



# **Les systèmes de gestion de base de données et Windev®**

*par Cyril Beaussier*

## Sommaire

---

<b>Introduction</b>	<b>3</b>
<b>Les bases de données classiques</b>	<b>3</b>
<b>Les bases de données multimédias</b>	<b>3</b>
<b>Définition</b>	<b>3</b>
<b>La base de données</b>	<b>3</b>
<b>Le SGBD</b>	<b>3</b>
<b>Préambule</b>	<b>4</b>
<b>Conception</b>	<b>4</b>
<b>Schéma de données</b>	<b>4</b>
<b>Opération sur les données</b>	<b>4</b>
<b>Concurrence d'accès</b>	<b>4</b>
<b>Le langage SQL</b>	<b>5</b>
<b>Principe</b>	<b>5</b>
<b>La normalisation SQL</b>	<b>5</b>
<b>Vocabulaire</b>	<b>5</b>
<b>Les mots réservés</b>	<b>5</b>
<b>Expression simple sur une table</b>	<b>5</b>
<b>Expression simple sur plusieurs tables</b>	<b>6</b>
<b>Expression complexe sur une table</b>	<b>7</b>
<b>Expression complexe ou imbriquée</b>	<b>7</b>
<b>Expression avec fonctions de calcul</b>	<b>7</b>
<b>Expression d'agrégation</b>	<b>8</b>
<b>Expression d'écriture</b>	<b>9</b>
<b>Optimisation du SGBD</b>	<b>9</b>
<b>Les index</b>	<b>10</b>
<b>Les vues</b>	<b>10</b>
<b>Limites de l'optimisation</b>	<b>10</b>
<b>Critère de choix d'un SGBD</b>	<b>11</b>
<b>Remerciement</b>	<b>11</b>
<b>Copyright</b>	<b>11</b>

## Introduction

L'informatique et les systèmes qui la compose permet de stocker des données représentant des informations. Ces données sont regroupées au sein d'une base appelé ainsi base de données (database). Les exemples d'application utilisant des bases de données sont nombreux. On peut cependant les diviser en deux catégories.

### **Les bases de données classiques**

Au sein desquelles on retrouve les applications les plus courantes : les bases de gestion (salaires, stocks...), les bases transactionnelles (comptes bancaires, centrales d'achats...), les bases de réservations (avions, trains...).

### **Les bases de données multimédias**

Plus récentes, on les retrouve dans les domaines de la documentation, la géographie, le génie logiciel ou la C.A.O. (conception assisté par ordinateur).

## Définition

### **La base de données**

Une base de données est donc un gros ensemble d'informations qui sont structurées et mémorisées sur un support permanent.

Un fichier texte composé de noms est déjà une base de données même si l'accès aux informations est pénible puisqu'il faudra lire à chaque recherche l'ensemble du fichier en le parcourant séquentiellement.

Pour accéder plus rapidement aux informations, on a donc recours à un index. Le format de base de données de Windev, l'Hyper File est un système de ce type en étant composé d'un fichier de données et d'un fichier d'index. On retrouve ce type de base pour l'ISAM, ou le format dBase.

Le système HF, malgré ce que dit son éditeur PC SOFT, a ses limites. Une base de données de ce type mise en réseau face à plusieurs dizaines d'utilisateurs subit une fragmentation de ses données très importante. En effet, si l'accès en lecture ne pose à priori pas de problème, c'est l'accès en écriture (création, suppression ou mise à jour) qui fait risquer à la base une incohérence entre ses données et son indexation. La base est en effet, directement attaqué par le programme, elle n'a pas d'intelligence propre.

### **Le SGBD**

Si la base de données n'est composé que de données, le système de gestion de base de données est en fait un logiciel couplé à une base de données. Un système de gestion de base de données (couramment appelé SGBD) est un logiciel de haut niveau qui permet de manipuler ses informations.

