

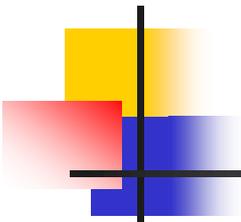


François Jacquenet
Professeur d'Informatique
Faculté des Sciences
Laboratoire Hubert Curien – UMR CNRS 5516
18 rue Benoit Luras
42000 Saint-Etienne
Tél : 04 77 91 58 07
Francois.Jacquenet@univ-st-etienne.fr

Licence de Sciences et Techniques

BASES DE DONNEES

<http://labh-curien.univ-st-etienne.fr/~fj/bd>



Organisation administrative

■ Cours

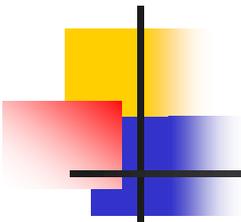
- Le mardi de 14h à 16h
- Enseignant : François Jacquenet
- Supports de cours sur <http://labh-curien.univ-st-etienne.fr/~fj/bd>

■ TD

- Le mardi de 16h15 à 18h15 et le mercredi de 14h à 16h
- Enseignant : François Jacquenet

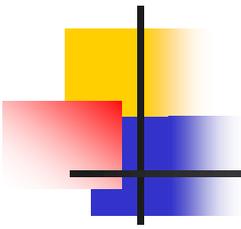
■ TP

- Groupe 1 : mardi de 16h15 à 18h15
- Groupe 2 : mercredi de 14h à 16h
- Enseignants : David Combe et Christophe Moullin



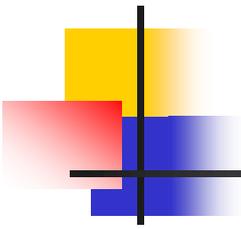
Organisation administrative

- Trois contrôles continus (même coefficient)
 - Début mars(2h)
 - Début avril (2h)
 - Début mai (2h)
- Cela implique
 - Qu'il faut travailler dès le début
 - Qu'il faut être assidu
 - Qu'il faut profiter de votre présence à la fac
 - être concentré pendant les cours, TD et TP
 - poser des questions
 - Qu'il faut travailler en dehors des cours
- A chaque cours
 - Au début : rappel sur le cours précédent
 - A la fin : synthèse de ce qui vient d'être vu
 - C'est vous qui ferez ce travail
 - **N'hésitez pas à poser des questions avant, pendant, après le cours !**
 - **N'hésitez pas à prendre contact avec moi !**
- Alors tout se passera bien



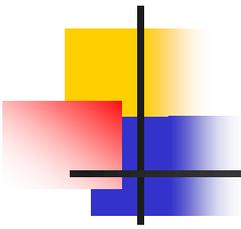
Plan du cours

- Introduction
- Introduction au modèle relationnel et à l'algèbre relationnelle
- Langage de manipulation : SQL
- Le SGBD Access
- Modélisation d'une base de données
- Normalisation d'une relation



Mes objectifs

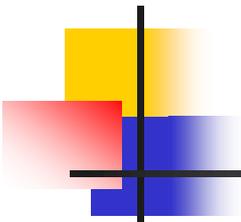
- **Vous faire réussir cette unité !!!**
- Faire en sorte qu'à la fin du cours vous soyez capable de :
 - Concevoir une base de données en utilisant un logiciel du commerce.
 - Interroger une base de données en utilisant un logiciel du commerce.
 - Modéliser une base de données en utilisant une technique courante de modélisation conceptuelle.



Références

Ce cours est inspiré de nombreuses références, notamment :

- Bases de données objet et relationnel, G. Gardarin, Eyrolles
- Bases de données, les systèmes et leurs langages, G. Gardarin, Eyrolles
- Introduction aux bases de données, C.J. Date, Vuibert
- Fondements des bases de données, S. Abiteboul, Vuibert
- Programmation SQL, R.A. Mata-Toledo, Schaum (Dunod)
- Cours de l'EPFL (Stéfano Spaccapietra)
- Cours du CNAM Paris (Philippe Rigaud)
- Cours de l'Université de Dauphine (Witold Litwin)
- Cours de l'Université du Wisconsin et de l'Université de Cornell (R. Ramakrishnan, J. Gehrke, J. Derstadt, L. Zhu)



Logiciels utilisables

■ ACCESS

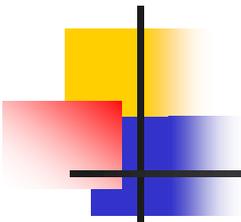
- Dans le pack office → payant sauf avec MSDNA
- Microsoft
- Interface conviviale
- Facilité de développement de petites applications

■ MySQL

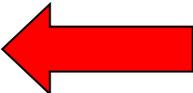
- Logiciel libre → gratuit
- Téléchargeable à : <http://www.mysql.fr>
- Très souvent associé à PHP dans le cadre de sites Web

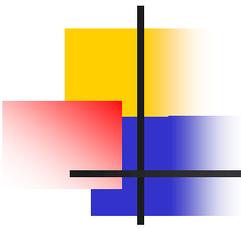
■ PostgreSQL

- Logiciel libre → gratuit
- Téléchargeable à : <http://www.postgresql.org>



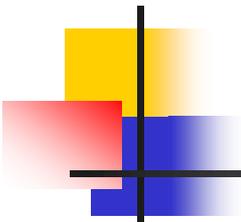
BD et informatique

- L'informatique est une vaste discipline
- L'IEEE (<http://www.ieee.org>) a répertorié les domaines à enseigner dans un cursus d'informaticien
 - Structures discrètes
 - Fondements de la programmation
 - Algorithmique et complexité
 - Langages de programmation
 - Architecture et organisation
 - Systèmes d'exploitation
 - Réseaux
 - Interfaces Homme/Machine
 - Informatique graphique
 - Systèmes intelligents
 - Management de l'information 
 - Génie Logiciel
 - Problèmes socio-professionnels



BD et informatique

- Management de l'information :
 - Modèles et systèmes d'information
 - Outils de bases de données
 - Modélisation des données
 - Bases de données relationnelles
 - Bases de données et langages de requêtes
 - Conception de bases de données relationnelles
 - Traitement transactionnel
 - Bases de données distribuées
 - Conception physique des bases de données
 - Fouille de données
 - Stockage et recherche d'information
 - Hypertexte et hypermédia
 - Information et systèmes multimédia
 - Librairies digitales



BD et emploi

- **Marché du travail : BD = gros pourcentage des offres d'emploi**
- **Exemples**

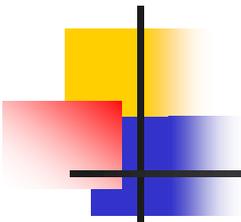
... Nous recherchons des **DBA Oracle** pour des projets de longue durée. A ce titre, vous optimisez le portage des **données Oracle 9i** en environnement AIX vers une **base Oracle 11g** en environnement Windows Server. De plus, vous créez l'environnement Oracle d'accueil des données. L'action concerne le stockage des données, l'accès aux données, les scripts, ainsi que les procédures stockées utilisées dans les jobs de chargement...

... AS+, société de conseil et d'ingénierie en pleine croissance, d'aujourd'hui 25 personnes recrute dans le cadre du développement de ses activités. Vous prendrez en charge chez le client **l'administration de la base de données Oracle** sous Unix ainsi que l'exploitation de l'application. Vous devrez gérer et résoudre les incidents d'exploitation de l'application et créer et renseigner les dossiers permettant le suivi des évènements...

... Nous recherchons un **ADMINISTRATEUR DE BASE DE DONNEES** Sybase & Oracle...

... De formation Bac+3 à Bac+5, vous justifiez d'une expérience d'au moins 1 an sur l'environnement J2EE et d'une réelle maîtrise des développements JAVA. A ce titre, vous maîtrisez :

- langages : JAVA, Javascript, HTML, **SQL**, STRUTS, JSP
- **base de données : DB2**
- serveur d'application : Websphère



BD et emploi

■ Exemples (suite)

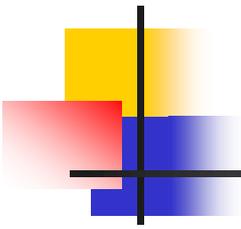
... Au sein de la société Free, vous participerez aux activités de conception et de réalisation des différents projets. Vous avez une expérience significative dans le développement de **systèmes d'information** et de la maîtrise des **bases de données**. Autonome, rigoureux et motivé, vous souhaitez apporter vos compétences à des projets innovants et dynamiques. Vous possédez impérativement les connaissances suivantes :

- environnement : GNU/Linux, UNIX
- langage : Perl, C, **SQL**, HTML
- **maîtrise des bases de données**

La maîtrise des langages et des outils de développement Internet, la connaissance des solutions Open Source et la maîtrise des scripts shell ou Perl seront appréciées...

... BAC +4/5 ou école d'ingénieur, connaissance des outils Websphere et Tomcat, et CFT Maîtrise des technologies Java, J2EE, Framework Struts, Html, XML, CSS Expérience d'utilisation de Jbuilder X, Crystalreport9, Starteam 6, PowerAMC Bonnes compétences **bases de données, SQL** et **Sybase** (Transact Sql) et Shell Unix...

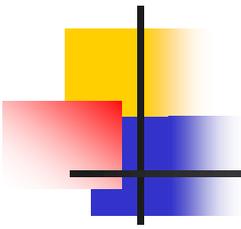
... Domaines de compétences requis : Langages : Java, PHP, C/C++, Technologies Java : J2EE, EJB 2.0, CORBA, RMI, JavaBeans, Servlets, JSP,... Serveurs d'Applications : BEA WebLogic, Jonas, JBoss... **SGBDR : Oracle, SQL Server, MySQL** ,... XML : XML, XSL, XML Schema,... Web Services : JAXM, JAX RPC, SOAP, Apache Axis, IBM WSDK OS : Linux, Solaris, Windows 200x ...



BD et emploi

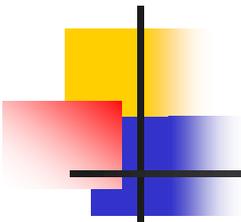
- Tout informaticien peut être (sera) amené à
 - Utiliser des bases de données
 - Concevoir de grosses bases de données
 - Administrer de grosses bases de données

- Tout non informaticien peut être amené à
 - Utiliser
 - Concevoir de petites bases



Exemples de bases de données

- Gestion des personnels, étudiants, cours, inscriptions, ... de l'UJM
- Système de réservation de places d'avions chez Air France, de places de trains à la SNCF
- Gestion des comptes clients de La Poste
- Gestion des commandes chez Amazon.com
- Gestion d'une bibliothèque
- Gestion des pages Web chez google.com
- ...



Que faut-il gérer ?

Exemple de la réservation de billets de trains

Billet =

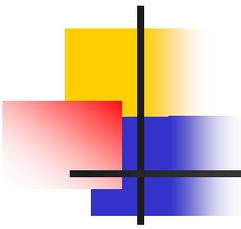
nom client
numéro train
date
classe
no wagon
numéros place
départ: - gare
 - heure
gare d'arrivée

Train =

numéro train
gare départ
heure départ
destination finale
heure d'arrivée
jours

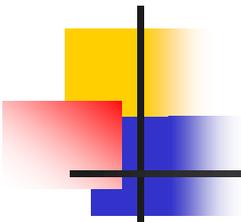
Arret =

numéro train
no arret
gare
heure départ
heure d'arrivée



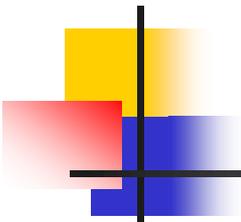
Quels sont les besoins dans le domaine de la gestion de l'information ?

- Décrire l'information
- Manipuler l'information
- Interroger la collection d'informations
- Exactitude et cohérence
- Garanties
 - Fiabilité
 - Contrôle de concurrence
- Confidentialité
- Efficacité



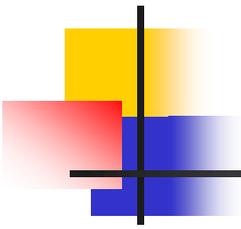
Besoin de pouvoir **décrire**

- Décrire les données de l'application (trains, trajets et réservations) sans faire référence à une solution informatique particulière
 - **modélisation conceptuelle**
- Élaborer une description équivalente pour le stockage des données dans le SGBD choisi
 - **modélisation logique**
 - **langage de description de données (LDD)**



Besoin de pouvoir manipuler

- Créer la base de données initiale avec les données représentant le réseau SNCF
 - langage permettant l'insertion de données
- Créer au fur et à mesure les données sur les réservations. Modifier si besoin et éventuellement supprimer toute donnée déjà rentrée
 - langage de manipulation de données (LMD)
(insertion, modification, suppression)



Besoin de pouvoir **interroger**

Répondre à toute demande d'information portant sur les données contenues dans la base.

Par exemple:

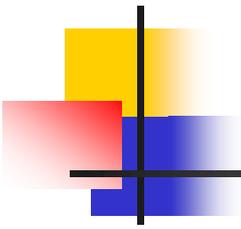
a) John Smith a-t-il une réservation pour aujourd'hui ?

- Si oui, donner les informations connues sur cette réservation.

b) Quels sont les horaires des trains de Saint-Etienne à Dijon entre 9h et 10h le dimanche ?

c) Donner les destinations au départ de Saint-Etienne sans arrêt intermédiaire.

→ langage de requête (langage d'interrogation)



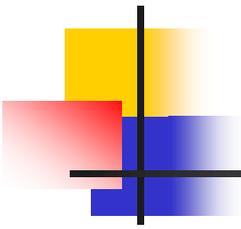
Besoin d'exactitude, de cohérence

Il faut pouvoir exprimer toutes les règles qui contraignent les valeurs pouvant être enregistrées de façon à éviter toute erreur qui peut être détectée.

Par exemple:

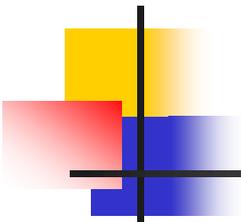
- Il ne faut jamais donner la même place dans le même train à 2 clients
- Les arrêts d'un train sont numérotés de façon continue (il ne peut y avoir pour un train donné un arrêt n° 3 s'il n'y a pas un arrêt n° 2 et un arrêt n° 1)
- La date de réservation pour un train doit correspondre à un jour de circulation de ce train
- Le numéro de train dans une réservation / arrêt doit correspondre à un train existant
- L'heure de départ d'une gare doit être postérieure à l'heure d'arrivée dans cette gare
- L'heure d'arrivée à un arrêt doit être postérieure à l'heure de départ de l'arrêt précédent

→ langage d'expression de contraintes d'intégrité



Besoin de **garanties**

- Il ne faut pas que les informations (par exemple, les réservations) soient perdues à cause d'un dysfonctionnement quelconque: erreur de programmation, panne système, panne de l'ordinateur, coupure de courant, etc
→ garantie de **fiabilité**
- Il ne faut pas qu'une action faite pour un utilisateur (par exemple l'enregistrement d'une réservation) soit perdue du fait d'une autre action faite simultanément pour un autre utilisateur (réservation de la même place).
→ garantie de **contrôle de concurrence**

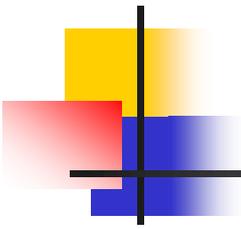


Besoin de **confidentialité**

- Toute information doit pouvoir être protégée contre l'accès par des utilisateurs non autorisés
 - en lecture
 - en écriture

Interdire par exemple aux clients de modifier les numéros des trains ou les horaires ou leur réservation.

