de résultats inclut des données de plusieurs tables).
- ◆ Vous entrez une valeur incorrecte (par exemple, une chaîne de caractères dans une colonne numérique ou une valeur NULL dans une colonne qui ne l’autorise pas).

Lorsqu’une modification échoue, un message d’erreur Interactive SQL s’affiche pour expliquer la cause de l’erreur et les valeurs de la table de la base demeurent inchangées.

Lorsque vous avez modifié les valeurs d’une table, vous devez entrer une instruction COMMIT pour rendre ces modifications permanentes. A l’inverse, si vous voulez annuler vos modifications, vous devez exécuter l’instruction ROLLBACK.

☞ Pour plus d’informations, reportez-vous aux sections [“Instruction COMMIT”](#) [ASA - Manuel de référence SQL, page 332] et [“Instruction ROLLBACK”](#) [ASA - Manuel de référence SQL, page 603].

Modifications des valeurs de table dans le jeu de résultats Interactive SQL

A partir d’Interactive SQL, vous pouvez modifier certaines valeurs ou toutes

les valeurs dans les lignes existantes des tables de base de données. Vous devez détenir l'autorisation UPDATE pour les colonnes concernées par la mise à jour. Lorsque vous modifiez le jeu de résultats, vous devez modifier les valeurs ligne à ligne.

❖ Pour modifier une ligne dans le jeu de résultats

1. Cliquez sur la valeur que vous voulez modifier.
2. Cliquez dessus avec le bouton droit de la souris et sélectionnez Modifier dans le menu contextuel. Vous pouvez également appuyer sur F2 pour modifier le jeu de résultats.

Un curseur clignotant apparaît dans la cellule de la table contenant la valeur.

3. Entrez la nouvelle valeur.

Vous ne pouvez pas entrer des types de données incorrects dans une colonne. Par exemple, vous ne pouvez pas entrer une chaîne de texte dans une colonne qui accepte le type de données INT.

Si vous avez fini de modifier les valeurs dans la ligne, appuyez sur Entrée pour mettre à jour la base de données. Si vous voulez modifier d'autres valeurs dans la ligne, appuyez sur la touche TAB ou sur MAJ+TAB pour les atteindre.

Vous pouvez appuyer sur la touche ECHAP pour annuler la modification qui a été apportée à la valeur sélectionnée.

4. Exécutez l'instruction COMMIT pour que vos modifications deviennent permanentes.

Modification des colonnes calculées

Lorsque vous avez modifié des valeurs dans le jeu de résultats, la base de données est mise à jour avec les valeurs modifiées. Les colonnes calculées sont recalculées en fonction des valeurs modifiées, qu'elles fassent partie ou non du jeu de résultats. Toutefois, si votre jeu de résultats contient des colonnes calculées et que vous modifiez une valeur dans l'une d'entre elles, la base de données est mise à jour avec la valeur modifiée.

Insertion de lignes dans la base de données du jeu de résultats Interactive SQL

Interactive SQL permet d'ajouter des lignes dans les jeux de résultats. Vous passez d'une colonne à l'autre à l'aide de la touche Tab dans le jeu de résultats pour ajouter des valeurs à la ligne. Lorsque vous ajoutez des valeurs à la table, des caractères sont stockés dans la même case à mesure qu'ils sont entrés. Pour ajouter des lignes, vous devez détenir l'autorisation INSERT pour la table.

❖ Pour insérer une ligne dans le jeu de résultats

1. Cliquez avec le bouton droit sur le jeu de résultats et sélectionnez Ajouter dans le menu contextuel.

Une nouvelle ligne vide apparaît avec un curseur clignotant sur sa première valeur.

Appuyez sur la touche TAB pour déplacer le curseur d'une colonne à l'autre le long de la ligne. Vous pouvez également insérer une valeur en cliquant dessus dans la ligne sélectionnée.

2. Entrez la nouvelle valeur.
Vous ne pouvez pas entrer des types de données incorrects dans une colonne. Par exemple, vous ne pouvez pas entrer une chaîne de texte dans une colonne qui accepte le type de données INT.
3. Appuyez sur la touche TAB pour atteindre la colonne suivante.
4. Reprenez les étapes 2 et 3 jusqu'à ce que vous ayez ajouté toutes les valeurs de colonne requises.
5. Appuyez sur ENTRÉE pour mettre la base de données à jour.
6. Exécutez l'instruction COMMIT pour que vos modifications deviennent permanentes.

Insertion de valeurs dans des colonnes calculées

Si le jeu de résultats contient une colonne calculée pour laquelle vous n'avez spécifié aucune valeur, cette dernière est calculée lorsque la base de données est mise à jour. En revanche, si vous spécifiez une valeur pour la colonne calculée, la base de données est mise à jour avec cette valeur et cette dernière n'est pas calculée.

Insertion de lignes à l'aide de l'instruction INPUT

Vous pouvez également insérer des lignes à partir du jeu de résultats dans Interactive SQL à l'aide de l'instruction INPUT avec la clause PROMPT. Lorsque la clause PROMPT est spécifiée, Interactive SQL vous invite à fournir la valeur de chaque colonne de la table. Par exemple, pour ajouter une ligne dans la table product et afficher une invite vous demandant de saisir les valeurs de chaque colonne, vous devez exécuter l'instruction suivante dans Interactive SQL :

```
INPUT INTO product  
PROMPT
```

Suppression de lignes de la base de données à l'aide d'Interactive SQL

Vous pouvez également supprimer des lignes d'une table de base de données dans Interactive SQL. Pour supprimer des lignes, vous devez détenir

l'autorisation DELETE pour la table. Vous ne pouvez sélectionner que des lignes qui se suivent dans le jeu de résultats.

❖ Pour supprimer une ligne dans le jeu de résultats

1. Sélectionnez la ou les lignes à supprimer. Pour sélectionner une ou plusieurs ligne(s) :
 - ◆ Appuyez sur la touche MAJ et maintenez-la enfoncée pendant que vous cliquez sur la ou les lignes.
 - ◆ Appuyez sur la touche MAJ et maintenez-la enfoncée pendant que vous utilisez la flèche vers le haut ou vers le bas.

Si les lignes ne se suivent pas, vous devez les supprimer une à une.
2. Cliquez sur le jeu de résultats avec le bouton droit de la souris et sélectionnez Supprimer dans le menu contextuel. Vous pouvez également supprimer la ou les lignes sélectionnée(s) en appuyant sur la touche SUPPR.

La ou les es ligne(s) est/sont retirée(s) de la base de données.
3. Exécutez l'instruction COMMIT pour que vos modifications deviennent permanentes.

Copie de lignes du jeu de résultats Interactive SQL

Vous pouvez copier directement des lignes du jeu de résultats dans Interactive SQL, puis les coller dans d'autres applications. La copie de lignes inclut aussi les en-têtes de colonne. Les données copiées sont délimitées par des virgules, ce qui permet à d'autres applications, telles que Microsoft Excel, de les formater correctement. Les données copiées sont au format ASCII et toutes les chaînes sont entourées de guillemets simples. Vous ne pouvez sélectionner que des lignes qui se suivent dans le jeu de résultats.

❖ Pour copier des lignes du jeu de résultats Interactive SQL

1. Sélectionnez la ou les lignes à copier : Pour sélectionner une ou plusieurs ligne(s) :
 - ◆ Appuyez sur la touche MAJ et maintenez-la enfoncée pendant que vous cliquez sur la ou les lignes.
 - ◆ Appuyez sur la touche MAJ et maintenez-la enfoncée pendant que vous utilisez la flèche vers le haut ou vers le bas.
2. Cliquez avec le bouton droit sur le jeu de résultats et sélectionnez Copier dans le menu contextuel. Vous pouvez également copier les lignes sélectionnées en appuyant sur CTRL+C.

Les lignes et leurs en-têtes de colonne sont copiées dans le Presse-papiers. Vous pouvez les coller dans d'autres applications.

Copie de valeurs spécifiques du jeu de résultats

Pour copier une seule valeur du jeu de résultats, sélectionnez la valeur à copier, cliquez avec le bouton droit sur le jeu de résultats et choisissez Copier cellule dans le menu contextuel. Dans ce cas, aucun en-tête de colonne n'est copié—seules les données sont copiées.

Leçon 4 : Utilisation des instructions SQL

Les sections suivantes décrivent certaines commandes utilisables dans Interactive SQL. Cette section présente les tâches générales permettant d'utiliser des commandes dans Interactive SQL.

Toutes les instructions SQL peuvent être saisies comme des commandes dans le volet supérieur de la fenêtre Interactive SQL. Lorsque vous avez fini la saisie, vous devez exécuter l'instruction pour afficher ses résultats.

☞ Vous pouvez entrer n'importe quelle instruction SQL sur la base de données à partir d'Interactive SQL. Pour obtenir la liste complète des instructions SQL, reportez-vous à la section “[Instructions SQL](#)” [[ASA - Manuel de référence SQL, page 255](#)].

❖ Pour exécuter une instruction SQL

1. Appuyez sur le bouton Exécuter l'instruction SQL, ou choisissez SQL ► Exécuter ou appuyez sur F5.

❖ Pour effacer le contenu du volet Instructions SQL

1. Choisissez Edition ► Effacer SQL ou appuyez sur ECHAP.

Annulation d'une commande Interactive SQL

Le bouton Interrompt l'instruction SQL sur la barre d'outils Interactive SQL annule une commande.

Une opération d'annulation interrompt le traitement en cours et vous invite à entrer une nouvelle commande. Si un fichier de commandes était en cours de traitement, ou s'il y avait plusieurs instructions dans le volet Instructions SQL, les choix suivants vous sont proposés : Interrompre fichier de commandes, Continuer ou Quitter Interactive SQL. Ces actions peuvent être contrôlées par l'option ON_ERROR d'Interactive SQL.

☞ Pour plus d'informations sur l'option ON_ERROR, reportez-vous à la section “[Option ON_ERROR \[Interactive SQL\]](#)” [[ASA - Guide d'administration, page 687](#)].

Messages signalés

Selon le moment où une interruption est détectée, l'un des trois messages suivants s'affiche :

1. Si l'interruption est détectée alors qu'Interactive SQL (et non pas le serveur de base de données) traite la requête, le message suivant apparaît :
Commande ISQL interrompue par l'utilisateur

Interactive SQL interrompt immédiatement le traitement, sans incidence sur la transaction de base de données en cours.

2. Si l'interruption est détectée par le serveur pendant le traitement d'une commande de manipulation de données (SELECT, INSERT, DELETE ou UPDATE), le message suivant apparaît :

Instruction interrompue par l'utilisateur.

Les effets de la commande en cours sont annulés, mais le reste de la transaction n'est pas concerné.

3. Si l'interruption est détectée alors que le serveur de base de données traite une commande de définition de données (CREATE, DROP, ALTER, etc.), le message suivant apparaît :

L'utilisateur a mis fin à l'opération - la transaction a été annulée

Comme toutes les commandes de définition de données exécutent automatiquement un COMMIT avant le début de la commande, l'instruction ROLLBACK annule simplement la commande en cours.

Ce message apparaît aussi lorsque le serveur de base de données fonctionne en mode "bulk" et exécute une commande qui modifie la base (INSERT, UPDATE et DELETE). Dans ce cas, ROLLBACK annule non seulement la commande en cours, mais également toutes les opérations depuis le dernier COMMIT. Dans certains cas, le serveur de bases de données peut nécessiter un temps considérable pour exécuter le ROLLBACK automatique.

Exécution d'instructions multiples

L'environnement Interactive SQL permet d'entrer plusieurs instructions à la fois. Il suffit de terminer chaque instruction par un délimiteur de commande. Le délimiteur des commandes est une option configurable dans Interactive SQL que vous pouvez modifier à l'aide de l'option `COMMAND_DELIMITER`. Par défaut, le point virgule (;) est utilisé.

☞ Pour plus d'informations, reportez-vous à la section "[Option `COMMAND_DELIMITER` \[Interactive SQL\]](#)" [*ASA - Guide d'administration*, page 655].

❖ Pour entrer plusieurs instructions dans le volet Instructions SQL

1. Entrez les trois commandes suivantes dans le volet Instructions SQL.

```
UPDATE employee
SET dept_id = 400,
    manager_id = 1576
WHERE emp_id = 467;

UPDATE employee
SET dept_id = 400,
    manager_id = 1576
WHERE emp_id = 195;

SELECT *
FROM employee
WHERE emp_id IN ( 195, 467 );
```

2. Dans la barre d'outils, cliquez sur le bouton Exécuter l'instruction SQL. Les trois instructions sont exécutées. Une fois l'exécution terminée, les commandes restent dans le volet Instructions SQL.
3. Annulez vos modifications en entrant ROLLBACK dans le volet Instructions SQL et en exécutant l'instruction.

Utilisation de go comme alternative

Vous avez le choix entre l'utilisation du point-virgule (;) ou la saisie du terme go tout seul sur une ligne, au début de la ligne.

```
UPDATE employee
SET dept_id = 400,
    manager_id = 1576
WHERE emp_id = 467
go

UPDATE employee
SET dept_id = 400,
    manager_id = 1576
WHERE emp_id = 195
go

SELECT *
FROM employee
WHERE emp_id IN ( 195, 467 )
go
```

Conseil

Vous pouvez appuyer sur F9 pour exécuter uniquement le texte sélectionné dans le volet Instructions SQL.

Consultation des tables, des colonnes et des procédures

Lorsque vous entrez des commandes dans Interactive SQL, vous pouvez consulter les noms des tables, des colonnes ou des procédures stockés dans la base courante et les insérer à l'endroit où se trouve le curseur.

❖ Pour consulter les noms des tables de la base

1. Choisissez Outils ► Consulter le nom de table ou appuyez sur F7.
2. Recherchez la table et sélectionnez-la.
3. Cliquez sur OK pour insérer le nom de la table dans le volet Instructions SQL à la position actuelle du curseur.

❖ Pour consulter les noms des colonnes de la base

1. Choisissez Outils ► Consulter le nom de table ou appuyez sur F7.
2. Recherchez et sélectionnez la table contenant la colonne.
3. Cliquez sur Afficher les colonnes.
4. Sélectionnez la colonne et cliquez sur OK pour insérer son nom dans le volet Instructions SQL à la position actuelle du curseur.

❖ Pour consulter les noms des procédures de la base

1. Choisissez Outils ► Consulter le nom de la procédure ou appuyez sur F8.
2. Recherchez et sélectionnez la procédure.
3. Cliquez sur OK pour insérer le nom de la procédure dans le volet Instructions SQL à la position actuelle du curseur.

Dans les boîtes de dialogue de consultation des tables et des procédures, vous pouvez entrer les premiers caractères du nom correspondant. Vous voyez apparaître dans la boîte de dialogue la liste des seuls éléments qui commencent par le texte indiqué.

Pour affiner la recherche, vous pouvez utiliser les caractères joker SQL % (signe de pourcentage) et “_” (trait de soulignement). Le signe % correspond à toute chaîne comprenant aucun ou plusieurs caractères, tandis que le trait de soulignement (“_”) correspond à tout type de caractère.

Par exemple, pour répertorier toutes les tables contenant le mot ‘profil’, tapez **%profile%**.

Si vous souhaitez rechercher un signe de pourcentage ou un caractère souligné dans un nom de table, vous devez ajouter un caractère d'échappement en guise de préfixe au signe pourcentage ou au caractère souligné. Le caractère d'échappement dépend du pilote JDBC que vous utilisez. Si vous êtes connecté via jConnect, ce caractère d'échappement est une barre oblique inverse “\” et, avec le pilote JDBC iAnywhere, il s'agit d'un tilde “~”.

Impression d'instructions SQL

Vous pouvez imprimer le contenu du volet Instructions SQL en appuyant sur CTRL+P ou en sélectionnant Imprimer dans le menu Fichier. Dans la boîte de dialogue Options d'Interactive SQL, vous avez la possibilité d'ajouter un en-tête ou un pied de page et de configurer d'autres options de mise en forme.

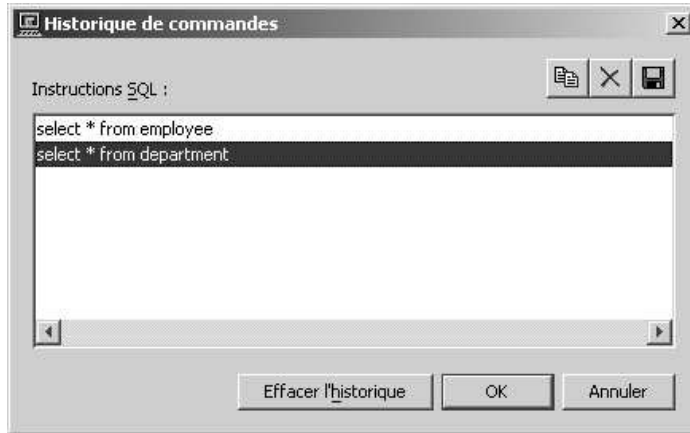
☞ Pour plus d'informations sur la configuration des options d'impression, reportez-vous à la section “[Onglet Impression](#)” [*SQL Anywhere Studio - Aide*, page 171].

☞ Pour plus d'informations sur l'impression de plans graphiques dans Interactive SQL, reportez-vous à la section “[Plans graphiques](#)” [*ASA - Guide de l'utilisateur SQL*, page 497].

Rappel de commandes

Lorsque vous exécutez une commande, Interactive SQL l'enregistre automatiquement dans une liste historique qui est conservée d'une session Interactive SQL à l'autre. Interactive SQL garde en mémoire les cinquante dernières commandes.

Vous pouvez afficher la liste complète de ces commandes dans la boîte de dialogue Historique de commandes. Pour y accéder, appuyez sur CTRL+H ou cliquez sur l'icône en forme de livre dans la barre d'outils.



Les commandes les plus récentes apparaissent en bas de la liste. Pour rappeler une commande, sélectionnez-la et cliquez sur OK. Elle apparaît dans le volet Instructions SQL d'Interactive SQL.

Vous pouvez aussi rappeler des commandes sans passer par la boîte Historique de commandes. Utilisez les flèches de la barre d'outils pour parcourir vos commandes ou appuyez sur ALT+DROITE et ALT+GAUCHE.

Remarque

Si vous exécutez une instruction SQL qui contient des informations relatives au mot de passe (GRANT CONNECT, GRANT REMOTE DBA, CONNECT ou CREATE EXTERNLOGIN), ces dernières apparaissent dans la boîte de dialogue Historique de commandes pendant la session Interactive SQL courante.

Lorsque vous consultez l'historique de commandes lors des sessions Interactive SQL suivantes, les mots de passe sont remplacés par ... dans toute instruction qui contient des informations sur le mot de passe. Par exemple, si vous exécutez l'instruction suivante dans Interactive SQL :

```
GRANT CONNECT TO testuser IDENTIFIED BY testpassword
```

Voici ce qui apparaît dans la boîte de dialogue Historique de commandes lors des sessions Interactive SQL suivantes :

```
GRANT CONNECT TO testuser IDENTIFIED BY ...
```

Lorsque Interactive SQL enregistre cette historique entre les sessions, il supprime les informations relatives au mot de passe à partir des instructions répertoriées ci-dessus.

Suppression de commandes dans la boîte de dialogue Historique de commandes

CTRL+C ou en cliquant sur le bouton Copier quand la boîte de dialogue est active. Pour copier des commandes sélectionnées dans le volet Instructions SQL, cliquez sur OK. Lorsque vous copiez plusieurs commandes, celles-ci sont séparées par un délimiteur de commande (par défaut, un point-virgule).

Le contenu de la boîte de dialogue Historique de commandes est conservé d'une session Interactive SQL à l'autre. Il existe deux façons de supprimer des commandes de la boîte de dialogue :

- ◆ Sélectionnez au moins une commande puis cliquez sur le bouton Supprimer ou appuyez sur la touche SUPPR pour supprimer les commandes sélectionnées de la boîte de dialogue. Cette action ne peut pas être annulée.
- ◆ Supprimez toutes les commandes de la boîte de dialogue en cliquant sur le bouton Effacer l'historique. Cette action ne peut pas être annulée.

Vous pouvez aussi enregistrer les commandes dans des fichiers texte afin de les utiliser au cours d'une session Interactive SQL ultérieure.

❖ Pour enregistrer l'historique de commandes dans un fichier

1. Ouvrez la boîte de dialogue Historique de commandes.
2. Cliquez sur le bouton Enregistrer ou appuyez sur CTRL+S.
3. Dans la boîte de dialogue Enregistrer sous, spécifiez un emplacement et un nom pour le fichier.
Le fichier d'historique de commande porte l'extension *.SQL*.
4. Cliquez sur Enregistrer.

Consignation des commandes

Grâce à la fonction de consignation d'Interactive SQL, vous pouvez garder un enregistrement des commandes à mesure de leur exécution.

Interactive SQL effectue cette consignation jusqu'à ce que vous y mettiez un terme ou jusqu'à la fin de la session. Les commandes sont ainsi enregistrées dans un fichier journal.

❖ Pour lancer la consignation des commandes Interactive SQL

1. Choisissez SQL ► Démarrer la consignation.
2. Dans la boîte de dialogue Enregistrer sous, spécifiez un emplacement et un nom pour le journal.
Un journal doit avoir l'extension *.SQL*.

3. Cliquez sur Enregistrer.

❖ **Pour arrêter la consignation des commandes Interactive SQL**

1. Choisissez SQL ► Arrêter la consignation.

Conseils

Vous pouvez aussi démarrer et arrêter la consignation en entrant les instructions SQL adéquates dans le volet Instructions SQL. Pour démarrer le processus, saisissez et exécutez **START LOGGING** 'c :|*nom_fichier.sql*', où *c :|nom_fichier.sql* indique le chemin, le nom et l'extension du journal. Un journal doit avoir l'extension *.SQL*. Si le chemin contient des espaces, vous devez mettre cette chaîne entre apostrophes. Pour arrêter la consignation, saisissez et exécutez **STOP LOGGING**.

Une fois que la consignation commence, toutes les commandes que vous tentez d'exécuter sont consignées, y compris celles qui ne se déroulent pas correctement.

Résumé

Ce chapitre vous avez montré comment démarrer Interactive SQL, exécuter des requêtes SQL et afficher les données d'une base de données.

CHAPITRE 14

Gestion des bases de données avec Sybase Central

Présentation

Ce chapitre présente Sybase Central, l'outil de gestion de bases de données à interface graphique d'Adaptive Server Anywhere. Il explique brièvement comment exploiter Sybase Central pour gérer les propriétés des bases de données.

Vous apprendrez comment démarrer Sybase Central et vous connecter à la base de données exemple, afficher et modifier la base de données exemple, modifier les tables, gérer les utilisateurs et les groupes, utiliser les procédures stockées et sauvegarder votre base de données.

Avant de commencer, faites une copie de la base de données exemple afin de pouvoir la restaurer à son état initial lorsque vous aurez terminé. Pour plus d'informations sur la copie de la base de données exemple, reportez-vous à la section "[Leçon 1 : Copie de la base de données exemple](#)" à la page 150.

Sommaire

Sujet :	page
Sybase Central	194
Leçon 1 : Démarrage de Sybase Central et connexion	195
Leçon 2 : Interface Sybase Central	196
Leçon 3 : Affichage de la base de données exemple	198
Leçon 4 : Création et modification de tables	203
Leçon 5 : Gestion des utilisateurs et des groupes	205
Leçon 6 : Visualisation et modification des procédures stockées	207
Leçon 7 : Sauvegarde d'une base de données	210
Restauration de la base de données exemple	211
Résumé	212

Sybase Central

Sybase Central est un outil de gestion de bases de données à interface graphique qui fournit sous une forme conviviale les utilitaires, les propriétés et les paramètres d'une base de données Adaptive Server Anywhere. Sybase Central permet également de gérer d'autres produits Sybase. Ce chapitre décrit comment utiliser Sybase Central avec la base de données exemple Adaptive Server Anywhere.

Sybase Central fournit une interface conviviale pour les deux types de tâches.

- ◆ les tâches exécutées par l'envoi d'instructions SQL au serveur ;
- ◆ les tâches exécutées par des utilitaires Adaptive Server Anywhere.

Leçon 1 : Démarrage de Sybase Central et connexion

Cette section explique comment démarrer Sybase Central et se connecter à la base de données exemple à l'aide de la source de données ASA 9.0 Sample.

❖ Pour démarrer Sybase Central

1. Choisissez Démarrer ► Programmes ► SQL Anywhere ► Sybase Central.

La fenêtre Sybase Central s'ouvre.

❖ Pour vous connecter à la base de données exemple

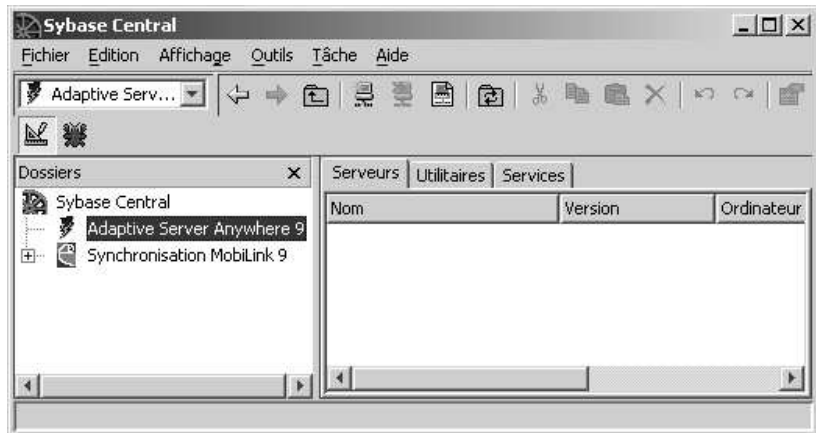
1. Choisissez Outils ► Connexion.
2. Si une boîte de dialogue vous invite à choisir un module de connexion, choisissez Adaptive Server Anywhere 9, puis cliquez sur OK.
La boîte de dialogue Connexion apparaît.
3. Sélectionnez l'option Nom de la source de données ODBC.
4. Dans la zone Nom de source de données ODBC, entrez ASA 9.0 Sample ou sélectionnez ce nom dans la liste déroulante s'il s'y trouve déjà.
Vous pouvez aussi cliquer sur le bouton Parcourir et sélectionner ASA 9.0 Sample dans la boîte de dialogue Noms des sources de données, puis cliquer sur OK.
5. Cliquez sur OK pour vous connecter à la base de données exemple.

Leçon 2 : Interface Sybase Central

Cette section explique comment naviguer dans l'interface utilisateur de Sybase Central.

Fenêtre principale de Sybase Central

La fenêtre principale de Sybase Central s'affiche.



Volets

La fenêtre principale de Sybase Central est scindée verticalement en deux volets.

Le volet gauche affiche une vue hiérarchique des objets de base de données ou **conteneurs**, sous la forme d'une structure arborescente. Un conteneur est un objet de base de données pouvant contenir d'autres objets, y compris d'autres dossiers.

Le volet droit de la fenêtre affiche le contenu du dossier sélectionné dans le volet gauche. Vous pouvez modifier l'apparence de ces deux volets à partir de la boîte de dialogue Options (accessible depuis le menu Outils).

Barre d'outils

La barre d'outils de la boîte de dialogue principale fournit des boutons pour les commandes usuelles. Pour afficher ou masquer la barre d'outils, sélectionnez Affichage ► Barre d'outils.

A partir de la barre d'outils, vous pouvez :

- ◆ naviguer dans l'arborescence des objets ;
- ◆ vous connecter à une base de données, un serveur ou un module et vous en déconnecter ;
- ◆ accéder à la fenêtre Profils de connexion (également accessible à partir du

menu Outils),

- ◆ actualiser le contenu du dossier sélectionné. Vous pouvez également appuyer sur la touche F5.
- ◆ couper, copier, coller ou supprimer des objets ;
- ◆ afficher la feuille des propriétés d'un objet sélectionné.

Lorsque vous placez le curseur au-dessus de ces boutons, vous obtenez une brève description des fonctions qu'ils exécutent.

Barre d'état

La barre d'état, en bas de la fenêtre principale, affiche une brève description des commandes de menu que vous utilisez. Pour afficher ou masquer la barre d'état, sélectionnez Affichage ► Barre d'état.

Leçon 3 : Affichage de la base de données exemple

Avec Sybase Central, vous pouvez effectuer un grand nombre de tâches d'administration de base de données, comme la création, la suppression et l'affichage de tables, de colonnes et de procédures, la gestion d'utilisateurs et de groupes d'utilisateurs, la sauvegarde de bases de données et la modification de données dans les tables. Cette section vous aide à explorer les tables et autres objets de la base de données exemple.

Affichage du schéma de base de données

Le **schéma** d'une base de données est la représentation de l'ensemble des objets qu'elle contient. Sybase Central affiche un schéma sous la forme d'une arborescence de dossiers assortis de leur contenu. Cette section décrit comment afficher le schéma d'une base de données.

Développement d'un dossier de base de données

Tout comme dans l'Explorateur Windows, Sybase Central propose plusieurs méthodes permettant de visualiser les objets d'une base de données, notamment :

- ◆ Cliquez sur un dossier dans le volet gauche pour le sélectionner. Le volet droit affiche alors son contenu.
- ◆ Cliquez une fois sur l'icône plus (+) ou moins (-) en regard d'un dossier figurant dans le volet gauche. Le dossier est alors développé ou réduit. L'absence d'icône plus ou moins en regard d'un dossier indique que ce dernier ne contient aucun objet s'étendant au-delà du niveau de ce dossier.
- ◆ Double-cliquez sur un dossier dans l'un ou l'autre des volets. Cette action développe l'arborescence dans le volet gauche et affiche son contenu dans le volet droit.