Le système de gestion de base de données est architecturé sur trois niveaux :

1. **Le niveau physique.** C'est ce que fait le SGBD physiquement. La gestion des données et des index. Le partage de ces données et de la concurrence des accès. La reprise sur panne. La distribution des données à travers le réseau.
2. **Le niveau logique.** C'est ce que fait le SGBD logiquement. La définition de la structure des données. La gestion de la confidentialité (sécurité). Le maintien de l'intégrité entre les données et les index. La consultation et la mise à jour des données.
3. **Le niveau externe.** C'est ce que voit l'utilisateur. Il opère sur le SGBD à travers différents outils. Les programmes de saisie ou d'édition d'états. Les outils de programmation et de débogage.

Le système de gestion de base de données va donc permettre de gérer un important volume d'informations pour un grand nombre d'utilisateurs. Ces informations seront persistantes (sur plusieurs années). Elles resteront fiables face à des pannes physique ou logique. Elles seront partageables (tant au niveau utilisateur qu'au niveau programme). Elles pourront être manipulées indépendamment de leur représentation physique.

## Préambule

Ce document est à coupler avec *la programmation en client/serveur avec Windev* disponible sur mon site à l'adresse [www.beaussier.free.fr](http://www.beaussier.free.fr).

Ce document ne rentre pas dans les détails de l'administration des systèmes de gestion de base de données sur serveur dédié de type Oracle par exemple.

## Conception

Il existe de nombreux systèmes de gestion de base de données. Certains peuvent tourner sur micro-ordinateur comme Access, d'autres doivent se placer sur des serveurs dédiés comme *Oracle*, *Sybase*, *DB2* ou *SQL/Server*. La décision du choix dépendra du nombre de transactions qui s'opérera sur le SGBD.

Bien entendu, je conseille fortement de ne pas se lancer sur des SGBD serveur. Sans faire le pro Microsoft, MS-ACCESS représente une bonne alternative qualité/prix. Le moteur du SGBD est performant. La conception du modèle de données est facilité par l'interface graphique.

### **Schéma de données**

C'est la façon de représenter les informations du monde réel dans le système de gestion de base de données. Comme pour la création de l'analyse des données dans Windev, le modèle de données dans Access est similaire avec :

- Une structuration des objets
- Les opérations sur ces objets

### **Opération sur les données**

Il existe quatre types d'opérations classiques. Ces opérations correspondent à des requêtes sur le système de gestion de base de données.

- La création (ou l'insertion)
- La modification (ou la mise à jour)
- La destruction
- La recherche

L'opération la plus complexe est la recherche en raison de la richesse des critères.

### **Concurrence d'accès**

Le système de gestion de base de données doit pouvoir répondre aux requêtes de plusieurs utilisateurs. Ces utilisateurs de leur côté doivent pouvoir accéder en même temps aux mêmes données. Le SGBD doit donc savoir :

- Gérer les conflits si deux utilisateurs font des mises à jour.
- Offrir un mécanisme de retour en arrière si l'utilisateur décide d'annuler des modifications en cours.
- Donner une image cohérente des données si un utilisateur fait des recherches et un autre fait une mise à jour.

Le but final est bien sûr d'éviter les blocages et les lenteurs, tout en empêchant des modifications anarchiques.

## Le langage SQL

Le SQL est l'abréviation de Structured Query Language (traduction pour Langage Structuré de Requête), c'est le mode de communication avec le système de gestion de base de données. Il a été standardisé par l'ANSI et adopté en 1986 par l'ISO/IEC.

### Principe

L'opération de recherche est la plus complexe. L'opération s'exprime par une projection, une sélection et une jointure par un bloc de commande.

**SELECT** < liste des attributs à projeter >  
**FROM** < liste des tables ou des arguments >  
**WHERE** < conditions sur un ou plusieurs attributs >

### La normalisation SQL

Le SQL en est à sa version 2 (ISO 91). Ses principales caractéristiques sont :

- La standardisation des codes réponses en ajoutant la variable de retour sqlState (géré par Windev avec la commande sqlErreur).
- La possibilité de renommer les colonnes résultat
- La possibilité d'utiliser les mots réservés comme nom de table ou d'attribut.

Attention, certains SGBD ont des mots supplémentaires qui ne sont pas standardisé SQLv.2. Il est bien sûr fortement recommandé de ne pas les utiliser.

Là encore Windev reconnaît que son moteur Hyper File n'est pas un véritable SGBD et qu'il n'est absolument pas compatible avec le SQLv.2. Il ne faut donc pas utiliser le langage SQL sur une base HF. Le pilote ODBC fourni pour Hyper File ne peut exécuter que des requêtes de recherche simple (SELECT).

