



LE LANGUAGE SQL

**LMD : Langage de Manipulation
des Données (1/2)**

M. RABI

Introduction

Le langage SQL (Structured Query Language) peut être considéré comme le langage d'accès normalisé aux bases de données. Il est supporté par la plupart des produits commerciaux que ce soit par les systèmes de gestion de bases de données micro tel que Access ou par les produits plus professionnels tels que Oracle.

Le succès du langage SQL est dû essentiellement à sa simplicité et au fait qu'il s'appuie sur le schéma relationnel pour énoncer des requêtes en laissant le SGBD responsable de la stratégie d'exécution.

De manière synthétique, on peut dire que SQL est un langage relationnel de manipulation de bases de données.

1. Présentation

1.2 Objectifs :

- Créer la structure de la base de données et de ses tables.
- Exécuter les tâches de la gestion des données, telle que l'insertion, la modification et la suppression de données des tables.
- Effectuer des requêtes simples ou complexes.

1.3 Définition

Le **SQL** (Structured Query Language) est un langage permettant la manipulation et la communication avec une base de données.

L'intérêt de ce langage est qu'il est très puissant et très bien documenté. Il permet, via des requêtes, de créer, interroger, modifier, supprimer et manipuler les données.

1. Présentation

1.4 Fonctionnalités

SQL est un langage composé de sous-ensembles.

LDD : Le Langage de Définition de Données (DDL : *Data Definition Language*) pour créer et supprimer des objets dans la base de données (tables, contraintes d'intégrité, etc...)

Exemple de commandes : CREATE DROP ALTER

LMD : Le Langage de Manipulation de Données (DML, *Data Manipulation Language*) pour la recherche, l'insertion, la mise à jour et la suppression de données.

Le LMD est basé sur les opérateurs relationnels, auxquels sont ajoutés des fonctions de calcul d'agrégats et des instructions pour réaliser les opérations d'insertion, mise à jour et suppression.

Exemple de commandes : INSERT UPDATE DELETE SELECT

LCD : Le Langage de Contrôle de Données (DCL, *Data Control Language*) pour gérer les droits sur les objets de la base (création des utilisateurs et affectation de leurs droits).

Exemple de commandes : GRANT REVOKE

2. Le langage de manipulation de données (LMD)

2.1 LA PROJECTION

La principale commande du *langage de manipulation de données* est la commande **SELECT**.

2.1.1 La commande SELECT

L'utilisation la plus courante de SQL consiste à lire des données issues de la base de données. Cela s'effectue grâce à la commande SELECT, qui retourne des enregistrements dans un tableau de résultat. Cette commande peut sélectionner une ou plusieurs colonnes d'une table.

Syntaxe : L'utilisation basique de cette commande s'effectue de la manière suivante.

```
SELECT nom_Champ  
FROM nom_Table
```

Cette requête va sélectionner (SELECT) le champ « nom_du_champ » provenant (FROM) de la table appelée « nom_du_tableau »

2. Le langage de manipulation de données (LMD)

2.1.1 La commande SELECT

Exemple : On considère la table « Client » :

identifiant	prenom	nom	ville
1	Pierre	Dupond	Paris
2	Sabrina	Durand	Nantes
3	Julien	Martin	Lyon
4	David	Bernard	Marseille
5	Marie	Leroy	Grenoble

Si on veut avoir la liste de toutes les villes des clients, il suffit d'effectuer la requête suivante :

SELECT ville FROM Client



ville
Paris
Nantes
Lyon
Marseille
Grenoble

2. Le langage de manipulation de données (LMD)

2.1.1 La commande SELECT

Obtenir plusieurs colonnes

Avec la même table « Client » il est possible de lire plusieurs colonnes à la fois. Il suffit de séparer les noms des champs souhaités par une virgule.

Pour obtenir les prénoms et les noms des clients :

```
SELECT nom, prenom  
FROM Client
```



prenom	nom
Pierre	Dupond
Sabrina	Durand
Julien	Martin
David	Bernard
Marie	Leroy

2. Le langage de manipulation de données (LMD)

2.1.1 La commande SELECT

Obtenir toutes colonnes d'une table

Il est possible de retourner automatiquement toutes les colonnes d'un tableau sans avoir à connaître le nom de toutes les colonnes. Au lieu de lister toutes les colonnes, il faut utiliser le caractère « * ».

```
SELECT * FROM Client
```

identifiant	prenom	nom	ville
1	Pierre	Dupond	Paris
2	Sabrina	Durand	Nantes
3	Julien	Martin	Lyon
4	David	Bernard	Marseille
5	Marie	Leroy	Grenoble

2. Le langage de manipulation de données (LMD)

2.1.2 La commande DISTINCT

Éliminer les redoublons

L'utilisation de la commande SELECT en SQL permet de lire toutes les données d'une ou plusieurs colonnes. Cette commande peut potentiellement afficher des lignes en doubles. Pour éviter des redondances dans les résultats il faut ajouter DISTINCT après le mot SELECT.

