

# **Introduction au Système de Gestion de Base de Données et aux Base de Données**

Formation

« Gestion des données scientifiques : stockage et consultation en  
utilisant des bases de données »

24 au 27 /06/08

**Dernière modif. : 17/06/2008**

**Version : 1.0**

**Pages: 47**

INTRODUCTION .....	4
I - INTRODUCTION AUX BASES DE DONNEES .....	5
1.1 - <i>Qu'est ce qu'une base de données</i> .....	5
1.2 – <i>Système de Gestion de Base de Données</i> .....	5
1.2.1 Définition et principes de fonctionnement .....	5
1.2.2 Objectifs .....	5
1.2.3 Niveaux de description des données ANSI/SPARC .....	6
1.3 <i>Quelques SGBD connus et utilisés</i> .....	7
1.4 <i>PostgreSQL</i> .....	7
II - SGBD RELATIONNEL.....	8
2.1 - <i>Introduction au modèle relationnel</i> .....	8
2.2 <i>Éléments du modèle relationnel</i> .....	9
2.3 <i>Algèbre relationnelle</i> .....	10
2.3.1 Introduction .....	10
2.3.2 Sélection .....	10
2.3.3 Projection.....	11
2.3.4 Union.....	11
2.3.5 Intersection .....	12
2.3.6 Différence.....	12
2.3.7 Produit cartésien .....	13
2.3.8 Jointure, theta-jointure, equi-jointure, jointure naturelle .....	14
a) Jointure .....	14
b)Theta-jointure .....	14
c) Equi-jointure.....	14
d) Jointure naturelle .....	15
2.3.9 Division .....	15
III – LANGAGE SQL.....	17
3.1 <i>Introduction SQL</i> .....	17
3.2 <i>Catégories d'instructions</i> .....	17
3.2.1 Langage de définition de données .....	17
3.2.2 Langage de manipulation de données.....	17
3.2.3 Langage de protections d'accès.....	18
3.2.4 Langage de contrôle de transaction .....	18
3.2.5 SQL intégré .....	18
3.3 <i>Définir/créer une base – Langage de Définition des Données (LDD)</i> .....	18
3.3.1 Introduction aux contraintes d'intégrité.....	18
Contrainte d'intégrité de domaine .....	18
Contrainte d'intégrité de table (ou de relation ou d'entité).....	18
Contrainte d'intégrité de référence .....	19
3.3.2 Créer une table : CREATE TABLE .....	19
Introduction .....	19
Création simple.....	19
Les types de données .....	20
3.3.3 Contraintes d'intégrité .....	21
Syntaxe .....	21
Contraintes de colonne .....	21
Contraintes de table .....	21
Complément sur les contraintes.....	22
3.3.4 Supprimer une table : DROP TABLE.....	22
3.3.5 Modifier une table : ALTER TABLE.....	23
Ajout ou modification de colonne .....	23
Ajout d'une contrainte de table.....	23
Renommer une colonne .....	23
Renommer une table.....	23
3.4 <i>Modifier une base – Langage de manipulation de données (LMD)</i> .....	23
3.4.1 Insertion de n-uplets : INSERT INTO .....	23
3.4.2 Modification de n-uplets : UPDATE .....	23
3.4.3 Suppression de n-uplets : DELETE.....	24
3.5 <i>Interroger une base – Langage de manipulation de données (LMD) : SELECT (1<sup>ère</sup> partie)</i> .....	25
3.5.1 Introduction à la commande SELECT.....	25
Introduction .....	25