### Vocabulaire

Dans une base de données classique, on parle de fichier, dans un système de gestion de base de données on parle de table. Même chose pour les champs d'un fichier qui sont appelés attributs dans une table. Une table est composé de colonnes, chaque colonne est qualifié par un attribut. Lorsque deux tables sont liés par un attribut, on appelle cela une jointure.

### Les mots réservés

SELECT, FROM, WHERE	Expression de base
ALL, DISTINCT	Permet d'avoir tout ou de supprimer les doublons
=, <>, <, >, <=, >=?	Opérateur de comparaison
BETWEEN, AND, OR	Opérateur d'intervalle
LIKE	Opérateur de comparaison de texte
ALL, ANY, IN, EXISTS	Permet d'opérer sur des sous-requêtes
UNION	Permet de croiser deux requêtes
COUNT, SUM, AVG, MAX, MIN	Fonction de calcul
GROUP BY, HAVING	Permet le regroupement de données
INSERT INTO	Création
UPDATE SET	Mise à jour
DELETE	Suppression

### Expression simple sur une table

Sur le schéma de données d'une gestion des commandes clients d'une société d'outillage, voici quelques requêtes SQL simples juste pour vous familiariser avec le langage. Sachant que le SGBD est composé d'une seule table décrite ci-dessous.

La table `Client` est composée de son numéro (`NumCl`), de son nom (`NomCl`), de son adresse (`AdrCl`), la ville (`VilleCl`) et du chiffre d'affaire qui lui est associé (`CaCl`).

Langage naturel	Expression SQL
Informations sur toutes les clients	<pre>SELECT NumCl, NomCl, AdrCl, CaCl FROM Client ;  OU... SELECT * FROM Client ;</pre>
Le nom des clients dont le C.A. est compris entre 10.000 et 50.000 F	<pre>SELECT NomCl FROM Client WHERE CaCl BETWEEN 10000 AND 50000 ;</pre>
Les nom et adresse des clients dont le nom commence par C	<pre>SELECT NomCl, AdrCl FROM Client WHERE NomCl LIKE 'C%' ;</pre>
Combien de clients dans la table ?	<pre>SELECT COUNT(*) FROM Client ;</pre>
Les noms des clients classés par chiffre d'affaire	<pre>SELECT NomCl FROM Client ORDER BY CaCl ;</pre>

Notons que le symbole étoile (\*) remplace la liste de tous les attributs, que le symbole pourcentage (%) remplace une chaîne de caractères quelconque et que les chaînes de caractères doivent être placées entre cote ('). Enfin le point-virgule (;) signale la fin de la requête.

### Expression simple sur plusieurs tables

Compliquons maintenant en ajoutant une table `Commande` et une table `Produit`. Un client passe une commande composée de produits. Le lien est entre la commande et les produits est une table `Ligne`. Le modèle des données est alors le suivant :



La table `Commande` est composée de son numéro (`NumCo`), du n° du client (`NumCl`) et du total hors taxe (`TotalCo`).

La table `Produit` est composé de son numéro (`NumPr`), de son nom (`NomPr`), de son prix (`PrixPr`) et de la quantité en stock (`QtePr`).

La table `Ligne` est composé des n° de commande (`NumCo`) et de produit (`NumPr`) et de la quantité commandée (`QteCo`).

Langage naturel	Expression SQL
Qui a passé la commande n° 15 ?	<pre>SELECT NomCl FROM Client Cl, Commande Co WHERE Cl.NumCl = Co.NumCl AND NumCo = 15 ;</pre>
Combien de commande a passé le client Martin ?	<pre>SELECT COUNT(*) FROM Client Cl, Commande Co WHERE Cl.NumCl = Co.NumCl AND NomCl = 'MARTIN' ;</pre>
Le nom des produits de la commande n° 24	<pre>SELECT P.NomPr FROM Produit P, Ligne L WHERE P.NumPr = L.NumPr AND NumCo = 24 ;</pre>

Langage naturel	Expression SQL
Les clients ayant commandé des tournevis	<pre>SELECT NomCl FROM Client Cl, Commande Co, Ligne L, Produit P WHERE Cl.NumCl = Co.NumCo AND Co.NumCo = L.NumCo AND L.NumPr = P.NumPr AND NomPr = 'Tournevis' ;</pre>

Dans le cas de jointure entre deux tables, il faut préciser dans la requête les attributs que l'on veut récupérer. Cependant, si les noms des attributs sont identiques, il faut préciser le nom de la table devant. Pour éviter des requêtes trop longues, on peut alors remplacer le nom de la table par un alias. C'est cet alias unique qui se placera devant l'attribut.