→ garantie de **confidentialité** (privacy)



Besoin d'efficacité

Le temps de réponse du système doit être conforme aux besoins :

- en interactif : pas plus de 3 secondes
- en programmation : assez rapide pour assumer la charge de travail attendue (nombre de transactions par jour)
 - mécanismes d'optimisation
 - éventuellement, répartition / duplication des données sur plusieurs sites

Comment gérer des données ?

■ Avec des **fiches en papier**

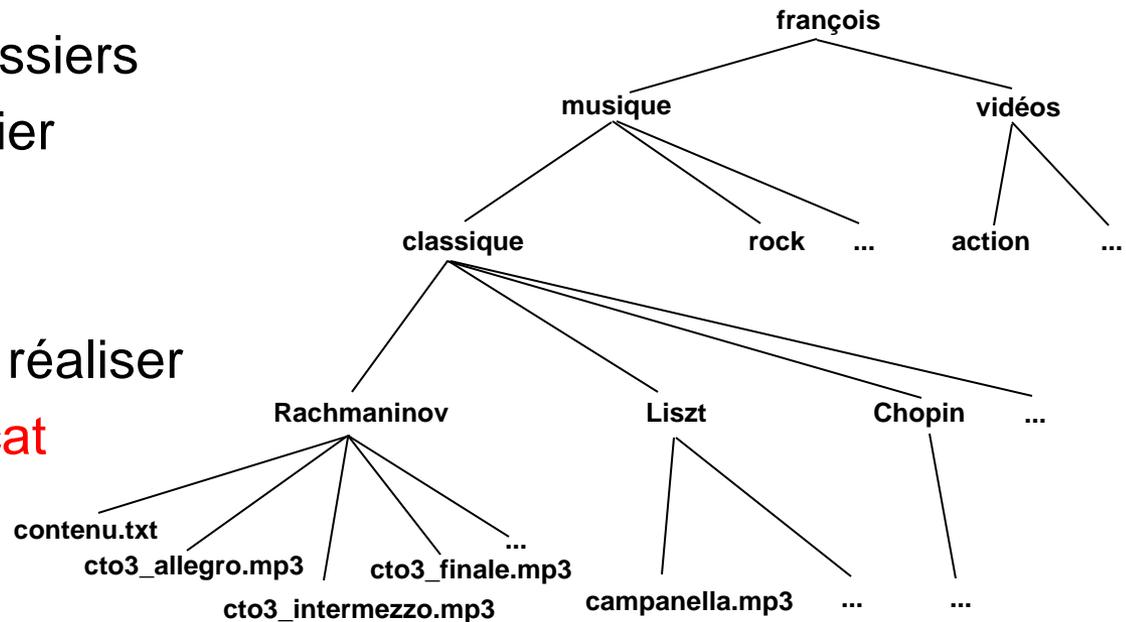
- Exemple des bibliothèques pendant très longtemps
- Recherche très fastidieuse
- Mise à jour longue
- Place très importante
- Problèmes de sécurité
- Faible fiabilité
- Aucune automatisation
- **Passage à l'échelle délicat**
- ...

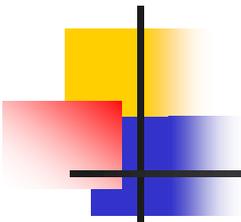


Comment gérer des données ?

■ Avec des fichiers informatiques

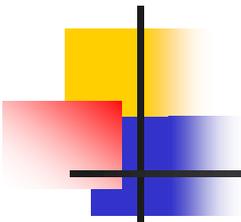
- C'est un peu ce que l'on fait sur son ordinateur..
- Fichiers organisés en dossiers
- Index dans chaque dossier
- Recherche fastidieuse
- Problèmes de sécurité
- Automatisation difficile à réaliser
- **Passage à l'échelle délicat**
- ...





Comment gérer des données ?

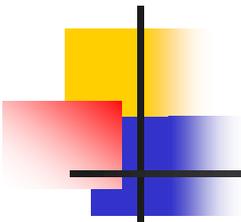
- Base de données
 - ensemble cohérent, intégré et partagé de données structurées
 - défini pour les besoins d'une application
- **S**ystème de **G**estion de **B**ase de **D**onnées (**SGBD**)
 - logiciel permettant de couvrir les besoins :
 - définir une représentation des informations apte à
 - stocker, interroger et manipuler (insérer, supprimer, mettre à jour) de
 - grandes quantités de données (plus que la mémoire vive)
 - dont il faut garantir la longévité et
 - l'accessibilité de manière concurrente (plusieurs utilisateurs simultanés)
 - et sûre.



Qu'est-ce qu'une base de données ?

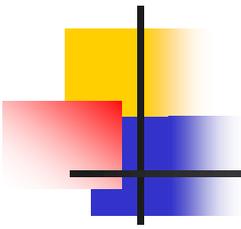
Une collection de données qui supporte les *manipulations* :

- **Recherche** de données
 - Interactive
 - Assertionnelle
 - L4G
 - Logiquement et physiquement performante
- **Insertion, Mise à jour, Suppression** de données
 - Cohérente
 - Partagée
 - Fiable



Pourquoi les bases de données ?

- **Intégration** des données
 - Moins de duplications
- **Partage** des données
- **Fiabilité** des données
 - Transactions, Reprises sur pannes, Tolérance de pannes
- **Sécurité** des données
- Langages **assertionnels** de requêtes
 - SQL, QBE
- Interfaces **conviviales**
 - L4G & Web



Pourquoi les bases de données ?

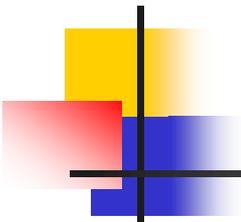
En pratique, il est impossible de concevoir ces fonctions soi-même

■ Théorie complexe

- Conception d'une BD
 - Formes normales...
- Langages de définition et de manipulation d'une BD
- Optimisation de requêtes assertionnelles
- Transactions et partage de données
-

■ Logiciel complexe

- des centaines de milliers ou des millions de lignes de programme



Qui utilise des Bases de Données ?

■ Les utilisateurs interactifs

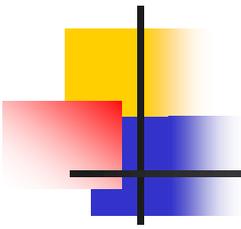
- Ils cherchent des informations sans connaître la Base de Données
- Utilisent des interfaces (formulaire, Web, ...)
- Peuvent à la rigueur utiliser des langages tels que QBE

■ Les programmeurs d'applications

- Construisent les interfaces pour les utilisateurs interactifs
- Spécialistes de SQL

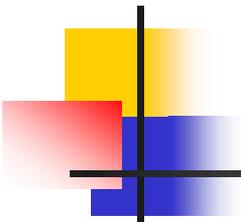
■ Les DBA (Database Administrators)

- Modélisent les bases de données
- Maintiennent les bases de données
- Ont la priorité sur tous les autres usagers
- Peuvent être très bien payé...



Différents types de BD

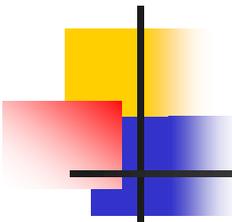
- Il y en a pour tous les usages
- BDs personnelles
 - MsAccess
 - 100 Ko – 500 Ko
- BDs professionnelles typiques
 - 500 Ko – 100 Go
- BDs professionnelles très grandes
 - Very Large Databases (VLDB)
> 100 Go



Culture générale

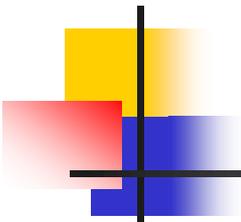
- Informatique = 0 ou 1 = bit
- 1 octet = 8 bits (= 1 byte)
- 1 Ko = 2^{10} octets = 1024 octets
- 1 Mo = 1024 Ko (méga octet) = 1048576 octets
- 1 Go = 1024 Mo (giga octet) = 1073741824 octets
- 1 To = 1024 Go (terra octet) = ...
- 1 Po = 1024 To (peta octet) = ...

- disques durs moyens sur PC : 500 Go à 1.5 To
- mémoire vive moyenne sur PC : 4 Go



VLDB'98 survey (taille)

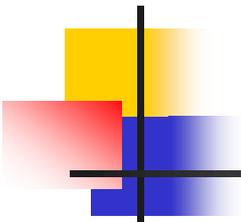
Rank	Organization	DBMS	Processor	Architecture	Storage	Data(GB)
1	UPS - PLD Repository (federated)	DB2	IBM S/390, Hitachi Skyline	SMP	EMC, IBM	16,800
1	TELSTRA	DB2	Hitachi Skyline	Cluster	HDS, EMC, IBM	4,350
2	Experian - Credit Reporting	DB2	Hitachi Skyline 62	SMP	EMC	2,739
3	SK Telecom	DB2	IBM S/390	MPP	HDS, IBM	1,955
4	IZB Software	Adabas	Amdahl 5995	SMP	EMC, Comparex	1,310
5	Caixa Economica Federal	CA-IDMS	IBM S/390	SMP	EMC, HDS, IBM	661
6	Victoria's Secret Catalogue	DB2	Amdahl	SMP	IBM, EMC	660
7	DBS Systems Corp.	Oracle RDB	Digital VAX7830	MPP	DEC	631
8	Mitsukoshi Information Service Co.	Teradata	NCR 5100	MPP	EMC	600
9	CentreLink	CCA Model 204	IBM S/390	Cluster	IBM, EMC	582
10	Metromail Corporation	Oracle	Sequent Symmetry 5000	SMP	EMC	405



VLDB'2003 survey (taille)

Normalized Data Volume - DSS - All

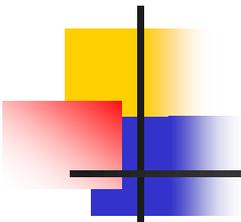
<i>Rank</i>	<i>Organization</i>	<i>Normalized Size (Gb)</i>	<i>DBMS</i>	<i>Server Vendor</i>	<i>Storage Vendor</i>
1	AT&T	94,305	Daytona	Sun	Sun
2	Amazon.com	34,219	Oracle	HP	HP
3	France Telecom	29,735	Oracle	HP	HP
4	Health Insurance Review Agency	29,299	Sybase IQ	HP	Hitachi
5	Barclays Bank	24,756	Teradata	NCR	LSI
6	FedEx Services	14,745	Teradata	NCR	EMC
7	Samsung Card.	14,567	Sybase IQ	HP	HP
8	Kmart	13,874	Teradata	NCR	LSI
9	Cho-Hung Bank	12,350	Sybase IQ	Sun	Hitachi
10	LG Card	12,313	Sybase IQ	Sun	EMC



VLDB'2005 survey (taille)

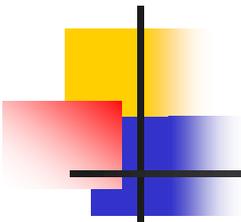
Norm. Data Volume, All Environments, DW *

Company/Organization	Norm. Data Volume (GB)	DBMS	Platform	Architecture	DBMS Vendor	System Vendor	Storage Vendor
AT&T	330,644	Daytona	UNIX	Federated/SMP	AT&T	HP	HP
AT&T	93,468	Daytona	UNIX	Federated/SMP	AT&T	Sun	Sun
Amazon.com	28,184	Oracle RAC	Linux	Centralized/Cluster	Oracle	HP	HP
Nielsen Media Research	17,969	Sybase IQ	UNIX	Centralized/SMP	Sybase	Sun	EMC
Yahoo!	17,014	Oracle	UNIX	Centralized/SMP	Oracle	Fujitsu Siemens	EMC
Amazon.com	14,849	Oracle RAC	Linux	Centralized/Cluster	Oracle	HP	HP
UBS AG	14,177	Oracle	UNIX	Centralized/SMP	Oracle	Sun	EMC
China Telecom Corporation Co.,Ltd. GuangZhou Research Institute	13,241	Sybase IQ	UNIX	Centralized/SMP	Sybase	Sun	Sun
USDA	11,876	SQL Server	Windows	Centralized/SMP	Microsoft	HP	EMC
Reliance Infocomm Ltd	11,500	Oracle	UNIX	Centralized/SMP	Oracle	Sun	EMC
Cellcom	10,345	Oracle RAC	UNIX	Centralized/Cluster	Oracle	HP	EMC



VLDB'98 survey (nb de tuples)

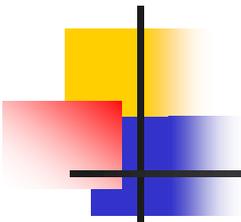
Rank	Organization	DBMS	Processor	Architecture	Storage	Rows/ Records (m)
1	UPS - PLD Repository (federated)	DB2	IBM S/390. Hitachi Skyline	SMP	EMC, IBM	324,000
1	TELSTRA	DB2	Hitachi Skyline	Cluster	HDS, EMC, IBM	51,000
2	Experian - Credit Reporting	DB2	Hitachi Skyline 62	SMP	EMC	15,018
3	SK Telecom	DB2	IBM S/390	MPP	HDS, IBM	5,870
4	Caixa Economica Federal	CA-IDMS	IBM S/390	SMP	EMC, HDS, IBM	5,000
5	Deere & Company	DB2	IBM RS/6000	SMP	IBM	2,503
6	Metromail Corporation	Oracle	Sequent Symmetry 5000	SMP	EMC	2,500
7	IZB Software	Adabas	Amdahl 5995	SMP	EMC, Comparex	1,900
8	DBS Systems Corp.	Oracle RDB	Digital VAX7830	MPP	DEC	915
9	Victoria's Secret Catalogue	DB2	Amdahl	SMP	IBM, EMC	800
10	CentreLink	CCA Model 204	IBM S/390	Cluster	IBM, EMC	734



VLDB'2003 survey (nb de tuples)