Le volet gauche n'affiche que des objets dossier. Le volet droit de la fenêtre affiche le contenu du dossier sélectionné dans le volet gauche. Par exemple, si vous sélectionnez le dossier Tables dans le volet gauche de la fenêtre, toutes les tables du dossier apparaissent dans le volet droit.

Visualisation des tables d'une base de données

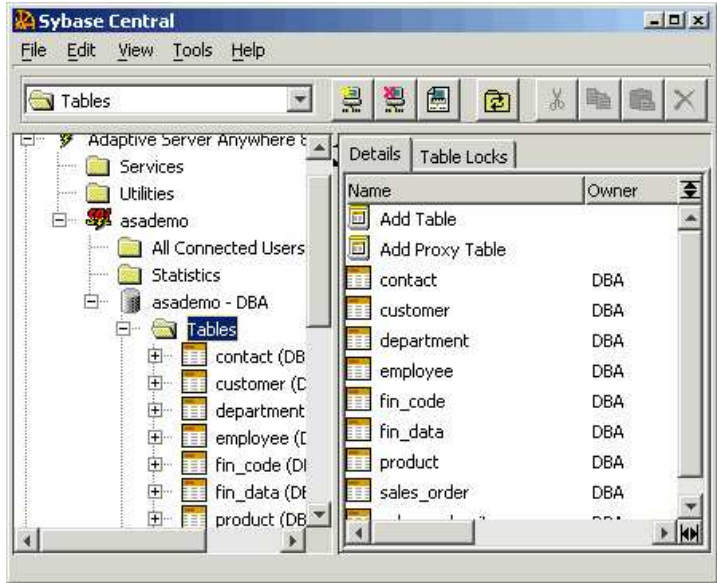
Une fois que vous êtes connecté à la base exemple, vous pouvez ouvrir les dossiers dans le volet gauche afin de visualiser les tables et autres objets de la base.

A chaque table est associé un **propriétaire**, indiqué entre parenthèses à côté du nom de la table dans le volet gauche. Si vous voyez plus de tables que n'en montre la figure, cliquez avec le bouton droit sur la base asademo et choisissez Filtrer les objets par propriétaire dans le menu contextuel. Cochez

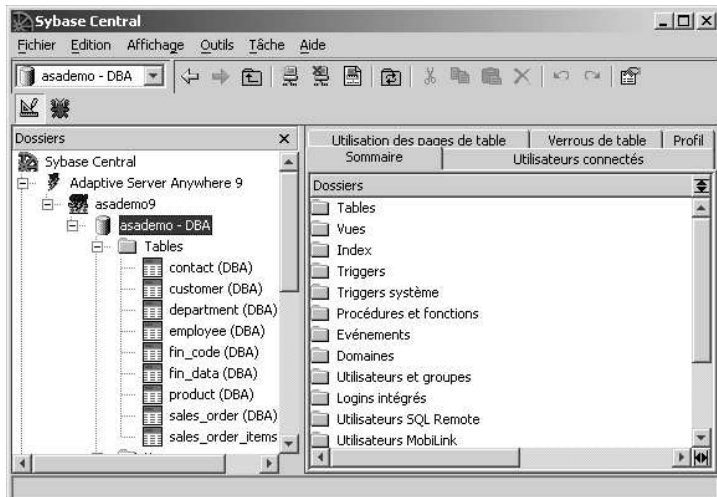
l'option DBA et supprimez la coche pour les autres ID utilisateur de la liste. Cliquez sur OK pour n'afficher que les objets appartenant à l'utilisateur DBA.

❖ **Pour examiner les tables de la base de données exemple**

1. Affichez le contenu du dossier Tables. Vous devrez peut-être développer d'abord asademo.



2. Chaque table du dossier Tables est elle-même un dossier. Sélectionnez une table dans le volet gauche.



Visualisation des propriétés d'un objet de base de données

3. Cliquez sur les différents onglets dans le volet droit pour afficher la page relative à la table sélectionnée.

Pour afficher les propriétés des objets de base de données, tels qu'une base de données ou une procédure stockée, vous avez le choix entre les méthodes suivantes.

- ◆ Cliquez sur un objet de base de données dans le volet gauche avec le bouton droit de la souris et sélectionnez Propriétés dans le menu contextuel.
- ◆ Sélectionnez un objet de base de données, puis cliquez sur Fichier ► Propriétés.

Vous pouvez naviguer dans une base de données en cliquant ou en double-cliquant sur les éléments.

Dossiers de base de données

Chaque base de données Adaptive Server Anywhere contient les dossiers suivants :

- ◆ **Tables** Tables sous-jacentes et tables temporaires globales stockées dans la base de données.
- ◆ **Vue** Tables calculées, stockées dans la base de données sous forme de requêtes, évaluées lorsqu'elles font l'objet d'un accès.
- ◆ **Index** Ordonne les lignes d'une table en fonction des valeurs de tout ou partie des colonnes.
- ◆ **Triggers** Forme spéciale de procédure stockée qui est exécutée automatiquement lorsqu'une requête modifiant les données est exécutée.
- ◆ **Triggers système** Triggers qui mettent en oeuvre des actions d'intégrité référentielle.
- ◆ **Procédures & fonctions** Pour utiliser un langage modulaire constitué de procédures SQL.
- ◆ **Événements** Pour créer et modifier des événements.
- ◆ **Domaines** Pour créer des types de données personnalisés.
- ◆ **Utilisateurs et groupes** Pour administrer les personnes habilitées à exploiter la base de données.
- ◆ **Logins intégrés** Pour autoriser les utilisateurs à se connecter à une base de données via leur nom utilisateur et leur mot de passe Windows NT.
- ◆ **Utilisateurs SQL Remote** Pour gérer la réplication SQL Remote des données dans la base.

- ◆ **Utilisateurs MobiLink** Pour gérer la synchronisation des données et des utilisateurs MobiLink.
Publications Objets de base de données décrivant les données à répliquer.
- ◆ **Projets UltraLite** Pour collecter les instructions SQL et les définitions de table utilisées dans les applications UltraLite.
- ◆ **Dbspaces** Pour créer plusieurs fichiers *.db* pour la base de données.
- ◆ **Serveurs distants** Pour identifier les serveurs distants de sorte que les utilisateurs locaux puissent exécuter des appels de procédure à distance ou récupérer des informations à partir du serveur distant.
Services Web Liste de services Web actuellement proposés dans la base de données.

Explorez la structure de la base de données exemple jusqu'à ce que vous puissiez localiser facilement les objets de base de données dans la fenêtre principale de Sybase Central.

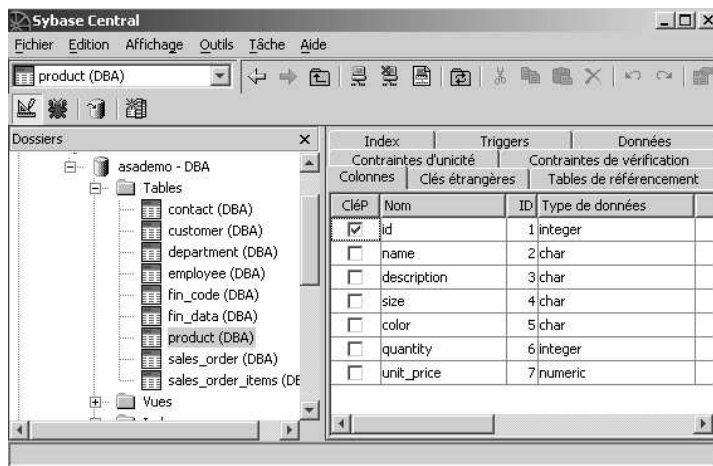
Affichage des colonnes de la table

Chaque table comporte, dans le volet droit, des onglets contenant des informations sur ses colonnes, ses clés étrangères, ses index, ses triggers et ses données. Ici, nous considérons uniquement les colonnes de la table product de la base exemple.

❖ **Pour afficher des informations sur les colonnes de la table product**

1. Ouvrez la table product. Dans le volet gauche, développez le dossier Tables puis sélectionnez la table product.

Des informations sur la table product apparaissent dans les onglets du volet droit.



2. Dans le volet droit, cliquez sur l'onglet Colonnes.

Le nom, le type de données et d'autres informations sur chaque colonne s'affichent dans le volet droit.

3. Affichez les propriétés de la colonne color.

Sélectionnez la colonne color, puis sélectionnez Propriétés dans le menu Fichier.

La feuille de propriétés de color apparaît. Elle contient une description détaillée de la colonne. Certaines propriétés, qui ne présentent d'intérêt que pour les utilisateurs expérimentés, ne sont pas décrites ici.

Visualisation des données dans la base

Outre le schéma de la base de données, vous pouvez visualiser les données des tables de la base à partir de Sybase Central.

❖ **Pour afficher le contenu des tables de base de données**

1. Affichez le contenu du dossier Tables.
2. Dans le volet gauche, cliquez sur la table contact.
3. Dans le volet droit, cliquez sur l'onglet Données.

Les données contenues dans la table apparaissent dans le volet droit.

Leçon 4 : Création et modification de tables

Ce didacticiel vous montre comment ajouter une table à la base de données exemple. Cette tâche implique d'ajouter des colonnes à une table existante.

Création d'une table dans Sybase Central

Dans cette leçon, nous créons une table "office" dans la base de données exemple.

❖ Pour créer une table nommée office

1. Dans le volet gauche de Sybase Central, cliquez sur le dossier Tables.

2. From the File menu, choose New ► Table.

Vous pouvez placer le curseur au-dessus des boutons pour obtenir une brève description des fonctions qu'ils exécutent

L'assistant de création de table apparaît.

3. Suivez les instructions de l'assistant : nommez la table **office** et spécifiez que vous ne souhaitez pas lui affecter une clé primaire. Conservez les paramètres par défaut dans les autres pages.

Modification de la table Office

Cette section décrit la procédure à suivre pour modifier une table existante. Ces étapes montrent comment ajouter une clé primaire à la table créée à la section précédente en convertissant la colonne office_id en clé primaire.

❖ Modification de la table Office

1. Cliquez sur le dossier Tables et sélectionnez la table office.

2. Nommez la première ligne office_id et affectez un type de données smallint.

3. La coche dans la colonne PKey indique que la colonne nommée office_id est la clé primaire de la table.

Conditions requises pour une clé primaire

Si la case correspondant à la colonne Valeurs NULL autorisées est cochée ou que des valeurs en double sont stockées dans la colonne office_id, celle-ci ne peut en aucun cas constituer une clé primaire.

4. Dans la barre d'outils, cliquez sur l'icône Enregistrer la table.

-
5. Dans la barre d'outils, cliquez sur l'icône Ajouter une colonne.
 6. Modifiez la nouvelle colonne de sorte que son nom soit office_name, son type char et sa police 20. Sélectionnez Nulls afin que le champ office_name puisse être vide.
 7. Cliquez sur Enregistrer la table.

La colonne figure maintenant dans la base de données, bien qu'elle ne contienne aucune donnée.

Remarque

- ◆ Dans Sybase Central, la plupart des objets, notamment les tables, les utilisateurs et les procédures stockées, possèdent des feuilles de propriétés.
- ◆ Lorsque vous cliquez sur la plupart des dossiers, les boutons **New item** apparaissent dans la barre d'outils. Un clic sur ces boutons crée des objets dans la base de données.
- ◆ Lorsque vous sélectionnez un objet dans Sybase Central, les commandes relatives à cet objet apparaissent à plusieurs endroits : dans le menu Fichier, dans le menu contextuel (auquel vous accédez en effectuant un clic droit sur l'objet) et sous forme de boutons sur la barre d'outils.
- ◆ Une fois que vous avez ouvert un dossier, son contenu est placé en mémoire cache pour que les performances soient améliorées. Vous pouvez accéder à deux commandes d'actualisation dans le menu Affichage pour mettre à jour la vue du dossier en cours ou de toute la fenêtre.

Suppression de la table office

Il est possible de supprimer des tables d'une base de données. Par exemple, vous pouvez supprimer la table office de façon à restaurer la base de données exemple dans son état initial.

❖ Pour supprimer la table office de la base de données exemple

1. Ouvrez le dossier Tables.
2. Cliquez avec le bouton droit sur la table office et sélectionnez Supprimer dans le menu contextuel.
La boîte de dialogue Confirmer la suppression apparaît.
3. Cliquez sur Oui pour supprimer la table office de la base de données.

Leçon 5 : Gestion des utilisateurs et des groupes

Dans Adaptive Server Anywhere, les utilisateurs et les groupes sont des objets contenus dans la base de données. Cependant, les groupes sont également des dossiers pouvant contenir des utilisateurs et d'autres groupes. Lorsque des utilisateurs figurent dans un groupe, ils deviennent des membres de ce groupe.

Les utilisateurs et les groupes qui sont membres d'un groupe héritent des autorisations octroyées à ce dernier. Adaptive Server Anywhere permet de créer des utilisateurs et des groupes autorisés à utiliser une base de données, et de définir l'appartenance à d'autres groupes. Utilisateurs et groupes peuvent être membres de plusieurs groupes.

Cette section montre comment créer un groupe dans la base de données, puis un utilisateur individuel et comment définir ce dernier comme membre de ce groupe.

Ajout d'un groupe à la base de données exemple

Cette section explique comment ajouter un groupe à la base de données exemple.

❖ Pour ajouter un groupe à la base de données exemple

1. Connectez-vous à la base de données exemple.
2. Sélectionnez le dossier Utilisateurs et groupes dans le volet gauche de la fenêtre.
3. Cliquez sur Nouveau groupe dans la barre d'outils.
L'assistant de création de groupe apparaît.
4. Suivez les instructions de l'assistant : nommez le nouveau groupe **Sales** et conservez les paramètres par défaut des autres options.

Le groupe Sales apparaît dans les deux volets.

Ajout d'un utilisateur à la base de données

Cette section explique comment ajouter un utilisateur à la base de données exemple.

❖ Pour ajouter un utilisateur à la base de données exemple

1. Connectez-vous à la base de données exemple.
2. Sélectionnez le dossier Utilisateurs et groupes dans le volet gauche de la fenêtre.
3. Cliquez sur Nouvel utilisateur dans la barre d'outils.
4. Suivez les instructions de l'assistant de création d'utilisateur : nommez le nouvel utilisateur **Sandy** et utilisez ce nom comme mot de passe. Conservez les valeurs par défaut des autres paramètres.

Une icône représentant le nouvel utilisateur apparaît dans les deux volets.

Ajout d'un utilisateur à un groupe

Cette section explique comment ajouter par copier-coller deux utilisateurs à un groupe.

❖ Pour ajouter des utilisateurs à un groupe

1. Ouvrez le dossier Groupes & d'utilisateurs dans le volet gauche de la fenêtre.
2. Cliquez sur le groupe Sales.
3. Cliquez sur l'onglet Membres du volet droit.
4. Choisissez Fichier ► Nouveau ► Membres.
5. Choisissez l'utilisateur Sandy, puis cliquez sur OK.

Leçon 6 : Visualisation et modification des procédures stockées

Le dossier Procédures et fonctions contient les procédures stockées pour une base de données. Ce didacticiel montre comment visualiser et modifier le contenu et les propriétés d'une procédure.

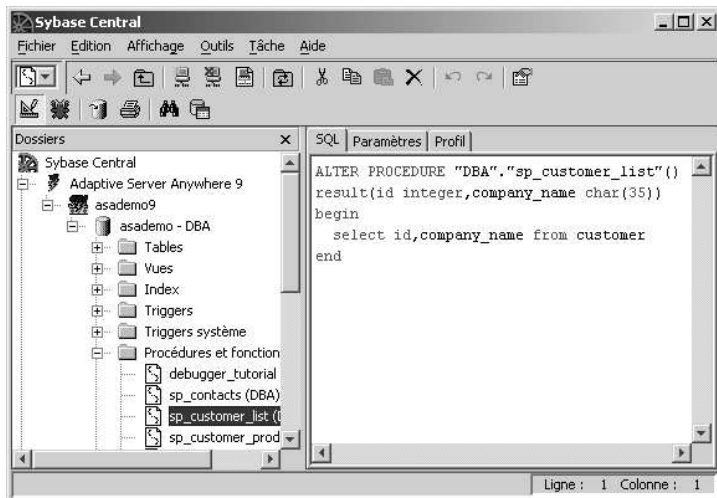
Visualisation des procédures stockées

La base de données exemple comporte plusieurs procédures stockées. Les étapes suivantes expliquent comment visualiser et modifier ces procédures.

❖ Pour afficher le contenu d'une procédure stockée

1. Dans le volet gauche de Sybase Central, ouvrez le dossier Procédures et fonctions de la base de données exemple.
2. Cliquez sur la procédure stockées sp_customer_list.

Vous pouvez visualiser le texte de la procédure sp_customer_list sur la page SQL dans le volet droit :



Cette procédure est conçue pour renvoyer un ensemble d'ID de clients et de noms de sociétés figurant dans la table customer.

Modification des propriétés de procédures stockées

Vous pouvez modifier les propriétés de procédures stockées à l'aide de leur feuille de propriétés. Dans cette section, nous allons modifier les

autorisations d'une procédure stockée afin que tous les commerciaux, notamment l'utilisateur créé dans la leçon précédente (Sandrine), puissent exécuter la procédure et obtenir ainsi une liste de clients.

Des autorisations sont associées aux procédures stockées. Pour exécuter une procédure, vous devez détenir l'autorisation correspondante ou être membre d'un groupe d'utilisateurs détenant ce droit.

❖ **Pour visualiser et modifier les permissions de la procédure `sp_customer_list`**

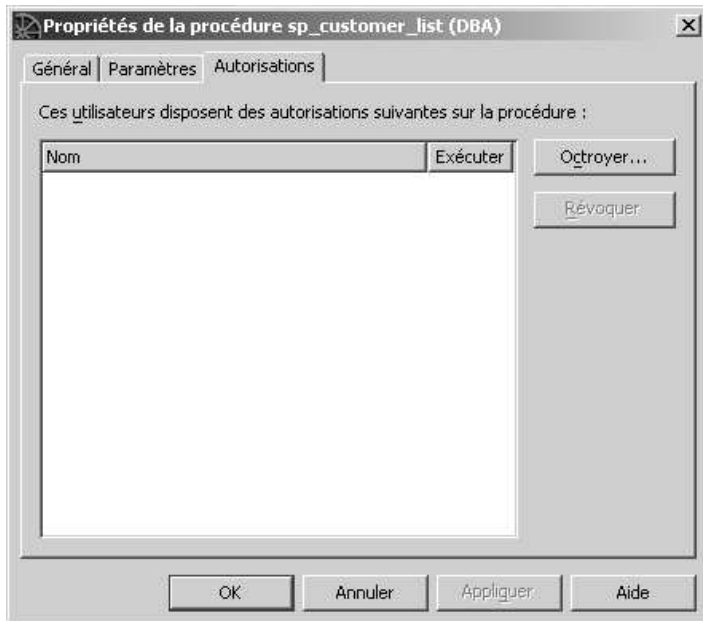
1. Dans le dossier Procédures & fonctions, sélectionnez `sp_customer_list`.
2. Cliquez sur Fichier ► Propriétés.
3. Cliquez sur l'onglet Autorisations pour afficher les ID utilisateur détenteurs d'autorisations sur cette procédure. Il n'en existe aucun pour l'instant car le seul utilisateur de la base de données exemple est DBA, lequel dispose automatiquement des autorisations d'exécution (en tant que propriétaire de la procédure).
4. Cliquez sur le bouton Octroyer.

La boîte de dialogue Octroyer des autorisations s'affiche.



5. Dans la boîte de dialogue Octroyer des autorisations, puis sélectionnez Sales et cliquez sur OK pour accorder à ce groupe l'autorisation d'exécuter la procédure `sp_customer_list`.

Le groupe Sales est ajouté à l'onglet Autorisations de la feuille de propriétés de la procédure avec une coche dans la colonne Exécuter :



Vous pouvez annuler l'autorisation accordée à ce groupe en cliquant sur la coche pour la faire disparaître ou en sélectionnant le groupe et en cliquant sur le bouton Révoquer situé sur le côté de l'onglet Autorisations.

6. Cliquez sur OK pour accepter les modifications apportées aux autorisations sp_customer_list.

Leçon 7 : Sauvegarde d'une base de données

Sybase Central comprend un ensemble d'utilitaires couvrant les tâches courantes d'administration de bases de données. Les assistants vous guident tout au long des diverses procédures.

Utilitaires

Pour afficher la liste des utilitaires pouvant être utilisés pour une base de données active, cliquez avec le bouton droit sur le contenu de la base de données exemple. Plusieurs options apparaissent dans le menu contextuel notamment Sauvegarder la base de données, Mettre la base de données à niveau et Valider la base de données. Chaque option de menu représente un utilitaire.

Dans cette section, nous allons sauvegarder la base de données exemple à l'aide d'un assistant. Cette opération peut être exécutée pour une base de données active.

❖ Pour sauvegarder une base de données active

1. Dans le volet gauche, cliquez avec le bouton droit sur l'icône asademo - DBA et sélectionnez Sauvegarder la base de données dans le menu contextuel. L'assistant de sauvegarde de la base de données apparaît.
2. Lisez l'écran introductif de l'assistant, puis cliquez sur Suivant.
3. Sélectionnez asademo dans la liste des bases de données que vous pouvez sauvegarder. Cliquez sur Suivant.
4. Sélectionnez Sur le disque, Dans le fichier suivant.

Dans la zone de texte, entrez un nom pour le fichier dans lequel vous sauvegarderez la base de données. Comme il ne s'agit ici que d'un exercice, choisissez un répertoire de fichiers temporaires tel que `c:\temp\backup`.

5. Cliquez sur Terminer pour sauvegarder la base de données.

Remarque

Ce type de sauvegarde s'appelle une **sauvegarde d'archivage**. Vous pouvez également effectuer des sauvegardes d'image qui sont des copies physiques du fichier de la base de données et des fichiers associés.

L'extension `.1` est ajoutée au nom de fichier spécifié dans l'assistant de sauvegarde de base de données.

Plusieurs autres tâches d'administration de base de données peuvent être effectuées à l'aide d'assistants. Essayez de créer une base de données en cliquant avec le bouton droit sur le serveur dans le volet gauche, puis en sélectionnant Créer une base de données dans le menu contextuel.

Restauration de la base de données exemple

Maintenant que le didacticiel est terminé, vous devez restaurer la base de données exemple, pour pouvoir la réutiliser dans son état initial. Remplacez la version de *asademo.db* que vous venez de modifier par la copie que vous avez réalisée avant de commencer le didacticiel. Supprimez *asademo.log*.

Résumé

Dans ce chapitre, vous avez appris à démarrer Sybase Central et à vous connecter à la base de données exemple. Vous avez également appris à afficher et modifier cette base de données, modifier les tables, gérer les utilisateurs et les groupes, utiliser les procédures stockées et sauvegarder votre base de données.

CHAPITRE 15

Conception et création d'une base de données

Présentation

Ce chapitre présente quelques principes de la conception d'une base de données et décrit comment créer une base de données à l'aide de Sybase Central. Il utilise la base de données exemple Adaptive Server Anywhere pour illustrer les principes impliqués et applique les compétences acquises dans les chapitres précédents : de l'utilisation de la base de données exemple à la création d'une base de données. Vous allez concevoir et créer une base très simple, sur le modèle des tables product, sales_order_items, sales_order et customer de la base exemple.

Sommaire

Sujet :	page
Conception d'une base de données	214
Conception et création d'une base de données simple	215
Leçon 1 : création d'un fichier de base de données	216
Leçon 2 : connexion à la base de données	218
Leçon 3 : conception et création d'une table	220
Leçon 4 :	222
Leçon 5 : définition des propriétés de colonne	224
Leçon 6 : conception et création des relations entre les tables	227
Résumé	230

Conception d'une base de données

Il n'est pas inutile de consacrer du temps à la phase de conception d'une base de données, même simple : il convient en effet de déterminer avec soin les tables à utiliser, les clés qui les relient etc.

Bien que la conception d'une base de données de taille petite ou moyenne ne présente pas de difficulté particulière, elle n'en demeure pas moins une tâche importante. Une conception incorrecte peut conduire à un système de base de données inefficace, voire peu fiable. Les applications clientes étant construites pour exploiter des éléments spécifiques d'une base et dépendant de la structure de celle-ci, il peut s'avérer difficile, par la suite, de remédier à une conception inadéquate.

La conception d'une grande base de données est une opération complexe. Des méthodes formelles, comme la modélisation conceptuelle des données, peuvent être utilisées. Il existe aussi des outils sophistiqués tels que Sybase PowerAMC et DataArchitect qui vous aident à appliquer ces techniques pour concevoir et maintenir des bases de données volumineuses.

Ce chapitre n'a pas pour objectif d'aborder les aspects de la conception des bases de données volumineuses. Le présent chapitre doit surtout vous permettre de choisir le type d'informations à rassembler dans une table et de déterminer les relations qui uniront les tables.

☞ Pour connaître quelques notions élémentaires, reportez-vous au chapitre [“Conception d'une base de données”](#) [ASA - Guide de l'utilisateur SQL, page 3]. Pour des opérations plus élaborées, reportez-vous à la documentation de Sybase PowerAMC ou à un ouvrage spécialisé sur ce sujet.

Conception et création d'une base de données simple

Lorsque vous concevez une base de données, vous devez prévoir les éléments sur lesquels vous allez stocker des informations, ainsi que les données que vous allez stocker. Vous déterminez aussi les relations qui unissent ces éléments. Ceux-ci sont les entités de la base de données et les liens qui les unissent sont appelés **relations**.

Leçon 1 : création d'un fichier de base de données

Dans cette leçon, vous allez créer un fichier de base de données.

☞ Pour plus d'informations, reportez-vous à la section "[Eléments d'un système de base de données](#)" à la page 67.

Concepts

Un fichier de base de données est un conteneur destiné à recevoir votre base. Il contient les tables système et d'autres objets système communs à toutes les bases, mais vous devez y ajouter toutes les tables et les données spécifiques de la base.

L'ensemble des tables, index et autres objets de la base, ainsi que les relations qui les unissent, constituent le **schéma**. Ce schéma correspond à la base, sans les données. Le présent didacticiel explique comment concevoir et créer un schéma très simple.

☞ Pour une présentation des éléments conceptuels, reportez-vous à la section "[Principes fondamentaux des bases de données relationnelles](#)" à la page 62.

Le nom de chaque objet de la base, notamment des tables, des colonnes et des index, est un **identificateur**. Il existe des règles pour définir les identificateurs. Ainsi, vous pouvez utiliser des lettres, des nombres ou des symboles. S'il contient des caractères autres que des lettres, des chiffres ou des traits de soulignement, s'il ne commence pas par une lettre ou s'il est identique à un mot-clé, vous devez l'inclure entre guillemets.

Si l'option de base de données QUOTED_IDENTIFIER est désactivée (OFF), les guillemets doubles servent à délimiter les chaînes SQL et ne peuvent donc pas être utilisés pour les identificateurs. Toutefois, vous pouvez toujours utiliser des crochets pour délimiter les identificateurs, quelle que soit la définition de QUOTED_IDENTIFIER.

☞ Pour plus d'informations, reportez-vous à la section "[Identificateurs](#)" [[ASA - Manuel de référence SQL](#), page 8].

Exercice

❖ Pour créer un fichier de base de données

1. Démarrez Sybase Central.
2. Dans le volet gauche, sélectionnez le module de connexion Adaptive Server Anywhere, puis cliquez sur l'onglet Utilitaires dans le volet droit. Dans le volet droit, double-cliquez sur Créer une base de données. L'assistant de création d'une base de données apparaît.

3. Lisez la page d'introduction et cliquez sur Suivant.
4. Sélectionnez Créer une base de données sur cet ordinateur et cliquez sur Suivant.
5. Choisissez un emplacement et un nom pour le fichier de base de données :
 - ◆ Entrez le nom de fichier *c : \temp\monexemple*. Si votre répertoire temporaire n'est pas *c : \temp*, spécifiez le chemin de votre choix.
6. Cliquez sur Terminer pour créer la base.

D'autres options, accessibles lorsque vous cliquez sur Suivant, sont disponibles lors de la création d'une base de données, mais les options par défaut conviennent en général à la plupart des situations.

La fenêtre Création de la base de données affiche la progression de la tâche. Lorsque le fichier est créé, cliquez sur OK pour fermer la fenêtre.

Leçon 2 : connexion à la base de données

Dans cette leçon, vous allez vous connecter à la base de données que vous venez de créer.

☞ Pour plus d'informations, reportez-vous à la section "[Interaction entre les différents éléments](#)" à la page 70.

Exercice

Une fois la base créée, vous pouvez vous y connecter afin de créer les tables et autres objets.

❖ Pour vous connecter à la base de données

1. Démarrez Sybase Central.
2. Choisissez Outils ► Connecter pour ouvrir la boîte de dialogue Nouvelle connexion. Si plusieurs modules de connexion sont chargés, vous devez choisir le module Adaptive Server Anywhere avant d'accéder à cette boîte de dialogue.