Syntaxe simplifiée de la commande SELECT .....	25
Délimiteurs : apostrophes simples et doubles .....	25
3.5.2 Traduction des opérateurs de projection, sélection, produit cartésien et équi-jointure de l'algèbre relationnelle (1 ère partie) .....	26
Traduction de l'opérateur de projection.....	26
Traduction de l'opérateur de sélection.....	26
Traduction de l'opérateur de produit cartésien .....	26
Traduction de l'opérateur d'équi-jointure.....	26
3.5.3 Syntaxe générale de la commande SELECT .....	27
3.5.4 La clause SELECT .....	27
Introduction .....	27
L'opérateur étoile (*) .....	28
Les opérateurs DISTINCT et ALL .....	28
Les opérations mathématiques de base.....	28
L'opérateur AS.....	28
L'opérateur de concaténation.....	28
3.5.5 La clause FROM (1 ère partie).....	29
Comportement .....	29
L'opérateur AS.....	29
Sous-requête .....	29
Les jointures .....	29
3.5.6 La clause ORDER BY.....	30
3.5.7 La clause WHERE.....	30
Comportement .....	30
Expression simple.....	31
Prédicat simple .....	31
Prédicat composé.....	32
3.5.8 Les expressions régulières.....	32
Introduction .....	32
Formalisme .....	33
Exemples .....	34
3.6 Interroger une base – Langage de manipulation de données (LMD) : SELECT (2 ème partie).....	36
3.6.1 La clause FROM (2 ème partie) : les jointures .....	36
Le produit cartésien .....	36
Syntaxe générale des jointures.....	37
Définition de deux tables pour les exemples qui suivent.....	38
Exemples de jointures internes .....	39
Exemples de jointures externes gauches.....	40
Exemples de jointures externes droites.....	40
Exemples de jointures externes bilatérales .....	41
3.6.2 Les clauses GROUP BY et HAVING et les fonctions d'agrégation .....	42
Notion de groupe .....	42
Syntaxe .....	42
La clause GROUP BY .....	42
Les fonctions d'agrégation .....	42
Exemples .....	43
La clause HAVING .....	44
Exemples .....	44
3.6.3 Opérateurs ensemblistes : UNION, INTERSECT et EXCEPT .....	45
3.6.4 Traduction des opérateurs d'union, d'intersection, de différence et de division de l'algèbre relationnelle (2 ème partie).....	45
Traduction de l'opérateur d'union .....	45
Traduction de l'opérateur d'intersection.....	45
Traduction de l'opérateur de différence.....	45
Traduction de l'opérateur de division.....	46

## **INTRODUCTION**

Aujourd'hui, la disponibilité de systèmes de gestion de base de données fiables permet aux organisations de toutes tailles de gérer des données efficacement, de déployer des applications utilisant ces données et de les stocker. Les bases de données sont actuellement au cœur du système d'information des entreprises.

Les bases de données relationnelles constituent l'objet de ce cours. Ces bases sont conçues suivant le modèle relationnel, dont les fondations théoriques sont solides, et manipulées en utilisant l'algèbre relationnelle. Il s'agit, à ce jour, de la méthode la plus courante pour organiser et accéder à des ensembles de données.

Cependant, il est difficile de modéliser un domaine directement sous une forme base de données relationnelle. Une modélisation intermédiaire est généralement indispensable. Le modèle entités-associations ou le diagramme des classes permettent une description naturelle du monde réel à partir des concepts d'entité et d'association ou classes/association.

Après une courte introduction (chapitre 1), nous présenterons le modèle relationnel, le passage du modèle conceptuel au modèle relationnel et enfin l'algèbre relationnelle (chapitre 2).

Le chapitre 3 est entièrement consacré au langage SQL (Structured Query Language) qui peut être considéré comme le langage d'accès normalisé aux bases de données relationnelles. Ce langage est supporté par la plupart des systèmes de gestion de bases de données commerciaux (comme Oracle) et du domaine libre (comme PostgreSQL).

Nous détaillons dans ce chapitre les instructions du langage de définition de données et celles du langage de manipulation de données.

Ce document a été réalisé à partir du support du cours « Base de Données et langage SQL » dispensé aux étudiants du département d'informatique de l'institut universitaire de technologie de Villetaneuse (<http://laurent-audibert.developpez.com/Cours-BD/>).

## I - Introduction aux bases de données

### 1.1 - Qu'est ce qu'une base de données

*Une base de données (BD) est un ensemble structuré et organisé permettant le stockage de grandes quantités d'informations afin d'en faciliter l'exploitation (ajout, mise à jour, recherche de données).*

### 1.2 – Système de Gestion de Base de Données

#### 1.2.1 Définition et principes de fonctionnement

*Un système de gestion de base de données (SGBD) est un ensemble de programmes qui permet la gestion et l'accès à une base de données. Il héberge généralement plusieurs bases de données, qui sont destinées à des logiciels ou des thématiques différentes.*

On distingue couramment les SGBD classiques, dits **SGBD relationnels** (SGBD-R), des **SGBD orientés objet** (SGBD-O). En fait, un SGBD est caractérisé par le **modèle de description des données** qu'il supporte (relationnel, objet etc.). Les données sont décrites sous la forme de ce modèle, grâce à un Langage de Description des Données (LDD). Cette description est appelée schéma.

Une fois la base de données spécifiée, on peut y insérer des données, les récupérer, les modifier et les détruire. Les données peuvent être manipulées non seulement par un Langage spécifique de Manipulation des Données (LMD) mais aussi par des langages de programmation classiques.

Actuellement, la plupart des SGBD fonctionnent selon un mode client/serveur. Le serveur (sous entendu la machine qui stocke les données) reçoit des requêtes de plusieurs clients et ceci de manière concurrente. Le serveur analyse la requête, la traite et retourne le résultat au client.