### **Expression complexe sur une table**

Langage naturel	Expression SQL
Les noms des clients et leur chiffre d'affaire qui sont supérieur au client n° 143	<pre>SELECT NomCl, CaCl FROM Client C1, Client C2 WHERE C2.NumCl = 143 AND C1.CaCl &gt; C2.CaCl ;</pre>

Revenons au système de l'alias, C1 et C2 représentent deux instances de la même table. La première condition permet d'éliminer les clients qui n'ont pas le n° 143 et la seconde permet d'obtenir le chiffre d'affaire du n° 143 pour le comparer aux autres.

### **Expression complexe ou imbriquée**

Le langage SQL peut aller très loin en croisant le résultat d'une requête avec une autre ou en récupérant le résultat d'une requête pour en refaire une sélection avec l'aide d'une seconde.

Langage naturel	Expression SQL
Les noms des produits non commandés	<pre>SELECT NomPr FROM Produit WHERE NumPr NOT IN (SELECT DISTINCT NumPr FROM Ligne) ;</pre>
Les noms des clients ayant passé au moins une commande	<pre>SELECT NomCl FROM Client WHERE NumCl EXISTS (SELECT NumCl FROM Commande) ;</pre>
Le nom des clients n'ayant jamais commandé	<pre>SELECT NomCl FROM Client WHERE NumCl NOT EXISTS (SELECT NumCl FROM Commande) ;</pre>

Notons que le SGBD va d'abord exécuter la requête se trouvant entre parenthèses. Le résultat sera alors mis en corrélation avec la requête principale tapé au début.

### **Expression avec fonctions de calcul**

Le langage SQL est très souple également puisqu'il peut calculer directement un résultat de requête et le renvoyer. On a vu plus haut que l'on pouvait déjà compter le nombre d'enregistrement par la fonction COUNT, le langage SQL permet également d'additionner (SUM), de faire une moyenne (AVG) ou renvoyer la valeur la plus grande (MAX) ou la plus petite (MIN). Le langage tolère également les opérateurs de comparaison égal (=), supérieur (>), supérieur ou égal (>=), inférieur (<), inférieur ou égal (<=) et différent (<>).

Voyons ci-dessous quelques exemples très simples qui récapitule ce que l'on a vu plus haut afin de vous montrer la souplesse du langage SQL :

Langage naturel	Expression SQL
Nombre de tournevis commandé	SELECT COUNT(DISTINCT NumPr) FROM Produit P, Ligne L WHERE L.NumPr = P.NumPr AND NomPr = 'Tournevis' ;
Quantité total de tournevis commandé	SELECT SUM(QteCo) FROM Ligne L, Produit P WHERE L.NumPr = P.NumPr AND NomPr = 'Tournevis' ;
Chiffre d'affaire moyen des clients	SELECT AVG(CaCl) FROM Client ;
Le plus gros chiffre d'affaire	SELECT MAX(CaCl) FROM Client ;
Le plus petit chiffre d'affaire	SELECT MIN(CaCl) FROM Client ;
Les clients ayant commandé plus que la moyenne des quantités commandées de tournevis	SELECT NomCl FROM Client Cl, CommandeCo, Ligne L WHERE Cl.NumCl = Co.NumCl AND Co.NumCo = L.NumCo AND QteCo > (SELECT AVG(QteCo) FROM Ligne L, Produit P WHERE L.NumPr = P.NumPr AND NomPr = 'Tournevis' ;

### Expression d'agrégation

Langage naturel	Expression SQL
Nombre de clients par ville	SELECT VilleCl, COUNT(NomCl) FROM Client GROUP BY VilleCl ;

Dans cette exemple, nous voyons qu'il est possible de regrouper les résultats d'une requête suivant l'attribut. Voyons ce que fait exactement la requête :

#### Contenu de la base

VilleCl	NomCl
Paris	MARTIN
Paris	HENRI
Paris	CHARLES
Lyon	ROBERT
Lyon	GERARD

#### Résultat de la requête

villeCl	COUNT(NomCl)
Paris	3
Lyon	2

Notons que s'il y a une condition GROUP BY, l'attribut contenu dans celle-ci doit figurer obligatoirement dans la clause SELECT.