3. Spécifiez l'ID utilisateur et le mot de passe.

Sur le premier onglet de la boîte de dialogue Connexion (onglet Identification) entrez l'ID utilisateur **DBA** et le mot de passe **SQL**. Il s'agit des valeurs créées pour les nouvelles bases de données et qui permettent d'y accéder.

Choisissez **Aucun** dans les options de profil au bas de l'onglet.

4. Entrez le nom de votre fichier de base de données.

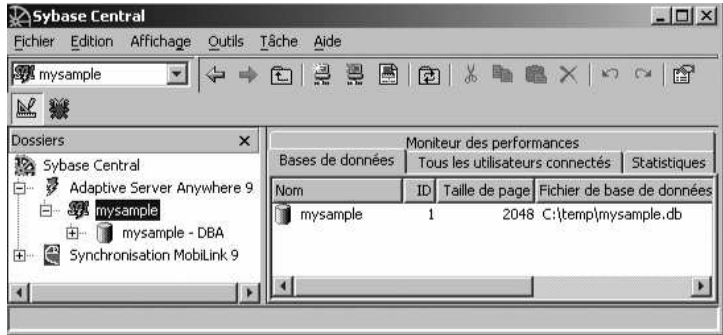
Cliquez sur l'onglet Base de données. Entrez le chemin complet d'accès au fichier de base de données dans le champ prévu à cet effet. Ainsi, si vous reprenez l'exemple de la première leçon, entrez :

```
c:\temp\mysample.db
```

5. Connectez-vous à la base de données.

Cliquez sur OK. Sybase Central se connecte à la base de données.

Ouvrez le conteneur du serveur de base de données dans le volet gauche pour consulter la base de données monexemple.



Moniteur des performances			
Bases de données			
Tous les utilisateurs connectés			
Statistiques			
Nom	ID	Taille de page	Fichier de base de données
mysample	1	2048	C:\temp\mysample.db

Leçon 3 : conception et création d'une table

Concepts

Dans cette leçon, vous allez ajouter une table à votre base de données.

Chaque table doit contenir des informations sur un même sujet. Dans le langage propre à la conception des bases de données, vous désignez une **entité**. Ainsi, la base exemple contient des informations sur les employés dans une table et des informations sur les produits dans une autre : les employés et les produits sont des entités.

☞ Pour une introduction aux tables, reportez-vous à la section [“Tables de base de données” à la page 62.](#)

Chaque colonne d'une table décrit une caractéristique de l'élément sur lequel vous souhaitez enregistrer des informations. Toujours dans la base exemple, la table `employee` possède des colonnes qui contiennent l'ID d'un employé, son nom et son prénom, une adresse et d'autres données qui lui sont propres.

Dans les représentations graphiques des bases de données, les tables sont signalées par un rectangle. Le nom de la table figure au-dessus du rectangle et les noms de colonne à l'intérieur.

Product	
<u>ID</u>	Integer
name	char(15)
description	char(30)
size	char(18)
color	char(6)
quantity	integer
unit_price	numeric(15,12)

Dans la table `product` de la base exemple ci-dessus, chaque produit est un article de sport.

Exercice

Création d'une version simplifiée de la table `product` contenant uniquement les colonnes identificateur (`id`) et nom (`name`).

❖ Pour créer la table `product`

1. Dans Sybase Central, connectez-vous à votre base de données, si ce n'est pas déjà fait.
2. Ouvrez le dossier `Tables` de votre base.
Ouvrez d'abord les conteneurs `serveur` et `base de données`, puis le dossier `Tables`.
3. Dans le menu `Fichier`, sélectionnez **► Nouvelle ► table**.
L'assistant de création de table apparaît.

4. Nommez votre table **product**.
5. Cliquez sur Terminer.
6. Sélectionnez la table product dans le volet gauche, puis cliquez sur l'onglet Colonnes dans le volet droit.
7. Créez les colonnes.

Dans le menu Fichier, sélectionnez Ajouter une colonne et ajoutez une colonne ayant les propriétés suivantes :

- ◆ **Nom de la colonne** Nommez la colonne **id**.
- ◆ **PKey** Cochez la case en regard du nom de la colonne : la coche indique que la colonne est une **primary key**.
- ◆ **Type de données** Attribuez à la colonne le type de données **integer** (entier).

Vous pouvez ignorer les autres propriétés. Ajoutez une deuxième colonne ayant les propriétés suivantes :

- ◆ **Nom de la colonne** Nommez la colonne **name**. Elle devra contenir le nom du produit.
- ◆ **Type de données** Choisissez le type de données **char**, pour indiquer que la colonne contiendra des chaînes de caractères et fixez une longueur maximale de **15** dans la colonne Taille.

8. Pour terminer, cliquez sur le bouton Enregistrer et fermer à l'extrémité droite de la barre d'outils.

Vous avez ainsi créé une table. Les données de la table sont contenues dans le fichier de base de données. Pour le moment, cette table est vide.

Les deux leçons suivantes vont vous en apprendre davantage sur les colonnes et les types de données.

Leçon 4 :

Dans cette leçon, vous allez apprendre à définir les clés primaires de vos tables. Aucun exercice n'est proposé.

Pour plus d'informations, reportez-vous à la section "[Les tables sont dotées d'une clé primaire](#)" à la page 64.

Concepts

La **clé primaire** est une ou plusieurs colonnes spéciales utilisées pour identifier une ligne de la table de manière univoque. Dans la table product, la colonne id identifie chaque produit de façon unique.

Product	
<u>ID</u>	<u>Integer</u>
name	char(15)
description	char(30)
size	char(18)
color	char(6)
quantity	integer
unit_price	numeric(15,12)

Chaque ligne possède une valeur unique dans la colonne id et les valeurs contenues dans chaque ligne appartiennent exclusivement à un identificateur de produit, soit sa valeur id. Deux produits peuvent avoir le même nom ou la même taille, mais pas le même identificateur ou id. Dans le diagramme, la colonne id est soulignée pour montrer qu'il s'agit d'une clé primaire.

La création d'une colonne destinée à contenir un identificateur qui n'a pas d'autre signification est une pratique courante. Vous pouvez, de la même façon, constater sur vos relevés bancaires ou d'électricité que chaque compte possède un identificateur unique.

Utilisation d'une clé primaire AUTOINCREMENT

Vous pouvez simplifier la saisie des clés primaires en attribuant à une colonne de type clé primaire une valeur par défaut d'AUTOINCREMENT. Ainsi, la valeur de cette colonne est automatiquement entrée chaque fois qu'une ligne est ajoutée et elle augmente de un par rapport à la valeur précédente.

❖ Pour créer une clé primaire de type AUTOINCREMENT

1. Sélectionnez la table product dans le volet gauche, puis cliquez sur l'onglet Colonnes dans le volet droit.
2. Sélectionnez la colonne clé primaire. Dans le menu Fichier, sélectionnez le bouton Propriétés afin d'ouvrir la feuille de propriétés de la colonne.
3. Cliquez sur l'onglet Valeur.

4. Sélectionnez l'option Valeur par défaut.
5. Cliquez sur Système et choisissez Autoincrémentation dans la liste déroulante.
6. Cliquez sur OK pour fermer la feuille de propriétés.
7. Enfin, sélectionnez Fichier ► Enregistrer la table.

Leçon 5 : définition des propriétés de colonne

Dans cette leçon, vous allez apprendre à choisir des types de données et d'autres attributs pour les colonnes de vos tables.

Concepts

Chaque colonne est associée à un type de données. Le **type de données** définit le type d'informations que la colonne contient. Choisissez un type adapté aux données de la colonne. Par exemple, les colonnes d'identificateur seront du type entier, celles contenant des noms ou des adresses auront le type caractère.

Les types de données sont regroupés en plusieurs catégories :

- ◆ **Types de données numériques** Il existe plusieurs types de données numériques. Certains sont exacts (ils ne sont pas arrondis lors des calculs), d'autres sont approximatifs.
Le type de données d'une colonne a un impact sur la taille maximale de la colonne. Par exemple, si vous spécifiez le type SMALLINT, la colonne peut contenir la valeur maximale 32 767. Si vous spécifiez INTEGER, la valeur maximale est 2 147 483 647.
☞ Pour obtenir la liste complète, reportez-vous à la section "[Types de données numériques](#)" [ASA - Manuel de référence SQL, page 61].
- ◆ **Types de données caractères** Il s'agit de chaînes de texte, par exemple des noms, des adresses. Pour ces types de données, une longueur maximale fixe le nombre de caractères que peut contenir une chaîne.
☞ Pour obtenir la liste complète, reportez-vous à la section "[Types de données caractères](#)" [ASA - Manuel de référence SQL, page 56].
- ◆ **Données binaires** Ces types contiennent des informations qui ont un sens pour une application mais qui sont codées en format binaire.
☞ Pour obtenir la liste complète, reportez-vous à la section "[Types de données binaires](#)" [ASA - Manuel de référence SQL, page 78].
- ◆ **Date/heure** Ces types correspondent aux dates et aux heures.
☞ Pour obtenir la liste complète, reportez-vous à la section "[Types de données date et heure](#)" [ASA - Manuel de référence SQL, page 71].
- ◆ **Type de données longs** On parle parfois d'objets **blob** (binary large object). Ils servent à stocker des chaînes de texte (parfois appelées champs mémo), des images ou d'autres informations binaires.
☞ Pour plus d'informations, reportez-vous aux scripts "[Type de données LONG BINARY \[Binaire\]](#)" [ASA - Manuel de référence SQL, page 78] et "[Type de données LONG VARCHAR \[Caractère\]](#)" [ASA - Manuel de référence SQL, page 58].

De plus, Adaptive Server Anywhere supporte des types de données définis par l'utilisateur et des types de données propres à Java. Le présent manuel ne traite pas de ces types de données.

NULL et NOT NULL

Si chaque ligne doit contenir une valeur pour cette colonne, vous devez définir cette dernière comme étant NOT NULL. Sinon, la colonne accepte les valeurs NULL, ce qui correspond à l'absence de valeur. Par défaut, les colonnes acceptent les valeurs NULL, mais vous devriez explicitement les définir comme NOT NULL.

☞ Pour plus d'informations sur la valeur NULL, reportez-vous à la section “Valeur NULL” [ASA - Manuel de référence SQL, page 52]. Pour plus d'informations sur son utilisation dans les comparaisons, reportez-vous à la section “Conditions de recherche” [ASA - Manuel de référence SQL, page 24].

❖ Pour spécifier le type de données d'une colonne

1. Sélectionnez la table product dans le volet gauche, puis cliquez sur l'onglet Colonnes dans le volet droit.
2. Sélectionnez la colonne clé primaire puis sélectionnez Fichier ► Propriétés pour ouvrir la feuille de propriétés de colonne.
La feuille de propriétés s'ouvre.
3. Cliquez sur l'onglet Types de données.
4. Sélectionnez un type de données dans la liste déroulante Type prédéfini.

Exercice

Les précédentes leçons vous ont présenté les concepts élémentaires nécessaires pour créer des tables. Vous pouvez les appliquer pour ajouter des tables supplémentaires à votre base de données. Ces tables seront utilisées dans les prochaines leçons du chapitre.

Ajoutez les tables suivantes à votre base de données :

- ◆ **customer** Ajoutez une table customer, avec les colonnes suivantes :
 - ◆ **id** Identificateur de chaque client. Cette colonne est de type **integer** et sert de **clé primaire**. Attribuez-lui la valeur d'auto-incrémentation.
 - ◆ **company_name** Nom de la société. Cette colonne est de type **character** et sa longueur maximale est de **35** caractères.
- ◆ **sales_order** Ajoutez une table nommée sales_order, avec les colonnes suivantes :
 - ◆ **id** Identificateur de commande. Cette colonne est de type **integer** et sert de **clé primaire**. Attribuez-lui la valeur d'auto-incrémentation.
 - ◆ **order_date** Date à laquelle la commande a été passée. Cette colonne est de type **date**.

-
- ◆ **cust_id** Identificateur du client qui a passé la commande. Cette colonne est de type **integer**.
 - ◆ **sales_order_items** Ajoutez une table nommée `sales_order_items` destinée à contenir les informations sur les articles, avec les colonnes suivantes :
 - ◆ **id** Identificateur de la commande à laquelle appartient l'article. Cette colonne est de type **integer** et doit être une **clé primaire**.
 - ◆ **line_id** Identificateur de commande. Cette colonne est de type **integer** et doit être une **clé primaire**.
 - ◆ **prod_id** Identificateur de l'article commandé. Cette colonne est de type **integer**.

Vous avez ainsi créé quatre tables. Pour l'instant, elles ne sont pas reliées entre elles. Dans la leçon suivante, vous allez définir des clés étrangères qui établiront ces relations.

Leçon 6 : conception et création des relations entre les tables

Dans cette leçon, vous allez concevoir et créer des relations entre les tables, au moyen de clés étrangères.

☞ Pour plus d'informations, reportez-vous à la section "[Tables liées par des clés étrangères](#)" à la page 64.

Concepts

Bien que chaque table contienne des informations sur un sujet unique, deux ou plusieurs tables peuvent contenir des informations ayant un lien entre elles. Par exemple, un employé appartient à un service ou une commande porte sur plusieurs produits. Les relations dans une base de données peuvent apparaître comme des relations par **clé étrangère** entre des tables ou comme des tables distinctes. Des exemples sont donnés dans la suite de ce chapitre.

Vous créez des relations dans la base de données afin d'établir des règles servant à régir les données de vos tables. Lorsqu'une relation est définie dans la structure d'une base de données, aucune exception n'est admise.

Les relations entre les tables sont classées comme suit :

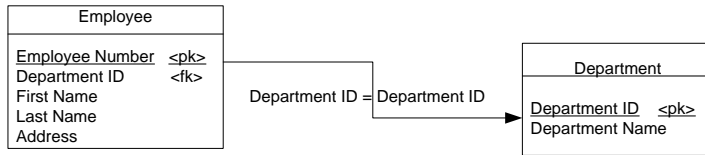
- ◆ **Relations 1-1** Chaque élément d'une entité correspond à aucun ou à un élément dans une autre entité. Ainsi, dans la base exemple, *un* employé gère *un* service. Le service ne peut avoir qu'un responsable. La duplication de l'entrée "département" supposerait la duplication de l'ID de département, ce qui n'est pas possible car il s'agit de la clé primaire.

Ce type de relation convient souvent pour combiner des éléments dans une même table. Il existe ainsi dans la table `department` une colonne désignant le responsable au lieu d'une table distincte nommée `manager`.

☞ Pour plus d'informations sur les cas où il convient de séparer les éléments, reportez-vous à la section "[Conception d'une base de données](#)" [*ASA - Guide de l'utilisateur SQL*, page 3].

- ◆ **Relations 1-n** Ce type de relation devient une relation de clé étrangère entre les tables. La clé primaire d'une entité apparaît comme clé étrangère dans la table *n*.

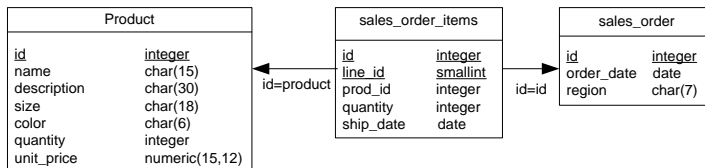
Par exemple, dans la base que vous venez de créer, *un* client peut passer *n* commandes mais un seul client peut passer chaque commande prise séparément. Pour représenter cette relation, vous avez besoin d'une **colonne clé étrangère** dans la table `sales_order` (`cust_id`) qui corresponde à la colonne clé primaire de la table `customer` (`id`). Il est souvent plus pratique de donner le même nom à ces colonnes.



Chaque entrée de la colonne cust_id de la table sales_order doit correspondre à l'une des entrées de la colonne id de la table customer. La table sales_order (qui contient la clé étrangère dans cette relation) est appelée table étrangère ou table qui référence. La table customer (qui contient la clé primaire référencée) est appelée table primaire ou table référencée.

- ◆ **Relations n-n** Une relation n-n est représentée par une table intermédiaire et il existe une relation de clé étrangère de la table intermédiaire vers chacune des entités reliées.

Ainsi, dans la base exemple, il existe une relation n-n entre les produits et les commandes. Une commande peut être passée pour plusieurs produits et un produit peut apparaître dans plusieurs commandes.



Parfois, la table intermédiaire (sales_order_items) contient des informations supplémentaires, comme le nombre d'articles commandés et la date de leur livraison. Dans notre cas, la table intermédiaire ne contient pas d'autres informations.

Exercice

Ajout de clés étrangères pour relier les tables de votre base de données.

Ajoutez les clés étrangères suivantes :

- ◆ Une clé étrangère de la colonne id dans la table sales_order_items, qui référence la colonne id dans la table sales_order. Cette clé établit une relation n-1 entre les tables sales_order et sales_order_items.
- ◆ Une clé étrangère de la colonne prod_id dans la table sales_order_items qui référence la colonne id dans la table product. Cette clé établit une relation n-1 entre les tables sales_order_items et product.
- ◆ Une clé étrangère de la colonne cust_id dans la table sales_order qui référence la colonne id dans la table customer. Cette clé établit une relation n-1 entre les tables sales_order et customer.

Les deux premières clés étrangères prises ensemble établissent une relation n-n entre les tables sales_order et product de la base de données.

❖ **Pour créer une clé étrangère**

1. Sélectionnez la table pour laquelle vous voulez créer une clé étrangère.
2. Cliquez sur l'onglet Clés étrangères du volet droit.
3. Dans le menu Fichier, sélectionnez Nouvelle ► clé étrangère pour ouvrir l'assistant de création de clé étrangère.
4. Suivez les instructions de l'assistant.

Cet exercice termine le chapitre introductif sur la conception et la création de bases de données relationnelles. Les autres chapitres du manuel expliquent comment ajouter et extraire des données dans les bases de données. Ils utilisent la base de données exemple Adaptive Server Anywhere, qui offre un modèle de base plus important que celle que vous venez de créer.

Résumé

Dans ce chapitre, vous vous êtes familiarisé avec les principes de la conception de bases de données. Vous avez ensuite utilisé ces principes pour créer une base de données à l'aide de Sybase Central.

CHAPITRE 16

Synchronisation de bases de données avec MobiLink

Présentation

Ce chapitre fournit un didacticiel qui vous guide dans la synchronisation de deux bases de données Adaptive Server Anywhere à l'aide de MobiLink. L'une de ces bases est la base de données consolidée et l'autre, une base de données distante. Le didacticiel vous montre comment créer ces bases de données et les synchroniser.

Sommaire

Sujet :	page
Présentation de MobiLink	232
Introduction	233
Leçon 1 : Création de bases de données	235
Leçon 2 : Préparation des bases de données pour la synchronisation	239
Leçon 3 : Démarrage du serveur de synchronisation MobiLink	242
Leçon 4 : Exécution de l'utilitaire client de synchronisation MobiLink	243
Résumé	245

Présentation de MobiLink

La réplication consiste à partager des données entre des bases physiquement distinctes. Sybase fournit trois technologies de réplication de données :

- ◆ MobiLink
- ◆ SQL Remote
- ◆ Replication Server

La synchronisation MobiLink permet la réplication entre une base de données consolidée compatible ODBC et une base de données distante Adaptive Server Anywhere ou UltraLite. Dans ce didacticiel, c'est une base distante Adaptive Server Anywhere qui est utilisée. La base de données consolidée peut être une base Sybase Adaptive Server Anywhere, Sybase Adaptive Server Enterprise, Oracle, Microsoft SQL Server ou IBM DB2.

MobiLink est conçu pour synchroniser un serveur de données consolidées et un grand nombre de bases de données distantes, avec notamment de nombreuses bases de données mobiles. Les impératifs en matière d'administration et de ressources sur les sites distants sont réduits au minimum. Le système est basé sur la connexion et un site distant peut se connecter aussi souvent qu'il le souhaite. Au terme de chaque connexion, les bases de données sont complètement synchronisées.

MobiLink travaille en regroupant les résultats de plusieurs transactions sur la base de données distante au sein d'un ensemble de modifications appliquées à la base de données consolidée. La synchronisation se produisant toujours au niveau d'un délimiteur de transaction, l'intégrité référentielle est préservée. L'ordre des modifications individuelles apportées lors des transactions au niveau du composant n'est pas maintenu : toutefois, comme les données non validées ne sont jamais synchronisées, l'intégrité des données est préservée.

☞ Pour plus d'informations sur les stratégies de synchronisation, notamment pour consulter l'ensemble de la documentation relative à MobiLink, reportez-vous au document "[Préface](#)" [*Synchronisation MobiLink - Guide de l'utilisateur, page ??*].

☞ Pour consulter une présentation de la synchronisation SQL Remote, reportez-vous au chapitre "[Réplication de données avec SQL Remote](#)" à la page 247.

Introduction

Dans ce didacticiel, vous allez créer une base de données consolidée et une base de données distante. Vous allez ensuite écrire des publications et des souscriptions de synchronisation. Ensuite, vous les synchroniserez à l'aide de la technologie de synchronisation MobiLink.

Configuration requise Les conditions requises pour ce didacticiel sont les suivantes.

Condition requise	Description
Durée	Le didacticiel dure environ 50 minutes.
Logiciel	<ul style="list-style-type: none"> ◆ Installation complète d'Adaptive Server Anywhere 9. ◆ Installation complète du serveur de synchronisation MobiLink.
Compétences et expérience	<p>Connaissances et/ou expérience du traitement par ligne de commande.</p> <p>Compétences en matière de connexion à une base de données au moyen d'ODBC et de Sybase Central.</p> <p>Pour plus d'informations sur ODBC, reportez-vous à la section "Leçon 1 : Création d'une source de données ODBC" à la page 160.</p>

Objectifs Ce didacticiel a pour objectif de vous faire acquérir des compétences et de vous familiariser avec les tâches suivantes :

- ◆ utilisation du serveur et du client de synchronisation MobiLink comme un système intégré ;
- ◆ exécution des commandes du serveur et du client de synchronisation MobiLink ;
- ◆ utilisation des lignes de commande et des options du serveur, ainsi que du client de synchronisation MobiLink.

Principaux concepts Le serveur de synchronisation MobiLink se connecte à la base de données consolidée via l'interface ODBC. Le client de synchronisation MobiLink se connecte à votre base de données distante. Le serveur et le client de synchronisation MobiLink gèrent conjointement le téléchargement et le transfert des données d'une base vers une autre.

- ◆ Les principaux concepts abordés dans ce didacticiel sont les suivants :

-
- ◆ Serveur et client de synchronisation MobiLink
 - ◆ Connexion et source de données ODBC, souscription et publication de synchronisation, serveur consolidé et bases de données distantes.

Leçon 1 : Création de bases de données

La synchronisation MobiLink exige que vous disposiez de bases de données consolidée et distante compatibles, de tables de base de données remplies et de sources de données ODBC pour chaque base.

Conseil

Dans ce didacticiel, toutes les opérations effectuées au niveau de la commande sont supposées avoir lieu dans un répertoire que vous nommerez `C:\MLTutorial`.

Création des fichiers de base de données

La première étape consiste à créer chacune des bases de données. Lors de cette procédure, vous créez une base consolidée et une base distante à partir de l'invite de commandes, au moyen de l'exécutable `dbinit`.

Conseil

La création d'un fichier de base de données à l'aide de `dbinit` s'apparente au formatage d'un disque dur : vous disposez d'un fichier de base de données sans table ni procédure utilisateur. Vous créez le schéma de la base de données lorsque vous définissez différentes tables et procédures utilisateur dans le nouveau fichier initialisé.

Pour plus d'informations sur l'utilitaire `dbinit`, reportez-vous à la section "[Création d'une base de données à l'aide de l'utilitaire `dbinit`](#)" [*ASA - Guide d'administration*, page 538].

❖ Pour créer vos fichiers de base de données

1. A l'invite de commandes, allez dans le répertoire que vous avez nommé `\MLTutorial`.

2. Créez un fichier pour la base de données consolidée. A l'invite de commandes, saisissez :

```
dbinit consol.db
```

3. Créez un fichier pour la base de données distante. A l'invite de commandes, saisissez :

```
dbinit remote.db
```

4. Vérifiez si ces fichiers de base de données ont bien été créés en saisissant la chaîne suivante à l'invite de commandes :

```
dir
```

-
5. Les deux fichiers de base de données doivent figurer dans la liste des répertoires. Dans le cas contraire, vérifiez votre procédure et reprenez au besoin l'étape 1 ou 2.

Création de sources de données ODBC

☞ Vous êtes maintenant prêt à créer les sources de données ODBC qui vous permettront de vous connecter aux bases de données Adaptive Server Anywhere.

☞ Pour plus d'informations sur la création de sources de données ODBC, reportez-vous à la section [“Utilitaire de source de données” \[ASA - Guide d'administration, page 520\]](#).

❖ Pour créer des sources de données ODBC

1. Créez une source de données ODBC pour la base consolidée en saisissant la chaîne suivante à l'invite de commandes :

```
dbdsn -w test_consol -y -c "uid=DBA;pwd=SQL;dbf=C:\
MLTutorial\consol.db;eng=Consol"
```

2. Créez une source de données ODBC pour la base distante en entrant la chaîne suivante à l'invite de commandes :

```
dbdsn -w test_remote -y -c "uid=DBA;pwd=SQL;dbf=C:\
MLTutorial\remote.db;eng=remote"
```

A présent, vous pouvez vérifier vos sources de données.

❖ Pour vérifier les nouvelles sources de données

1. Démarrez l'administrateur ODBC.
 - ◆ Choisissez Démarrer ► Programmes ► SQL Anywhere 9 ► Adaptive Server Anywhere ► Administrateur ODBC.L'administrateur de sources de données ODBC apparaît.
2. Cliquez sur l'onglet Sources de données utilisateur.
3. Faites défiler la liste pour rechercher vos nouvelles sources de données.
4. Sélectionnez une source de données et cliquez sur Configurer.
La boîte de dialogue Configuration ODBC pour Adaptive Server Anywhere apparaît.
5. Dans l'onglet ODBC, testez votre source de données en cliquant sur le bouton Tester la connexion.

Le serveur et le client de synchronisation MobiLink utilisent les sources de données ODBC pour se connecter aux bases de données consolidée et distante, respectivement.

Remplissage des bases de données

Vous pouvez désormais créer des tables pour chaque nouvelle base de données initialisée, en exécutant des instructions SQL dans des scripts au moyen d'Interactive SQL. Les scripts contiennent des instructions SQL qui créent des tables dans les bases consolidée et distante et y insèrent des données. De plus, ils génèrent des souscriptions et des publications de synchronisation sur la base distante.

❖ Pour exécuter des scripts depuis Interactive SQL

1. Démarrez Interactive SQL.

- ◆ Dans le menu Démarrer, choisissez Programmes ► SQL Anywhere 9 ► Adaptive Server Anywhere ► Interactive SQL.

ou

- ◆ A l'invite de commandes, tapez *dbisql*.

2. Connectez-vous à la base de données consolidée.

3. Créez une table dans la base consolidée et ajoutez-y quelques lignes :

- ◆ Entrez les instructions suivantes :

```
CREATE TABLE cust (
  cust_id int default autoincrement primary key,
  emp_id int,
  cust_name varchar( 128 )
);
-- add data to cust table
INSERT INTO cust ( emp_id, cust_name ) VALUES ( 1,
  'cust1' );
INSERT INTO cust ( emp_id, cust_name ) VALUES ( 1,
  'cust2' );
INSERT INTO cust ( emp_id, cust_name ) VALUES ( 2,
  'cust3' );
COMMIT;
```

4. Vérifiez si la création de la table a réussi :

- ◆ Entrez la commande suivante et vérifiez que trois lignes ont bien été renvoyées :

```
SELECT * FROM cust
```

5. Répétez les étapes 2, 3 et 4 pour la base de données distante *remote.db* en tapant les instructions SQL suivantes. Aucune ligne n'est ajoutée à la table de la base de données distante.

```
CREATE TABLE cust (cust_id int default autoincrement primary
                    key,
                    emp_id int,
                    cust_name varchar( 128 )
                    )
```

La requête `SELECT * FROM cust` ne doit renvoyer aucune ligne.

Conseil

Pour démarrer une base de données à tout moment, vous pouvez exécuter les commandes suivantes, pour les bases de données consolidée et distante, à partir du répertoire `C :|MLTutorial` :

```
dbeng9 consol.db dbeng9
remote.db
```

Leçon 2 : Préparation des bases de données pour la synchronisation

La synchronisation est gérée par les éléments suivants :

- ◆ **Publications, utilisateurs et souscriptions de synchronisation** Ces éléments sont définis dans chaque base de données distante.
- ◆ **Scripts de synchronisation** Ces scripts sont écrits en langage SQL et stockés dans la base de données consolidée. Vous pouvez également écrire des scripts de synchronisation en langage Java et les stocker dans un emplacement accessible au serveur de synchronisation MobiLink. Dans ce didacticiel, nous utilisons des scripts SQL.

Vous pouvez écrire, visualiser et modifier des scripts de synchronisation, ainsi que des publications et des souscriptions, à partir de Sybase Central.

Création d'une souscription et d'une publication de synchronisation

Une publication, un utilisateur et une souscription de synchronisation MobiLink sont indispensables à la synchronisation MobiLink. Chacun de ces éléments est défini dans la base de données distante.