Number of Rows - DSS - All

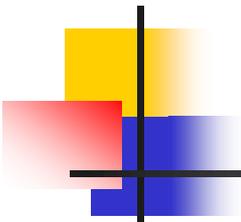
<i>Rank</i>	<i>Organization</i>	<i>Number of Rows (Millions)</i>	<i>DBMS</i>	<i>Server Vendor</i>	<i>Storage Vendor</i>
1	AT&T	496,041	Daytona	Sun	Sun
2	Nielsen Media Research	250,918	Sybase IQ	Sun	Sun
3	France Telecom	156,787	Oracle	HP	HP
4	SBC	144,416	Teradata	NCR	LSI
5	Kmart	133,079	Teradata	NCR	LSI
6	FedEx Services	89,342	Teradata	NCR	EMC
7	Anonymous	77,837	DB2 for Unix	IBM	IBM
8	LG Card	74,702	Sybase IQ	Sun	EMC
9	Health Insurance Review Agency	72,092	Sybase IQ	HP	Hitachi
10	AT&T Wireless Services	64,534	Teradata	NCR	NCR



VLDB'2005 survey (nb de tuples)

Number of Rows, All Environments, DW *

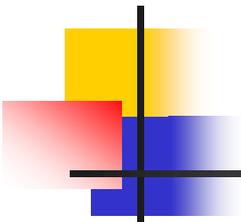
Company/Organization	Number of Rows (Millions)	DBMS	Platform	Architecture	DBMS Vendor	System Vendor	Storage Vendor
Sprint	2,847,553	NonStop SQL	NonStop OS	Centralized/Cluster	HP	HP	HP
AT&T	1,882,638	Daytona	UNIX	Federated/SMP	AT&T	HP	HP
AT&T	533,723	Daytona	UNIX	Federated/SMP	AT&T	Sun	Sun
Nielsen Media Research	502,407	Sybase IQ	UNIX	Centralized/SMP	Sybase	Sun	EMC
Yahoo!	385,318	Oracle	UNIX	Centralized/SMP	Oracle	Fujitsu Siemens	EMC
Turkcell	181,083	Oracle	UNIX	Centralized/SMP	Oracle	Sun	Hitachi
Anonymous	167,173	Sybase IQ	UNIX	Centralized/SMP	Sybase	Sun	EMC
KT IT-Group	136,641	DB2	UNIX	Centralized/Cluster	IBM	IBM	Hitachi
Anonymous	134,880	Sybase IQ	UNIX	Centralized/SMP	Sybase	HP	HP
China Telecom Corporation Co.,Ltd. GuangZhou Research Institute	133,924	Sybase IQ	UNIX	Centralized/SMP	Sybase	Sun	Sun



VLDB'98 survey

(nb de transactions par seconde)

Rank	Organization	DBMS	Processor	Architecture	Storage	TPS
1	Roadway Express Inc., Inc.	Model 204	IBM S/390	SMP	IBM, HDS	1,820
2	TELSTRA	DB2	Hitachi Skyline	Cluster	HDS, EMC, IBM	300
2	Pacific Telecom, Inc.	CA-IDMS	Amdahl 5995 M	SMP	IBM, Spectris, EMC	300
3	CentreLink	CCA Model 204	IBM S/390	Cluster	IBM, EMC	272
4	SK Telecom	DB2	IBM S/390	MPP	HDS, IBM	250
5	UPS - PLD Repository	DB2	IBM S/390. Hitachi Skyline	SMP	EMC, IBM	220
6	Caixa Economica Federal	CA-IDMS	IBM S/390	SMP	EMC, HDS, IBM	205
7	Progressive Corp.	CA-IDMS	IBM S/390	SMP	EMC	140
8	IZB Software	Adabas	Amdahl 5995	SMP	EMC, Comparex	135
9	DBS Systems Corp.	Oracle RDB	Digital VAX7830	MPP	DEC	110
10	Metromail Corporation	Oracle	Sequent Symmetry 5000	SMP	EMC	92



VLDB'2003 survey (nb de transactions par seconde)

Peak Workload - OLTP - All

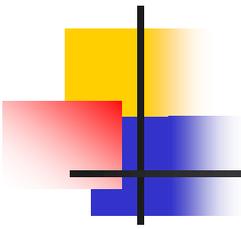
<i>Rank</i>	<i>Organization</i>	<i>Trans. per Second</i>	<i>DBMS</i>	<i>Server Vendor</i>	<i>Storage Vendor</i>
1	Bureau of Customs & Border Protection	51,448	CA-Datacom	IBM	Hitachi
2	Anonymous	4,010	SQL Server	HP	HP
3	Internet Auction Co., Ltd.	3,634	SQL Server	Unisys	EMC
4	Commander Communications Ltd	2,604	SQL Server	Dell	Dell
5	Scottish and Southern Energy plc	2,152	SQL Server	HP	HP
6	2001OUTLET Co. Ltd	1,204	SQL Server	HP	EMC
7	Anonymous	1,200	SQL Server	HP	EMC
8	Caixa Econômica Federal	1,133	CA-IDMS	IBM	EMC
9	CheckFree Corporation	739	DB2 for z/OS	IBM	EMC
10	Land Registry	702	DB2 for z/OS	IBM	Hitachi

VLDB'2005 survey (nb de transactions par heure)

Peak Workload, Mainframe and other, OLTP

Company/Organization	Peak Workload	DBMS	Platform	Architecture	DBMS Vendor	System Vendor	Storage Vendor
UPS	1,134,034,718	DB2	z/OS	Federated/SMP	IBM	IBM	IBM
US Bureau of Customs & Border Protection	340,838,403	CA-Datacom	z/OS	Distributed/SMP	CA	IBM	Hitachi
Experian	202,214,000	DB2	z/OS	Centralized/SMP	IBM	IBM	EMC
State Street Corporation	195,430,140	CA-Datacom	z/OS	Federated/SMP	CA	IBM	EMC
Caixa Econômica Federal	131,847,300	CA-IDMS	z/OS	Centralized/SMP	CA	IBM	EMC
CheckFree Corporation	66,046,711	DB2	z/OS	Centralized/Cluster	IBM	IBM	EMC
LG Credit Card	36,639,038	DB2	z/OS	Centralized/Cluster	IBM	IBM	EMC
Land Registry	6,464,623	DB2	z/OS	Centralized/Cluster	IBM	IBM	IBM
US Department of Treasury/FMS	331,509	CA-Datacom	z/OS	Centralized/SMP	CA	IBM	IBM

315 009 par seconde

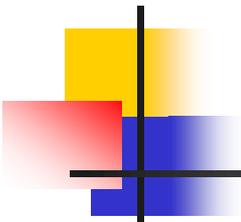


Evolution des BD

■ Entre 1998 et 2005

- Taille : 16 800 Go → 330 644 Go (330 To)
- Nombre de tuples : 324 000 Millions → 2 847 553 Millions
- Nombre de transactions par seconde : 1820 → 315 009

■ Problème de passage à l'échelle (scalability)



Composants logiciels d'une base de données

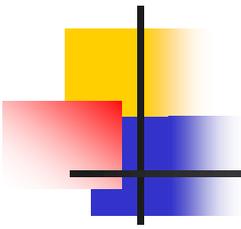
■ SGBD

- Gère le niveau logique et physique de la base
 - Selon l'architecture ANSI-SPARC

■ Les outils frontaux (L4G)

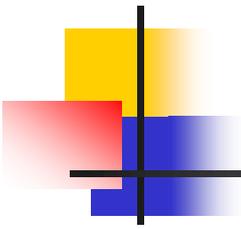
- Générateurs : de formes, de rapports, des applications
- Intégrés au SGBD ou externes
 - Powerbuilder, Borland...
- Interfaces WEB : HTML, XML...
- Interfaces OLAP & Data Mining
 - Intelligent Data Miner (IBM)

■ Utilitaires : chargement, statistiques, aide à la conception...



Composants matériels d'une base de données

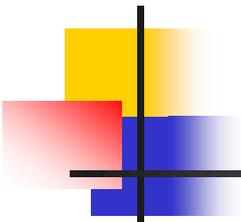
- Ordinateur générique avec
 - son CPU (Central Processing Unit)
 - sa RAM (Random Access Memory)
 - son disque dur pour la BD
 - ses bandes magnétiques, ses graveurs pour la sauvegarde
- La RAM est considéré traditionnellement trop petite pour une BD
 - Ce n'est plus toujours vrai
- Problème d'organisation d'E/S pour une BD



Composants matériels d'une base de données

Machine spécialisée (MBD - Machine BD)

- Ne supporte que la BD
- En général multiprocesseur
 - Teradata massivement parallèle
- Les applications sont sur d'autres ordinateurs
- Liaison par LAN (Local Area Network)



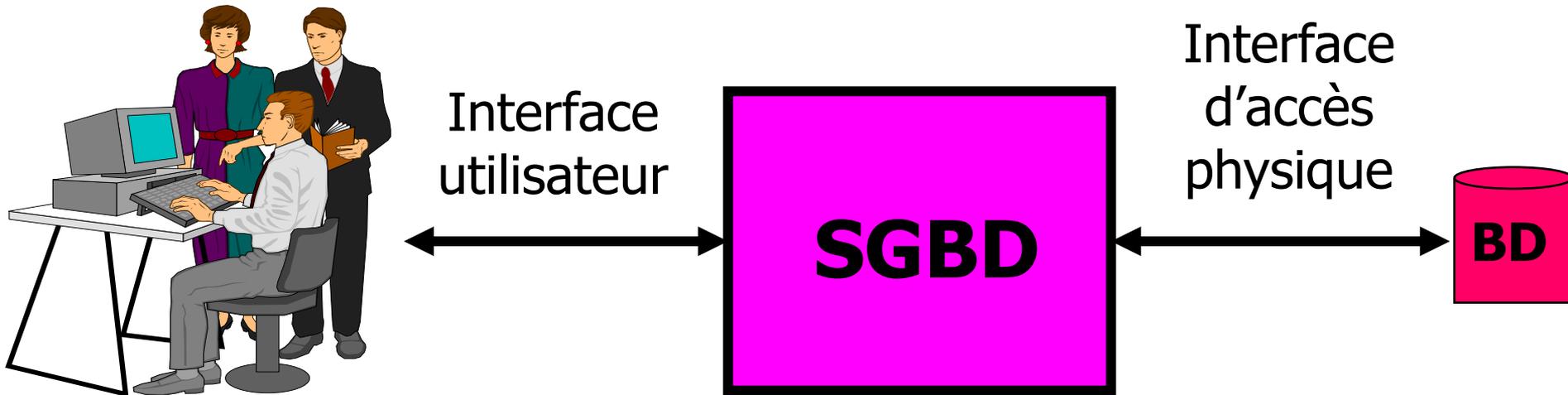
SGBD

SGBD : **S**ystème de **G**estion de **B**ases de **D**onnées

DBMS : **D**ata**B**ase **M**anagement **S**ystem

- Ne pas confondre avec Base de Données
- Logiciel gérant une base de données
- Mono ou multi-ordinateurs
- Peut gérer plusieurs bases de données
- Peut accéder aux BD d'autres SGBD
- Nombreux SGBD sur le marché
 - ACCESS
 - MySQL, PostgreSQL
 - ORACLE
 - DB2
 - SQLSERVER
 - ...

Architecture d'un SGBD



Analyse/vérification des requêtes

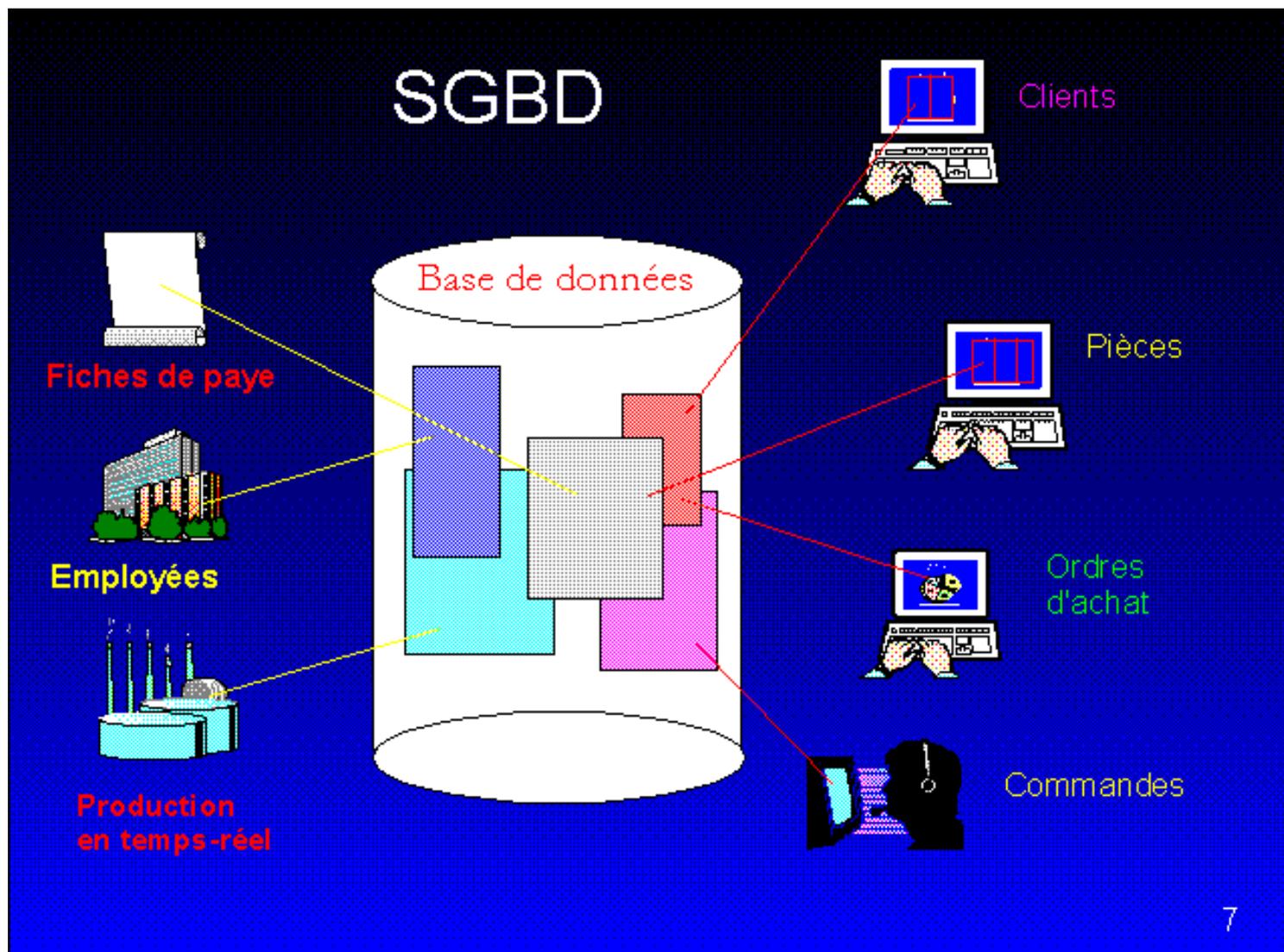
Convivialité de l'interface

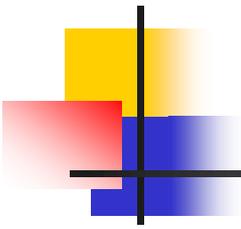
Puissance des langages

Stockage / accès aux données

Optimisation des performances

SGBD

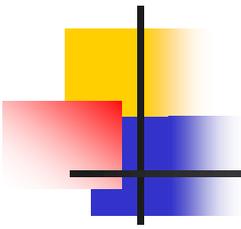




SGBD (par modèle de données)