Continuons avec d'autres exemples...

Langage naturel	Expression SQL
La quantité moyenne de chaque produit	SELECT NomPr, AVG(QtePr) FROM Produit GROUP BY QtePr ;



Langage naturel	Expression SQL
Le classement des produits par nombre commandé	<pre>SELECT NomPr FROM Produit P, Ligne L WHERE P.NumPr = L.NumPr ORDER BY QteCo GROUP BY NomPr ;</pre>

Compliquons maintenant la requête avec la clause HAVING. Celle-ci permet d'éliminer des partitionnements.

Langage naturel	Expression SQL
Produits commandés par plus de deux clients en quantité supérieure à 5	<pre>SELECT NomPr FROM Produit P, Ligne L, Commande C WHERE P.NumPr = L.NumPr AND C.NumCo = L.NumCo AND QteCo &gt; 5 GROUP BY NomPr HAVING COUNT(NumCl) &gt;= 2 ;</pre>

Prenons de nouveau le contenu de notre base...

Avant la clause HAVING			Après la clause HAVING		
NomPr	NumCl	QteCo	NomPr	NumCl	QteCo
Marteau	12	6	Tournevis	12	6
Tournevis	12	6	Tournevis	18	8
Tournevis	18	8			

### Expression d'écriture

Comme nous l'avons vu plus haut dans le chapitre Opération sur les données. Les requêtes d'écriture à la différence des requêtes de recherche, modifient le contenu de la base. Il existe trois requêtes d'écriture : la création (INSERT), la modification (UPDATE) et la destruction (DELETE). Ces requêtes doivent être manipulées avec précaution car une mauvaise rédaction peut entraîner des catastrophes irréversibles.

Langage naturel	Expression SQL
Ajout du produit Pince au prix de 50 F. et en quantité 10	<pre>INSERT INTO Produit(NomPr, PrixPr, QtePr) VALUES('Pince',50,10) ;</pre>
Modification de l'adresse du client n° 17	<pre>UPDATE Client SET AdrCl = '5 rue Petit', VilleCl = 'PARIS' WHERE NumCl = 17 ;</pre>
Destruction du client dénommé MARTIN	<pre>DELETE FROM Client WHERE NomCl = 'MARTIN' ;</pre>

Notons qu'une requête INSERT ressemble beaucoup à une requête SELECT notamment dans la rédaction des valeurs à ajouter.

Il est très important de préciser la clause WHERE dans les requêtes UPDATE et DELETE car l'oubli ou la mauvaise rédaction des attributs peut causer des dégâts importants. Oublier la clause WHERE dans une requête DELETE, effacera **tous** les enregistrements de la table.

### Optimisation du SGBD

Pour optimiser un système de gestion de base de données, l'administrateur ou le concepteur de la base peut ajouter des index ou des vues sur ces tables afin d'accroître les temps d'accès et de réponse.

## **Les index**

Comme pour une base de données classique, l'index va permettre au SGBD d'accéder plus rapidement à l'enregistrement voulu. Il permet également d'assurer l'unicité des clés primaire ou secondaire. En résumé, un index est une structure contenant l'adresse physique de chaque enregistrement d'une table. Un index permet l'accès direct à l'information.

Que doit-on indexer :

- Toujours indexer les clés primaires des tables dont le nombre dépasse 100 lignes. Une clé primaire est une clé dont le numéro n'aura aucun doublon (exemple : le numéro de client).
- Toujours indexer les attributs intervenants dans des jointures entre tables. Par exemple, le n° de produit que l'on retrouve dans les tables Produit et Ligne.
- Indexer les attributs intervenants dans les clauses ORDER BY, GROUP BY et les fonctions de calcul MIN, MAX, etc.