❖ Pour ajouter une publication et une souscription de synchronisation dans la base de données distante

1. Démarrez Sybase Central et connectez-vous à la base de données distante :
 - ◆ Dans le menu Démarrer, choisissez Programmes ► SQL Anywhere 9 ► Sybase Central.
 - ◆ Cliquez avec le bouton droit sur Adaptive Server Anywhere 9 et sélectionnez Connecter dans le menu contextuel.
 - ◆ Entrez le nom de source de données ODBC test_remote et cliquez sur OK pour vous connecter.
2. Ajoutez une publication à la base de données distante :
 - ◆ Dans Sybase Central, ouvrez la base de données distante.
 - ◆ Ouvrez le dossier Publications.
 - ◆ Dans le menu Fichier, sélectionnez Nouvelle ► publication. L'assistant de création de publication apparaît.
 - ◆ Nommez la publication Customer et cliquez sur Suivant.
 - ◆ Double-cliquez sur la table **cust** pour l'ajouter à la liste des tables sélectionnées, puis cliquez sur Terminer pour créer la publication.
3. Ajoutez un nom d'utilisateur MobiLink à la base de données distante :
 - ◆ Dans Sybase Central, ouvrez la base de données distante.

-
- ◆ Ouvrez le dossier Utilisateurs MobiLink.
 - ◆ Dans le menu Fichier, sélectionnez Nouvel ► utilisateur MobiLink. L'assistant de création d'utilisateur MobiLink apparaît.
 - ◆ Nommez l'utilisateur **ml_user** et cliquez sur Terminer pour le créer.
4. Faites souscrire l'utilisateur MobiLink à la publication :
- ◆ Dans le volet gauche, ouvrez le dossier Publications afin que la publication Customer soit visible.
 - ◆ Dans le volet droit, cliquez sur le nom d'utilisateur MobiLink ml_user.
 - ◆ Faites glisser l'utilisateur MobiLink sur la publication Customer.
 - ◆ Un message vous demande si vous voulez faire souscrire l'utilisateur à la publication. Cliquez sur Oui.
5. Ajoutez des informations d'adresse à la souscription :
- Les informations d'adresse permettent au client MobiLink de localiser le serveur de synchronisation MobiLink approprié.
- ◆ Cliquez avec le bouton droit sur l'utilisateur MobiLink ml_user et sélectionnez Propriétés dans le menu contextuel.
 - ◆ Dans l'onglet Connexion, entrez l'information suivante, en laissant les autres champs vides :
 - ◆ **Host** localhost.
Si le serveur de synchronisation MobiLink est exécuté sur un autre ordinateur, vous devez entrer le nom ou le numéro IP de cette machine au lieu de **localhost**.
 - ◆ Cliquez sur OK pour ajouter les informations d'adresse.

La préparation de la base de données distante est terminée. L'étape suivante consiste à ajouter des scripts de synchronisation à la base de données consolidée.

Ajout de scripts de synchronisation à la base de données consolidée

Chaque script appartient à une **version de script** donnée. Avant d'ajouter des scripts, vous devez ajouter une version de script à la base de données consolidée.

❖ Pour ajouter une version de script

1. Démarrez Sybase Central et connectez-vous à la source de données test_consol au moyen du module de connexion MobiLink.
2. Ouvrez le dossier Versions.
3. Double-cliquez sur Ajouter une version. Nommez la nouvelle version **default**.

❖ **Pour ajouter des tables synchronisées à la base de données consolidée**

1. Ouvrez le dossier Tables situé sous l'icône de synchronisation MobiLink.
2. Ouvrez le dossier DBA. Cliquez avec le bouton droit sur la table cust et ajoutez-la à la liste des tables synchronisées.

Vous avez désigné ces tables comme tables à synchroniser. A présent, vous pouvez ajouter un nouveau script de table pour chaque téléchargement et transfert dans la base de données consolidée.

❖ **Pour ajouter des scripts de table à chaque table synchronisée**

1. Dans le dossier Tables synchronisées, double-cliquez sur la table cust.
2. Dans la partie droite de la fenêtre, double-cliquez sur Ajouter un script de table. L'assistant d'ajout de script de table synchronisée apparaît.
3. Dans la liste déroulante, sélectionnez l'événement **upload_insert**.
4. Cliquez sur Terminer.
5. Dans la boîte de dialogue qui apparaît, tapez le code suivant :

```
INSERT INTO cust ( cust_id, emp_id, cust_name )  
VALUES ( ?, ?, ? )
```

6. Enregistrez le script et fermez la boîte de dialogue.
7. Répétez cette étape pour l'événement download_cursor à l'aide du script suivant :

```
SELECT cust_id, emp_id, cust_name  
FROM cust
```

Vous venez de générer des scripts qui effectuent une synchronisation par clic de votre base de données, en téléchargeant de nouvelles lignes dans la base consolidée et en transférant des lignes de la base consolidée vers la base distante. Dans une installation MobiLink complète, vous ajouteriez des scripts pour d'autres événements.

Leçon 3 : Démarrage du serveur de synchronisation MobiLink

☞ Dans cette leçon, vous allez démarrer le serveur de synchronisation MobiLink afin de pouvoir synchroniser les données des bases consolidée et distante.

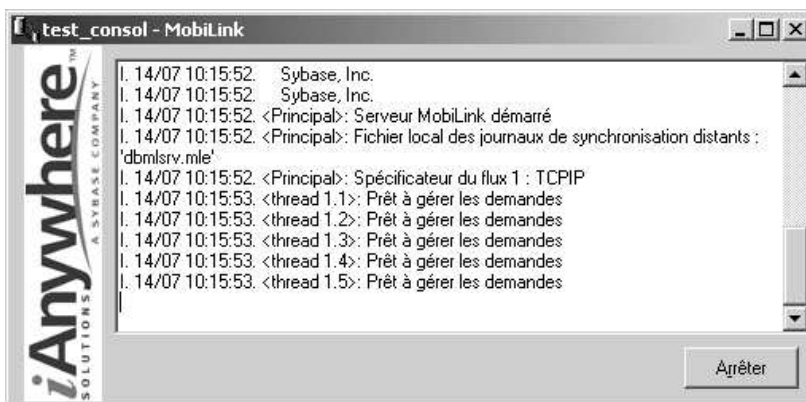
❖ Pour démarrer le serveur de synchronisation MobiLink

1. A l'invite de commandes, tapez :

```
dbmlsrv9 -c "dsn=test_consol" -o mlserver.mls -v+ -dl -zu+
```

☞ Pour plus de détails sur la signification des options, reportez-vous à la section “Options du serveur de synchronisation MobiLink” [*Synchronisation MobiLink - Manuel de référence, page 3*].

Une fois la commande du serveur de synchronisation MobiLink exécutée, les résultats suivants s'affichent.



L'affichage de cet écran signifie que vous êtes prêt à passer à la leçon suivante du didacticiel.

Leçon 4 : Exécution de l'utilitaire client de synchronisation MobiLink

Au cours de cette leçon, vous allez démarrer le client de synchronisation MobiLink. Vous indiquez des paramètres de connexion sur la ligne de commande de l'utilitaire client `dbmlsync` en utilisant l'option `-c`. Ces paramètres concernent la base de données distante.

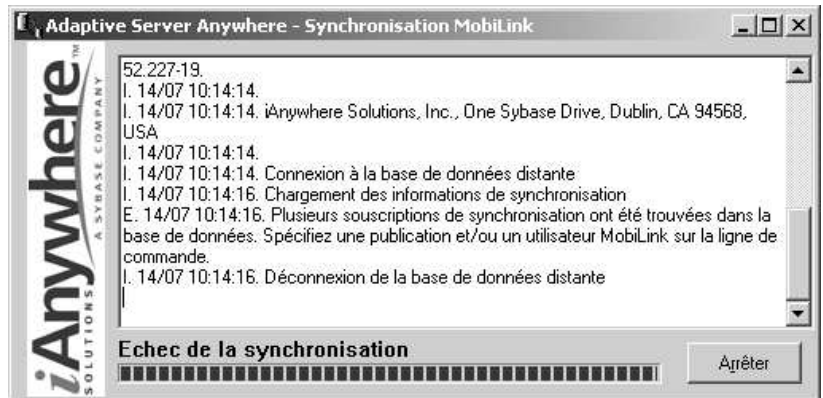
❖ Pour exécuter le client de synchronisation MobiLink

1. Assurez-vous que le serveur de synchronisation MobiLink est démarré.
2. A l'invite de commandes, tapez :

```
dbmlsync -c "dsn=test_remote" -o dbmlsync.out -v
```

☞ Pour plus d'informations, reportez-vous à la section "Client de synchronisation MobiLink" [*Synchronisation MobiLink - Manuel de référence*, page 42].

Une fois la commande du serveur de synchronisation MobiLink exécutée, les résultats suivants s'affichent.



L'affichage de cet écran signifie que la synchronisation a réussi. Vous pouvez ensuite vous connecter à la base de données distante et vous assurer que la table `cust` comporte trois lignes. Ces lignes ont été ajoutées à la base de données consolidée.

Vous pouvez ajouter d'autres lignes aux bases distante et consolidée, puis les synchroniser.

Nettoyage

Au terme de ce didacticiel, n'oubliez pas de supprimer les sources de données et les bases de données correspondantes.

Résumé

Vous disposez à présent de deux bases de données dont vous avez synchronisé le contenu. Vous pouvez à tout moment répéter ces opérations en exécutant le serveur et le client de synchronisation.

Ce didacticiel vous a montré comment :

- ◆ créer des bases de données Adaptive Server Anywhere en vue de les utiliser comme bases de données consolidées et distantes ;
- ◆ écrire des définitions de publication et de souscription de synchronisation ;
- ◆ créer des scripts pour contrôler le téléchargement et le transfert des données ;
- ◆ démarrer le serveur et le client de synchronisation MobiLink et synchroniser la base de données distante avec la base consolidée.

Connaissances acquises Grâce à ce didacticiel, vous savez :

- ◆ utiliser le serveur et le client de synchronisation MobiLink comme un système intégré, ainsi que leurs lignes de commande et leurs options ;
- ◆ exécuter les commandes du serveur et du client de synchronisation MobiLink ;
- ◆ écrire des scripts de synchronisation.

Etape suivante

Il est possible que vous deviez vous familiariser davantage avec le fonctionnement de MobiLink pour en tirer le meilleur parti. Dans cette optique, les supports suivants constituent de bons points de départ :

- ◆ Essayez un autre didacticiel de notre série intitulé **“Introduction”** [*Synchronisation MobiLink - Guide de l'utilisateur*, page 388].
- ◆ Lisez le chapitre de présentation de MobiLink **“Présentation de la synchronisation MobiLink”** [*Synchronisation MobiLink - Guide de l'utilisateur*, page 3].
- ◆ Lisez le chapitre **“Exécution de MobiLink en dehors de la session en cours”** [*Synchronisation MobiLink - Guide de l'utilisateur*, page 343] qui concerne l'exécution du serveur de synchronisation MobiLink.
- ◆ Familiarisez-vous avec l'utilitaire client MobiLink dans **“Client de synchronisation MobiLink”** [*Synchronisation MobiLink - Manuel de référence*, page 42].

CHAPITRE 17

Réplication de données avec SQL Remote

Présentation Ce chapitre explique comment configurer un système de réplication simple par le biais de SQL Remote.

Sommaire	Sujet :	page
	Présentation de SQL Remote	248
	Leçon 1 : Mise en route	249
	Leçon 2 : Configuration de la base de données consolidée	251
	Leçon 3 : Configuration de la base de données distante	255
	Leçon 4 : Réplication des données	258
	Leçon 5 : Restitution de la base de données et de ses paramètres	261
	Résumé	263

Présentation de SQL Remote

La réplication consiste à partager des données entre des bases physiquement distinctes. Les modifications apportées aux données partagées d'une base sont répliquées dans les autres bases de données. Sybase fournit trois stratégies distinctes de réplication de données :

- ◆ SQL Remote
- ◆ MobiLink
- ◆ Replication Server

SQL Remote est conçu pour la réplication bidirectionnelle impliquant un serveur de données consolidées et un grand nombre de bases de données distantes, avec notamment de nombreuses bases de données mobiles. Les impératifs en matière d'administration et de ressources sur les sites distants sont réduits au minimum. Ce système est un système à base de messages.

Dans une installation SQL Remote, la base de données centrale doit être Sybase Adaptive Server Anywhere ou Sybase Adaptive Server Enterprise.

SQL Remote réplique les données en parcourant le journal de transactions pour identifier les opérations qu'il doit transférer et prépare un ou plusieurs messages. Il classe ensuite ces messages dans l'ordre et les envoie au site distant ou consolidé. Pour le traitement des messages reçus, SQL Remote procède toujours en respectant l'ordre dans lequel ils ont été appliqués à l'autre base de données. Lorsque cela s'avère nécessaire, il reporte automatiquement le traitement d'un message jusqu'à ce que tous les messages antérieurs aient été appliqués.

☞ Pour plus d'informations sur les stratégies de synchronisation, notamment pour consulter l'ensemble de la documentation relative à SQL Remote, reportez-vous au document "[Préface](#)" [[SQL Remote - Guide de l'utilisateur, page ??](#)].

☞ Pour une présentation de la synchronisation MobiLink, reportez-vous au chapitre "[Synchronisation de bases de données avec MobiLink](#)" à la [page 231](#).

Leçon 1 : Mise en route

Ce didacticiel explique comment configurer un système de réplication SQL Remote simple.

Avec SQL Remote, vous pouvez effectuer une réplication bidirectionnelle entre une base de données centrale (dite **base de données consolidée**) et un ensemble de **bases de données distantes**. Ces bases de données peuvent résider sur des ordinateurs portables, alors que la base de données consolidée peut se trouver sur un réseau. Toutes les tâches de configuration et d'administration sont exécutées au niveau de la base de données consolidée.

Ce didacticiel présente un cas très simple : la réplication d'une table de la base de données exemple vers une seule autre base de données résidant sur la même machine.

Dans ce didacticiel, vous jouez le rôle de l'administrateur système d'une base de données Adaptive Server Anywhere consolidée. Le système de réplication consiste en une table simple issue de la base de données exemple. La table que nous répliquons est la table `department`, qui est l'une des plus simples de la base de données exemple.

Au cours du didacticiel, vous effectuerez les opérations suivantes :

1. Configuration de la base de données consolidée.
2. Création d'un système de réplication de partage de fichiers avec une seule base de données distante.
3. Réplication de données entre les deux bases.

Création des répertoires pour le didacticiel

Avant de commencer, vous devez créer les répertoires qui hébergeront les bases de données et autres fichiers que vous allez créer au cours du didacticiel. Vous devez également créer une copie de la base de données exemple au cas où vous devriez la recréer dans sa forme initiale.

❖ Pour vous préparer pour le didacticiel

1. Créez un répertoire pour les fichiers que vous créerez avec le didacticiel, par exemple, `c:\tutorial`.
2. Créez deux sous-répertoires de `tutorial` nommés `DBA` et `field`, à savoir `c:\tutorial\DBA` et `c:\tutorial\field`. Ces sous-répertoires stockent les messages pour chacun des deux ID utilisateur du système de réplication.

-
3. Avant tout, faites une copie de la base de données exemple. Créez un dossier pour la copie, par exemple *demoback*, puis copiez-y *asademo.db*.

Etape suivante

Vous êtes maintenant prêt à créer la base de données consolidée.

Leçon 2 : Configuration de la base de données consolidée

Cette section vous explique comment préparer la base de données consolidée d'un système de réplication SQL Remote élémentaire. Pour ce faire, vous devez disposer de l'autorisation DBA.

Pour vous apprendre à préparer une base de données consolidée pour la réplication, cette leçon se divise en 5 étapes :

1. Création du type de message à utiliser pour la réplication.
2. Octroi des autorisations PUBLISH à un ID utilisateur pour identifier la source des messages sortants.
3. Octroi des autorisations REMOTE à tous les ID utilisateur destinataires des messages.
4. Création d'une publication décrivant les données à répliquer.
5. Création de souscriptions décrivant les destinataires de la publication.

Ajout d'un type de message SQL Remote

Tous les messages envoyés dans le cadre de la réplication se fondent sur un type de message. Un type de message se décompose en deux parties :

- ◆ Une liaison de message supportée par SQL Remote. Dans ce didacticiel, nous utilisons le lien FILE. Il s'agit d'une méthode de partage de fichiers, dans laquelle les messages sont conservés dans un fichier sur disque, et lus par le destinataire prévu. SQL Remote prend également en charge les protocoles ftp et e-mail.
- ◆ Une adresse pour le lien de message, permettant d'identifier la source des messages sortants. Dans ce didacticiel, nous utilisons un chemin d'accès au fichier pour spécifier où le message doit être conservé.

Les types de message sont créés dans toutes les nouvelles bases de données, mais vous devez fournir une adresse pour le type de message que vous souhaitez utiliser.

❖ Pour ajouter une adresse à un type de message

1. A partir de Sybase Central, connectez-vous à la base de données exemple sous l'ID utilisateur **DBA** avec le mot de passe **SQL**, et ouvrez le dossier de la base de données exemple (**asademo**) dans le volet gauche.

-
2. Cliquez sur le dossier SQL Remote dans le volet gauche.
 3. Cliquez sur l'onglet Types de message dans le volet droit.
 4. Avec le bouton droit de la souris, cliquez sur le type de message FILE et sélectionnez Propriétés.
 5. Entrez une adresse d'éditeur pour fournir une adresse de retour aux utilisateurs à distance. L'adresse de l'éditeur est le répertoire créé dans la leçon 1 pour conserver les messages dans la base de données consolidée (DBA). Par exemple, `c:\tutorial\DBA`.
 6. Cliquez sur OK pour enregistrer le type de message.

Ajout de l'éditeur à la base de données

Chaque base de données du système de réplication SQL Remote requiert un ID utilisateur unique qui identifie l'**éditeur** des données. Nous adoptons l'ID DBA comme ID utilisateur de l'éditeur.

❖ Pour définir l'éditeur

1. Sélectionnez le dossier Utilisateurs et groupes dans le volet gauche de la fenêtre.
2. Avec le bouton droit de la souris, cliquez sur DBA et sélectionnez Opter pour le type Editeur dans le menu contextuel.
Editeur apparaît dans la colonne Type à côté de DBA.

Une base de données ne peut avoir qu'un seul éditeur. Vous pouvez déterminer à tout moment quel est cet éditeur en ouvrant le dossier Groupes & d'utilisateurs.

Ajout d'un utilisateur à la base de données

Chaque base de données distante est identifiée dans la base consolidée par un ID utilisateur possédant les autorisations REMOTE.

Lorsqu'un utilisateur distant est ajouté à une base de données, le système de messages qu'il utilise et son adresse dans ce système doivent être enregistrés avec son ID utilisateur de base de données.

❖ Ajout d'un utilisateur distant

1. Ouvrez le dossier SQL Remote dans le volet gauche.
2. Choisissez Fichier ► Nouvel utilisateur ► SQL Remote.
L'assistant Création d'utilisateurs distants s'affiche.

3. Entrez **field** comme nom du nouvel utilisateur distant et cliquez sur Suivant.
4. Vérifiez que l'utilisateur est autorisé à se connecter et entrez le mot de passe **field**. Confirmez le mot de passe en le saisissant de nouveau. Cliquez sur Suivant.
5. Sélectionnez les autorisations DBA et DBA distant pour l'utilisateur distant.
6. Sélectionnez le type de message FILE et entrez l'adresse distante **field** dans la zone de texte. Cliquez sur Suivant.
7. Sélectionnez Envoyer et Fermer. Cliquez sur Suivant.
8. Cliquez sur Terminer pour créer l'utilisateur distant.
L'utilisateur distant field apparaît dans le dossier Utilisateurs SQL Remote.

Remarque

Si vous oubliez de définir les autorisations DBA dans l'assistant, il vous est encore possible de le faire en effectuant un clic droit sur l'utilisateur et en cochant les options DBA souhaitées dans l'onglet Autorisation.

Vous venez de créer les utilisateurs qui vont exploiter ce système.

Ajout de publications et de souscriptions

Cette section explique comment ajouter une publication à une base de données et établir une souscription à cette publication pour un utilisateur. La publication réplique toutes les lignes de la table department.

❖ Pour ajouter une publication

1. Ouvrez le dossier Publications.
2. Dans le menu File, sélectionnez New ► Publication.
L'assistant de création de publication apparaît.
3. Sur la première page de l'assistant, attribuez à la publication le nom DepartmentPub. Cliquez sur Suivant.
4. Sélectionnez department dans la liste Tables correspondantes. Cliquez sur Ajouter.
La table apparaît dans la liste Tables sélectionnées, à droite.
5. Cliquez sur Terminer pour créer la publication.

Pour ajouter une souscription

Chaque ID utilisateur devant recevoir les modifications apportées à une publication doit disposer d'une **souscription** à cette publication. Les souscriptions ne peuvent être créées que pour un utilisateur distant valide. Vous devez ajouter une souscription à la publication DepartmentPub pour l'utilisateur de la base de données distante field.

❖ **Pour ajouter une souscription**

1. Ouvrez le dossier Publications.
2. Cliquez sur l'onglet Souscriptions SQL Remote dans le volet droit.
3. Dans le menu Fichier, sélectionnez Nouvelle souscription ► SQL Remote.
La boîte de dialogue Créer une souscription SQL Remote pour un utilisateur distant apparaît.
4. Choisissez de créer une souscription pour l'utilisateur field. Cliquez sur Suivant.
5. Cliquez sur Terminer.

Vous venez de configurer la base de données consolidée.

Etape suivante

Vous pouvez maintenant créer la base de données distante.

Leçon 3 : Configuration de la base de données distante

Pour qu'il soit possible d'envoyer et de recevoir des messages et de figurer dans une configuration SQL Remote, la base de données distante doit être créée et configurée.

L'utilitaire d'extraction de base de données permet d'exécuter toutes les opérations nécessaires pour créer une base distante comportant les souscriptions et les ID utilisateur requis.

Création de la base de données distante

Vous devez extraire une base de données de la base consolidée pour l'utilisateur distant field.

❖ Extraction de la base de données distante

1. Avec le bouton droit de la souris, cliquez sur asademo (DBA) et choisissez Extraire la base de données dans le menu contextuel.
L'assistant d'extraction d'une base de données s'affiche.
2. Cliquez sur Suivant sur la page introductive de l'assistant.
3. Choisissez d'extraire la base de données asademo. Cliquez sur Suivant.
4. Choisissez d'extraire au niveau d'isolement 3. Cliquez sur Suivant.
5. Sélectionnez Démarrer automatiquement les souscriptions pour l'utilisateur. Cliquez sur Suivant.
6. Sélectionnez Extraire et recharger dans une nouvelle base de données. Cliquez sur Suivant.
7. Procédez à l'extraction de la structure et des données. Cliquez sur Suivant.
8. Cochez les cases pour les clés étrangères, les procédures et fonctions, les triggers et les vues.
9. Désélectionnez la case Extraire des définitions de publication entièrement qualifiées. Cliquez sur Suivant.
10. Désélectionnez la case Se connecter à la nouvelle base de données. Cliquez sur Terminer.

La fenêtre de message Extraire la base de données s'affiche en indiquant l'avancement de l'extraction. Une fois l'extraction terminée, fermez la fenêtre.

Remarque

Dans une configuration SQL Remote correcte, la base de données distante **field** aurait été chargée sur l'ordinateur qui l'utilise, avec un serveur de base de données et les applications clientes requises. Dans le cadre de ce didacticiel, nous laissons la base de données telle quelle et recourons à Interactive SQL pour entrer des données et les répliquer.

Vérification de la base de données créée

Pour vérifier les actions effectuées par l'utilitaire d'extraction, connectez-vous à la base de données *field* et vérifiez que tous les objets de base de données sont créés.

❖ Pour naviguer dans la base de données distante

1. A partir de Sybase Central, cliquez sur le bouton Connecter.
La boîte de dialogue Nouvelle connexion apparaît.
2. Sélectionnez Adaptive Server Anywhere 9, puis cliquez sur OK.
3. Dans l'onglet Identification, entrez l'ID utilisateur **field** et le mot de passe **field**.
4. Sous l'onglet Base de données, entrez le chemin d'accès à la base de données (par exemple, dans le champ Fichier de base de données `c:\tutorial\field.db`). Cliquez sur OK pour vous connecter à la base de données.
La base de données *field* s'affiche dans le volet gauche de Sybase Central.
5. Ouvrez le dossier Tables. La table *department*, propriété de l'utilisateur DBA, figure dans la liste.
6. Dans le volet gauche, cliquez sur la table *department*. Dans le volet droit, cliquez sur l'onglet Données pour afficher les cinq lignes de la table *department*.
7. Ouvrez le dossier Utilisateurs SQL Remote et cliquez sur l'onglet Types de message dans le volet droit. **Field** est désigné comme adresse d'éditeur. Toute donnée envoyée à partir de cette base provient de l'utilisateur **field**, de la même façon que les données de la base consolidée proviennent de l'utilisateur DBA.

Sur l'onglet SQL Remote, vous constatez que l'utilisateur ID DBA est désigné comme utilisateur consolidé. Ce qui signifie que DBA est l'éditeur de la base de données consolidée (au-dessus de la base de données *field* dans la hiérarchie).

8. Ouvrez le dossier Publications. Vous pouvez constater la présence de la publication DepartmentPub.

Etape suivante

Le système est maintenant prêt pour la réplication.

Laissez Sybase Central actif pour une utilisation ultérieure dans le didacticiel.

Leçon 4 : Réplication des données

Le système de réplication est à présent configuré. Dans cette section, nous répliquerons les données de la base consolidée vers la base distante et inversement.

Ajout des données à la base de données consolidée

Entrez d'abord une ligne à la base de données consolidée.

❖ Pour entrer des données dans la base consolidée

1. Ajoutez une ligne à la table department.

- ◆ Dans Sybase Central, ouvrez le dossier Tables de la base de données consolidée (asademo), sélectionnez la table department, puis cliquez sur Données dans le volet droit.
- ◆ Cliquez sur le signe plus (+) de la barre d'outils pour ajouter une ligne. Entrez les valeurs suivantes :

Column	Valeur
dept_id	202
dept_name	Eastern Sales
dept_head_id	Laissez la valeur (NULL)

- ◆ Appuyez sur Entrée. Si une boîte de dialogue de confirmation apparaît, cliquez sur Oui pour mettre à jour la ligne.

L'étape suivante consiste à envoyer la nouvelle ligne à la base de données distante.

Pour envoyer des données à la base de données distante, exécutez l'agent de message à partir de la base consolidée. Le programme *dbremote* est l'agent de message d'Adaptive Server Anywhere.

❖ Pour envoyer les données vers la base de données distante

1. A partir d'une invite de commandes, allez dans le répertoire de votre didacticiel. Par exemple, *c :|tutorial*.
2. Exécutez la commande suivante pour exécuter l'agent de message sur la base de données consolidée :

```
dbremote -c "dbn=asademo;uid=DBA;pwd=SQL"
```

La fenêtre SQL Remote apparaît et affiche des messages concernant l'état de la réplication.

3. Lorsque SQL Remote affiche le message `Exécution terminée`, cliquez sur Arrêter.

Pour recevoir l'instruction d'insertion sur la base de données distante, vous devez exécuter l'agent de message `dbremote` sur cette base.

❖ Réception de données par la base distante

1. A l'invite de commandes, positionnez-vous dans le répertoire de votre didacticiel. Par exemple, `c :|tutorial`.
2. Exécutez la commande suivante pour exécuter l'agent de message sur la base de données **field** :

```
dbremote -c "dbn=field;uid=field;pwd=field"
```

La fenêtre SQL Remote s'affiche, vous informant de l'état de la réplication. La fenêtre indique qu'un message a été reçu de DBA.

3. Lorsque SQL Remote affiche le message `Exécution terminée`, cliquez sur Arrêter.

Remarque

- ◆ La fenêtre SQL Remote affiche des informations d'état pendant l'exécution. Ces informations peuvent être enregistrées dans un journal à des fins d'archivage. Vous constatez que l'agent de message reçoit d'abord un message de `asdemo`, puis envoie un message. Ce message de retour contient la confirmation de la réception de la mise à jour de la réplication ; ces confirmations font partie du système de suivi des messages SQL Remote, qui en assure la livraison, même en cas d'erreur du système de messages.
- ◆ Selon l'état actuel de la base de données, il existe trois paramètres de connectivité susceptibles d'être utilisés pour la connexion à une base de données :
 - ◆ **dbf** Connexion à la base de données via le fichier de base de données. Ce paramètre requiert la spécification du fichier de base de données lui-même. Si aucun serveur n'est actif, un serveur est lancé, sur lequel la base de données est chargée. Si un serveur est actif, la base de données est chargée sur le serveur par défaut.
 - ◆ **dbn** Connexion à la base de données via le nom de la base. Ce paramètre requiert la spécification du nom de la base de données. Vous ne pouvez spécifier ce paramètre que lorsque la base de données est déjà active.
 - ◆ **dbS** Connexion à la base de données via la source de la base. Une source de données est un ensemble de paramètres stockés dans le registre du système ou dans un ensemble de fichiers. La source est simplement référencée par son nom.

☞ Pour plus d'informations sur les types de données, reportez-vous à la section "Paramètre de connexion DataSourceName [DSN]" [ASA - Guide d'administration, page 209].

Vérification de l'arrivée des données

A partir de Sybase Central, examinez la table department pour vérifier que la ligne a été reçue.