Quelque soit le modèle, un des problèmes fondamentaux à prendre en compte est la cohérence des données. Par exemple, dans un environnement où plusieurs utilisateurs peuvent accéder concurrentement à une colonne d'une table par exemple pour la lire ou pour l'écrire, il faut s'accorder sur la politique d'écriture. Cette politique peut être : les lectures concurrentes sont autorisées mais dès qu'il y a une écriture dans une colonne, l'ensemble de la colonne est envoyée aux autres utilisateurs l'ayant lue pour qu'elle soit rafraîchie.

#### 1.2.2 Objectifs

**Les principaux objectifs fixés aux SGBD** afin de résoudre les problèmes causés par une gestion sous forme de fichiers à plat sont les suivants :

- **Indépendance physique** : La façon dont les données sont définies doit être indépendante des structures de stockage utilisées.
- **Indépendance logique** : Un même ensemble de données peut être vu différemment par des utilisateurs différents. Toutes ces visions personnelles des données doivent être intégrées dans une vision globale.
- **Accès aux données** : L'accès aux données se fait par l'intermédiaire d'un Langage de Manipulation de Données (LMD). Il est crucial que ce langage permette d'obtenir des réponses aux requêtes en un temps « raisonnable ». Le LMD doit donc être optimisé, minimiser le nombre d'accès disques, et tout cela de façon totalement transparente pour l'utilisateur.
- **Administration centralisée des données (intégration)** : Toutes les données doivent être centralisées dans un réservoir unique commun à toutes les applications. En effet, des visions

différentes des données (entre autres) se résolvent plus facilement si les données sont administrées de façon centralisée.

- **Non redondance des données** : Afin d'éviter les problèmes lors des mises à jour, chaque donnée ne doit être présente qu'une seule fois dans la base.
- **Cohérence des données** : Les données sont soumises à un certain nombre de contraintes d'intégrité qui définissent un état cohérent de la base. Elles doivent pouvoir être exprimées simplement et vérifiées automatiquement à chaque insertion, modification ou suppression des données. Les contraintes d'intégrité sont décrites dans le Langage de Description de Données (LDD).
- **Partage des données** : Il s'agit de permettre à plusieurs utilisateurs d'accéder aux mêmes données au même moment de manière transparente. Si ce problème est simple à résoudre quand il s'agit uniquement d'interrogations, cela ne l'est plus quand il s'agit de modifications dans un contexte multi-utilisateurs car il faut : permettre à deux (ou plus) utilisateurs de modifier la même donnée « en même temps » et assurer un résultat d'interrogation cohérent pour un utilisateur consultant une table pendant qu'un autre la modifie.
- **Sécurité des données** : Les données doivent pouvoir être protégées contre les accès non autorisés. Pour cela, il faut pouvoir associer à chaque utilisateur des droits d'accès aux données.
- **Résistance aux pannes** : Que se passe-t-il si une panne survient au milieu d'une modification, si certains fichiers contenant les données deviennent illisibles ? Il faut pouvoir récupérer une base dans un état « sain ». Ainsi, après une panne intervenant au milieu d'une modification deux solutions sont possibles : soit récupérer les données dans l'état dans lequel elles étaient avant la modification, soit terminer l'opération interrompue.

### 1.2.3 Niveaux de description des données ANSI/SPARC

Pour atteindre certains de ces objectifs (surtout les deux premiers), trois niveaux de description des données ont été définis par la norme ANSI/SPARC :

- **Le niveau externe** correspond à la perception de tout ou partie de la base par un groupe donné d'utilisateurs, indépendamment des autres. On appelle cette description le schéma externe ou vue. Il peut exister plusieurs schémas externes représentant différentes vues sur la base de données avec des possibilités de recouvrement. Le niveau externe assure l'analyse et l'interprétation des requêtes en primitives de plus bas niveau et se charge également de convertir éventuellement les données brutes, issues de la réponse à la requête, dans un format souhaité par l'utilisateur.
- **Le niveau conceptuel** décrit la structure de toutes les données de la base, leurs propriétés (i.e. les relations qui existent entre elles : leur sémantique inhérente), sans se soucier de l'implémentation physique ni de la façon dont chaque groupe de travail voudra s'en servir. Dans le cas des SGBD relationnels, il s'agit d'une vision tabulaire où la sémantique de l'information est exprimée en utilisant les concepts de relation, attributs et de contraintes d'intégrité. On appelle cette description le schéma conceptuel.
- **Le niveau interne ou physique** s'appuie sur un système de gestion de fichiers pour définir la politique de stockage ainsi que le placement des données. Le niveau physique est donc responsable du choix de l'organisation physique des fichiers ainsi que de l'utilisation de telle ou telle méthode

Ceci est un exemple, cliquez sur le lien de téléchargement pour obtenir le cours complet.