Que ne doit-on pas indexer :

- Les colonnes contenant peu d'enregistrements
- Les colonnes contenant peu de valeurs différentes. En règle générale, lorsque 20 % ou plus des occurrences d'une satisfait le critère de recherche.
- Les colonnes très souvent modifiés et très peu utilisées lors de recherche.

Pour la création d'index, reportez vous au manuel d'utilisation de Microsoft Access.

## **Les vues**

Une vue est une perception logique sur les données d'une ou de plusieurs tables ou même d'autres vues. Elle est définie à partir d'une requête d'interrogation du langage SQL. Elle hérite des mêmes caractéristiques que les objets auxquels elle se réfère.

Une vue est définie pour :

- Fournir un niveau de sécurité supplémentaire sur les données d'une table. Par exemple, un employé A ne peut consulter les informations d'un employé B dans un logiciel de gestion de personnel (salaire, prime, etc).
- Donner à l'utilisateur l'impression que la base lui appartient.
- Cacher une certaine complexité des données. Une vue pouvant être créer à partir de plusieurs jointures entre des tables différentes.
- Simplifier la formulation de requête SQL.
- Présenter les données d'une table sous un autre aspect sans en modifier la définition.
- Sauvegarder d'une façon indirecte les requêtes complexes.

Les vues ne peuvent être que créer ou remplacer.

Un exemple de vue :

<b>Langage naturel</b>	<b>Expression SQL</b>
Avoir directement les gros clients	<pre>CREATE VIEW GrosClient(NomGc, AdrGc, VilleGc) AS SELECT NomCl, AdrCl, VilleCl FROM Client WHERE CaCl &gt; 100000</pre>

## **Limites de l'optimisation**

Malgré tout la performance du SGBD peut très vite aller en décroissant. Celle-ci se révèle souvent par une mauvaise définition de la structure logique. Une mauvaise analyse donnera un mauvais schéma de données. Il en sera de même, si les requêtes SQL sont mal

formulées. Le support physique est également important, un serveur sous-dimensionné (manque de place disque ou de mémoire) ou un réseau mal conçu grèvera les temps d'accès et de réponse du SGBD.

## Critère de choix d'un SGBD

La question du choix du système de gestion de base de données est essentiel. Celui-ci représente le cerveau de votre application. Une partie client (les programmes) excellentement bien développé ne servira à rien si le SGBD n'est pas conforme.

Voici quelques questions essentiels à se poser :

1. Quels sont les systèmes d'exploitation supportés ?
2. Quel est le volume disque nécessaire pour supporter les modules du SGBD ?
3. Peut-on gérer la même base sur plusieurs serveurs ?
4. Quel est le nombre maximum d'utilisateurs ?
5. Quelle est la taille maximum d'une base ?
6. Quel est le nombre maximum de tables autorisé et leur taille maximum ?
7. Y a-t-il un nombre limite d'enregistrement, de lignes ou de colonnes ?
8. Combien y a-t-il de types de données différentes (entier, texte, date, monétaire, etc.) ?
9. Quelles sont les limites des clés et des index ?
10. Peut-on créer des vues multi-tables ?
11. Y a-t-il des outils de maintenance du SGBD ?
12. Comment fonctionne la sécurité (blocage des enregistrements, des tables, etc.) ?

## Remerciement

Je tiens par le présent document à remercier Monsieur SCHOLL auprès duquel j'ai suivi pendant une année les cours du C.N.A.M. ([www.cnam.fr](http://www.cnam.fr)) sur les systèmes de gestion de base de données.

## Copyright

Toutes les marques citées dans le présent document sont déposées par leurs sociétés respectives. L'auteur décline toute responsabilité quant à la mauvaise utilisation qui pourrait être faite des informations contenues dans ce document.

Toute reproduction même partielle de ce document est interdite sauf autorisation de l'auteur.

© Cyril Beaussier - Courriel : [cyril.beaussier@mail.dotcom.fr](mailto:cyril.beaussier@mail.dotcom.fr)

Création initiale : 29/02/2000

Dernière date de révision : 25/06/2001