❖ Pour vérifier l'arrivée des données

1. Ouvrez le dossier *field* de la base de données.
2. Dans le volet gauche, sélectionnez la table department. Dans le volet droit, cliquez sur Données pour visualiser les lignes de la table.
Vous constatez que la table department contient le service Easter Sales dans la base de données consolidée.

Réplication de la base de données distante dans la base de données consolidée

Vous allez maintenant essayer d'accéder aux données situées sur la base distante pour les envoyer à la base consolidée. Seules les grandes lignes de cette procédure sont présentées ici.

❖ Pour répliquer les données de la base de données distante dans la base de données consolidée

1. Insérez une ligne dans la base de données distante, pour un service ayant 203 comme dept_id et Western Sales comme dept_name.
2. Exécutez *dbremote* pour envoyer le message vers la base de données consolidée :

```
dbremote -c "dbn=field;uid=field;pwd=field"
```

3. Exécutez *dbremote* pour recevoir le message dans la base de données consolidée :

```
dbremote -c "dbn=asademo;uid=DBA;pwd=SQL"
```

4. Affichez les données de la table department table dans la base de données consolidée : Elle contient la ligne Western Sales.

Etape suivante

Le didacticiel est désormais terminé, mais il est conseillé de passer à la section suivante pour restaurer la base de données exemple à son état initial.

Leçon 5 : Restitution de la base de données et de ses paramètres

A la fin du didacticiel, vous devez annuler les modifications que vous avez apportées à la base de données exemple. Veillez à exécuter les étapes suivantes pour vous assurer que les paramètres sont correctement réinitialisés.

❖ Pour supprimer l'utilisateur distant

1. Dans le volet gauche de Sybase Central, ouvrez le dossier Groupes & d'utilisateurs de la base de données exemple.
2. Cliquez avec le bouton droit sur l'utilisateur field et sélectionnez Supprimer dans le menu contextuel. Cliquez sur Oui pour supprimer l'utilisateur de la liste.

❖ Pour supprimer la publication

1. Ouvrez le dossier Publications.
2. Cliquez à l'aide du bouton droit sur DepartmentPub dans le volet gauche et sélectionnez Supprimer dans le menu contextuel. Cliquez sur Oui pour confirmer la suppression.

❖ Pour retirer l'état d'éditeur de la base de données exemple

1. Sélectionnez le dossier Utilisateurs et groupes dans le volet gauche de la fenêtre.
2. Cliquez avec le bouton droit sur l'utilisateur DBA dans le volet gauche et sélectionnez Révoquer éditeur dans le menu contextuel.

Révocation d'un état pendant l'exécution de la base de données

L'état d'un utilisateur ne peut pas être modifié tant qu'il exécute la base de données. Vérifiez que l'utilisateur est inactif avant de révoquer ou d'invoquer un état.

❖ Pour restaurer les paramètres initiaux du type de message

1. Sélectionnez le dossier SQL Remote, puis cliquez sur l'onglet Types de message dans le volet de droite.
2. Avec le bouton droit de la souris, cliquez sur le type de message FILE et sélectionnez Propriétés.

-
3. Supprimez l'adresse de l'éditeur et cliquez sur OK pour restaurer les paramètres.

❖ **Pour supprimer la base de données distante**

1. Dans le volet gauche, sélectionnez le module de connexion Adaptive Server Anywhere.
2. Dans le volet droit, cliquez sur l'onglet Utilitaires.
3. Dans le volet droit, double-cliquez sur Supprimer la base de données. L'assistant de suppression d'une base de données s'affiche.
4. Suivez les instructions de l'assistant.
5. Saisissez le nom de la base de données que vous voulez supprimer ou cliquez sur Parcourir pour la sélectionner manuellement. Par exemple, entrez `c:\tutorial\field.db`. Cliquez sur Terminer pour supprimer le fichier de base de données.

Remarque

Maintenant que votre apprentissage est terminé avec le didacticiel, il est conseillé de supprimer son répertoire (`c:\tutorial`) pour libérer de l'espace.

Restauration des données dans la base

Le plus important dans le processus de remise à zéro est de vérifier que toutes les modifications apportées à la base de données exemple sont annulées. L'intégrité de la base de données exemple est en effet essentielle pour exécuter les autres didacticiels proposés plus loin dans ce manuel.

❖ **Pour supprimer les données insérées de la base de données exemple**

1. A partir de Sybase Central, affichez les données de la table `department` de la base de données `asademo`.
2. Supprimez les lignes avec les valeurs `dept_id` 202 et 203 afin de restaurer les paramètres initiaux de la table.

Résumé

Ce didacticiel vous a appris à

- ◆ préparer la base de données consolidée d'un système de réplication élémentaire ;
- ◆ créer et configurer une base de données distante ;
- ◆ répliquer des données dans les deux sens entre les deux bases ;
- ◆ restaurer la base de données et ses paramètres ;
- ◆ vérifier toutes les étapes.

Etape suivante

☞ Pour plus d'informations sur SQL Remote, reportez-vous à la section "Préface" [[SQL Remote - Guide de l'utilisateur, page ??](#)].

☞ Pour une consulter une présentation de la synchronisation MobiLink, reportez-vous au chapitre "[Synchronisation de bases de données avec MobiLink](#)" à la page 231.

CHAPITRE 18

Conception de bases de données avec PowerAMC

Présentation SQL Anywhere Studio comprend PowerAMC Développeur SQL, un module du puissant outil de conception de base de données Sybase, PowerAMC. Ce module permet de générer et de modifier des bases de données à l'aide d'une représentation graphique de la structure des bases de données. Vous pouvez optimiser votre base de données en personnalisant les tables, les colonnes, les index, l'intégrité référentielle, les vues, le stockage physique, les triggers et les procédures stockées.

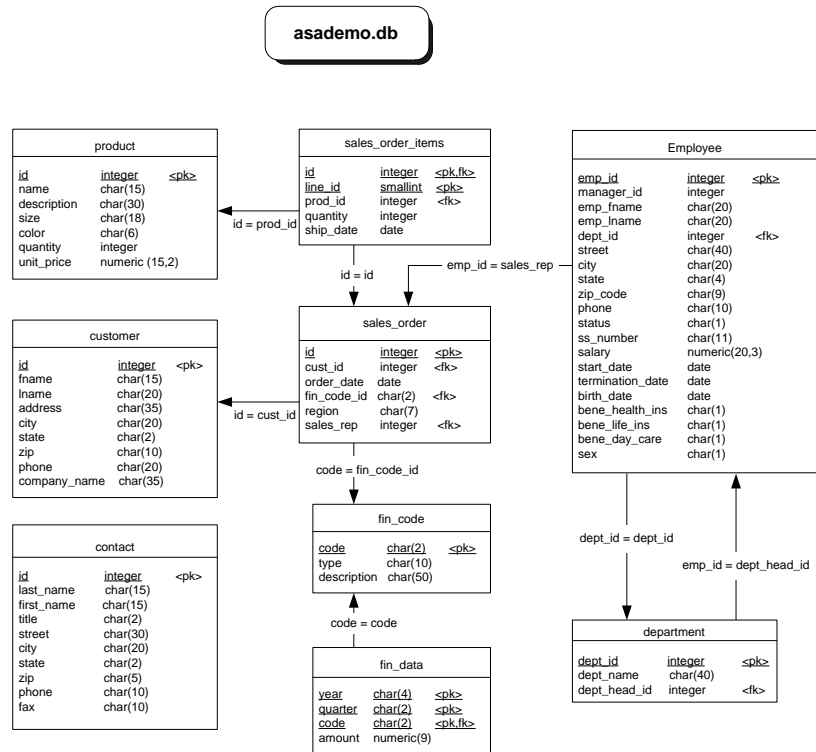
Sommaire	Sujet :	page
	Présentation de PowerAMC	266
	Leçon 1 : Mise en route	268
	Leçon 2 : Ajout d'une colonne	273
	Leçon 3 : Vérification de votre travail	275
	Leçon 4 : Enregistrement des modifications et génération de la base de données	276
	Résumé	279

Présentation de PowerAMC

La structure de votre base de données (tables, relations, vues, triggers etc.) est appelée **schéma** de la base de données. Pour créer et organiser ces éléments, vous utilisez des instructions SQL, mais sans outil graphique, il est parfois difficile de s’y retrouver.

PowerAMC vous offre une représentation graphique de la structure de la base de données. Mieux, vous pouvez modifier entièrement la structure d’une base de données ou en créer une, en traçant simplement de nouvelles tables ou en saisissant des informations. Une fois votre conception au point, PowerAMC peut générer un script SQL pour créer la base de données.

Le diagramme suivant, qui montre la structure de la base de données exemple, est facile à créer avec PowerAMC.



Les performances de votre base de données dépendent étroitement de sa conception. D’une façon générale, il convient de stocker dans des tables distinctes les informations sur des objets de types différents (employés, produits, etc.).

Vous pouvez identifier les relations entre les tables par le biais de références, les clés étrangères d'une table identifiant des lignes particulières d'une autre table. Nombre de relations de type n-1 et 1-n peuvent être représentées par une référence. Les relations n-n requièrent deux références et une table supplémentaire.

☞ Pour plus d'informations sur la conception d'une base de données, reportez-vous à la section [“Conception d'une base de données” \[ASA - Guide de l'utilisateur SQL, page 3\]](#).

☞ Pour plus d'informations sur PowerDesigner, cliquez sur Help dans la barre d'outils de la fenêtre principale de PowerAMC pour accéder aux trois manuels suivants : *Guide des fonctionnalités générales de PowerAMC*, *Guide de l'utilisateur du Modèle Physique de Données* et *Guide de l'utilisateur des Rapports*.

☞ Pour d'autres didacticiels PowerAMC, sélectionnez Démarrer ► Programmes ► SQL Anywhere 9 ► PowerAMC 8 ► MPD Mise en route. Cela vous permet d'accéder au document *Modèle Physique de Données - Mise en route*.

Leçon 1 : Mise en route

PowerAMC peut lire la structure d'une base de données à partir d'un fichier script qui crée la base de données. Il est toutefois souvent plus simple de vous connecter à votre base de données à partir de PowerAMC et de lui laisser le soin d'en déduire lui-même la conception avec la fonctionnalité rétro-ingénierie.

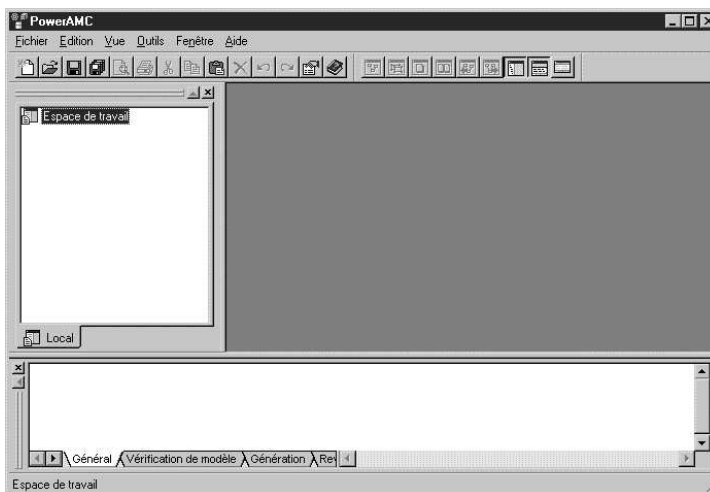
Le didacticiel suivant part de la base de données exemple. Il illustre les possibilités de PowerAMC en mettant en oeuvre une modification qui améliore la conception de la base de données exemple.

Le prix de chaque produit est toujours dans la table product. En conséquence, la mise à jour du prix modifie le prix de vente de cet article dans tous les bons de commande antérieurs. L'ajout d'une colonne `unit_price` à la table `sales_order_items` permet de corriger cette anomalie. Le prix réel de vente à chaque client peut alors être enregistré distinctement. Le prix indiqué dans la table `product` correspond au tarif actuel.

❖ Pour démarrer PowerAMC

1. Dans le menu Démarrer, choisissez Programmes ► SQL Anywhere 9 ► PowerAMC 8 ► PowerAMC.

La fenêtre principale de PowerAMC s'affiche.



La fenêtre principale PowerAMC inclut un explorateur situé sur la gauche et une fenêtre de résultats en bas.

2. Choisissez Fichier ► Nouveau.

3. Dans la boîte de dialogue Nouveau, sélectionnez Modèle Physique de Données et cliquez sur OK.
La boîte de dialogue Choix du SGBD s'affiche.
4. Dans la liste déroulante, sélectionnez Sybase ASAanywhere9.
5. Assurez-vous que Partage est sélectionné et cliquez sur OK.
Le nom de modèle MODL_1 s'affiche dans l'explorateur et la barre de titres.

Rétro-ingénierie de la base de données

Dans cette section, vous générez un Modèle Physique de Données de la base de données exemple en effectuant une rétro-ingénierie.

❖ Pour effectuer une rétro-ingénierie de la base de données

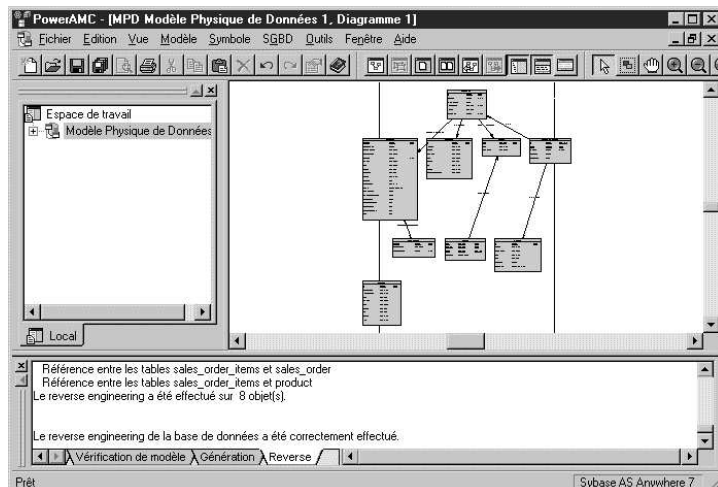
1. Cliquez dans la fenêtre du diagramme (le grand volet central de PowerAMC).
2. Dans le menu SGBD de la barre d'outils, choisissez Reverse Engineering.
La boîte de dialogue Reverse Engineering d'une base de données s'affiche.
3. Vérifiez que l'option A l'aide d'une source de données ODBC est sélectionnée et que la source de données est ASA 9.0 Sample.
Si ASA 9.0 Sample n'apparaît pas, cliquez sur l'icône à droite de la source des données. La boîte de dialogue Connexion à une source de données ODBC s'affiche. Sélectionnez la source de données de la machine puis ASA 9.0 Sample dans la liste déroulante. Indiquer l'ID utilisateur **DBA** et le mot de passe **SQL**. Cliquez sur Connecter pour retourner à la boîte de dialogue Reverse Engineering d'une base de données.
4. Dans la boîte de dialogue Reverse Engineering d'une base de données, cliquez sur OK.
La boîte de dialogue Reverse Engineering via ODBC s'affiche.



Dans la partie inférieure de la boîte de dialogue, six cases à cocher vous permettent de sélectionner des clés, des index etc. Ils s'agit des **options de rétro-ingénierie**. Assurez-vous qu'elles sont toutes activées (valeur par défaut). Vérifiez également que toutes les tables sont sélectionnées (valeur par défaut également).

5. Cliquez sur OK pour effectuer une rétro-ingénierie de la base de données.

Une représentation graphique de la base de données exemple apparaît dans la fenêtre du diagramme et les objets du modèle s'affichent dans l'explorateur :



6. Examinez le schéma affiché. Vous pouvez effectuer un zoom avant à l'aide de F6 ou un zoom arrière à l'aide de F7, pour rendre les tables plus lisibles.

- ◆ Entraînez-vous à utiliser les boutons de la barre d'outils :



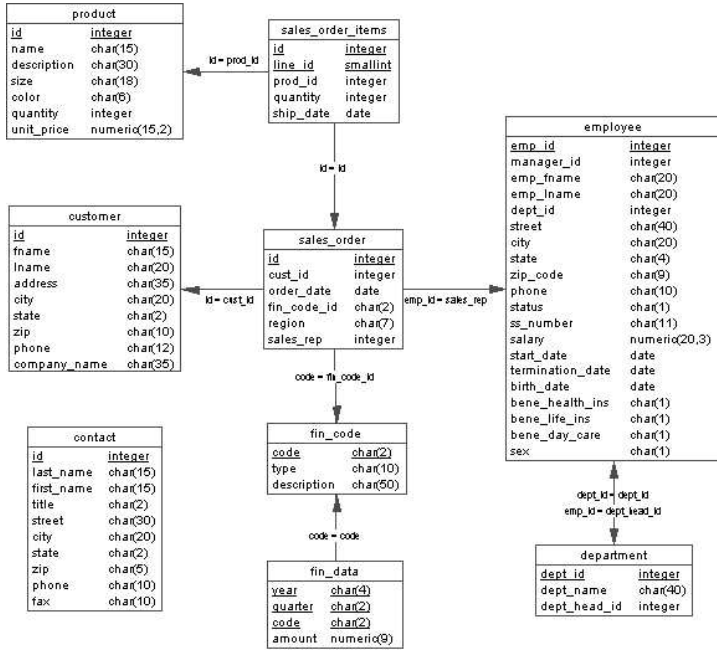
- ◆ Choisissez Outils ► Préférences d'affichage pour afficher les options permettant de modifier l'affichage. Cliquez sur chaque option se trouvant à gauche pour visualiser les pages des paramètres de préférence d'affichage. Cliquez sur le bouton Aide en bas de chaque page pour obtenir des informations complètes.

Chaque table de la base de données est représentée par une case. Le nom de la table s'affiche en haut. En dessous se trouve une liste de colonnes. Les noms de colonne soulignés correspondent aux colonnes définies comme éléments de la clé primaire de la table. Le type de données de chaque colonne est indiqué à droite. Il est possible que certaines tables soient superposées après la rétro-ingénierie, mais vous pouvez les réorganiser à l'aide de l'outil Disposition automatique du menu Symbole.

Les références entre les tables sont représentées par des flèches. La flèche pointe sur la table parente, c'est-à-dire la table contenant la clé primaire. Une équation en regard de chaque flèche identifie la référence.

7. Réorganisez les tables de sorte que le diagramme soit plus lisible. Pour cela, faites-les glisser avec votre souris. Les flèches de référence suivent automatiquement le mouvement. Vous pouvez sélectionner plusieurs objets en maintenant enfoncée la touche MAJ.

Une disposition possible est présentée ci-dessous.



Leçon 2 : Ajout d'une colonne

Vous êtes maintenant prêt à ajouter la colonne `unit_price` à la table `sales_order_items`. Pour ce faire, vous pouvez accéder à la liste des colonnes via la feuille de propriétés de la table.

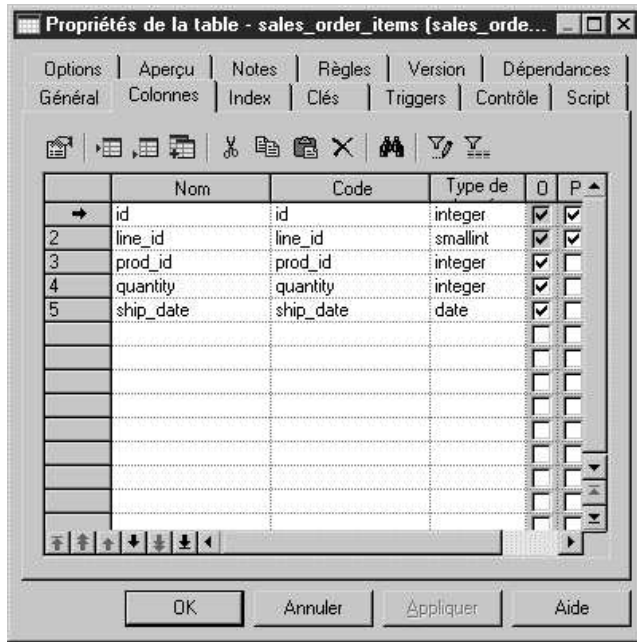
❖ Pour ajouter une colonne

1. Affichez les propriétés de la colonne.
 - ◆ Cliquez avec le bouton droit sur la table `sales_order_items` et sélectionnez Propriétés dans le menu contextuel.

La feuille de propriétés apparaît.



- ◆ Cliquez sur l'onglet Colonnes.
La liste des colonnes s'affiche.



2. Ajoutez une nouvelle colonne pour le prix unitaire.
 - ◆ Cliquez sur l'outil Insérer une ligne.
Une flèche apparaît en début de ligne, ainsi qu'une colonne avec un nom par défaut (Colonne_6).
 - ◆ Tapez **unit_price** dans la colonne name. Le nom est automatiquement dupliqué comme code.
 - ◆ Dans la liste déroulante, sélectionnez numeric. (Le champ Type de données est quelquefois trop étroit pour permettre la lecture. Vous pouvez élargir la colonne en tirant sur ses côtés.)
 - ◆ Les propriétés de colonne P, E et O correspondent à Clé primaire, Clé étrangère et Obligatoire, comme suit :
 - ◆ Clé primaire désigne une colonne dont les valeurs identifient de façon univoque une ligne de la table.
 - ◆ Clé étrangère désigne une colonne dépendante d'une colonne clé primaire d'une autre table et qui en est dérivée.
 - ◆ Obligatoire désigne une colonne à laquelle une valeur doit être affectée.
 - ◆ Sélectionnez Obligatoire et cliquez sur OK.
3. Examinez l'effet de vos modifications sur le schéma de la base de données. La table sales_order_items inclut désormais une nouvelle colonne intitulée unit_price.

Leçon 3 : Vérification de votre travail

PowerAMC permet de détecter rapidement les erreurs de conception de base de données dans le nouveau modèle.

❖ Pour vérifier votre nouveau schéma

1. Choisissez Outils ► Vérifier le modèle.

La boîte Paramètres de vérification de modèle s'affiche. Vous pouvez utiliser les paramètres par défaut.

2. Cliquez sur OK.

Les résultats de la vérification du modèle apparaissent dans la Liste de résultats.

Leçon 4 : Enregistrement des modifications et génération de la base de données

Dans PowerAMC, les modèles qui décrivent les composants physiques de votre conception de base de données, notamment les tables et les colonnes, sont appelés **Modèles Physiques de Données (PDM, Physical Data Models)**. PowerAMC enregistre ces fichiers avec l'extension *.PDM*.

❖ Pour sauvegarder le modèle physique de données

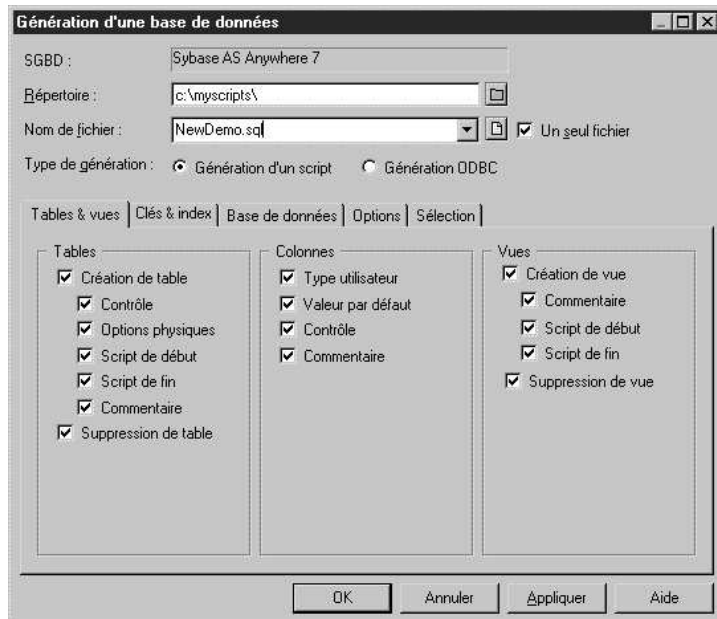
1. Sélectionnez Fichier ► Enregistrer sous.
2. Tapez un nom de fichier portant l'extension *.PDM* et sélectionnez un emplacement.
3. Cliquez sur OK.

Vous pouvez utiliser PowerAMC pour générer un script SQL qui mettra en oeuvre tous les composants de votre modèle. Vous pouvez alors utiliser le script SQL pour générer une base de données.

❖ Pour générer un script SQL qui crée la nouvelle base de données

1. Sélectionnez SGBD ► Générer une base de données.

La boîte de dialogue Génération d'une base de données s'affiche.



2. Nommez le script **NewDemo.SQL** et choisissez un répertoire adapté.
3. Vérifiez que Génération d'un script est sélectionné.
4. Cliquez sur l'onglet Base de données et vérifiez que l'onglet Création de la base est sélectionné.

Explorez les autres onglets pour observer les options qui vous permettent de contrôler les nombreuses autres propriétés du script généré.

5. Cliquez sur OK.

Lorsque le script est créé, la fenêtre de résultat s'affiche.

6. Cliquez sur Editer pour afficher le script.

Vérifiez que vos modifications ont été intégrées dans le script. Par exemple, voici la définition de la nouvelle table office ci-dessous.

```
/* ===== */
/* Table: office */
/* ===== */
create table office
(
    id            integer            not null
                default autoincrement
                check (
                    id >= 100),
    name          char(15)           not null,
    street        char(30)           not null,
    city          char(20)           not null,
    state         char(2)            not null,
    zip           char(5)            not null,
    phone        char(10)            ,
    fax          char(10)            ,
    primary key (id)
);
```

7. Lorsque vous avez terminé, fermez la boîte de dialogue : dans la fenêtre de résultat, cliquez sur Fermer.

Vous pouvez créer la nouvelle base de données à partir d'Interactive SQL.

❖ Pour créer une base de données

1. Démarrez Interactive SQL.
2. Connectez-vous à la base de données exemple. Vous pouvez utiliser la même connexion ODBC.
3. Créez une base de données vide.

-
- ◆ Exécutez l'instruction SQL suivante, en substituant tout répertoire adéquat.

```
CREATE DATABASE 'c:\\My Scripts\\newdemo.db'
```

Conseil

Pour exécuter une instruction SQL dans Interactive SQL, saisissez-la ou copiez-la dans le volet Instructions SQL, puis appuyez sur F5 ou choisissez SQL ► Exécute.

4. Fermez la connexion à la base de données exemple en choisissant SQL ► Déconnecter.
5. Cliquez sur SQL ► Connecter pour vous connecter à la base de données.
 - ◆ Entrez **DBA** comme ID utilisateur.
 - ◆ Entrez **SQL** comme mot de passe
 - ◆ Cliquez sur l'onglet Base de données, et dans la zone Fichier de base de données, entrez le chemin complet et le nom de fichier du nouveau fichier de base de données.
 - ◆ Cliquez sur OK.
6. Utilisez l'instruction read. N'oubliez pas que cette instruction requiert de placer le nom du fichier entre guillemets.
 - ◆ Exécutez l'instruction SQL suivante :

```
READ "c:\\My Scripts\\newdemo.SQL"
```

Vous pouvez de la même façon modifier d'autres bases de données.

Résumé

Ce didacticiel a uniquement présenté les fonctionnalités élémentaires de PowerAMC. En fait, PowerAMC est capable de gérer l'intégralité de la conception ou de la modification du schéma d'une base de données, notamment la totalité des tables, vues, index, références, triggers et procédures.

Domaines

D'autres fonctionnalités simplifient la conception de bases de données de taille plus importante. Vous pouvez par exemple spécifier des **domaines** particuliers. Un domaine contient un type particulier de données, un numéro de téléphone par exemple. Un type de données lui est associé, mais sa spécificité ne s'arrête pas là. Par exemple, vous pouvez créer un domaine de numéros d'identification. Lorsque vous avez besoin d'un numéro d'identification dans une table, vous pouvez associer la colonne concernée au domaine du numéro d'identification. Toutes les propriétés et les contrôles associés au domaine sont automatiquement rattachés.

Les domaines minimisent les définitions répétitives. Ce faisant, ils réduisent non seulement votre travail, mais également le risque d'utiliser par erreur une autre définition de type ou de procédure de contrôle. Au lieu d'identifier une colonne comme un simple entier, vous pouvez spécifier le type particulier de données qu'elle contient. Toutes les instances de ce type de données partagent la même définition de contenu de la colonne.

Règles de gestion

Une règle de gestion est une expression écrite indiquant le mode de fonctionnement d'une activité. Par exemple, *la date de livraison doit être supérieure ou égale à la date de la commande*.

Les règles de gestion se répartissent en quatre catégories :

- ◆ **Définition** Expriment les propriétés inhérentes à un objet. Les définitions décrivent normalement des entités.
- ◆ **Fait** Expriment la certitude ou l'existence. Les faits décrivent généralement des relations.
- ◆ **Validation** Contrainte sur une valeur.
- ◆ **Formule** Calcul générant une valeur.

Les règles de gestion sont particulièrement pratiques car elles sont directement liées à la tâche qu'un client demande à la base de données d'exécuter. En enregistrant des règles de gestion et en les rattachant à des objets particuliers, vous pouvez vous assurer qu'une base de données exécute les tâches requises.

Etape suivante

☞ Pour plus d'informations sur PowerAMC, cliquez sur Help dans la barre d'outils de la fenêtre principale de Power AMC pour accéder aux trois manuels suivants : *Guide des fonctionnalités générales de PowerAMC*, *Guide de l'utilisateur du Modèle Physique de Données* et *Guide de l'utilisateur des Rapports*.