1ère génération 1950 – 1965

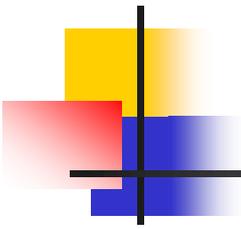
- SGF(Systèmes de Gestion de Fichiers)
- Permettent de stocker et d'organiser les données sur mémoire secondaire
- Différentes organisations
 - séquentielles
 - séquentielles triées
 - indexées
 - séquentielles indexées



SGBD (par modèle de données)

Exemple d'utilisation d'un SGF

- Exemple de la base viticole
 - un fichier par entité : vins, viticulteurs, buveurs
 - un fichier par association : productions, commandes
- Exemple de manipulation :
 - ouvrir le fichier vins
 - ouvrir le fichier productions
 - prendre le premier vin
 - 1. move nv dans var1
 - lire dans productions var1
 - afficher le numéro de producteur
 - prendre le vin suivant et retourner en 1 s'il existe
 - fermer les fichiers



SGBD (par modèle de données)

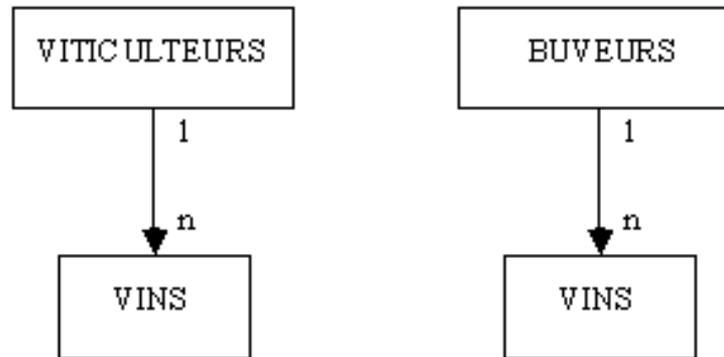
2ème génération 1965 - 1970

- SGBD navigationnel
 - Hiérarchique (IMS)
 - Réseaux (Codasyl, IDS2)
- Principe : relier les données provenant de différents fichiers
- Caractéristiques : liens logiques entre les données
 - hiérarchiques (modèle hiérarchique)
 - hiérarchiques ++ (modèle réseau)
- Liens physiques entre les enregistrements informatiques (pointeurs)

SGBD (par modèle de données)

Exemple d'un SGBD hiérarchique

- Lien de cardinalité 1-n
- Un type de fils ne peut pas avoir plusieurs types de pères

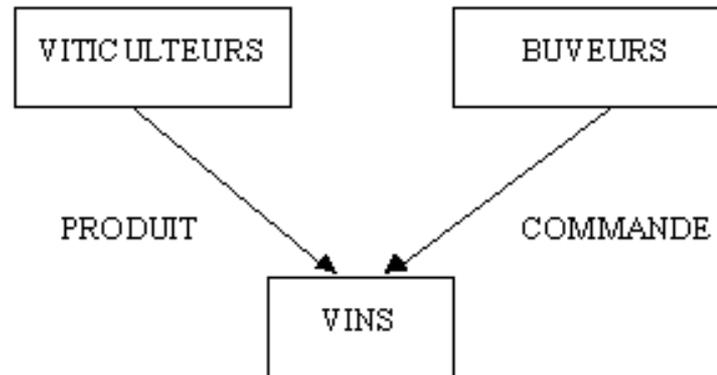


- Association m-n non modélisable
- Duplication de structures de données
- Accès par la racine de la hiérarchie

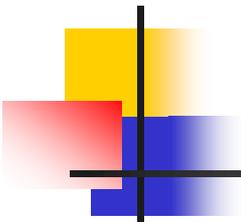
SGBD (par modèle de données)

Exemple d'un SGBD réseau

- Lien de cardinalité 1-n
- Un type de fils peut avoir plusieurs types de pères
- Notion d'articles et d'ensembles
- Association m-n modélisable



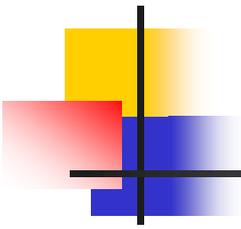
- Pas d'ensemble récursif
- Pas de partage d'objets pour un même ensemble



SGBD (par modèle de données)

3ème génération 1969 - ...

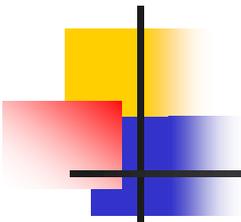
- SGBD relationnel (DB2, Oracle, Informix, MsAccess...)
- Principe :
 - langages assertionnels
 - **quoi au lieu de comment**
- SQL
 - langage de définition de bases de données
 - langage de manipulation de bases de données
 - normes SQL1, SQL2, SQL3



Culture générale

Différents types de langages de programmation

- Programmation impérative : on exprime comment résoudre un problème
- Programmation fonctionnelle : tout est exprimé sous forme de fonctions
- Programmation logique : on exprime le problème mais pas comment le résoudre
- Programmation objet : tout est exprimé sous forme d'objets qui communiquent entre eux pour résoudre un problème
- Déclaratif (quoi) contre Procédural (comment)



SGBD (par modèle de données)

Exemple d'utilisation de SQL

Langage de définition de données :

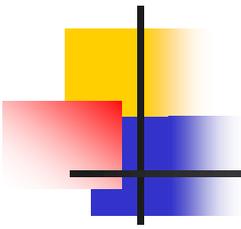
```
create table vins (n°v number, cru
  char(20),
  millesime number, n°vt number)
```

Langage de manipulation de données :

```
select n°v, cru
from vins
where millesime = 1984

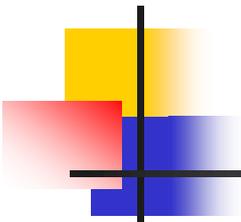
select distinct nom
from viticulteurs, vins, productions
where viticulteurs.n°vt = productions.nvt
and productions.nv = vins.n°v
and cru = 'muscadet'

insert into vins values (8, 'beaujolais', 1981, 4)
```



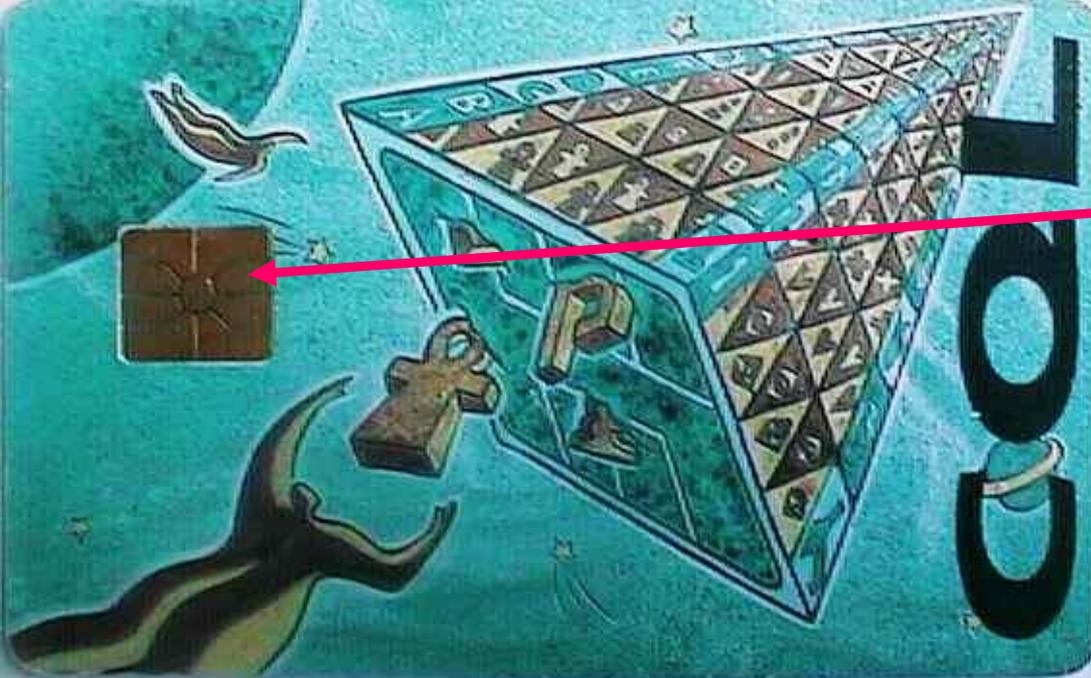
SGBD (par modèle de données)

- SGBD Orienté Objet (1990 - 1999)
- En pratique : une impasse (O2, Objectstore, Objectivity..)
- SGBD relationnel – objet (RO) 1993 - ...
- Évolution probable de tout SGBD relationnel



SGBD (par capacité)

- Mainframe ou cluster de mainframes
 - DB2, Oracle, Adabas, ...
- Machine base de données
 - Teradata
- Serveurs Unix & Windows 2000
 - DB2, Informix, Oracle, Sybase, Unify, SQL Server, Ingres,...
- Personnels
 - MsAccess, Paradox...
- Carte de Crédit
 - CQL



**SGBD Relationnel
et sa Base
Max.: 64 KO**

LA CARTE A PUCE RELATIONNELLE

Avec CQL, GEMPLUS présente la première carte qui gère les données au format relationnel. Utilisant le langage CQL, elle communique avec les serveurs ORACLE de votre système d'information.

ORACLE®



GEMPLUS

**Dossier médical
Personnel
Monnaie
Électronique
....**

Traitement d'un texte écrit dans un langage naturel

- Analyse lexicale

- Quels sont les mots du texte ?

- Analyse syntaxique

- Chacune des phrases a-t-elle été générée par la grammaire ?

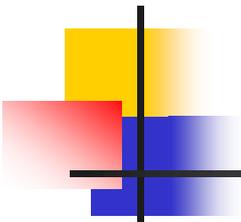
- Analyse sémantique

- Chaque phrase a-t-elle un sens ?

- Analyse pragmatique

- Le texte en entier a-t-il un sens ?

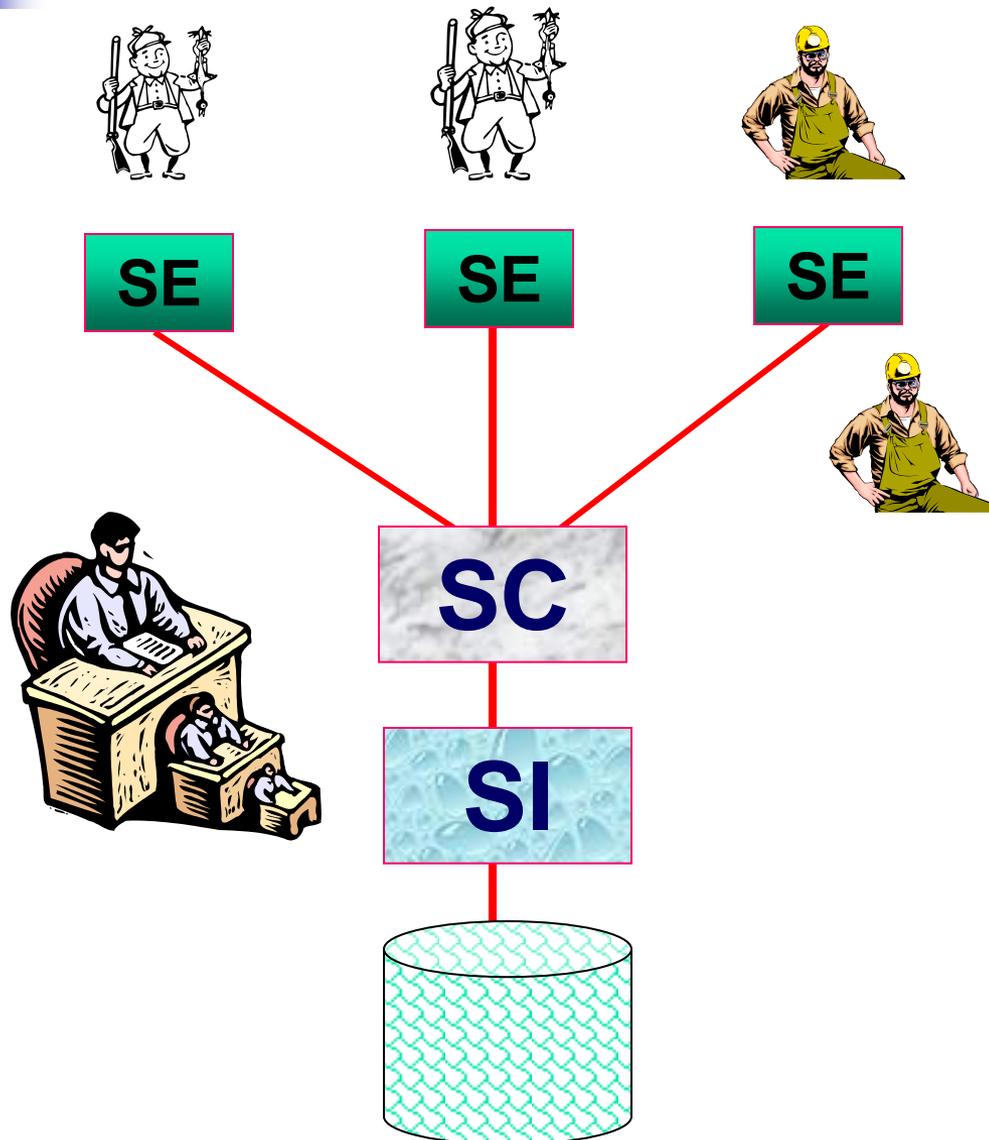
- Domaine de recherche très actif (partie de l'IA)



Principe de fonctionnement d'un SGBD

- Analyse lexicale, syntaxique et sémantique d'une requête (interaction avec la BD)
- Traduction au niveau logique
- Contrôles de confidentialité, concurrence...
- Si la requête est acceptée, optimisation et découpage en sous-requêtes élémentaires transférées au niveau interne
- Au niveau interne, traduction des sous-requêtes en requêtes physiques correspondantes.

