Table des matières

[I. Introduction 2](#_Toc175589262)

[II. La définition des données - LDD 4](#_Toc175589263)

[III. Les fonctions 5](#_Toc175589264)

[IV. Le contrôle des transactions 5](#_Toc175589265)

[V. Approfondissement 5](#_Toc175589266)

Toute donnée est stockée dans une structure de base de données. On parle de BDD pour désigner le stockage des données et de SGBD pour désigner les éléments (système) qui sont mis à la disposition du développeur pour manipuler ces données.

Le langage SQL pour Structured Query Language – langage de requête structuré se décompose en plusieurs sous-ensembles :

* DDL- Data Définition Language, qui se composent principalement des ordres CREATE, ALTER, DROP. Il regroupe les ordres (commandes) utilisés pour créer, modifier et supprimer les structures (index, tables, vues, etc.) de la base de données.
* DML – Data Manipulation Language, qui se composent principalement des ordres SELECT, INSERT, DELETE, UPDATE. Il regroupe les ordres (commandes) utilisés pour manipuler les données dans la base.
* DCL – Data Control Language, qui se composent principalement des ordres GRANT et REVOKE. Il regroupe les ordres (commandes) utilisés pour gérer la sécurité des accès aux données.
* TCL – Transaction Control Language, qui se composent principalement des ordres COMMIT et ROLLBACK. Il regroupe les ordres (commandes) utilisés pour gérer la validation ou non de la mise à jour de la base données.

Il existe plusieurs normes SQL. La norme SQL2 ou SQL92 est la plus importante. C’est cette dernière qui est implémentée dans la majorité des SGBDR (système de gestion de base de données relationnelles). Ainsi, chaque fournisseur de SGBDR a implémenté sa façon le langage SQL et a ajouté ses propres extensions.

# Introduction

Le modèle relationnel repose sur la notion d’ensemble. Un ensemble peut être représenté par une table (aussi appelée une relation). Cet ensemble a des attributs (colonnes) et des lignes (tuples). Une fois ces tables définies, il faut disposer d’un langage pour les manipuler. Il s’agit de l’algèbre relationnelle. Ainsi, l’algèbre relationnelle est mise en œuvre par le SQL et les systèmes de gestion de bases de données relationnelles implémentent le modèle relationnel. A l’aide des opérateurs, on peut interroger les relations existantes et créer de nouvelles relations. On parle d’opérateurs ensemblistes : union, intersection, différence, produit cartésien, division et jointure.

Le modèle relationnel est caractérisé par trois concepts :

* La Relation, fondement du modèle relationnel. La relation permet de mettre en relation des domaines suivant certains critères. Le degré est le nombre d’attribut d’une relation.
* La Domaine, ensemble de valeurs caractérisées par un nom.
* Le Produit cartésien, représentant la jonction entre deux domaines.

L'algèbre relationnelle est la base théorique des opérations que l'on peut réaliser sur les bases de données relationnelles, et SQL (Structured Query Language) en est la concrétisation pratique. Le SQL est devenu le standard en ce qui concerne la gestion des données. Il permet la manipulation des tables et des colonnes. Son principe repose sur la création de nouvelles tables (tables résultantes) à partir des tables existantes. Ces nouvelles tables devenant des objets utilisables immédiatement. Les principales opérations de l’algèbre relationnelle entre deux tables table1 et table2 sont :

* L’Union entre deux relations de même structure (degré et domaines). La commande UNION renvoie toutes les lignes de table1 et table2, sans les doublons.
* L’Intersection entre deux relations de même structure (degré et domaines). La commande INTERSECT renvoie seulement les lignes qui apparaissent dans les deux tables
* La Différence entre deux relations de même structure (degré et domaines). La commande EXCEPT renvoie les lignes de table1 qui ne sont pas dans table2.
* La Division entre deux relations est possible à condition que la relation diviseur soit totalement incluse dans la relation dividende. Le quotient de la division correspond à l’information qui, présente dans le dividende, n’est pas présente dans le diviseur.
* La Restriction (Sélection) repose sur une condition. Elle produit, à partir d’une relation, une relation de même schéma n’ayant que les éléments de la relation initiale qui répondent à la condition. Elle s’utilise avec la commande WHERE condition.
* La Projection d’une relation sur un groupe d’attributs donne une relation résultante ayant comme schéma uniquement ces attributs, et comme éléments les n-uplets distincts composés par les valeurs associées de ces attributs. Elle permet de supprimer les doublons. Extrait certaines colonnes d'une table, en supprimant les doublons. La commande DISTINCT colonne1, colonne2 FROM table1 renvoie les valeurs distinctes des colonnes spécifiées.
* La Produit cartésien combine toutes les lignes de deux relations pour produire toutes les combinaisons possibles. SELECT \* FROM table1, table2 crée une combinaison de chaque ligne de table1 avec chaque ligne de table2.
* La Jointure entre deux relations est produite par la restriction sur le produit cartésien. La Jointure combine les lignes de deux relations basées sur une condition (généralement une égalité entre colonnes). SELECT \* FROM table1 JOIN table2 ON table1.colonne1 = table2.colonne2 combine les lignes de table1 et table2 lorsqu'il y a correspondance des valeurs spécifiées
* La Calcul d’agrégats projection sur relation associée à un ou des calculs statistiques portant sur un attribut pour tous les éléments de la relation ou du regroupement lié à la projection afin de créer un ou plusieurs nouveaux attributs. COUNT, SUM, AVG, MAX, MIN.