☞ Pour d'autres didacticiels PowerAMC, sélectionnez Démarrer ► Programmes ► Sybase SQL Anywhere 9 ► PowerAMC 7 ► MPD Mise en route. Cela vous permet d'accéder au document *Modèle Physique de Données - Mise en route*.

☞ Pour plus d'informations sur la conception d'une base de données, reportez-vous à la section "[Conception d'une base de données](#)" [*ASA - Guide de l'utilisateur SQL*, page 3].

CHAPITRE 19

Génération d'états avec InfoMaker

Présentation Ce chapitre comporte un bref didacticiel sur InfoMaker qui vous aide à démarrer et vous indique les techniques élémentaires requises pour être efficace dans l'environnement InfoMaker.

Sommaire	Sujet :	page
	Présentation d'InfoMaker	282
	Leçon 1 : Mise en route	283
	Leçon 2 : Génération d'un état élémentaire	284
	Leçon 3 : Amélioration de l'état	287
	Résumé	291

Présentation d'InfoMaker

InfoMaker est un outil puissant qui permet la génération d'états et la maintenance des données. Avec InfoMaker, vous pouvez créer les objets suivants :

- ◆ Etats permettant de visualiser les données.
- ◆ Formulaires permettant de visualiser et de modifier les données.
- ◆ Requêtes permettant de récupérer automatiquement des données pour des états ou des formulaires.
- ◆ Pipelines permettant de conduire les données d'une base de données (ou SGBD) à une autre base.
- ◆ Applications permettant de regrouper les états et formulaires et de les distribuer aux utilisateurs.

InfoMaker fournit une connectivité intégrée à une large gamme de bases de données de bureau ou basées sur serveur. Lorsque vous utilisez InfoMaker, vous travaillez dans un environnement graphique, ce qui signifie que vous n'avez pas besoin de comprendre SQL pour travailler dans cet environnement. InfoMaker génère toutes les instructions SQL en arrière-plan pendant que vous construisez graphiquement vos états et autres objets.

☞ Pour plus d'informations sur InfoMaker, choisissez Démarrer ► Programmes ► Sybase SQL Anywhere 9 ► InfoMaker 8.0 ► Online Help Files. Vous pouvez avoir accès à la documentation complète au sein de l'application en cliquant sur le menu Help.

Leçon 1 : Mise en route

Pour utiliser ce didacticiel InfoMaker, vous devez être connecté à la base de données exemple Adaptive Server Anywhere. Pour vous connecter à la base de données exemple SQL Anywhere Studio 8.0, vous devez au préalable créer un profil de base de données.

❖ Pour démarrer InfoMaker

1. Dans le menu Démarrer, choisissez Programmes ► Sybase SQL Anywhere 9 ► InfoMaker 8.0 ► InfoMaker.

La première fenêtre InfoMaker s'affiche. Elle propose la PowerBar, qui comporte des boutons donnant accès aux Painters InfoMaker et à l'aide en ligne.

❖ Pour créer un profil de base de données pour la base de données exemple de SQL Anywhere Studio avec InfoMaker

1. Dans InfoMaker, ouvrez la boîte de dialogue Database Profiles en cliquant sur DB Profile dans la PowerBar.
2. Sélectionnez ODBC et cliquez sur New.
3. La boîte de dialogue Database Profile Setup s'affiche.
 - ◆ Entrez le nom de profil **Anywhere**.
 - ◆ Sélectionnez ASA 9.0 Sample comme Data Source (source de données).
 - ◆ Entrez **DBA** comme ID utilisateur.
 - ◆ Entrez **SQL** comme mot de passe.
 - ◆ Cliquez sur OK.

Le profil Anywhere doit maintenant être affiché sous ODBC.

❖ Pour vous connecter à la base de données exemple d'Adaptive Server Anywhere une fois la source de données créée

1. Si ce n'est pas le cas, ouvrez la boîte de dialogue Database Profiles en cliquant sur DB Profile dans la PowerBar.
2. Sélectionnez Anywhere dans le groupe ODBC et cliquez sur Connect.

Etape suivante

Vous allez maintenant créer une table à l'aide d'un assistant InfoMaker.

Leçon 2 : Génération d'un état élémentaire

Cette section explique comment générer un état élémentaire, comment le prévisualiser et l'enregistrer, et comment modifier les paramètres de votre environnement de conception.

Génération d'un état

Vous pouvez générer différents types d'états. Cette section explique comment créer une table.

❖ Pour générer l'état, procédez comme suit

1. Cliquez sur le bouton New.
2. Dans la boîte de dialogue New, sélectionnez l'onglet Object.
3. Double-cliquez sur l'icône Tabular. Vous sélectionnez ainsi le style de présentation tabulaire.
L'assistant de création des états tabulaires apparaît.
4. Sélectionnez la source de données Quick Select et vérifiez que Retrieve on Preview est sélectionné. Cliquez sur Suivant.
La boîte de dialogue Quick Select s'affiche. Vous pouvez y sélectionner une table de base de données et l'ensemble ou une partie de ses colonnes.
5. Sélectionnez la table contact.
6. Sélectionnez les colonnes suivantes : last_name, first_name, title, phone et fax. (Vous devrez peut-être faire défiler la fenêtre vers le bas.)
InfoMaker affiche les colonnes sélectionnées dans la grille située en bas de la boîte de dialogue. Dans cette grille, vous pouvez réorganiser les colonnes et indiquer des critères de tri et de sélection.
7. Cliquez sur OK.
La boîte de dialogue Select Color and Border Settings s'affiche.
8. Vous pouvez utiliser les paramètres par défaut. Cliquez sur Next.
Une boîte de dialogue récapitulant vos spécifications s'affiche.
9. Cliquez sur Terminer.

Aperçu de l'état

Dans cette section, vous allez prévisualiser l'état pour savoir comment il se présentera une fois imprimé.

La fenêtre InfoMaker principale comporte des barres d'outils dans sa section supérieure. Sous ces barres d'outils se trouve un volet de prévisualisation qui occupe le reste de la fenêtre. Une barre de titres se trouve tout en haut du volet. A sa gauche figure le nom de l'état prévisualisé tandis qu'à sa droite se trouvent les boutons Minimize/Maximize qui permettent de régler la taille du volet de prévisualisation.

❖ **Pour prévisualiser votre état**

1. Placez la pointe du pointeur dans l'angle supérieur droit du volet de prévisualisation et cliquez sur le bouton Maximize.

Le volet de prévisualisation occupe toute la fenêtre. Il contient désormais les informations d'en-tête relatives à l'état, ainsi que des informations sur la base de données.

2. Pour apprendre à désactiver les règles, cliquez dans la section contenant les données et choisissez File ► Print Preview Rulers.
3. Cliquez sur le bouton Minimize dans le coin supérieur droit du volet de prévisualisation pour redonner au panneau de prévisualisation sa taille et son emplacement d'origine.

Enregistrement d'un état

❖ **Pour enregistrer l'état, procédez comme suit**

1. Choisissez File ► Save.
2. Dans la zone Reports, saisissez **contacts_by_jobrole**. Cela devient le nom de l'état.
3. Cliquez dans la zone Comments et saisissez **This report shows my contacts grouped by job role**.
4. Appuyez sur Entrée.

Configuration de l'environnement de conception

Dans cette section, vous allez apporter des modifications aux contrôles, à la grille et à la règle.

❖ **Pour configurer l'environnement de conception, procédez comme suit**

1. Sélectionnez Options dans le menu Design.
La feuille de propriétés Report Options s'affiche.

-
2. Assurez-vous que les options suivantes sont sélectionnées : Show Grid, Show Ruler et Show Edges.
 3. Assurez-vous que Snap to Grid n'est pas sélectionné.
 4. Cliquez sur Aide pour afficher la description de toutes les options.
 5. Cliquez sur OK.

Etape suivante

Vous êtes maintenant prêt à personnaliser la présentation de votre état.

Leçon 3 : Amélioration de l'état

Dans cette section, vous allez apprendre à apporter un certain nombre d'améliorations à votre état, notamment

- ◆ Tri des données.
- ◆ Création et formatage d'en-têtes et de titres.
- ◆ Ajout de champs calculés du type dates, numéros de page et totaux.

Définition du tri et du regroupement

Dans cette section, vous allez regrouper les contacts par titre. Pour cela, vous allez trier les données par titre puis spécifier le regroupement par titre.

❖ Pour regrouper les contacts

1. Définissez l'ordre de tri. L'option de tri croissant est l'option par défaut.
 - ◆ Sélectionnez Sort dans le menu Rows.
La boîte de dialogue Specify Sort Columns s'affiche.
 - ◆ Faites glisser la zone title de la zone Source Data à la zone Columns.
 - ◆ Faites glisser last_name et first_name dans la zone Columns.
 - ◆ Cliquez sur OK.
2. Définissez le regroupement en fonction de la colonne title pour que tous les contacts ayant le même titre soient regroupés :
 - ◆ Dans le menu Rows, choisissez Create Group.
La boîte de dialogue Specify Group Columns s'affiche.
 - ◆ Faites glisser le titre dans la zone Columns et cliquez sur OK.

Le regroupement est terminé. Développez le volet de prévisualisation comme vous l'avez fait précédemment dans la leçon et vous pourrez constater que les données sont triées. Le regroupement n'apparaît pas encore.

Amélioration de l'état

Dans cette section, nous allons améliorer l'état en réorganisant les contrôles, en ajoutant un titre et une date, en ajoutant des numéros de page et en ajoutant un total.

❖ Pour améliorer l'état

1. Déplacez le rôle de travail (job rôle) sur l'en-tête de groupe et modifiez sa police :
 - ◆ Déplacez le pointeur sur la barre grise intitulée 1 : Header Group Title. Le pointeur prend la forme d'une double flèche.
 - ◆ Faites glisser la section vers le bas de cinq points de grille.
 - ◆ Faites glisser la colonne title du bas du volet de prévisualisation vers le bas de l'en-tête de groupe.
 - ◆ La colonne title étant toujours sélectionnée, cliquez sur B (bold pour gras) et I (italiques) sur la barre des styles.
Examinez l'état dans le volet de prévisualisation.
2. Corrigez les en-têtes :
 - ◆ Faites glisser les boîtes de texte First Name et Last Name dans la section En-tête de page, sur la droite, pour remplir l'espace vacant.
 - ◆ Faites glisser les colonnes first_name et last_name dans la section Détail, sur la droite, pour remplir l'espace vacant.
 - ◆ Dans le menu Edit, choisissez Select ► Select Text.
Tous les en-têtes sont alors sélectionnés. Il est parfois difficile de s'en rendre compte vu l'étroitesse de la section.
 - ◆ Dans la barre de styles, cliquez sur le bouton Left Justification. Si les en-têtes ne sont pas en gras, cliquez sur Bold.
Examinez l'état dans le volet de prévisualisation.
3. Ajoutez un titre à l'état :
 - ◆ Faites glisser la barre grise Header vers le bas de 10 points de grille.
 - ◆ Dans le menu Edit, choisissez Select ► Select Text. Tous les en-têtes sont alors sélectionnés.
 - ◆ Faites glisser tous les en-têtes vers le bas près de la bande grise.
 - ◆ Dans le menu Insert, choisissez Control ► Text.
 - ◆ Placez la pointe du pointeur sur la boîte First Name, un point en dessous du haut de la page. Cliquez. Cette opération positionne la boîte de texte pour le titre.
 - ◆ Tapez **My Contacts**.
 - ◆ Sélectionnez 14 dans la liste déroulante dans la barre de styles.
4. Ajoutez une date à l'état :
 - ◆ Dans le menu Insert, choisissez Control ► Today().

- ◆ Placez l'extrémité du pointeur dans le coin supérieur gauche de l'état et cliquez. La date est alors intégrée à l'état.
5. Ajoutez des numéros de page à l'état :
- ◆ Faites glisser la barre grise Footer vers le bas de quatre pas de grille. Si nécessaire, servez-vous de la barre de défilement pour ménager de l'espace en bas de la vue Design. Le fait de faire glisser vers le bas la barre de bas de page réserve de l'espace pour le numéro de page dans la bande de bas de page.
 - ◆ Dans le menu Insert, choisissez Control ► Page n of n.
 - ◆ Placez le pointeur à environ deux pas de grille en-dessous du centre de la bande de bas de page. Cliquez.
 - ◆ Sélectionnez la taille 10 dans la barre de style.
6. Ajoutez un total en faisant le total du nombre de noms de famille :
- ◆ Faites glisser la barre grise Summary vers le bas de six points de grille. L'espace est ainsi ménagé pour le total.
 - ◆ Dans le menu Insert, choisissez Control ► Computed Field.
 - ◆ Placez le pointeur à environ quatre pas de grille en-dessous du centre de la bande de récapitulatif. Cliquez.
La boîte de dialogue Modify Expression s'affiche.
 - ◆ Dans la zone Functions, cliquez sur Count(#x for all).
 - ◆ Dans la zone Columns, cliquez sur last_name.
 - ◆ Cliquez sur Verify.
Vous recevez un message indiquant que votre expression est correcte. Cliquez sur OK.
 - ◆ Cliquez sur OK pour terminer la définition de l'expression pour le champ calculé.
7. Formatez le total :
- ◆ Dans le menu Insert, choisissez Control ► Text.
 - ◆ Positionnez le pointeur à gauche du champ calculé et cliquez une fois.
 - ◆ Tapez **Total contacts :**.
 - ◆ Faites glisser la boîte de texte de façon à l'aligner avec le champ calculé.
 - ◆ Encadrez le texte et le champ calculé en dessinant un cercle autour d'eux avec le pointeur tout en maintenant le bouton gauche de la souris enfoncé, puis en relâchant le bouton de la souris.
 - ◆ Dans la barre de styles, sélectionnez la taille de police 10, puis cliquez sur B et I.

❖ **Pour imprimer, enregistrer et fermer l'état**

1. Dans le menu Fichier, choisissez Print Report.
2. Dans la boîte de dialogue Windows Print, cliquez sur OK.
3. Dans le menu File, cliquez sur Close.
4. Si vous êtes invité à enregistrer des modifications, cliquez sur Yes.

Résumé

Ce didacticiel vous a appris à

- ◆ vous connecter à InfoMaker,
- ◆ générer un état élémentaire,
- ◆ prévisualiser l'état,
- ◆ formater l'état en triant les données, puis en y ajoutant des en-têtes, des titres, des dates, des numéros de page et un total.

Etape suivante

Outre des états, InfoMaker fournit les fonctionnalités permettant de créer des graphiques, des requêtes et des formulaires de saisie de données.

☞ Pour plus d'informations sur InfoMaker, choisissez Démarrer ► Programmes ► Sybase SQL Anywhere 9 ► InfoMaker 8.0 ► Online Help Files. Vous pouvez avoir accès à la documentation complète au sein de l'application en cliquant sur le menu Help.





CINQUIÈME PARTIE

ANNEXE

Cette annexe décrit les systèmes d'exploitation supportés par les composants de SQL Anywhere Studio.

CHAPITRE 20

Glossaire

Adaptive Server Anywhere (ASA)	Composant serveur de la base de données relationnelle de SQL Anywhere Studio, destiné aux environnements mobiles et embarqués et conçu pour être utilisé comme serveur dans les entreprises de petite et moyenne taille.
administrateur de base de données (DBA)	Utilisateur détenant les autorisations requises pour la maintenance de la base de données. Le DBA est généralement responsable de toutes les modifications apportées au schéma d'une base de données et de la gestion des utilisateurs et des groupes d'utilisateurs. Le rôle de l'administrateur de base de données est automatiquement intégré dans les bases avec l'ID utilisateur DBA doté du mot de passe SQL.
administrateur ODBC	Programme Microsoft fourni avec les systèmes d'exploitation Windows pour définir des sources de données ODBC.
agent de réplication	Dans Replication Server, programme, également nommé Gestionnaire de transfert de journal (LTM, Log Transfer Manager), qui lit le journal de transactions d'une base de données et transmet les modifications validées à Replication Server.
alias	Nom d'une table ou d'une vue qui est utilisé dans la clause FROM d'une requête, c'est-à-dire soit le nom d'origine, soit un alias défini dans la clause FROM.
arborescence	Dans Sybase Central, hiérarchie des objets de base de données. La partie supérieure de l'arborescence affiche tous les produits que votre version de Sybase Central supporte. Chaque produit peut être développé pour afficher sa propre sous-arborescence.  Voir aussi : “Sybase Central” à la page 309 .
article	Dans SQL Remote ou MobiLink, objet de base de données qui représente une table complète ou un sous-ensemble de colonnes et de lignes dans une table. Les articles sont regroupés dans une publication.  Voir aussi : “réplication” à la page 307 , “publication” à la page 306 .
autorisation distante	Dans la réplication SQL Remote, autorisation d'échanger des messages de réplication à l'aide de la base éditrice. L'octroi d'autorisations distantes à un utilisateur lui confère le statut d'utilisateur distant. Vous devez spécifier un type de message, une adresse distante appropriée et une fréquence de

réplication. En général, les autorisations distantes peuvent également désigner tout utilisateur impliqué dans une réplication SQL Remote (par exemple, l'éditeur consolidé et l'éditeur distant).

☞ Voir aussi : [“réplication” à la page 307](#).

base de données

Ensemble de tables reliées par des clés primaires et étrangères. Les tables stockent les informations de la base de données. Les tables et les clés définissent ensemble la structure de la base de données. Un système de gestion de base de données accède à ces informations.

☞ Voir aussi : [“clé étrangère” à la page 297](#), [“clé primaire” à la page 297](#), [“système de gestion de base de données \(SGBD\)” à la page 309](#), [“système de gestion de base de données relationnelle ou SGBDR” à la page 309](#).

base de données consolidée

En réplication de base de données, base de données qui stocke la copie maîtresse des données. La base de données consolidée contient toutes les données, alors que les bases de données distantes ne contiennent généralement que des sous-ensembles de données. En cas de conflit ou de contradiction, la base de données consolidée est celle qui est censée détenir la copie principale de toutes les données.

Dans MobiLink, la base de données consolidée peut être Oracle, IBM DB2, Microsoft SQL Server, Adaptive Server Anywhere ou Adaptive Server Enterprise.

☞ Voir aussi : [“réplication” à la page 307](#).

base de données de référence

Dans MobiLink, base de données Adaptive Server Anywhere utilisée pour le développement de clients UltraLite. Vous pouvez utiliser une seule base de données Adaptive Server Anywhere à la fois comme base de référence et comme base consolidée pendant le développement. Les bases de données créées avec d'autres logiciels ne peuvent pas être utilisées comme bases de référence.

base de données distante

Dans la réplication SQL Remote ou la synchronisation MobiLink, base de données qui échange des données avec une base consolidée. Les bases de données distantes peuvent partager des données ou toutes les données de la base consolidée.

☞ Voir aussi : [“réplication” à la page 307](#), [“base de données consolidée” à la page 296](#).

classe Java

Principale unité structurelle du code dans Java. Elle se définit comme une collection de procédures et de variables groupées ensemble car elles se rapportent toutes à une catégorie précise identifiable.

classement

Combinaison d'un jeu de caractères et d'un ordre de tri qui définit les propriétés du texte dans la base de données. Pour les bases de données

	<p>Adaptive Server Anywhere, le classement par défaut est déterminé par le système d'exploitation et par la langue dans laquelle le serveur s'exécute ; par exemple, le classement par défaut sous les systèmes Windows en anglais est 1252LATIN1. Un classement, également appelé séquence de tri, permet de comparer et de trier des chaînes.</p>
clé étrangère	<p>Une ou plusieurs colonne(s) dans une table qui duplique(nt) les valeurs de clé étrangère dans une autre table. Les clés étrangères établissent les relations entre les tables.</p> <p>☞ Voir aussi : “clé primaire” à la page 297, “table étrangère” à la page 309.</p>
clé primaire	<p>Colonne ou liste de colonnes dont les valeurs identifient de façon unique chaque ligne de la table.</p> <p>☞ Voir aussi “clé étrangère” à la page 297 :</p>
client/serveur	<p>Architecture logicielle dans laquelle une application (le client) obtient des informations de la part d'une autre application (le serveur) et lui en envoie. Les deux applications résident souvent sur des ordinateurs différents connectés par un réseau.</p>
client MobiLink	<p>Il existe deux types de clients MobiLink. Pour les bases de données distantes Adaptive Server Anywhere, le client MobiLink est l'utilitaire dbmlsync. Pour les bases de données distantes UltraLite, le client MobiLink est intégré à la bibliothèque d'exécution UltraLite.</p>
concurrence	<p>Exécution simultanée de plusieurs processus indépendants, éventuellement concurrents. Adaptive Server Anywhere utilise automatiquement le verrouillage pour isoler les transactions et garantir que chaque application concurrente traite un ensemble de données cohérent.</p> <p>☞ Voir aussi : “transaction” à la page 310, “verrou” à la page 312, “niveau d'isolement” à la page 305.</p>
condition de jointure générée	<p>Restriction sur les résultats de la jointure définie par le mot-clé KEY ou NATURAL. Dans le cas d'une jointure naturelle, la condition de jointure générée est basée sur le nom commun des colonnes dans les deux tables. Dans le cas d'une jointure par clé, elle est basée sur une relation de clé étrangère entre les deux tables.</p> <p>☞ Voir aussi : “jointure” à la page 303, “condition de jointure” à la page 297.</p>
condition de jointure	<p>Restriction qui affecte les résultats de la jointure. Pour spécifier une condition de jointure, insérez une clause ON ou une clause WHERE immédiatement après la jointure. Dans le cas des jointures naturelles et des jointures par clé, Adaptive Server Anywhere génère une condition de jointure.</p>

	<p>☞ Voir aussi : “jointure” à la page 303, “condition de jointure générée” à la page 297.</p>
conflit	Accès conflictuel aux mêmes ressources. Par exemple, dans le contexte d’une base de données, deux utilisateurs qui tentent de modifier la même ligne d’une base de données se disputent les droits pour modifier cette ligne.
connexion à une base de données	Canal de communication entre une application cliente et la base de données. Pour établir une connexion, un ID utilisateur correct et un mot de passe sont requis. Les privilèges octroyés à un ID utilisateur déterminent les actions qui peuvent être exécutées pendant la connexion.
contrainte de clé étrangère	<p>Restriction sur une colonne ou un ensemble de colonnes qui spécifie les relations entre les données d’une table et celles d’une autre table.</p> <p>Lorsqu’une contrainte de clé étrangère est imposée sur un ensemble de colonnes, celles-ci deviennent la clé étrangère.</p> <p>☞ Voir aussi : “contrainte” à la page 298, “contrainte de vérification” à la page 298, “contrainte de clé primaire” à la page 298, “contrainte d’unicité” à la page 298.</p>
contrainte de clé primaire	<p>Contrainte d’unicité sur les colonnes de clé primaire. Une table ne peut avoir qu’une seule contrainte de clé primaire.</p> <p>☞ Voir aussi : “contrainte” à la page 298, “contrainte de vérification” à la page 298, “contrainte de clé étrangère” à la page 298, “contrainte d’unicité” à la page 298, “intégrité” à la page 302.</p>
contrainte de vérification	<p>Restriction qui impose des conditions précises pour une colonne ou un ensemble de colonnes.</p> <p>☞ Voir aussi : “contrainte” à la page 298, “contrainte de clé étrangère” à la page 298, “contrainte de clé primaire” à la page 298, “contrainte d’unicité” à la page 298.</p>
contrainte d’unicité	<p>Restriction sur une colonne ou un ensemble de colonnes qui exige que toutes les valeurs non NULL soient différentes. Une table peut comporter plusieurs contraintes d’unicité.</p> <p>☞ Voir aussi : “contrainte de clé étrangère” à la page 298, “contrainte de clé primaire” à la page 298, “contrainte” à la page 298.</p>
contrainte	Restriction sur les valeurs contenues dans un objet de base de données particulier, tel qu’une table ou une colonne. Par exemple, une colonne peut avoir une contrainte d’unicité qui exige que toutes les valeurs de la colonne soient différentes. Une table peut avoir une contrainte de clé étrangère, qui spécifie les relations entre ses informations et celles contenues dans une autre table.

	<p>☞ Voir aussi : “contrainte de vérification” à la page 298, “contrainte de clé étrangère” à la page 298, “contrainte de clé primaire” à la page 298, “contrainte d’unicité” à la page 298.</p>
curseur	<p>Liaison nommée avec un jeu de résultats, utilisée pour accéder aux lignes et les mettre à jour à partir d’une interface de programmation. Dans Adaptive Server Anywhere, les curseurs permettent les déplacements vers l’avant et vers l’arrière dans les résultats d’une requête. Les curseurs se composent de deux parties : le jeu de résultats du curseur, généralement défini par une instruction SELECT, et la position du curseur.</p> <p>☞ Voir aussi : “jeu de résultats du curseur” à la page 303, “position du curseur” à la page 306.</p>
dbspace	<p>Fichier de base de données supplémentaire qui crée davantage d’espace pour les données. Une base de données peut être contenue dans un maximum de 13 fichiers distincts (un fichier initial et 12 dbspaces). Mais chaque table, index compris, doit être contenue dans un seul et même fichier de base de données. La commande SQL CREATE DBSPACE ajoute un nouveau fichier à la base de données.</p> <p>☞ Voir aussi : “fichier de base de données” à la page 300.</p>
télécharger	<p>Le téléchargement d’une base de données exporte la structure et/ou les données d’une base vers des fichiers texte (fichiers de commandes SQL pour la structure et fichiers ASCII avec la virgule comme délimiteur pour les données). Vous téléchargez une base de données avec l’utilitaire de ligne de commande dbunload.</p> <p>Par ailleurs, vous pouvez télécharger des parties sélectionnées de vos données à l’aide de l’instruction UNLOAD.</p>
DLL (data definition language, langage de définition des données)	<p>Sous-ensemble d’instructions SQL permettant d’assurer la modélisation de la structure d’une base de données. Les instructions DDL créent, modifient et suppriment les objets de base de données, y compris les utilisateurs.</p>
DML (data manipulation language, langage de manipulation des données)	<p>Sous-ensemble d’instructions SQL permettant d’assurer l’extraction et la mise à jour du contenu d’une base de données.</p> <p>Niveau d’autorisation qui permet à un utilisateur d’effectuer des tâches d’administration dans la base de données. L’utilisateur DBA dispose par défaut des droits DBA.</p> <p>☞ Voir aussi : “administrateur de base de données (DBA)” à la page 295.</p>
droits DBA distants	<p>Dans SQL Remote, niveau d’autorisation requis par l’agent de message. Dans MobiLink, niveau d’autorisation requis par le client de synchronisation Adaptive Server Anywhere (<i>dbmlsync</i>). Lorsque l’agent de message ou le client de synchronisation se connecte comme un utilisateur détenant ces</p>