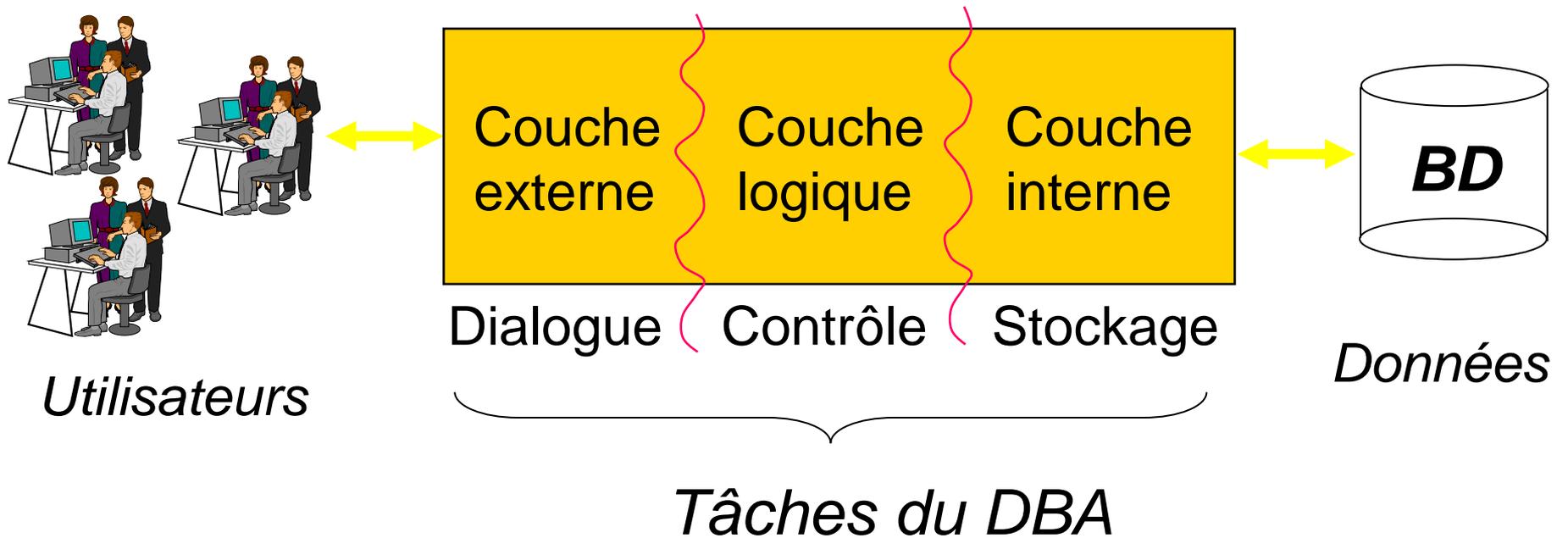
L'architecture ANSI-SPARC

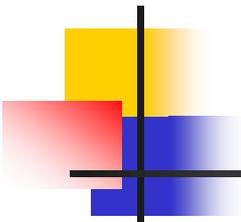


ANSI
Standards
Planning
And
Requirements
Committee

American
National
Standards
Institute

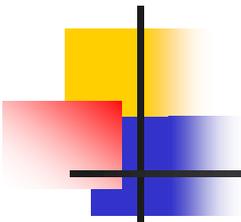
L'architecture ANSI-SPARC





L'architecture ANSI-SPARC

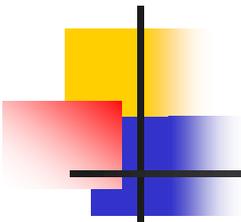
- C'est un standard pour tout SGBD digne de ce nom
- Proposé vers 1965 par Charles Bachman
 - Auteur du concept de la BD
 - Concepteur de IDMS-2
 - Le 1er SGBD moderne
 - Créateur du modèle de données réseau (Codasyl)
 - Plus tard de l'architecture Open System Interconnection (OSI)
 - **Prix Turing** de l'ACM (Association for Computing Machinery)
 - La plus haute récompense scientifique en informatique en USA



Culture générale

Turing award (<http://www.acm.org/awards/taward.html>)

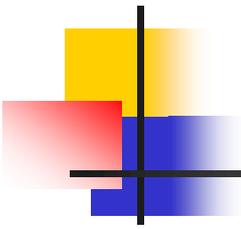
- 1966 A.J. Perlis (techniques de programmation avancées et compilation)
- 1967 Maurice V. Wilkes (inventeur du premier ordinateur stockant un programme)
- 1968 Richard Hamming (codage, détection d'erreur)
- 1969 Marvin Minsky (IA)
- 1970 J.H. Wilkinson (analyse numérique)
- 1971 John McCarthy (IA, LISP)
- 1972 E.W. Dijkstra (Algorithmique)
- 1973 Charles W. Bachman (bases de données)
- 1974 Donald E. Knuth (Algorithmique)
- 1975 Allen Newell (IA)
- 1975 Herbert A. Simon (idem)
- 1976 Michael O. Rabin (automates)
- 1976 Dana S. Scott (automates)
- 1977 John Backus (conception de langage de haut niveau, FORTRAN)
- 1978 Robert W. Floyd (analyse de programmes)
- 1979 Kenneth E. Iverson (langage de haut niveau)
- 1980 C. Antony R. Hoare (conception de langage)
- 1981 Edgar F. Codd (bases de données)
- 1982 Stephen A. Cook (complexité des programmes)
- 1983 Ken Thompson (implémentation d'UNIX)
- 1983 Dennis M. Ritchie (idem)



Culture générale

Turing award (<http://awards.acm.org/homepage.cfm?srt=all&awd=140>)

- 1984 Niklaus Wirth (EULER, ALGOL-W, MODULA, PASCAL)
- 1985 Richard M. Karp (algorithmique, flot dans les réseaux)
- 1986 John Hopcroft (algorithmique et structures de données)
- 1986 Robert Tarjan (idem)
- 1987 John Cocke (compilation, architecture RISC)
- 1988 Ivan Sutherland (info graphique, Sketchpad)
- 1989 William (Velvel) Kahan (analyse numérique)
- 1990 Fernando J. Corbato' (systèmes d'exploitation)
- 1991 Robin Milner (théorie programmation fonctionnelle)
- 1992 Butler W. Lampson (parallélisme)
- 1993 Juris Hartmanis (théorie de la complexité)
- 1993 Richard E. Stearns (idem)
- 1994 Edward Feigenbaum (IA)
- 1994 Raj Reddy (IA)
- 1995 Manuel Blum (théorie de la complexité, cryptographie)
- 1996 Amir Pnueli (logique temporelle)
- 1997 Douglas Engelbart (IHM, souris, <http://sloan.stanford.edu/mousesite/1968Demo.html>)
- 1998 James Gray (bases de données, transactions)



Culture générale

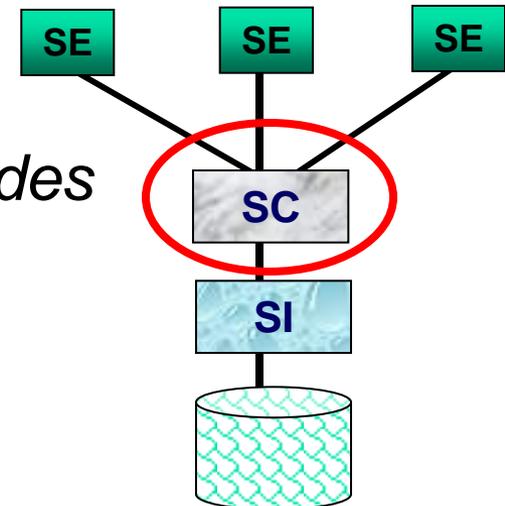
Turing award (<http://awards.acm.org/homepage.cfm?srt=all&awd=140>)

- 1999 Frederick P. Brooks, Jr. (architecture, systèmes d'exploitation, génie logiciel)
- 2000 Andrew Chi-Chih Yao (complexité, cryptographie)
- 2001 Ole-Johan Dahl (programmation objet, simula 67)
- 2001 Kristen Nygaard (idem)
- 2002 Ronald L. Rivest (cryptographie, RSA (1978))
- 2002 Adi Shamir (idem)
- 2002 Leonard M. Adleman (idem)
- 2003 Alan Kay (Smalltalk)
- 2004 Vinton G. Cerf (Internet _ TCP/IP)
- 2004 Robert E. Kahn (idem)
- 2005 Peter Naur (Langages, Algol 60, Compilation)
- 2006 Frances E. Allen (optimisation de compilateurs)
- 2007 Edmund M. Clarke, E. Allen Emerson et Joseph Sifakis (Model checking logiciels et matériels)
- 2008 Barbara Liskov (techniques fondamentales de programmation : abstraction, tolérance aux fautes, informatique distribuée)
- 2009 Charles P. Thacker (pour son travail chez Xerox)
- 2010 Leslie Valiant (Apprentissage automatique, théorie de la complexité, calcul distribué et parallèle)

L'architecture ANSI-SPARC

Le Schéma conceptuel

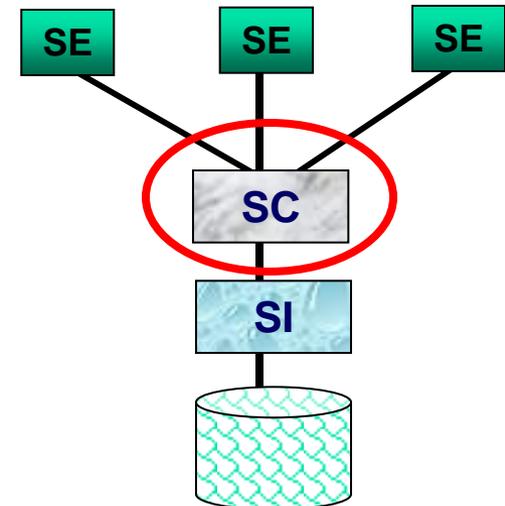
- D'une manière abstraite: un *modèle* conceptuel de l'univers réel de la BD
 - dit aussi *entreprise*
 - L'université de Saint-Etienne
 - La FNAC
 - Mes boissons préférées
 - L'Univers
- L'analyse du réel est le domaine des *méthodes de conception* de la BD



L'architecture ANSI-SPARC

Le schéma conceptuel

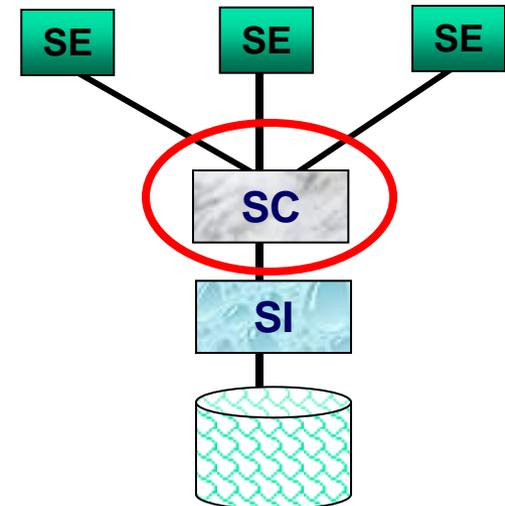
- D'une manière appliquée : la définition logique de la BD
 - Une et une seule
 - Les données logiques, leurs structures et types
 - Relations, attributs, domaines
 - Entités...
 - Objets, Types, Classes
 - Leur manipulations
 - procédures, fonctions, méthodes...

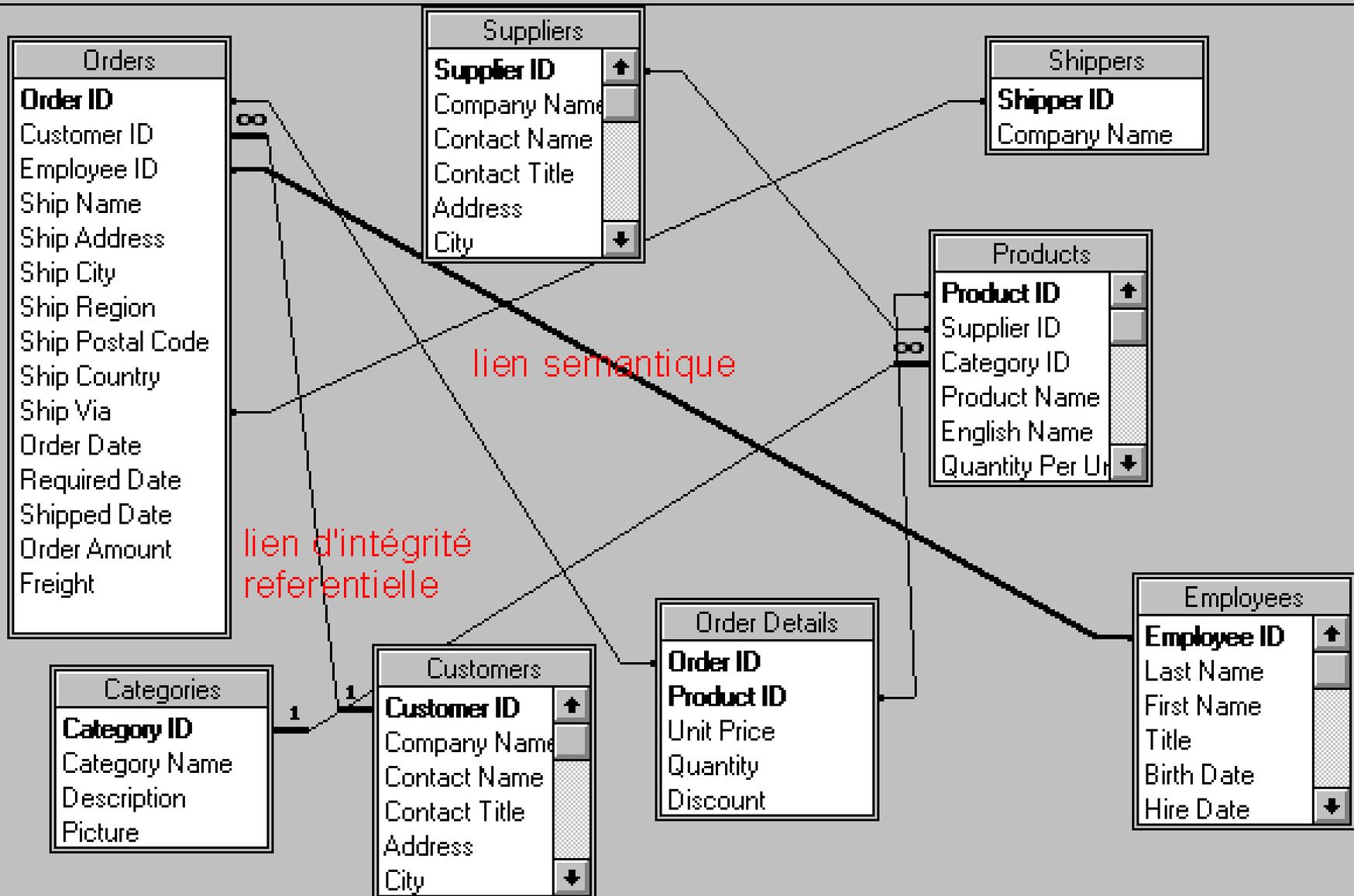


L'architecture ANSI-SPARC

Le Schéma conceptuel

- La définition logique de la BD (suite)
 - Les liens sémantiques
 - Données d'un même objet réel
 - Les contraintes d'intégrité
 - Mono-valeur
 - Référentielles
 - ...
 - Les contraintes de sécurité
 - Qui peut manipuler quoi