# La définition des données – LDD

Il existe trois grandes de familles de données : numérique, caractère (ou alphanumérique) et temporelle (dates et heures) :

* Numériques : les types numériques principaux de la nome SQL2 sont NUMERIC pour les données à virgule, INTERGER pour les entiers long et SMALLINT pour les entiers courts. La plupart, des SGBDR fusionnent le type NUMERIC et le type INTERGER en un seul type nommé NUMBER (ou DECIMAL). Notons cependant, qu’il existe d’autre types ajoutés par des SGBDR tels que TINYINT, SMALLINT, LONG, FLOAT.
* Caractères : le type alphanumériques classique se note CHAR pour les données de taille fixe et VARCHAR pour les chaînes de caractères de longueur variable. Dans MySQL, la taille maximum d’une chaîne CHAR et VARCHAR est normalement de 255 caractères. Si la saisie dépasse cette limite, MySQL transforme automatiquement le type en TEXT, MEDIUMTEXT ou LONGTEXT. Ces types sont de longueur fixe.
* Dates ou heures : Les types de format temporel sont principalement DATE, TIME et TIMESTAMP (ou DATETIME). La manipulation des dates et des heures est très différente d’un SGBDR.

## La création de tables

CREATE est l’ordre de base en langage SQL pour créer un élément (TABLE, INDEX, VUE, SYNONYME). Cette commande est indépendante du stockage physique de la table. En effet, chaque SGBDR a sa propre syntaxe dans ce domaine. Dans la majorité des cas, ce sont les DBA (Database Administrator – administrateur de base de données) qui spécifient les normes de stockage et les options à appliquer sur les tables (en termes de performance et de sécurité de la base de données).

La clause COMMENT permet d’ajouter des commentaires sur les colonnes et tables directement dans la base de données au moment de sa création. Compte tenu du fait qu’une base de données à une durée de vie importante et va être utilisée par une population importante et hétéroclite, ajouter les commentaires participe à la maintenance et à l’évolution de la base.

Dans une table, il y’a souvent une colonne qui fait office d’identifiant unique (une sorte de compteur) et qui est la plupart du temps de type numérique. Pour laisser le SGBDR gérer l’incrémentation de cette colonne, on peut utiliser un objet de type SEQUENCE et l’associer à une colonne. Par ailleurs, certains SGBDR permettent de créer une séquence indépendamment d’une colonne de table. Oracle implémente l’objet séquence. Quant à MySQL, il ne l’implémente pas, il existe néanmoins la fonction AUTO\_INCREMENT sur une colonne lors de la création de la table. Cependant, cette colonne doit être la PRIMARY KEY.

Dans une base de données, les tables sont rattachées à la personne (user) qui les a créées (souvent le DBA) qui donne le droit (ordre GRANT) aux autres intervenants d’utiliser ses tables. Par défaut, une table est préfixée par l’utilisateur qui l’a créée (ABOUBAKAR.TELEPHONE). Cependant on peut utiliser l’ordre SYNONYM pour simplifier un nom de table qui est normalisé, ou pour pointer sur une archive dont le nom n’a pas à être connu par les utilisateurs.

Une autre méthode pour créer une table consiste à dupliquer ou à s’inspirer de tables existantes. Pour récupérer uniquement la structure de la table d’origine, il faut ajouter une clause (condition) qui n’est jamais vérifiée.

|  |
| --- |
| Ligne de commandes |
| * MySQL : create table table2 select \* from table1 * MySQL : create table table2 like table1 * ORACLE : create table table2 as select \* from table1 |

## La suppression de tables

La suppression de tables est une opération simple mais qu’il faut manier avec prudence, en effet, cette dernière est définitive. L’ordre DROP permet de supprimer définitivement une table (contenu, index, contraintes, commentaires). Cependant, cet ordre ne détruit pas les synonymes. On utilise souvent l’ordre DROP juste avant la création d’une table.

## La modification de tables

L’ordre ALTER est utilisé pour réaliser plusieurs actions (supprimer ou ajouter une colonne d’une table d’une table, ajouter ou supprimer une contrainte ou ajouter une valeur par défaut à une colonne, et même modifier le type (CAST) d’une colonne).

L’ordre RENAME permet de renommer une table. Cette commande peut être utilisée dans le cas où la table doit être recréée tout en conservant la version actuelle en archive.

|  |
| --- |
| **Ligne de commandes** |
| * MySQL : alter table table\_1 add/drop nom\_2\_col ou nom\_2\_contrain * MySQL : rename table to sav\_table |

## Les vues

Les utilisateurs ou développeurs ont des besoins d’extractions spécifiques dans une base de données. Ces extractions sont matérialisées sous la forme de requêtes lancées manuellement ou incluses dans des programmes. Dans le cas, ou ces demandes sont répétitives ou communes à plusieurs utilisateurs, il est parfois nécessaire de créer une vue.

Une vue est une représentation logique de la base qui résulte d’une requête pour un besoin spécifique et répétitif. Contrairement à une table, une vue n’est pas stockée sur le disque. Les vues permettent de créer des tables « virtuelles » spécifiques pour un domaine ou pour une classe d’utilisateurs. Il est possible de créer un ensemble de vues par le type d’utilisateur ou métier de l’entreprise.

L’avantage majeure d’une vue est qu’elle est en permanence mise à jour. En effet, une vue est mise à jour automatiquement lorsque les tables qu’elles utilisent sont modifiées. Pour un utilisateur, une vue se comporte comme une table. Il peut réaliser ses requêtes sur la vue comme sur une table.

|  |
| --- |
| **Ligne de commandes** |
| * create view <nom\_vue> as select |

## Les index

## L’intégrité des données et les clés

# Les fonctions

# Le contrôle des transactions

# Approfondissement

|  |
| --- |
| Ligne de commandes |
| * sudo apt update * sudo apt install mysql-server * systemctl start mysql / service mysql start * systemctl stop mysql / service mysql stop |