	<p>droits, il dispose d'un accès DBA total. L'ID utilisateur n'a pas d'autorisation supplémentaire lorsque la connexion n'est pas établie depuis l'agent de message ou le client de synchronisation.</p> <p>☞ Voir aussi : “droits DBA” à la page 299.</p>
éditeur	<p>Dans la réplication SQL Remote, utilisateur unique d'une base de données qui peut échanger des messages de réplication avec d'autres bases de données en cours de réplication.</p> <p>☞ Voir aussi : “réplication” à la page 307.</p>
Embedded SQL	<p>Interface de programmation pour les applications en langage C. Embedded SQL d'Adaptive Server Anywhere est une mise en oeuvre des normes ANSI et IBM.</p>
émulation ou passthrough	<p>Dans la réplication SQL Remote, mode par lequel l'éditeur de la base de données consolidée peut directement modifier les bases distantes avec des instructions SQL. Le mode d'émulation est configuré pour des bases de données distantes spécifiques. En mode d'émulation normal, toutes les modifications effectuées dans la base de données consolidée sont transmises aux bases distantes sélectionnées. En mode “émulation uniquement”, les modifications sont apportées aux bases de données distantes, mais n'ont pas lieu dans la base consolidée.</p>
extraction	<p>Dans la réplication SQL Remote, opération consistant à télécharger la structure et les données appropriées de la base de données consolidée. Ces informations sont utilisées pour initialiser la base de données distante.</p> <p>Dans la synchronisation MobiLink, opération consistant à télécharger la structure et les données appropriées d'une base de données de référence.</p> <p>☞ Voir aussi : “réplication” à la page 307.</p>
fichier de base de données	<p>Une base de données est contenue dans un ou plusieurs fichiers de base de données. Il y a un fichier initial et les fichiers suivants se nomment dbspaces. Chaque table, notamment ses index, doit être contenue dans un seul fichier de base de données.</p> <p>☞ Voir aussi : “dbspace” à la page 299.</p>
fichier de base de données compacté	<p>Fichier de base de données qui a été réduit à une taille physique inférieure à l'aide de l'utilitaire dbshrink. Les bases de données compactées sont en lecture seule. Pour modifier un fichier de base de données compacté, vous devez utiliser un fichier d'écriture associé. Vous pouvez décompacter les fichiers de base de données compactés en fichiers de base de données normaux à l'aide de l'utilitaire dbexpand.</p>
fichier de commandes	<p>Fichier texte contenant des instructions SQL. Les fichiers de commandes</p>

	peuvent être créés manuellement ou automatiquement par des utilitaires de base de données. L'utilitaire dbunload, par exemple, crée un fichier de commandes constitué des instructions SQL requises pour reconstruire une base de données particulière.
fichier d'écriture	Fichier utilisé pour enregistrer les modifications apportées à une base de données en lecture seule. Souvent utilisé dans le cas de bases de données compactées. ☞ Voir aussi : “fichier de base de données compacté” à la page 300.
fichier JAR	Fichier d'archive Java. Format de fichier compacté constitué d'un package ou d'un ensemble de packages utilisé pour les applications Java. Il inclut toutes les ressources requises pour installer et exécuter un programme Java dans un fichier compacté unique.
fichier journal	Journal de transactions conservé par Adaptive Server Anywhere. Le fichier journal permet de garantir que la base de données est récupérable en cas de panne de disque ou de panne système, d'améliorer les performances de la base de données et il autorise la réplication des données à l'aide de SQL Remote. ☞ Voir aussi : “journal de transactions” à la page 303 , “miroir de journal de transactions” à la page 304 , “sauvegarde intégrale” à la page 307.
FILE	Dans une réplication SQL Remote, système de messagerie qui utilise des fichiers partagés pour échanger des messages de réplication. Cela est particulièrement utile pour les tests et les installations sans système de transfert de message explicite (tel que MAPI). ☞ Voir aussi : “réplication” à la page 307 , “MAPI” à la page 304.
flux de communication	Dans MobiLink, protocole réseau utilisé pour communiquer entre le client MobiLink et le serveur de synchronisation MobiLink.
fréquence de réplication	Dans la réplication SQL Remote, paramètre pour chaque utilisateur distant qui détermine à quelle fréquence l'agent de message de l'éditeur doit envoyer les messages de réplication à cet utilisateur distant. ☞ Voir aussi : “réplication” à la page 307 , “utilisateur distant” à la page 312.
ID de connexion	Nombre unique qui identifie une connexion donnée entre une application cliente et la base de données. Vous pouvez déterminer l'ID de la connexion en cours en utilisant l'instruction SQL suivante : <pre>SELECT connection_property('Number')</pre>
identificateur	Chaîne de caractères utilisée pour référencer un objet de base de données, tel qu'une table ou une colonne. Un identificateur peut contenir tout caractère

	de A à Z, de a à z, de 0 à 9, le caractère de soulignement (<u>_</u>), ainsi que les signes arobas (@), dièse (#) et dollar (\$).
index	Ensemble trié de clés et de pointeurs associé à une ou plusieurs colonnes dans une table sous-jacente. Un index sur une ou plusieurs colonnes dans une table peut améliorer les performances.
InfoMaker	Outil de génération d'états et de maintenance des données qui permet de créer des formulaires élaborés, des états, des graphiques, des tableaux croisés et des tables, ainsi que des applications qui utilisent ces états comme composants.
instruction SQL	Chaîne contenant des mots-clés SQL conçue pour transmettre des instructions à un SGBD. ☞ Voir aussi : “schéma” à la page 307 , “SQL” à la page 308 , “système de gestion de base de données (SGBD)” à la page 309 .
intégrité	Conformité aux règles qui gèrent le contrôle des données proprement dites (elles doivent être correctes et exactes) et celui de la structure relationnelle de la base de données. ☞ Voir aussi : “intégrité référentielle” à la page 302 .
intégrité référentielle	Conformité aux règles qui gèrent la cohérence des données, notamment les relations entre les valeurs de la clé primaire et de la clé étrangère dans des tables différentes. Pour avoir une intégrité référentielle, les valeurs de chaque clé étrangère doivent correspondre aux valeurs de clé primaire d'une ligne de la table référencée. ☞ Voir aussi : “clé primaire” à la page 297 , “clé étrangère” à la page 297 .
intégrité transactionnelle	Dans MobiLink, conservation des transactions garantie à travers le système de synchronisation. Soit une transaction est synchronisée intégralement, soit aucune partie de la transaction n'est synchronisée.
Interactive SQL	Application Adaptive Server Anywhere qui vous permet d'interroger et de modifier les données de votre base de données ou de modifier la structure de cette base. Interactive SQL comporte un volet de saisie des instructions SQL, ainsi que des volets affichant des informations sur le traitement de la requête et le jeu de résultats.
jConnect	Mise en oeuvre Java de la norme JavaSoft JDBC. Elle fournit aux développeurs Java qui créent des bases de données natives un accès aux environnements multiniveaux et multiplate-formes. ☞ Voir aussi : “JDBC” à la page 302 .
JDBC	Java Database Connectivity ou connectivité Java. Interface de programmation en langage SQL qui permet aux applications Java d'accéder

	aux données relationnelles.
jeu de résultats du curseur	<p>Ensemble de lignes résultant de l'exécution d'une requête associée au curseur.</p> <p>☞ Voir aussi : "curseur" à la page 299, "position du curseur" à la page 306.</p>
jointure	Opération élémentaire dans un système relationnel qui lie les lignes de deux ou plusieurs tables en comparant les valeurs des colonnes spécifiées.
jointure externe	<p>Jointure qui préserve toutes les lignes dans une table. Adaptive Server Anywhere supporte les jointures externes gauches, droites et complètes. Une jointure externe gauche préserve les lignes de la table à gauche de l'opérateur de jointure et renvoie NULL lorsqu'une ligne de la table de droite ne remplit pas la condition de jointure. Une jointure externe complète préserve toutes les lignes des deux tables.</p> <p>☞ Voir aussi : "jointure" à la page 303, "jointure interne" à la page 303.</p>
jointure interne	<p>Jointure dans laquelle les lignes n'apparaissent dans le jeu de résultats que si les deux tables satisfont à la condition de jointure. Les jointures internes sont les jointures par défaut.</p> <p>☞ Voir aussi : "jointure" à la page 303, "jointure externe" à la page 303.</p>
journal de transactions	<p>Fichier stockant toutes les modifications apportées à une base de données, dans l'ordre de leur exécution. Il augmente les performances et permet de restaurer les données lorsque le fichier de base de données est endommagé. Pour obtenir de meilleurs résultats, le journal de transactions doit résider sur un device distinct de celui des fichiers de base de données.</p>
journal des annulations	<p>Enregistrement des modifications effectuées pendant chaque transaction non validée. Dans le cas d'une requête ROLLBACK ou d'une défaillance du système, les transactions non validées sont retirées de la base de données, qui retrouve son état précédent. Chaque transaction a un journal d'annulations distinct, qui est supprimé lorsqu'elle est achevée.</p> <p>☞ Voir aussi : "transaction" à la page 310.</p>
login externe	<p>Nom de login et mot de passe de remplacement utilisés pour communiquer avec un serveur distant. Par défaut, Adaptive Server Anywhere utilise les noms et mots de passe de ses clients chaque fois qu'il se connecte à un serveur distant au nom de ces clients. Mais il est possible de créer des logins externes, qui priment alors sur ces logins par défaut. Les logins externes sont des noms et des mots de passe alternatifs utilisables pour communiquer avec un serveur distant.</p>
login intégré	<p>Fonctionnalité de login qui permet de ne gérer qu'un seul et même ID utilisateur, avec son mot de passe, pour accéder aux bases de données, au</p>

	réseau et au système d'exploitation.
LTM	Log Transfer Manager ou gestionnaire de transfert de journal. Voir aussi “agent de réplication” à la page 295 :
MAPI	Interface de programmation d'applications de messagerie de Microsoft. Système de messagerie utilisé dans plusieurs systèmes de courrier électronique répandus tels que Microsoft Mail.
message de réplication	Dans SQL Remote ou Replication Server, communication transmise entre une base de données éditrice et une base abonnée. Les messages contiennent des données, des instructions passthrough et les informations requises par le système de réplication. ☞ Voir aussi : “émulation ou passthrough” à la page 300 , “réplication” à la page 307 , “mise à jour de publication” à la page 304 .
métadonnées	Données sur les données. Les métadonnées décrivent la nature et le contenu d'autres données. ☞ Voir aussi : “schéma” à la page 307 .
miroir de journal de transactions	Copie conforme facultative du journal de transactions, les deux étant mis à jour simultanément. Chaque modification apportée à la base de données est consignée à la fois dans le journal de transactions et dans son “miroir”. Le journal en miroir doit résider sur un device distinct de celui du journal de transactions, de sorte que si l'un des devices tombe en panne, l'autre copie du journal de transactions contient les données pour la reprise. ☞ Voir aussi : “journal de transactions” à la page 303 .
mise à jour de publication	Dans la réplication SQL Remote, liste des modifications apportées à une ou plusieurs publications dans une base de données. Une mise à jour de publication est régulièrement envoyée dans le cadre d'un message de réplication à une ou plusieurs base(s) de données distant(e)s. ☞ Voir aussi : “réplication” à la page 307 , “publication” à la page 306 .
MobiLink	Technologie de synchronisation basée sur les sessions et conçue pour synchroniser UltraLite et les bases de données Adaptive Server Anywhere avec un grand nombre de systèmes de gestion de base de données SQL, conformes aux normes de l'industrie et développés par Sybase et d'autres fournisseurs. ☞ Voir aussi : “UltraLite” à la page 312 .
module de connexion	Dans Sybase Central, façon d'accéder à un produit et de l'administrer. En règle générale, les modules de connexion sont installés et enregistrés automatiquement avec Sybase Central lorsque vous installez un produit. Un

	<p>module de connexion apparaît généralement comme un dossier de niveau supérieur dans la fenêtre principale de Sybase Central, avec le nom du produit lui-même, par exemple, Adaptive Server Anywhere.</p> <p>☞ Voir aussi : “Sybase Central” à la page 309.</p>
NetWare	<p>Système d’exploitation en réseau couramment utilisé défini par Novell. NetWare emploie généralement le protocole IPX/SPX, mais le protocole TCP/IP peut également être utilisé.</p>
niveau d’isolement	<p>Degré de visibilité des opérations dans une seule transaction pour les opérations dans les transactions concurrentes. Il existe quatre niveaux d’isolement, numérotés de 0 à 3. Le niveau 3 constitue le plus haut niveau d’isolement. Le niveau 0 est le paramètre par défaut.</p>
nom de base de données	<p>Nom attribué à une base de données lorsqu’elle est chargée par un serveur. Le nom de base de données par défaut est la racine du fichier de base de données initial.</p> <p>☞ Voir aussi : “fichier de base de données” à la page 300.</p>
nom de rôle	<p>Nom d’une clé étrangère. On parle de nom de rôle car il désigne la relation entre la table étrangère et la table primaire. Par défaut, le nom de rôle est le nom de la table, à moins qu’une autre clé étrangère n’emploie déjà ce nom, auquel cas le nom de rôle par défaut est le nom de la table suivi d’un nombre à trois chiffres unique. Vous pouvez également créer le nom vous-même.</p> <p>☞ Voir aussi : “clé étrangère” à la page 297.</p>
normalisation	<p>Réajustement de la structure d’une base de données pour éliminer la redondance et améliorer l’organisation conformément aux règles issues de la théorie des bases de données relationnelles.</p>
objet de base de données	<p>Composant d’une base de données qui contient ou reçoit des informations. Les tables, les index, les vues, les procédures et les triggers sont des objets de base de données.</p>
objet système	<p>Objet de base de données appartenant à SYS ou à dbo.</p>
ODBC	<p>Open Database Connectivity. Interface standard de Windows pour les systèmes de gestion de base de données. ODBC est l’une des nombreuses interfaces supportées par Adaptive Server Anywhere.</p>
option GRANT	<p>Niveau d’autorisation qui permet à un utilisateur d’octroyer des autorisations aux autres utilisateurs.</p>
package	<p>Dans Java, ensemble de groupes de classes reliées entre elles.</p>
point de reprise	<p>Stade auquel toutes les modifications apportées à la base de données sont enregistrées dans le fichier de base de données. Le reste du temps, les</p>

	modifications validées sont uniquement enregistrées dans le journal de transactions.
position du curseur	<p>Pointeur sur une ligne à l'intérieur du jeu de résultats du curseur.</p> <p>☞ Voir aussi : “curseur” à la page 299, “jeu de résultats du curseur” à la page 303.</p>
PowerAMC	Application de modélisation de base de données. PowerAMC fournit une approche structurée à la conception d'une base de données ou d'un entrepôt de données.
PowerDynamo	Produit Sybase conçu pour élaborer et gérer une application Web liée à une base de données.
PowerJ	Produit Sybase pour le développement d'applications Java.
prédicat	Expression conditionnelle qui, associée aux opérateurs logiques AND et OR (facultatif), constitue l'ensemble des conditions d'une clause WHERE ou HAVING. Dans SQL, tout prédicat dont la valeur est UNKNOWN est interprété comme ayant la valeur FALSE.
procédure stockée	Programme constitué d'une séquence d'instructions SQL, stocké dans la base de données et conçu pour effectuer une tâche particulière.
profil de connexion	Ensemble de paramètres requis pour se connecter à une base de données, tels que le nom d'utilisateur, le mot de passe et le nom de serveur, qui est stocké et utilisé à loisir.
propriétaire de base de données (dbo, database owner)	<p>Utilisateur spécial possédant les objets système qui n'appartiennent pas à SYS.</p> <p>☞ Voir aussi : “administrateur de base de données (DBA)” à la page 295, “SYS” à la page 309.</p>
publication	<p>Dans SQL Remote ou MobiLink, objet de base de données qui identifie les données répliquées. Dans MobiLink, les publications n'existent que sur les clients. Une publication se compose d'articles. Les modifications apportées à chaque publication sont régulièrement répliquées vers tous les souscripteurs à cette publication. Les utilisateurs de SQL Remote peuvent recevoir une publication en s'y abonnant. Les utilisateurs MobiLink peuvent synchroniser une publication en créant pour elle une souscription de synchronisation.</p> <p>☞ Voir aussi : “réplication” à la page 307, “article” à la page 295, “mise à jour de publication” à la page 304.</p>
redirecteur	Module de connexion de serveur Web qui achemine les requêtes et les réponses entre un client et le serveur de synchronisation MobiLink. Ce module de connexion met également en oeuvre les mécanismes de répartition de la charge et de reprise sur le serveur secondaire.

règle de gestion	<p>Règle fondée sur les exigences du monde réel. Les règles de gestion sont généralement appliquées par l'intermédiaire des contraintes de vérification, des types de données définis par l'utilisateur et l'utilisation adéquate des transactions.</p> <p>☞ Voir aussi : “contrainte” à la page 298, “type de données défini par l'utilisateur” à la page 311.</p>
Replication Server	<p>Technologie de réplication basée sur les connexions Sybase qui fonctionne avec Adaptive Server Anywhere et Adaptive Server Enterprise. Elle est conçue pour une réplication presque en temps réel entre un petit nombre de bases de données.</p>
reprise	<p>Basculement vers un serveur répliqué ou de sauvegarde, un système ou un réseau à la suite d'une panne ou d'un arrêt inattendu d'un serveur, système ou réseau actifs. Les reprises se produisent automatiquement et sont souvent partie intégrante de systèmes à disponibilité continue.</p>
requête	<p>Instruction SQL ou groupe d'instructions SQL qui permettent d'accéder aux données d'une base et/ou de les manipuler.</p> <p>☞ Voir aussi : “SQL” à la page 308.</p>
réplication	<p>Partage des données entre des bases de données physiquement distinctes. Sybase intègre trois technologies de réplication : MobiLink, SQL Remote et Replication Server.</p>
rôle	<p>Dans un modèle conceptuel de données, verbe ou phrase décrivant une relation d'un point of vue unique. Vous pouvez décrire chaque relation par le biais de deux rôles. Parmi les exemples de rôle, on peut citer “contient” et “appartient à”.</p>
sauvegarde incrémentielle	<p>Sauvegarde du journal de transactions uniquement, généralement effectuée entre les sauvegardes intégrales.</p> <p>☞ Voir aussi : “journal de transactions” à la page 303.</p>
sauvegarde intégrale	<p>Sauvegarde de l'ensemble de la base de données et, de manière facultative, du journal de transactions. Une sauvegarde intégrale contient toutes les informations de la base de données et offre ainsi une protection en cas de panne système ou de disque.</p> <p>☞ Voir aussi : “sauvegarde incrémentielle” à la page 307.</p>
schéma	<p>Structure d'une base de données comprenant les tables, les colonnes et les index, ainsi que les relations qui les unissent.</p>
script	<p>Dans MobiLink, code écrit pour gérer les événements MobiLink. Les scripts contrôlent l'échange des données pour répondre aux besoins du marché.</p>

serveur de base de données	Programme informatique qui régule tout accès à l'information dans une base de données. Adaptive Server Anywhere fournit deux types de serveur : les serveurs réseau et les serveurs personnels.
serveur personnel	Serveur de base de données qui s'exécute sur le même ordinateur que l'application cliente. Un serveur de base de données personnel est généralement utilisé par un seul utilisateur sur un seul ordinateur, mais il peut supporter plusieurs connexions concurrentes de cet utilisateur.
serveur réseau	<p>Serveur de base de données qui accepte les connexions d'ordinateurs partageant un réseau commun.</p> <p>☞ Voir aussi : “serveur personnel” à la page 308.</p>
service	Dans les systèmes d'exploitation Windows, mode d'exécution des applications lorsque l'ID utilisateur exécutant l'application n'est pas connecté.
source de données ODBC	Spécification des données auxquelles l'utilisateur veut accéder via ODBC et informations requises pour obtenir ces données.
souscription	<p>Dans la réplication SQL Remote, lien entre une publication et un utilisateur distant, permettant à ce dernier d'échanger des mises à jour dans le cadre de cette publication avec la base de données consolidée.</p> <p>Dans la synchronisation MobiLink, une souscription de synchronisation est un lien dans une base de données cliente entre une publication et un utilisateur MobiLink qui permet aux données décrites par la publication d'être synchronisées.</p> <p>☞ Voir aussi : “publication” à la page 306, “utilisateur distant” à la page 312, “utilisateur MobiLink” à la page 312.</p>
sous-requête	<p>Instruction SELECT qui est imbriquée dans une autre instruction SELECT, INSERT, UPDATE ou DELETE, ou dans une autre sous-requête.</p> <p>Il existe deux types de sous-requête : la sous-requête corrélée et la sous-requête imbriquée.</p>
SQL	Langage utilisé pour communiquer avec les bases de données relationnelles. ANSI a défini des normes pour SQL, dont la dernière est SQL-99 (ou SQL3). SQL signifie Structured Query Language.
SQL Remote	Technologie de réplication par messages conçue pour la réplication bidirectionnelle entre la base de données consolidée et les bases distantes. La base de données consolidée doit être de type Adaptive Server Anywhere ou Adaptive Server Enterprise. Les bases de données distantes doivent être de type Adaptive Server Anywhere.
statistique de performances	Valeur reflétant les performances du système de base de données. La

	statistique CURRREAD, par exemple, représente le nombre de lectures de fichier émises par le moteur et non terminées.
Sybase Central	Outil de gestion de base de données à interface graphique qui fournit sous une forme conviviale les utilitaires, les propriétés et les paramètres d'une base de données Adaptive Server Anywhere. Sybase Central peut également être utilisé pour gérer d'autres produits Sybase, notamment MobiLink.
synchronisation	<p>Processus de répllication des informations entre les bases de données utilisant la technologie MobiLink.</p> <p>Dans SQL Remote, la synchronisation désigne exclusivement le processus de création d'une base de données distante avec un ensemble de données initiales.</p> <p>☞ Voir aussi : “MobiLink” à la page 304, “SQL Remote” à la page 308.</p>
synchronisation exécutée par le serveur	Méthode d'exécution de la synchronisation MobiLink à l'aide de scripts depuis la base de données consolidée.
synchronisation orientée session	Type de synchronisation générant une représentation cohérente des données modifiées au niveau des bases de données consolidées et distantes. MobiLink est orienté session.
SYS	Utilisateur spécial qui possède la plupart des objets système. Vous ne pouvez pas vous connecter en tant que SYS.
système de gestion de base de données (SGBD)	<p>Ensemble de programmes qui vous permettent de créer et d'utiliser des bases de données.</p> <p>☞ Voir aussi : “système de gestion de base de données relationnelle ou SGBDR” à la page 309.</p>
système de gestion de base de données relationnelle ou SGBDR	<p>Type de système de gestion de base de données qui stocke les données sous forme de tables reliées entre elles.</p> <p>☞ Voir aussi : “système de gestion de base de données (SGBD)” à la page 309.</p>
système de messagerie	<p>Dans la répllication SQL Remote, protocole conçu pour échanger des messages entre la base de données consolidée et les bases de données distantes. Adaptive Server Anywhere comprend le support pour les systèmes de messagerie suivants : FILE, MAPI, FTP, SMTP et VIM.</p> <p>☞ Voir aussi : “répllication” à la page 307, “FILE” à la page 301, “MAPI” à la page 304.</p>
table de travail	Zone de stockage interne des résultats provisoires pendant l'optimisation des requêtes.
table étrangère	Table contenant la clé étrangère.

	<p>☞ Voir aussi : “clé étrangère” à la page 297.</p>
table primaire	Table contenant la clé primaire dans une relation de clé étrangère.
table proxy	Table locale contenant des métadonnées permettant d’accéder à une table sur un serveur de base de données distant comme s’il s’agissait d’une table locale.
	<p>☞ Voir aussi : “métadonnées” à la page 304.</p>
table sous-jacente	Table permanente pour les données. On parle de table sous-jacente par opposition aux tables temporaires et aux vues.
	<p>☞ Voir aussi : “table temporaire” à la page 310, “vue” à la page 313.</p>
table système	Table, appartenant à SYS ou à dbo, qui contient des métadonnées. Les tables système, également connues sous le nom de référentiel de données, sont créées et gérées par le serveur de base de données.
table temporaire	Table créée pour le stockage temporaire des données. Il existe deux types de table temporaire : les tables temporaires globales et les tables temporaires locales.
	<p>☞ Voir aussi : “table temporaire locale” à la page 310, “table temporaire globale” à la page 310.</p>
table temporaire globale	Type de table temporaire pour laquelle toutes les définitions de données sont visibles par tous les utilisateurs jusqu’à ce qu’elle soit supprimée explicitement. Les tables temporaires globales permettent à chaque utilisateur d’ouvrir sa propre instance identique d’une table. Par défaut, les lignes sont supprimées à l’exécution d’un commit et elles sont toujours supprimées lorsque la connexion est terminée.
	<p>☞ Voir aussi : “table temporaire” à la page 310, “table temporaire locale” à la page 310.</p>
table temporaire locale	Type de table temporaire qui existe seulement pendant la durée d’exécution d’une instruction composée ou jusqu’à la fin de la connexion. Les tables temporaires locales sont utiles pour charger un ensemble de données en une fois. Par défaut, les lignes sont supprimées à l’exécution d’un commit.
	<p>☞ Voir aussi : “table temporaire” à la page 310, “table temporaire globale” à la page 310.</p>
téléchargement	Etape de la synchronisation durant laquelle les données sont transférées de la base de données distante à la base de données consolidée.
transaction	Séquence d’instructions SQL qui inclut une unité de travail logique. Une transaction est traitée dans son intégralité ou pas du tout. Adaptive Server Anywhere supporte le traitement transactionnel, avec des fonctions de

	verrouillage construites pour permettre aux transactions concurrentes d'accéder à la base sans altérer les données. Les transactions se terminent soit par une instruction COMMIT, qui rend les modifications apportées aux données permanentes, soit par une instruction ROLLBACK, qui annule toutes les modifications effectuées pendant la transaction.
transfert	Etape de la synchronisation durant laquelle les données sont transférées de la base de données consolidée à la base de données distante.
transfert orienté fichier	Dans MobiLink, méthode de synchronisation selon laquelle les transferts sont distribués comme des fichiers, ce qui permet la distribution hors ligne des modifications de synchronisation.
trigger	Forme spéciale de procédure stockée automatiquement exécutée lorsqu'un utilisateur effectue une requête qui modifie les données. ☞ Voir aussi : “trigger de niveau ligne” à la page 311 , “trigger de niveau instruction” à la page 311 , “trigger de résolution de conflit” à la page 311 , “intégrité” à la page 302 .
trigger de niveau instruction	Trigger qui s'exécute à la fin de l'instruction de déclenchement. ☞ Voir aussi : “trigger” à la page 311 , “trigger de niveau ligne” à la page 311 .
trigger de niveau ligne	Trigger qui s'exécute une fois pour chaque ligne modifiée. ☞ Voir aussi : “trigger” à la page 311 , “trigger de niveau instruction” à la page 311 .
trigger de résolution de conflit	Dans une réplication SQL Remote, trigger qui se déclenche lorsqu'un conflit de mise à jour est détecté, avant exécution de la mise à jour. Les triggers de résolution de conflit sont déclenchés lorsque les valeurs de la clause VERIFY d'une instruction UPDATE ne correspondent pas aux valeurs en cours dans la base de données. ☞ Voir aussi : “réplication” à la page 307 , “trigger” à la page 311 .
type de données	Format des données, tel que CHAR ou NUMERIC. Selon la norme ANSI SQL, les types de données peuvent également inclure une restriction sur la taille, le jeu de caractères et le classement. ☞ Voir aussi : “type de données défini par l'utilisateur” à la page 311 .
type de données défini par l'utilisateur	Type de données créé par l'utilisateur pour spécifier un type de données de base et, de façon facultative, une valeur par défaut, une condition CHECK et une condition NULL ou NOT NULL. Les types de données définis par l'utilisateur, également nommés domaines, peuvent être appliqués à des colonnes pour renforcer la cohérence des données dans la base.

	<p>☞ Voir aussi : “type de données” à la page 311.</p>
type de jointure	<p>Adaptive Server Anywhere fournit quatre types de jointure : la jointure croisée, la jointure par clé, la jointure naturelle et les jointures utilisant une clause ON.</p> <p>☞ Voir aussi : “jointure” à la page 303.</p>
type de message	<p>Dans la réplication SQL Remote, objet de base de données qui spécifie le mode de communication des utilisateurs distants avec l’éditeur d’une base de données consolidée. Une base de données consolidée peut être associée à plusieurs types de message ; ainsi, plusieurs utilisateurs distants peuvent communiquer avec elle via des systèmes de messagerie différents.</p> <p>☞ Voir aussi : “réplication” à la page 307, “base de données consolidée” à la page 296, “MAPI” à la page 304.</p>
UltraLite	<p>Technologie de déploiement des bases de données Adaptive Server Anywhere, conçue pour les périphériques mobiles et embarqués de petite taille. Les plates-formes concernées sont des téléphones portables, des récepteurs de radiomessagerie et des organisateurs personnels.</p>
utilisateur distant	<p>Dans la réplication SQL Remote, utilisateur de base de données dans la base consolidée auquel ont été octroyées des autorisations distantes et associé à une base distante particulière dans la configuration de réplication. Pour créer un utilisateur distant, il faut octroyer des autorisations distantes à un utilisateur de base de données ordinaire. Cela n’identifie pas seulement l’existence d’une base de données distante particulière, mais spécifie également le type de message et l’adresse avec lesquels communiquer avec le site distant concerné.</p> <p>Lorsque des bases de données distantes sont créées via l’extraction d’une base consolidée, chaque utilisateur distant dans la base consolidée devient l’éditeur des données dans une base distante particulière.</p> <p>☞ Voir aussi : “SQL Remote” à la page 308, “base de données consolidée” à la page 296, “éditeur” à la page 300.</p>
utilisateur MobiLink	<p>Un utilisateur MobiLink est un nom qui identifie une base de données distante MobiLink de manière unique dans le système de synchronisation. Le client fournit ce nom et, de manière facultative, un mot de passe associé lorsqu’il se connecte au serveur de synchronisation MobiLink. Les noms d’utilisateur MobiLink sont totalement indépendants des noms d’utilisateur de base de données.</p>
valider	<p>Tester des types de corruption de fichier particuliers dans une base de données, une table ou un index.</p>
verrou	<p>Mécanisme de contrôle de concurrence qui protège l’intégrité des données</p>

durant l'exécution simultanée de différentes transactions. Adaptive Server Anywhere applique automatiquement des verrous pour éviter que deux connexions ne modifient les mêmes données en même temps et que d'autres connexions ne lisent des données qui sont en cours de modification.

La gestion du verrouillage consiste à définir le niveau d'isolement.

☞ Voir aussi : [“niveau d'isolement” à la page 305](#), [“concurrency” à la page 297](#), [“intégrité” à la page 302](#).

vue	Instruction SELECT qui est conservée dans la base de données en tant qu'objet. Elle permet aux utilisateurs de voir un sous-ensemble de lignes ou de colonnes dans une ou plusieurs tables. Chaque fois qu'un utilisateur emploie la vue d'une table particulière ou d'une combinaison de tables, cette vue est recalculée à partir des informations stockées dans ces tables. Les vues s'avèrent utiles en matière de sécurité et pour personnaliser la présentation des informations de la base afin d'en simplifier l'accès.
vue système	Type de vue, inclus dans chaque base de données, qui présente les informations contenues dans les tables système dans un format simple et convivial.
Windows	Famille des systèmes d'exploitation Microsoft Windows, comprenant Windows 95, Windows 98, Windows Me, Windows CE, Windows NT, Windows 2000, et Windows XP.
Windows CE	Famille de systèmes d'exploitation conçue par Microsoft pour les périphériques mobiles.