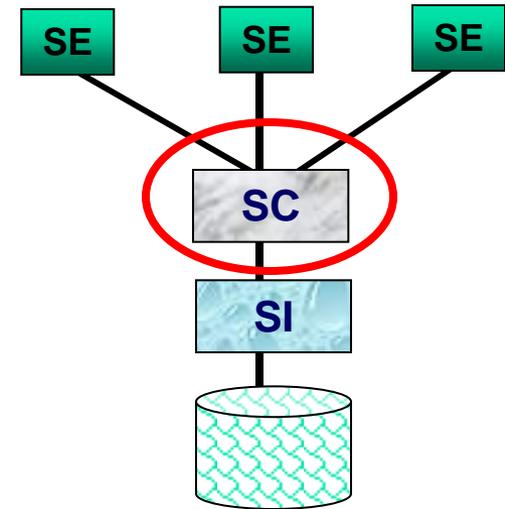




L'architecture ANSI-SPARC

Le schéma conceptuel

- Le schéma conceptuel est défini par le DBA seul
 - Complexité
 - Sécurité
- Par **l'intégration** de données des différentes applications de la BD
 - Plusieurs méthodes de conception, plus ou moins formelles
 - Entité-Relations
 - Objets & Fonctions
 - Merise
 - Normalisation relationnelle



L'architecture ANSI-SPARC

Le schéma conceptuel

- La BD (donc le SC) est définie en utilisant :
 - Le *langage de définition de données (LDD)*
 - La BD est manipulée au niveau du SC à travers le *langage de manipulation de données (LMD)*
- Les deux sous-langages forment:
 - Le *langage de base de données*
 - En général incomplet au sens de la machine de Turing
 - SQL pour une BD relationnelle

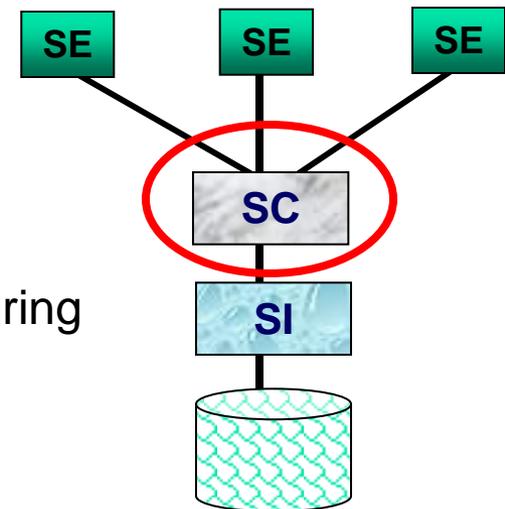
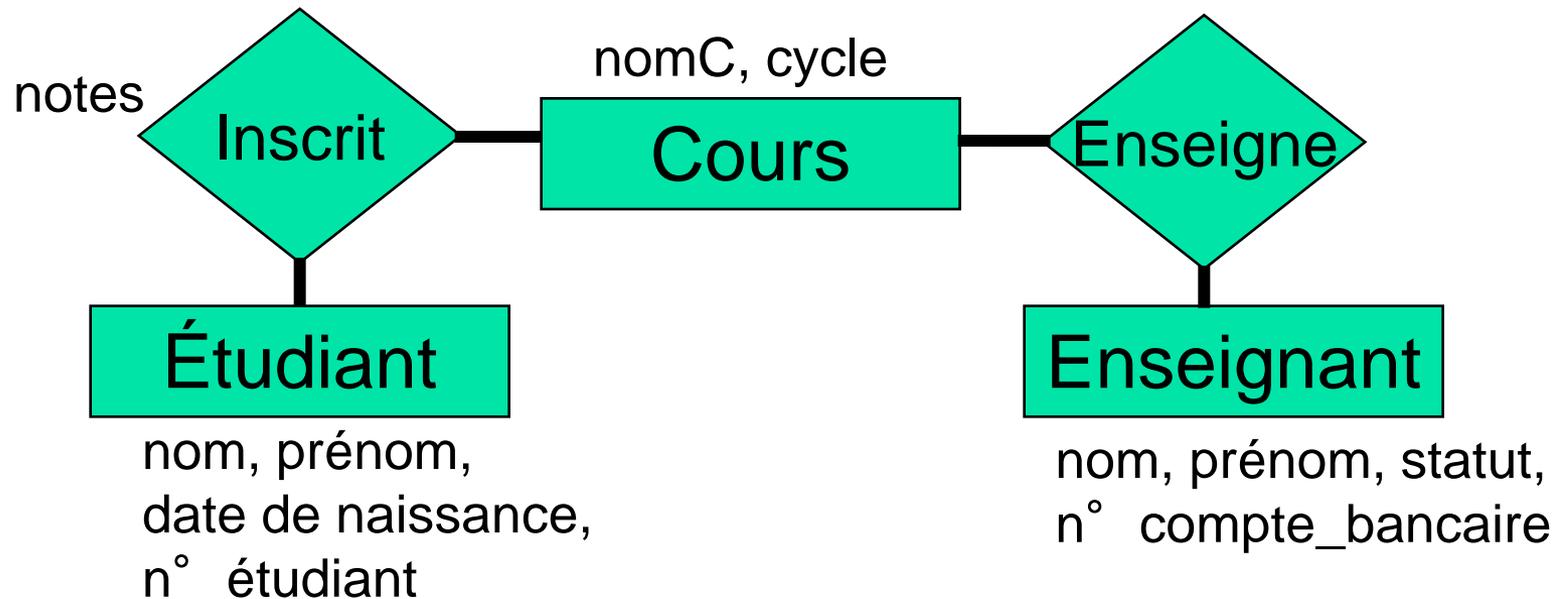


Schéma Conceptuel - exemple

Un institut de formation permanente

Schéma conceptuel (SC) entité-association :



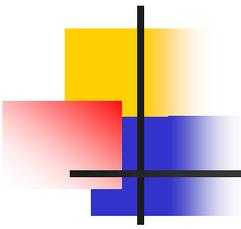


Schéma Conceptuel - exemple

- Traduction du **schéma conceptuel** en un **schéma logique** dans les concepts du modèle utilisé par le SGBD choisi
- On appelle **modèle logique** le modèle sur lequel est construit un SGBD.
 - relationnel
 - objet ...
- ***Attention*** aux confusions de terminologie entre **schéma logique** et **conceptuel** !!!

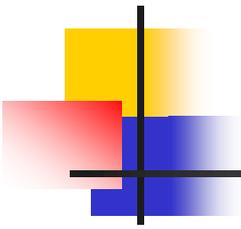


Schéma Conceptuel - exemple

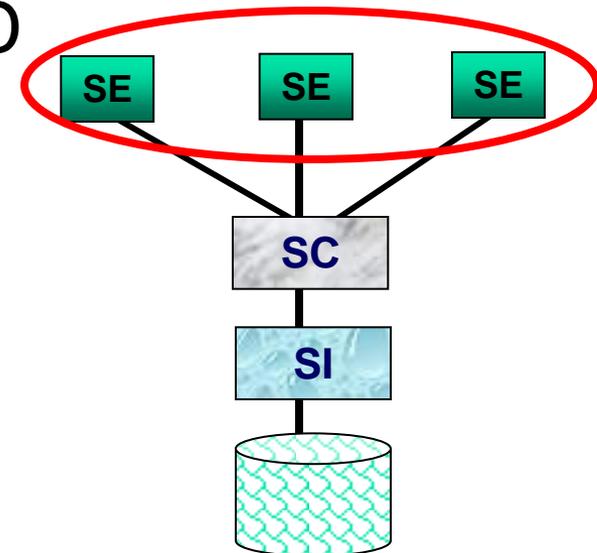
Institut de formation permanente

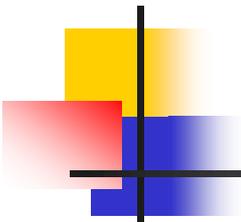
- Schéma logique (SL) relationnel :
 - 4 relations
 - Étudiant : nom, prénom, date_de_naissance, n° étudiant
 - étudiant = nom de la relation
 - nom,prénom,date_de_naissance_n° étudiant = noms d'attributs
 - Enseignant : nom, prénom, statut, n° compte_bancaire
 - Cours : nomC, cycle, nom_enseignant
 - Inscription : n° étudiant, nom_cours, note1, note2

L'architecture ANSI-SPARC

Les schémas externes

- Un SE = un sous-schéma d'une BD
- Dérivé du SC
 - La dérivation est définie dans le SE
- Définit une *vue* de la BD
 - Une sous-base virtuelle
 - En général partielle
 - Pour des usagers de la base
 - Vues L4G, orientés Web notamment (HTML, XML...)
 - Pour des programmeurs d'application
 - Vues SQL, pour SQL imbriqué ou Vbasic, ou procédures stockées...





L'architecture ANSI-SPARC

Les schémas externes

- Une BD est en général munie de plusieurs SE différents
- Mais tous ont le SC comme racine commune
 - Donc tous doivent être accepté par le DBA
 - Perte d'autonomie de l'utilisateur



Form Header

=[First Name] & " " & [Last N

Detail

Employee ID: Employee

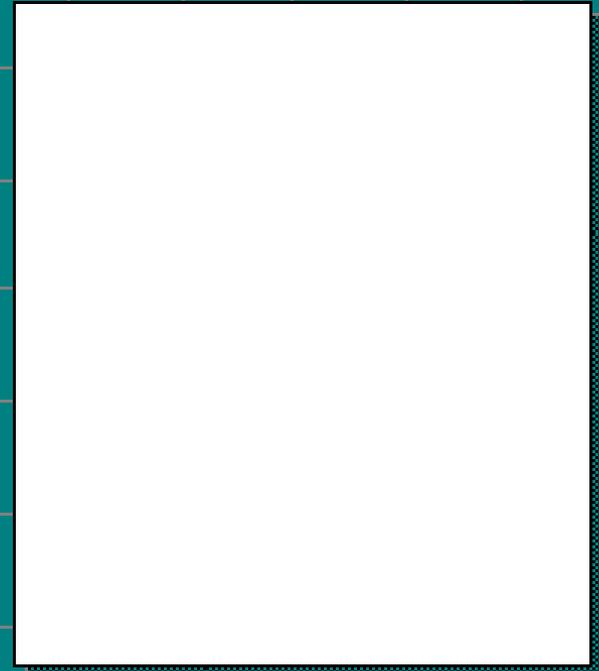
Last Name: Last Name

First Name: First Name

Title: Title

Extension: Extensi

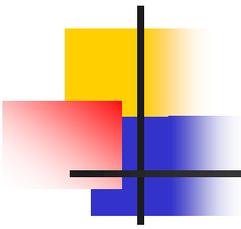
Hire Date: Hire Date



[Personal Info - Next Page]

Photo

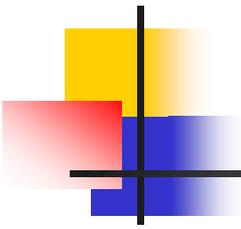
Personal Info:



Schémas externes - exemple

Schéma externe pour le professeur de base de données :

- **Étudiant _BD** : nom, prénom, note1, note2, note_finale
 - tel que
 - **Étudiant _BD** résulte de la combinaison de **Étudiant** et **Inscription** du Schéma Logique, tels qu'il existe une Inscription de cet étudiant pour le cours BD (n° étudiant dans **Étudiant** = n° étudiant dans **Inscription** et nom_cours dans **Inscription** = 'BD'),
 - et tel que $note_finale = (note1 + note2)/2$



Schémas externes - exemple

Schéma externe pour le service de gestion du personnel enseignant :

Professeur : nom, prénom, n° compte_bancaire, nombre_de_cours, liste(nom_cours)

- tel que **Professeur** résulte de la combinaison de *Enseignant* et *Cours* du Schéma Logique,

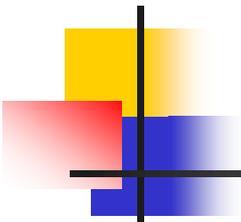
tels que liste(nom_cours) est la liste de nomC

qui se trouvent dans *Cours* tel que nom_enseignant

dans *Cours* = nom dans *Enseignant*, et

tel que

nombre_de_cours = Cardinalité(liste(nom_cours))



L'architecture ANSI-SPARC

Le schéma interne

- Définit la représentation interne de la BD
 - Niveau interne ou *physique*
 - *Les disques, fichiers hachés, arbres-B... contenant la BD*
 - *La représentation physique des valeurs de données*
 - *Réel, entier, texte, ...*
 - *Encodage...*
- Définit l'application du SC sur le SI
 - Selon le principe de l'indépendance des niveaux logique et physique

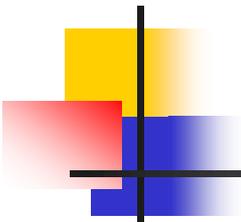
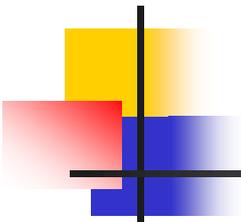


Schéma interne - exemple

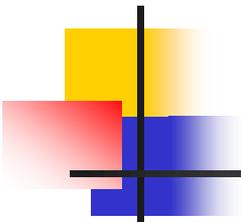
- Étudiant : **fichier** FEtud,
contenu : nom, prénom, date de naissance, n° étudiant
indexé sur n° étudiant,
index secondaire sur nom+prénom
- Enseignant + Cours : **fichier** FEnsCours,
contenu : nom, prénom, statut, n° compte_bancaire, liste(nomC, cycle)
tel que nom_enseignant dans Cours = nom dans Enseignant
indexé sur nom,
deux index secondaires, l'un sur nomC, l'autre sur cycle
- Inscription : **fichier** FInscrits,
contenu : n° étudiant, nom_cours, note1, note2
indexé sur n° étudiant,
index secondaire sur nom_cours



L'architecture ANSI-SPARC

Origine de l'idée

- Eviter les problèmes des SGF de 1ère génération
 - Existence de fichiers multiples et indépendants avec les données
 - dupliquées
 - hétérogènes
 - Difficulté d'assurer
 - les mises à jour cohérentes de duplicata par les applications indépendantes
 - sécurité de l'ensemble des données
- L'existence du niveau conceptuel et physique réels uniques et, par contre, de vues externes multiples mais virtuelles est une réponse théorique à ce besoin.