Index

A

accessibilité		
activation	5	
ActiveSync		
version supportée	52	
Adaptive Server Anywhere		
applications	15	
comparaison avec UltraLite	18, 19	
conditions système	13	
définition	295	
éléments internes	81	
généralités	18	
interface de programmation	76	
objectifs de conception	18	
présentation	10	
spécificités	12	
utilisation	11	
Adaptive Server Anywhere et UltraLite, présentation	9	
administrateur de base de données		
définition	295	
administrateur ODBC		
définition	295	
ADO		
outils de développement	80	
agent de message		
plates-formes UNIX supportées	48	
plates-formes Windows supportées	43	
agent de réplication		
plates-formes supportées	43	
systèmes d'exploitation supportés	48	
aide en ligne		
utilisation	xiii	
aide sur l'aide		
accès aux informations	xiii	
ajout		
ligne	135	
nouvelles lignes dans Interactive SQL	179	
ajout d'un script à une table synchronisée, assistant		
utilisation	241	
ajout d'une version, assistant		
utilisation	240	
alias		
définition	295	
pour les colonnes	92	
ALTER, instruction		
commit automatique	138	
amélioration		
états avec InfoMaker	287	
annulation		
transaction	138	
annulation d'une commande Interactive SQL	183	
API		
Adaptive Server Anywhere	76	
ADO	80	
Embedded SQL	78	
JDBC	80	
ODBC	78	
OLE DB	80	
Open Client	79	
application autonome		
définition	70	
arborescence		
définition	295	
architecture à trois niveaux		
présentation	73	
architecture multiniveau		
présentation	73	
architecture n-niveaux		
présentation	73	
architecture physique	265	
ARM, mode V4T		
Windows CE	54	
ARM, processeur		
Windows CE	54	
ARMV4T		
support plate-forme	57	
arrêt		
serveur de base de données	155	
article		
définition	295	
ASA		
définition	295	
asademo.db, fichier		

généralités	148	rétro-ingénierie	268
assistant		sauvegarde	210
ajout d'une version	240	serveur	67
ajout de script de table synchronisée	241	SQL	86
création d'une base de données	216	tables système	87
création d'utilisateur	205	visualisation du schéma	198
création d'utilisateur MobiLink	239	base de données consolidée	
création de clé étrangère	228	configuration	251
création de groupe	205	définition	296
création de publication	239, 253	SGBDR supporté	50
sauvegarde de base de données	210	base de données de référence	
Sybase Central	210	définition	296
Assistant de paramétrage d'index		base de données distante	
Interactive SQL	172	configuration pour SQL Remote (didacticiel)	255
attribut		définition	296
tables	63	base de données embarquée	
AUTO_COMMIT, option		conditions requises	11
regroupement des modifications dans Interactive		définition	70
SQL	138	base de données exemple	
autorisation		connexion à asademo.db	170
définition	207	démarrage d'asademo.db	151
PUBLISH	251	généralités	148
REMOTE	251	base de données multi-utilisateur	
autorisation distante		définition	72
définition	295	base de données relationnelle	
autorisation, publication		concepts	62
autorisation REMOTE	252	généralités	63
		terminologie	63
B		base de données, configuration hiérarchique	23
barre d'outils		base de données, définition	
Interactive SQL	171	PowerAMC	265
base de données		base de données, plusieurs	
administration	193	exécution sur un serveur unique	74
application cliente	67	base de données, schéma	
composants	67	visualisation	198
concept de création	215	base de données, serveur	
conception	213	arrêt	155
connexion	170, 195	BETWEEN, condition	
création	216	clause WHERE	104
définition	296	binary large object (BLOB)	
fichier	82	généralités	224
gestion	193	bitmaps	
interface de langage	67	objet BLOB	224
objet	65	BLOB	
propriétés des objets	200	généralités	224
relationnelle	62	boîte de dialogue	
requêtes	86	historique de commandes	187

C

caractère joker			
masque de correspondance	101		
caractéristiques			
Replication Server	33		
SQL Remote	33		
synchronisation MobiLink	32		
cardinalité			
relation	227		
CE			
processeurs supportés	54		
versions supportées	54		
Certicom			
commande du logiciel de cryptage	5		
classe Java			
définition	296		
classement			
définition	296		
clause ON			
présentation	110		
clé			
étrangère	63		
généralités	63		
primaire	63		
clé étrangère			
définition	63, 297		
généralités	64		
insertion	141		
clé primaire	63		
création	203		
définition	297		
client MobiLink			
définition	297		
client/serveur			
définition	72, 297		
colonne			
acceptant la valeur NULL	225		
alias	92		
calculée	92		
clé primaire	203		
consultation dans Interactive SQL	186		
généralités	62		
sélection dans une table	92		
tri	92		
colonne calculée			
ajout à de nouvelles lignes sous Interactive SQL			
mise à jour dans Interactive SQL	179		
recalculée dans Interactive SQL	179		
combinaison			
instructions multiples dans Interactive SQL	184		
Command Sequence, protocole de communication			
diagramme	76		
généralités	77		
commande			
annulation dans Interactive SQL	183		
arrêt dans Interactive SQL	183		
chargement dans Interactive SQL	90		
consignation dans Interactive SQL	189		
enregistrement dans Interactive SQL	90		
exécution dans Interactive SQL	90, 183		
interruption dans Interactive SQL	183		
modification dans Interactive SQL	187		
rappel dans Interactive SQL	187		
récupération dans Interactive SQL	90		
COMMENT, instruction			
commit automatique	138		
commit automatique			
instruction ALTER	138		
instruction COMMENT	138		
instruction DROP	138		
instructions de définition de données	138		
COMMIT, instruction			
généralités	139		
transactions	138		
comparaison			
généralités	99		
présentation	100		
utilisation des sous-requêtes	121		
composant			
disponibilité	15		
conception			
base de données	213, 215		
utilisation de PowerAMC	265		
conception et création d'une base de données	213		
concurrence			
définition	297		
condition			
filtrage	101		
recherche	99, 104		
condition de jointure			
définition	297		
condition de jointure générée			
définition	297		

condition de recherche		définition	298
clause GROUP BY	130	contrainte unique	
comparaison des dates	100	définition	298
filtrage	101	contrôle	
présentation	99	intégrité des données	141
raccourci	104	contrôle des performances	
sous-requête	118	spécificité d' Adaptive Server Anywhere	12
condition de recherche composée		convention	
utilisation	103	documentation	xvi
condition LIKE		copie	
présentation	101	ligne dans Interactive SQL	181
condition système		couche de transport, sécurité	
Adaptive Server Anywhere	13	commande	5
configuration		COUNT, fonction	
Adaptive Server Anywhere	13	application aux données groupées	128
base de données consolidée	251	création	
base de données distante pour SQL Remote		base de données	216
(didacticiel)	255	bases de données distante pour SQL Remote	255
configuration matérielle		états avec InfoMaker	281
spécificité d' Adaptive Server Anywhere	12	formulaires avec InfoMaker	281
configuration requise en matière de logiciel réseau	13	groupes	205
configuration système		profil de base de données InfoMaker	283
Adaptive Server Anywhere	40	source de données ODBC	159
conflit		tables	203
définition	298	utilisateurs	205
connexion		création d' utilisateur, assistant	
à partir de Sybase Central	195	utilisation	205
base de données exemple asademo.db	170	création d' utilisateurs MobiLink, assistant	
présentation	158	utilisation	239
connexion à une base de données		création de base de données, assistant	
définition	298	utilisation	216
connexion de votre application à sa base de données	157	création de clé étrangère, assistant	
consignation		utilisation	228
commandes dans Interactive SQL	189	création de groupe, assistant	
consultation		utilisation	205
colonnes dans Interactive SQL	186	création de publication, assistant	
procédures dans Interactive SQL	186	création de publications MobiLink	239
tables dans Interactive SQL	186	création de publications SQL Remote	253
conteneur	198	CURRENT PUBLISHER, instruction	252
contrainte		curseur	
définition	298	définition	299
contrainte de clé étrangère		D	
définition	298	date	
contrainte de clé primaire		combinaison	103
définition	298	composée	103
contrainte de vérification		condition de recherche	100, 104

dbeng9		définition	299
démarrage de serveurs de base de données	151	documentation	
restrictions	14	conventions	xvi
dbisql, utilitaire		SQL Anywhere Studio	x
généralités	167	domaine	
dbremote		présentation	279
plates-formes UNIX supportées	48	données distantes, accès	
plates-formes Windows supportées	43	plates-formes UNIX supportées	46
dbspace		plates-formes Windows supportées	40
définition	299	présentation	75
DDL		données groupées	127
commit automatique	138	droits DBA	
définition	299	définition	299
décharger		droits DBA distants	
définition	299	définition	299
DELETE, instruction		DROP, instruction	
erreur	142	commit automatique	138
exemple	142		
généralités	137	E	
démarrage		éditeur	
Interactive SQL	170	création	252
Sybase Central	195	définition	300
démarrage rapide		éditeur de procédure	
Interactive SQL	169	Sybase Central	207
détection et résolution des problèmes		éditeur de requête	
changement dans le jeu de résultats	96	affichage dans Interactive SQL	172
clause GROUP BY	129	Interactive SQL	172
jointure naturelle	114	effacement	
sous-requête	122	volet Instructions SQL	183
développement		éléments internes	
instruction SQL	168	Adaptive Server Anywhere	81
didacticiel		serveur de base de données	81
conception d'une base de données	215	élimination	
concepts de la synchronisation MobiLink	231	table	204
connexion à une base de données exemple	147	Embedded SQL	
InfoMaker	281	définition	300
PowerAMC	265	outils de développement	78
SQL Remote	247	enregistrement	
Sybase Central	193	états avec InfoMaker	285
disponibilité		résultats de transaction	138
composant d'Adaptive Server Anywhere	15	enregistrement de la licence	
distinction majuscules/minuscules		Java	5
nom de table	90	option de sécurité	5
SQL	90	erreur	
distribution d'états et de formulaires		Interactive SQL	183
avec InfoMaker	281	état	
DML		amélioration avec InfoMaker	287

création avec InfoMaker	281	fonction d'agrégat	
enregistrement avec InfoMaker	285	application aux données groupées	128
génération avec InfoMaker	284	présentation	127
impression d'états InfoMaker	289	formulaire	
prévisualisation avec InfoMaker	284	création avec InfoMaker	281
regroupement d'états InfoMaker	287	forum	
tri des états InfoMaker	287	support	xx
exécution		support technique	xx
commandes dans Interactive SQL	183	fréquence de réplication	
commandes dans Interactive SQL	183	définition	301
requête exécutée plus d'une fois	96	FTP, type de message	
extraction		plates-formes UNIX supportées par SQL Remote	
définition	300	48	
		plates-formes Windows supportées par SQL	
		Remote	43
F		G	
Federal Rehabilitation Act		génération	
section 508	5	états avec InfoMaker	284
fichier d'écriture		glossaire	295
définition	301	go	
fichier de base de données		délimiteur d'instruction	185
définition	300	GRANT PUBLISH, instruction	
présentation	82	présentation	252
fichier de base de données compacté		GRANT REMOTE, instruction	
définition	300	présentation	252
fichier de commandes		GROUP BY, clause	
création	184	erreurs	129
définition	300	fonction d'agrégat	128
présentation	184	groupe	
volet Instructions SQL	184	ajout	205
fichier JAR		ajout d'utilisateurs à partir de Sybase Central	206
définition	301	création	205
fichier journal		groupe de programmes	
définition	301	Adaptive Server Anywhere	6
fichier temporaire			
présentation	82	H	
FILE		Handheld PC	
définition	301	Windows CE	54
FILE, type de message		HAVING, clause	
définition	301	clause GROUP BY	130
filtrage		clause WHERE	130
présentation	101	Historique de commandes, boîte de dialogue	
fin transaction	138	rappel de commande dans Interactive SQL	187
fin transactions	138	utilisation dans Interactive SQL	187
flux de communication			
définition	301		
fonction			
SOUNDEX	102		

I			
icône			
icônes utilisées ans ce manuel	xvii		
ID de connexion			
définition	301		
ID utilisateur			
généralités	170		
nouvelle base de données	218		
utilisation	170		
identificateur			
définition	301		
impression			
états InfoMaker	289		
IN, condition			
utilisation en tant que raccourci pour les conditions de recherche composées	104		
index			
définition	302		
présentation	97		
inégalité			
test	100		
InfoMaker			
définition	302		
didacticiel	281		
généralités	281		
informations			
documentation	xx		
obtention	xx		
informatique de groupe de travail			
conditions requises	11		
informatique mobile			
conditions requises	11		
INPUT, instruction			
insertion de lignes dans Interactive SQL	180		
INSERT, instruction			
exemple	141		
présentation	135		
insertion			
lignes dans les tables dans Interactive SQL	179		
installation			
SQL Anywhere Studio	6		
instruction SQL			
définition	302		
intégrité			
contrôle	141		
définition	302		
intégrité référentielle			
		définition	302
		intégrité transactionnelle	
		définition	302
		Interactive SQL	
		affichage d'une liste de tables	108, 186
		affichage de l'éditeur de requête	172
		boîte de dialogue Historique de commandes	187
		configuration	172
		copie de lignes	181
		définition	302
		démarrage	170
		démarrage rapide	169
		effets d'une sortie	138
		exécution de texte sélectionné uniquement dans le volet Instructions SQL	185
		exécution de tout le texte dans le volet Instructions SQL	172
		fenêtre principale	170
		généralités	167
		interruption de commande	183
		option	172
		présentation	15, 168
		Interactive SQL	
		affichage d'une liste de procédures	186
		affichage des données	176
		annulation de commande	183
		arrêt de commande	183
		barre d'outils	171
		chargement de commandes	90
		combinaison de plusieurs instructions	184
		consignation des commandes	189
		consultation de nom de colonne	186
		consultation de nom de procédure	186
		consultation de nom de table	186
		enregistrement de commandes	90
		erreurs signalées	183
		exécution de commandes	90, 183
		insertion de lignes	179
		mise à jour de colonne calculée	179
		modification des valeurs de table	178
		ouverture de fenêtres multiples	171
		présentation des commandes	183
		raccourci clavier	172
		rappel de commandes	187
		récupération de commandes	90
		regroupement des modifications dans des transactions	138

suppression de lignes	180	définition	303
touche de fonction	172	jointure naturelle	
volet Instructions SQL	90	erreur	114
Interactive SQL, utilisation, affichage des données		présentation	114
176		jointure par clé	
interface		présentation	112
serveur de base de données	152	journal de transactions	
interface de programmation		définition	303
Adaptive Server Anywhere	76	présentation	82
ADO	80	journal des annulations	
Embedded SQL	78	définition	303
JDBC	80		
ODBC	78	L	
OLE DB	80	lancement	
Open Client	79	transactions	138
supportée dans Adaptive Server Anywhere	76	ligne	
J		ajout	135
Java, logique de synchronisation		ajout à l'aide d'Interactive SQL	179
plates-formes UNIX supportées	47	copie dans Interactive SQL	181
plates-formes Windows supportées	42	généralités	62
Java, utilisation dans la base de données		insertion dans Interactive SQL	179
achat	5	modification des valeurs dans Interactive SQL	178
licence distincte	5	sélection dans une table	99
jConnect		suppression à l'aide d'Interactive SQL	180
définition	302	ligne de commande, utilitaire	
JDBC		présentation	15
définition	302	Linux	
outils de développement	80	éléments d'Adaptive Server Anywhere	46
jeu de résultats		plates-formes supportées par l'agent de	
copie de lignes	181	réplication	48
détection et résolution des problèmes	96	liste de sélection	
exécution répétée d'une requête	96	colonne calculée	92
insertion de lignes	179	nom de colonne	92
modification de valeur de table dans Interactive		livraison en différé	
SQL	178	réplication SQL Remote	26
suppression de lignes	180	login externe	
jeu de résultats curseur		définition	303
définition	303	login intégré	
jointure		définition	303
définition	303	LTM	
présentation	108	définition	295
sous-requête	123	M	
jointure externe		manuel en ligne (PDF)	
définition	303	utilisation	xiii
présentation	116	MAPI	
jointure interne		définition	304

MAPI, type de message	
définition	304
plates-formes UNIX supportées par SQL Remote	48
plates-formes Windows supportées par SQL Remote	43
menu Démarrer	
Adaptive Server Anywhere	6
message de réplication	
définition	304
métadonnées	
définition	304
tables système	87
MIPS, processeur	
Windows CE	54
miroir de journal de transactions	
définition	304
mise à jour	
données	133
valeur dans Interactive SQL	178
mise à jour de la base de données	133
mise à jour de publication	
définition	304
mise en route	
InfoMaker	283
MobiLink	
définition	304
Didacticiel	231
MobiLink, base de données consolidée	
SGBDR supporté	50
MobiLink, synchronisation	
autres approches	21
plates-formes UNIX supportées	47
plates-formes Windows supportées	42
modèle conceptuel de base de données	
définition	215
modèle physique de bases de données	265
modélisation des bases de données	265
modification	
valeur de table dans Interactive SQL	178
module de connexion	
définition	304
mot de passe	
connexion à une nouvelle base de données	218
utilisation	170
moteur de base de données	
définition	70

N

.NET, logique de synchronisation	
plates-formes UNIX supportées	47
plates-formes Windows supportées	42
NetWare	
définition	305
éléments d'Adaptive Server Anywhere	40
niveau d'isolement	
définition	305
nom de base de données	
définition	305
nom de rôle	
définition	305
Nom de table de consultation, boîte de dialogue	
affichage d'une liste de tables	108
normalisation	
définition	305
norme	
section 508	5
Novell NetWare	
éléments d'Adaptive Server Anywhere	40
NULL	
autorisation dans les colonnes	135, 225
nuplet	63

O

objet de base de données	
définition	305
généralités	65
objet système	
définition	305
ODBC	
définition	305
généralités	160
outils de développement	78
présentation d'une source de données	159
sources de données	160
OLE DB	
outils de développement	80
Open Client	
outils de développement	79
optimisation des requêtes	
spécificité d'Adaptive Server Anywhere	12
option	
définition pour Interactive SQL	172
option de sécurité	

achat	5	plate-forme hôte	
licence distincte	5	plates-formes supportée par UltraLite Windows41	
option GRANT		plate-forme supportée	
définition	305	Adaptive Server Anywhere	13
ORDER BY, clause		composants SQL Anywhere Studio sous UNIX	46
exemple	95	composants SQL Anywhere Studio sous Windows	40
requis pour que les lignes apparaissent toujours dans le même ordre	96	généralités	37
utilisation des index pour améliorer les performances	97	plates-formes supportées par MobiLink UNIX	47
ordre alphabétique		plates-formes supportées par MobiLink Windows	42
clause ORDER BY	95	systèmes d'exploitation supportés	38
ouverture de fenêtres multiples Interactive SQL	171	version d'Adaptive Server Anywhere	39
P		Pocket PC	
package		Windows CE	54
définition	305	point de reprise	
Palm Computing, plate-forme		définition	305
environnement de développement UltraLite	42	position du curseur	
panne système		définition	306
transactions	140	PowerAMC	
paramètre		définition	306
fonction	127	didacticiel	265
partage de fichiers, type de message		généralités	265
plates-formes UNIX supportées par SQL Remote	48	PowerDynamo	
plates-formes Windows supportées par SQL Remote	43	définition	306
passthrough		PowerJ	
définition	300	définition	306
PDF		prédicat	
accès aux manuels en ligne	xiii	définition	306
performances		présentation	103
avantages de la réplication	22	prévisualisation	
pilote OLE DB		états avec InfoMaker	284
processeurs supportés	54	procédure	
pipeline		consultation dans Interactive SQL	186
création avec InfoMaker	281	procédure stockée	
plate-forme		définition	306
supportée	13	définition des autorisations	207
systèmes d'exploitation MobiLink UNIX	47	visualisation	207
systèmes d'exploitation MobiLink Windows	42	processeur ARM	
systèmes d'exploitation supportés	38	support	57
plate-forme cible		processeur XScale	
UltraLite	52	support	57
plate-forme de développement		produit cartésien	
UltraLite	41	présentation	109
		profil de base de données	
		InfoMaker	283
		profil de connexion	

définition	306	configurations hiérarchiques de bases de données	
projection		23	
définition	87	définition	307
propriétaire de base de données		didacticiel SQL Remote	247
définition	306	par connexion	26
propriété		par message	26
objets de base de données	200	réplication bidirectionnelle	25
protocole de communication		réplication par message	26
Adaptive Server Anywhere	77	Replication Server	
publication		caractéristiques	33
définition	306	définition	307
R		reprise	
raccourci clavier		définition	307
Interactive SQL	172	requête	
rappel		définition	86, 307
commande dans Interactive SQL	187	instruction SELECT	89
recherche		Interactive SQL	183
clause GROUP BY	130	restauration	
commande dans Interactive SQL	187	spécificité d' Adaptive Server Anywhere	12
redirecteur		restauration des données	
définition	306	transaction	140
plates-formes Windows supportées	43	restriction	
référence externe		définition	87
définition	123	rétro-ingénierie	
règle de gestion	279	bases de données	268
définition	307	rôle	
regroupement		définition	307
états InfoMaker	287	ROLLBACK, instruction	
regroupement des données	126	généralités	139
regroupement des modifications dans des		présentation	137
transactions	138	transactions	138
relation		S	
1-n	227	saisie	
1-1	227	commandes Interactive SQL	183
cardinalité	227	instructions multiples dans Interactive SQL	184
entités	63	sauvegarde	
généralités	227	base de données active	210
n-n	228	présentation	210
relation 1-1		Sybase Central	210
définition	227	sauvegarde de base de données, assistant	
relation 1-n		utilisation	210
définition	227	sauvegarde incrémentielle	
réplication		définition	307
autres approches	21	sauvegarde intégrale	
comparaison des technologies	29	définition	307
		schéma	

conception de base de données	266	plates-formes Windows supportées par SQL	
définition	216, 307	Remote	43
visualisation	198	SOUNDEX, fonction	
script		généralités	102
définition	307	source de données	
section 508		présentation	159
conformité	5	source de données ODBC	
SELECT, instruction		définition	308
Interactive SQL	176	souscription	
présentation	89	définition	308
sous-requête	118	sous-requête	
sélection		comparaison	121
définition	87	corrélée	123
sélection de données à l'aide de sous-requêtes	117	définition	308
sélection de données dans des tables de base de données	85	détection et résolution des problèmes	122
sélection de données dans plusieurs tables	107	jointure	123
sélection de lignes dans une table	99	présentation	118
sélection des données d'agrégat	125	sous-requête corrélée	
serveur de base de données		définition	123
connexion	147, 157	SQL	
définition	308	définition	308
démarrage	151	développement de requêtes	168
différences serveur personnel/serveur réseau	14	généralités	86
éléments internes	81	SQL Anywhere Studio	
exécution	147, 157	documentation	x
fenêtre	152	SQL et base de données	86
généralités	148	SQL Modeler	
interface	152	voir PowerDesigner	265
serveur personnel		SQL Remote	
définition	308	caractéristiques	33
restrictions	14	configuration d'une base de données consolidée	
support de plate-forme	14	251	
serveur réseau		configuration d'une base de données distante	
définition	308	(didacticiel)	255
support de plate-forme	14	définition	308
serveur, nom		didacticiel	247
présentation	153	octroi d'autorisation PUBLISH	252
service		octroi d'autorisation REMOTE	252
définition	308	plates-formes UNIX supportées	48
SGBD		plates-formes Windows supportées	43
définition	309	SQL Anywhere Studio	
SGBDR		éléments	4
définition	62, 309	présentation	4
SMTP, type de message		ssqueue	
plates-formes UNIX supportées par SQL Remote	48	plates-formes UNIX supportées	48
		plates-formes Windows supportées	43
		ssremote	

plates-formes UNIX supportées	48	synchronisation orientée session	
plates-formes Windows supportées	43	définition	309
statistique de performances		SYS	
définition	308	définition	309
support technique		système d'exploitation	
forum	xx	accès aux données distantes sous UNIX	46
suppression		accès aux données distantes sous Windows	40
ligne à l'aide d'Interactive SQL	180	Adaptive Server Anywhere sous UNIX	46
lignes des tables	180	Adaptive Server Anywhere sous Windows	40
table	204	agent de réplication	43, 48
Sybase Central		Embedded SQL sous UNIX	46
ajout d'utilisateurs à des groupes	206	Embedded SQL sous Windows	40
ajout de scripts de table à la base de données		Interactive SQL sous UNIX	46
consolidée	241	Interactive SQL sous Windows	40
assistants	210	Java sous Windows	40, 46
connexion à une base de données	195	plates-formes supportées par MobiLink UNIX	47
création d'une version de script	240	plates-formes supportées par Mobilink Windows	42
création d'utilisateurs	205	plates-formes supportées par UltraLite UNIX	47
création d'utilisateurs MobiLink	239	plates-formes supportées par UltraLite Windows	41
création de groupe	205	SQL Remote sous UNIX	48
création de publications SQL Remote	253	SQL Remote sous Windows	43
création de souscriptions de synchronisation	239	supporté	13, 38
définition	309	Sybase Central sous UNIX	46
démarrage	195	Sybase Central sous Windows	40
didacticiel	193	UltraLite	52
éditeur de code	207	utilitaires d'administration sous UNIX	46
fenêtre principale	198	utilitaires d'administration sous Windows	40
généralités	193	versions supportées	54
présentation	15	système de gestion de bases de données	
sauvegarde de base de données	210	relationnelles	
visualisation des données	202	définition	62
synchronisation		système de messagerie	
autres approches	21	définition	309
comparaison des fonctions MobiLink	29		
comparaison des technologies	29	T	
configurations hiérarchiques de bases de données		table	
MobiLink	23	ajout	203
définition	309	ajout de colonne	203
didacticiel MobiLink	231	caractéristiques	63
synchronisation exécutée par le serveur		clés étrangères	64
définition	309	conception	220
synchronisation MobiLink		consultation dans Interactive SQL	186
comparaison des fonctions	29	création	203
didacticiel	231	création d'une clé primaire	203
didacticiel d'écriture de publications	233	élimination	204
didacticiel d'écriture de scripts	233		
didacticiel d'écriture de souscriptions	233		

généralités	62	transfert orienté fichier	
modification	203	définition	311
suppression	204	tri	
visualisation	198	états InfoMaker	287
table de base	65	résultats d'une requête	95
table de travail		trigger	
définition	309	définition	311
table étrangère		visualisation	207
définition	309	trigger de niveau instruction	
table primaire		définition	311
définition	310	trigger de niveau ligne	
table proxy		définition	311
définition	310	trigger de résolution de conflit	
présentation	75	définition	311
table sous-jacente		type de données	
définition	310	définition	311
table système		type de données défini par l'utilisateur	
définition	87, 310	définition	311
table temporaire		type de jointure	
définition	310	définition	312
table temporaire globale		type de message	
définition	310	définition	312
table temporaire locale			
définition	310	U	
taille de base de données		UltraLite	
base de données de plusieurs gigaoctets	11	comparaison avec Adaptive Server Anywhere	18, 19
TDS, protocole de communication		définition	312
diagramme	76	généralités	19
technologie de réplication	29	objectifs de conception	19
technologie de réplication, caractéristiques	32	plates-formes cibles	52
technologie de réplication, choix	29	plates-formes de développement	41
technologie de réplication, défis	22	plates-formes UNIX supportées	47
téléchargement		plates-formes Windows supportées	41
définition	310	UNIX	
touche de fonction		éléments d'Adaptive Server Anywhere	46
Interactive SQL	172	plates-formes supportées par l'agent de réplication	48
traitement des transactions		UPDATE, instruction	
restauration des données	140	erreur	143
spécificité d'Adaptive Server Anywhere	12	exemple	143
transaction		présentation	136
définition	310	utilisateur	
fin	138	ajout	205
lancement	138	ajout dans un groupe à partir de Sybase Central	
regroupement des modifications	138	206	
restauration des données	140	création	205
transfert			
définition	311		

gestion	205	condition BETWEEN	104
utilisateur distant		exemples	99
définition	312	filtrage	101
utilisateur MobiLink		modification de lignes dans une table	136
définition	312	suppression de lignes	137
utilisation d'Interactive SQL	167	Windows	
utilisation des instructions SQL dans		définition	313
Interactive SQL	183	systèmes d'exploitation supportés	13
utilitaire		Windows 98/Me	
présentation	15	éléments d'Adaptive Server Anywhere	40
utilitaire de base de données		plates-formes supportées par l'agent de	
assistants	210	réplication	43
V		Windows CE	
V4T, mode		définition	313
processeurs ARM	54	éléments d'Adaptive Server Anywhere	40
valeur		OLE DB	54
modification dans Interactive SQL	178	plates-formes supportées par l'agent de	
valeur de table		réplication	43
modification dans Interactive SQL	178	processeurs supportés	54
valider		versions supportées	54
définition	312	Windows NT/2000/XP	
verrou		éléments d'Adaptive Server Anywhere	40
définition	312	plates-formes supportées par l'agent de	
verrouillage au niveau ligne		réplication	43
spécificité d'Adaptive Server Anywhere	12	X	
version de déploiement		x86, processeur	
généralités	39	Windows CE	54
version intégrale			
généralités	38		
VIM, type de message			
plates-formes UNIX supportées par SQL Remote	48		
plates-formes Windows supportées par SQL			
Remote	43		
visualisation			
données à l'aide de Sybase Central	202		
vue			
définition	313		
visualisation	207		
vue système			
définition	313		
W			
WHERE, clause			
clause HAVING	130		
comparaison des dates	100		