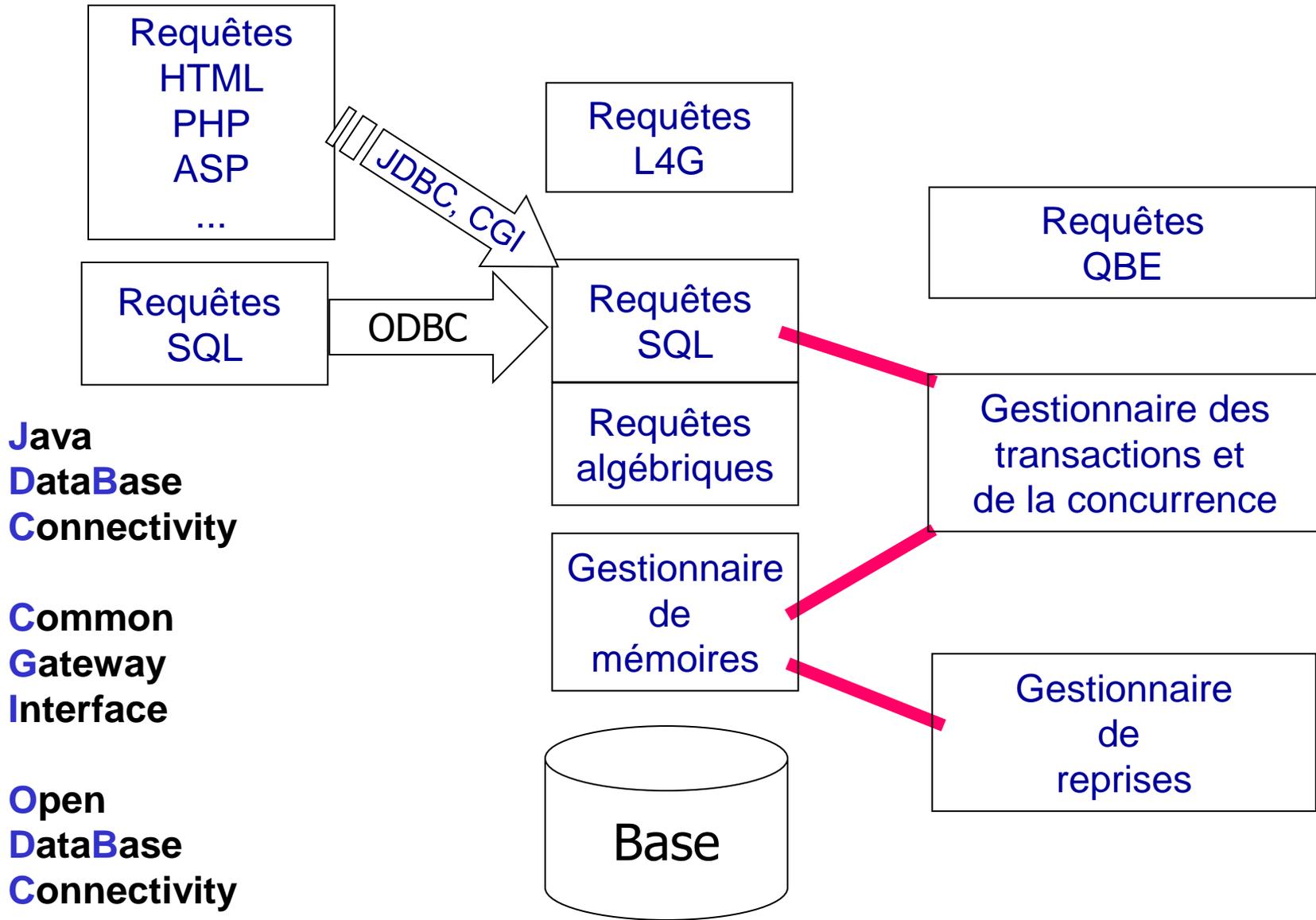


L'architecture ANSI-SPARC

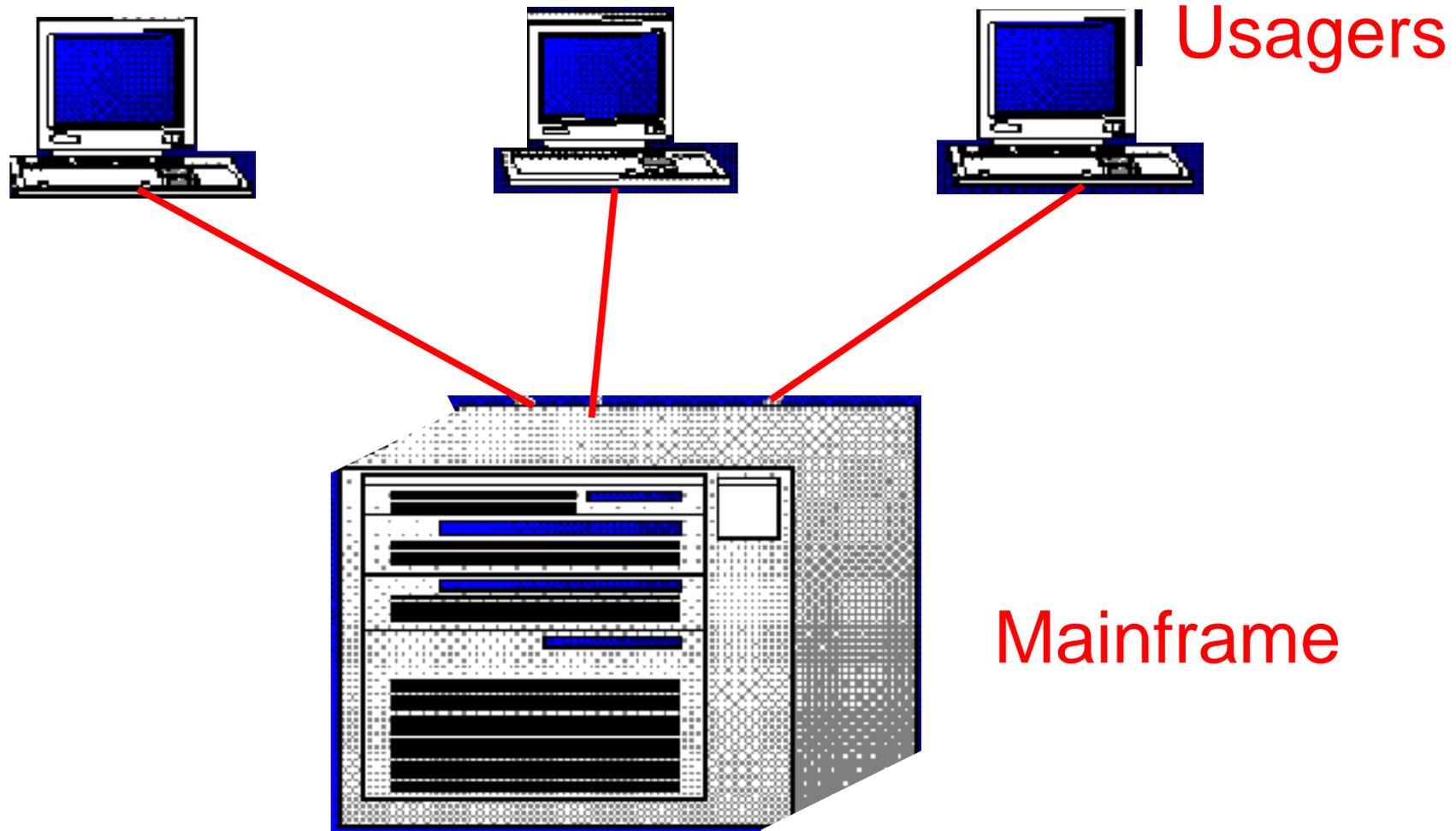
Limitations

- Notion d'une base unique au niveau d'une entreprise
 - en général impossible en pratique
 - complexité organisationnelle
 - complexité technique
- Imprécisions de concepts fondamentaux
 - entreprise, BD, ...
- Notion de vue (schéma externe)
 - limites théoriques sur la mise à jour des vues
- Distinction entre les langages de définition et de manipulation
 - une vue est souvent définie par un langage de manipulation
 - les objets sont définis par des manipulations qu'ils supportent
- Néanmoins l'architecture ANSI-SPARC était une idée capitale pour l'informatique moderne

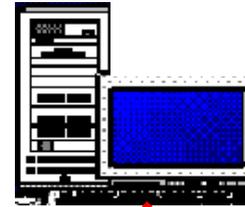
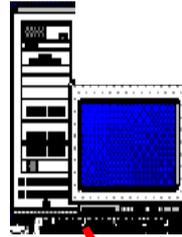
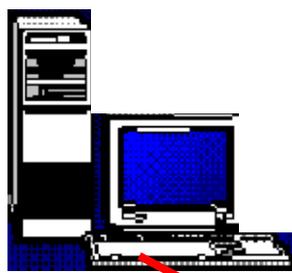
Architecture fonctionnelle d'un SGBD



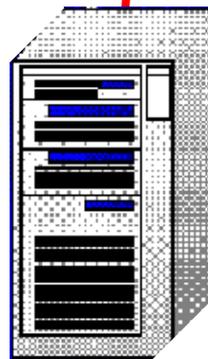
Architecture BD centralisée



Architecture client-serveur

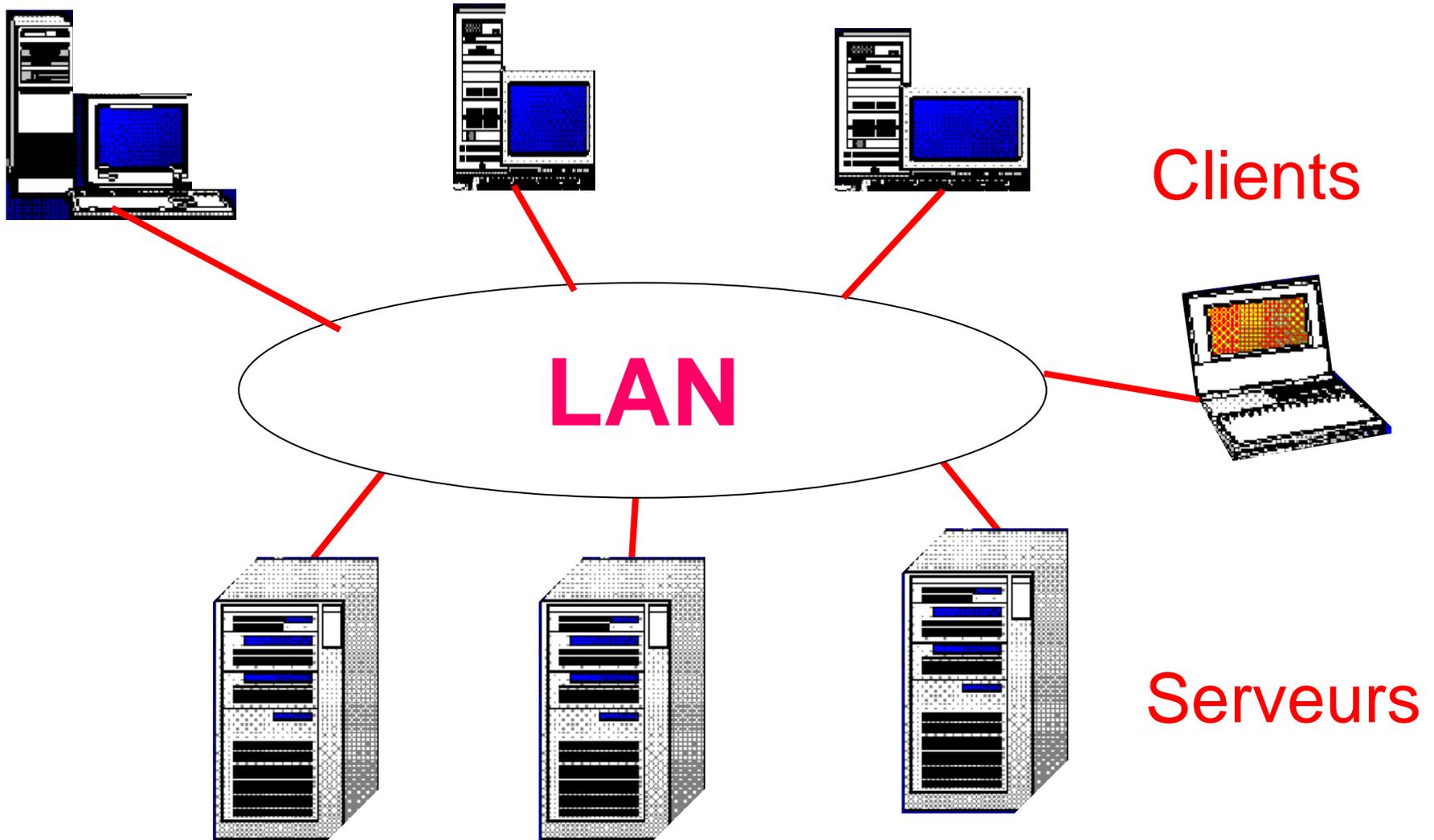


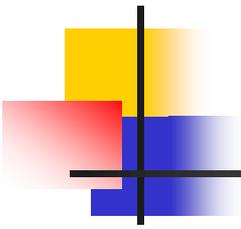
Clients



Serveur

Architecture BDR/BDP/MDB





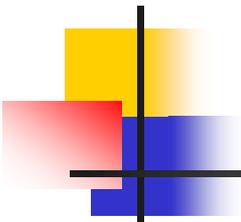
Architecture BDR/BDP/MDB

■ Base de Données Répartie (BDR)

- Une BD avec les fragments dans les SGBDs sur différents sites d'un réseau
- Si c'est un réseau local et la répartition est une partition au niveau physique, alors c'est une BD *parallèle* (BDP)

■ Multibase

- Une collection de bases de données munie d'un langage multibase
- Sur différents sites ou le même
- On les appelle aussi BDs *intéropérables*
- En général ces BDs sont autonomes et hétérogènes



Architecture BDR/BDP/MDB

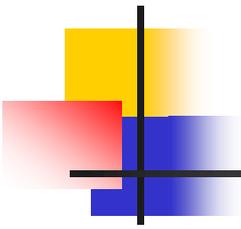
■ Fragmentation type BDP

- par hachage statique
- par intervalles pré-définis d'un attribut ordonné
 - Serveur 1 : recherche des villes de A à C
 - Serveur 2 : recherche des villes de D à G, etc

■ Fragmentation type BDR

- par prédicat SQL
 - **Serveur 1 :**
 - recherche des hôtels de paris de deux étoiles
 - **Serveur 2 :**
 - recherche des hôtels de paris de plus de deux étoiles

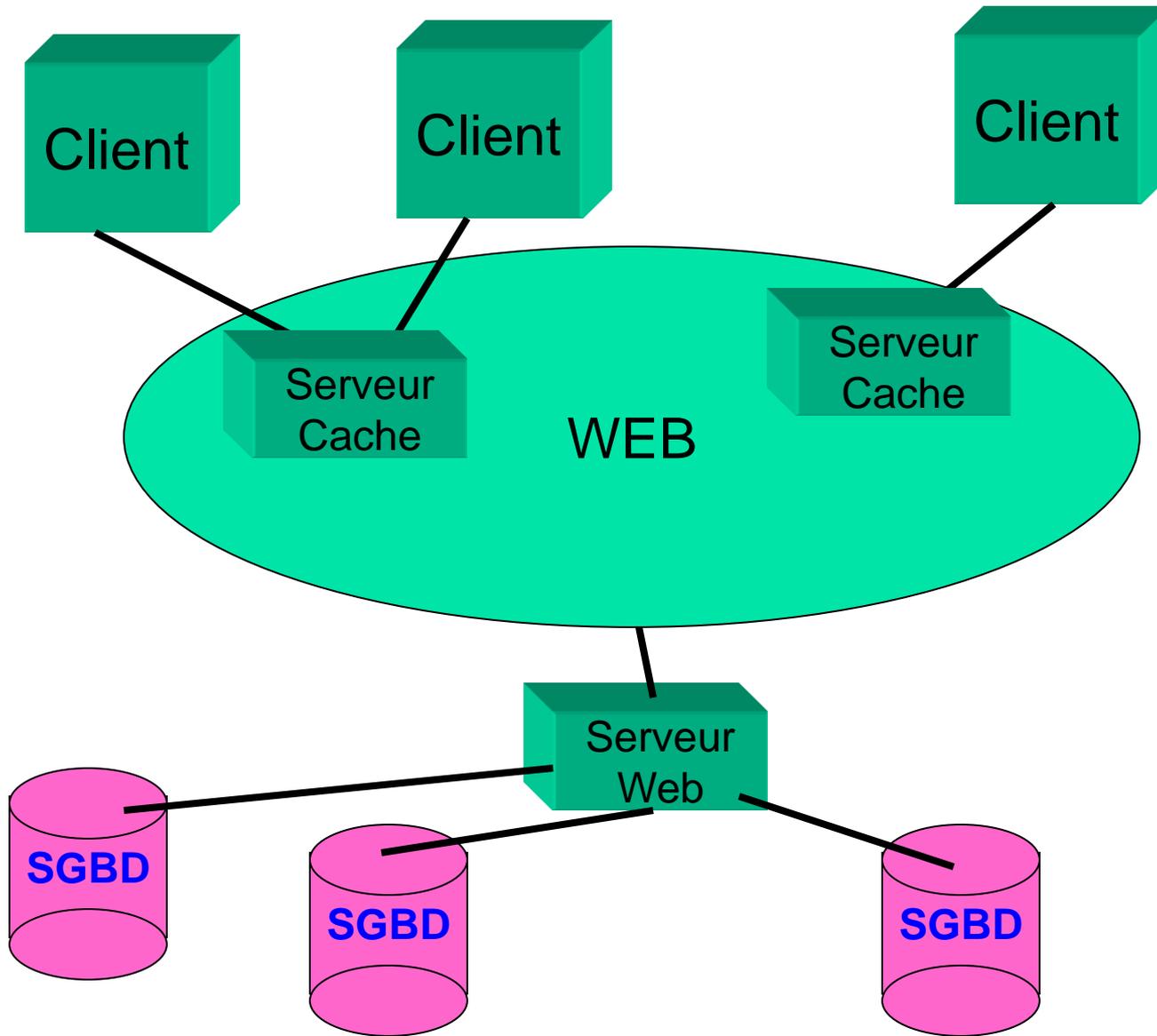
- **Fragmentation BDR est plus générale mais moins courante**

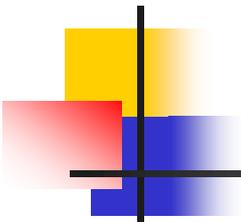


Architecture BDR/BDP - exemples

- SQL Server
- Sybase
- DB2
- Informix
- Oracle

Architecture Web DB





Architecture Web DB

■ Serveur Cache

- met en cache les pages souvent utilisées
 - Exemple : serveur squid

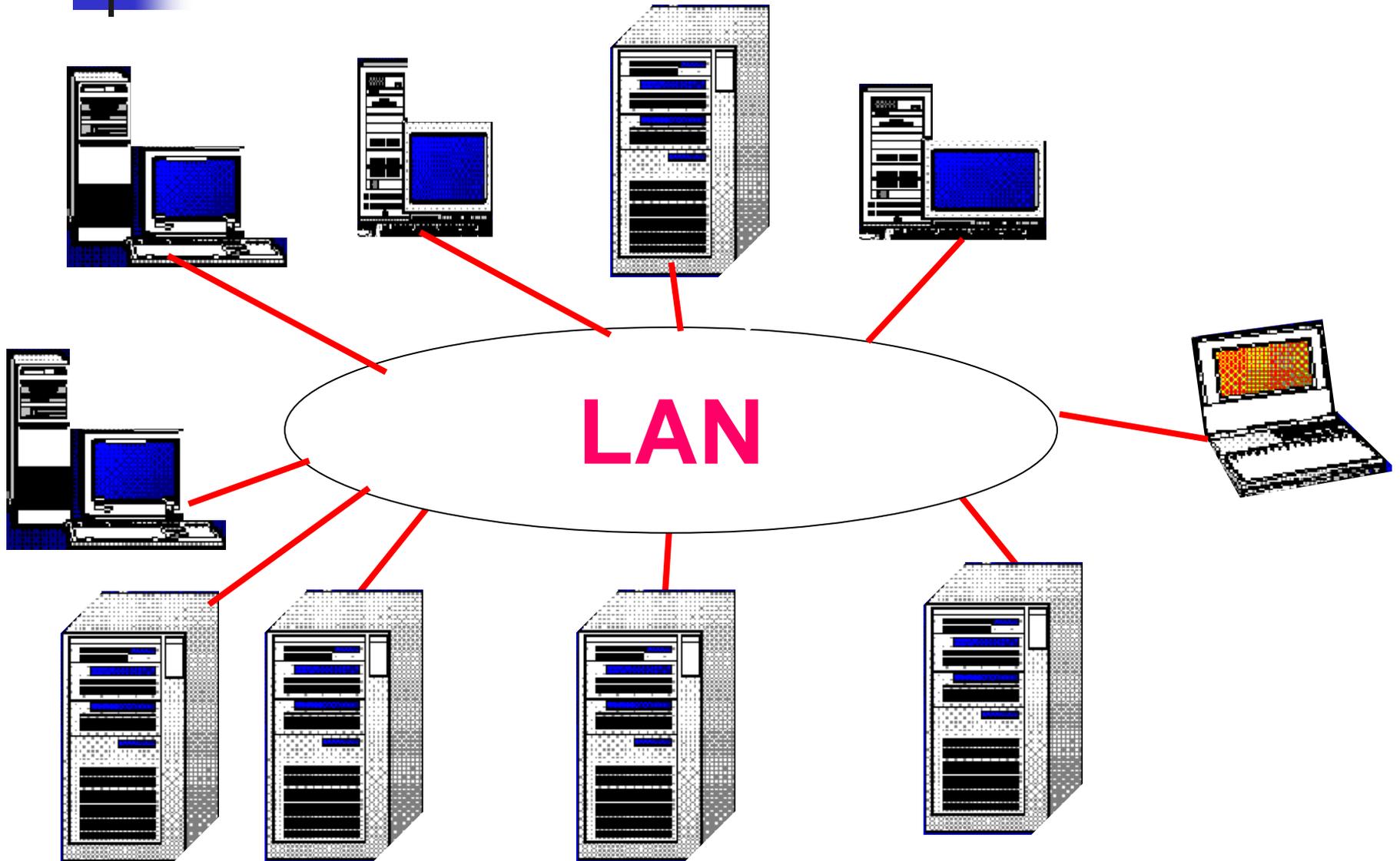
■ Serveur Web

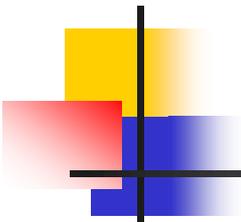
- gère les données en HTML, XML, PHP, etc
- transforme les données et les requêtes CGI \Leftrightarrow SQL
- distribue la charge

■ SGBD

- contient la base
 - DB2, Oracle, SQL Server...

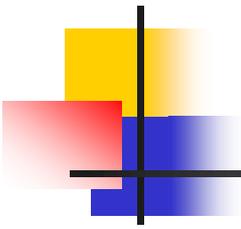
Architecture BDS (scalable)





Multiordinateur réseau

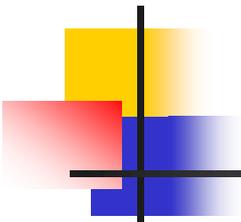
- Peut contenir des milliers de sites
- La BD utilise les ressources cumulées
 - Go de RAM, To de disques durs
- Les données et les traitements de la BD se répartissent sur autant de sites qu'il faut
- Le nombre de sites serveurs de la BD peut croître dynamiquement avec sa taille
 - D'une manière transparente pour les applications
 - La BD devient *scalable* (BDS)



Cycle de vie d'une base de données

4 phases :

- Conception de la base (-> SC)
- Implantation des données (-> SL, SI, population)
- Utilisation (interrogation, mises à jour)
développement des programmes d'application
(-> SEs)
- Maintenance (correction, évolution)

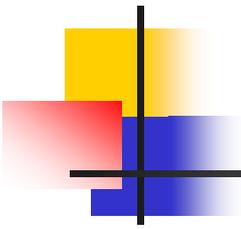


Quelle sera la suite ?

- Quelle théorie sous-jacente ? **algèbre relationnelle**
- Comment parler BD avec la machine ? **SQL**
- Un SGBD convivial : **ACCESS**
- Le centre du modèle ANSI-SPARC :
 - modèle conceptuel : **modèle entité association**
 - modèle logique : **modèle relationnel**
- Comment faire de belles BDs ? **normalisation**

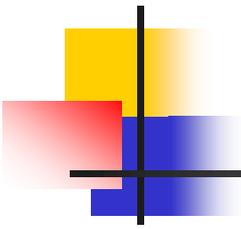
Comme promis

RECAPITULONS



Récapitulons

- Exemples de bases de données

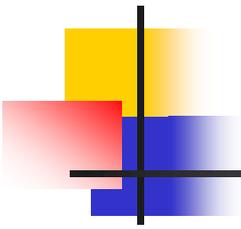


Récapitulons

- Définition du concept de BD

**ensemble cohérent,
intégré
partagé
de données structurées**

défini pour les besoins d'une application



Récapitulons

- Les principales possibilités d'un SGBD

définir une représentation des informations apte à

stocker, interroger et manipuler
(insérer, supprimer, mettre à jour) de

grandes quantités de données (plus que la mémoire vive)

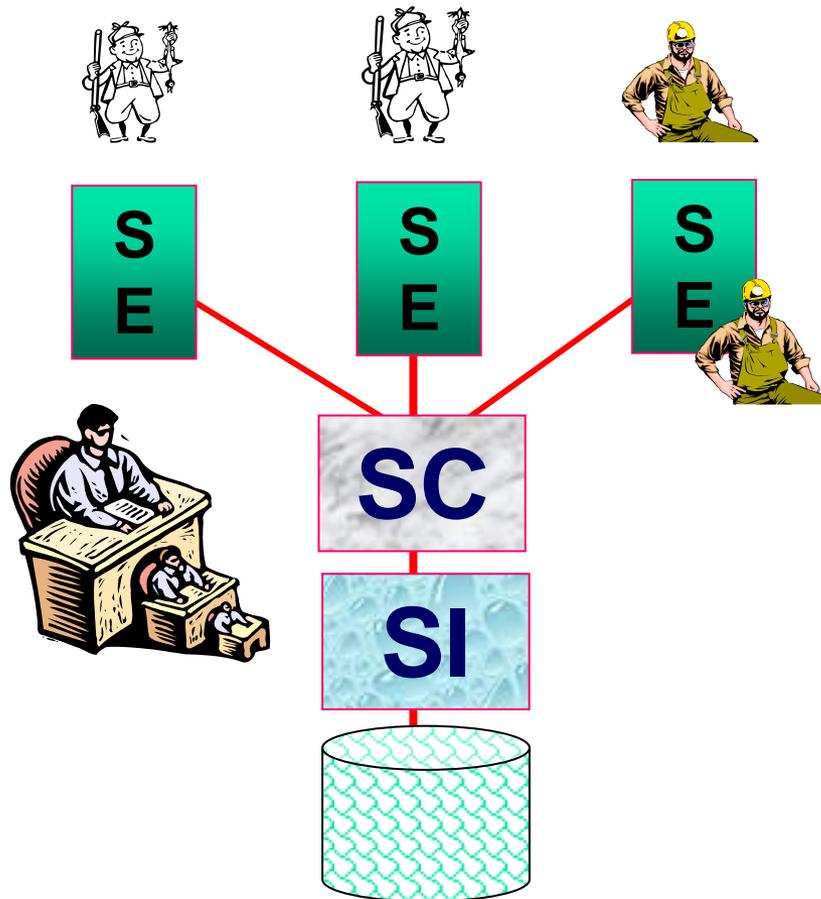
dont il faut garantir la longévité et

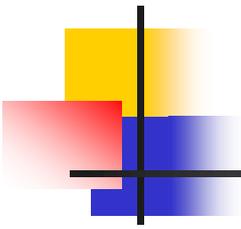
l'accessibilité de manière concurrente
(plusieurs utilisateurs simultanés)

et sûre.

Récapitulons

- Les principes de l'architecture ANSI-SPARC





Récapitulons

- Les diverses architectures de bases de données

Bases de données centralisées

Bases de données client-serveur

Base de données réparties

Bases de données parallèles

Multibase