Syntaxe :

```
SELECT DISTINCT  
nom_Champ  
FROM nom_Table
```

2. Le langage de manipulation de données (LMD)

2.1.2 La commande DISTINCT

Exemple :

identifiant	prenom	nom
1	Pierre	Dupond
2	Sabrina	Bernard
3	David	Durand
4	Pierre	Leroy
5	Marie	Leroy

SELECT nom FROM Personne

nom
Dupond
Bernard
Durand
Leroy
Leroy

SELECT **DISTINCT** nom FROM Personne

nom
Dupond
Bernard
Durand
Leroy

2. Le langage de manipulation de données (LMD)

2.2 LA RESTRICTION

Une restriction consiste à sélectionner les lignes satisfaisant à une condition logique effectuée sur leurs attributs.

En SQL, les restrictions s'expriment à l'aide de la clause **WHERE** suivie d'une condition logique.

2.2.1 La commande WHERE

La commande WHERE dans une requête SQL permet d'extraire les lignes d'une base de données qui respectent une condition. Cela permet d'obtenir uniquement les informations désirées.

Syntaxe :

```
SELECT nom_Champ  
FROM nom_Table  
WHERE Condition
```

2. Le langage de manipulation de données (LMD)

2.2.1 La commande WHERE

Exemple : On considère la table « Client » suivante.

id	nom	nbr_commande	ville
1	Paul	3	paris
2	Maurice	0	rennes
3	Joséphine	1	toulouse
4	Gérard	7	paris

Pour obtenir seulement la liste des clients qui habitent à Paris:

```
SELECT *  
FROM Client  
WHERE Ville ='Paris'
```

id	nom	nbr_commande	ville
1	Paul	3	paris
4	Gérard	7	paris

2. Le langage de manipulation de données (LMD)

2.2.1 La commande WHERE

Opérateurs : Il existe plusieurs opérateurs à utiliser avec la clause WHERE. La liste ci-jointe présente les opérateurs les plus couramment utilisés.

Opérateurs logiques :

- AND
- OR
- NOT

Comparateur logique :

- IN
- BETWEEN
- LIKE

Opérateur arithmétiques :

- +
- -
- *
- /
- %
- ...

Comparateurs arithmétiques :

- =
- !=
- >
- <
- >=
- <=
- ...

2. Le langage de manipulation de données (LMD)

2.2.1 La commande WHERE

Opérateurs Logiques: (AND , OR)

Une requête SQL peut être restreinte à l'aide de la condition WHERE. Les opérateurs logiques AND et OR peuvent être utilisées au sein de la commande WHERE pour combiner des conditions.

Les opérateurs sont à ajouter dans la condition WHERE. Ils peuvent être combinés à l'infini pour filtrer les données comme souhaites.

L'opérateur AND permet de s'assurer que la condition1 ET la condition2 sont vraies :

```
SELECT nom_Champ FROM  
nom_Table WHERE  
Condition1 AND Condition2
```

L'opérateur OR vérifie quant à lui que la condition1 OU la condition2 est vrai :

```
SELECT nom_Champ FROM  
nom_Table WHERE  
Condition1 OR Condition2
```

Ces opérateurs peuvent être combinés à l'infini :

```
SELECT nom_Champ FROM nom_Table WHERE  
Condition1 AND (Condition2 OR Condition3)
```

2. Le langage de manipulation de données (LMD)

2.2.1 La commande WHERE

Exemple Opérateur AND :

Pour illustrer les prochaines commandes, nous allons considérer la table « **Produit** » suivante

id	nom	categorie	stock	prix
1	ordinateur	informatique	5	950
2	clavier	informatique	32	35
3	souris	informatique	16	30
4	crayon	fourniture	147	2

```
SELECT * FROM produit  
WHERE categorie = 'informatique' AND stock < 20
```

id	nom	categorie	stock	prix
1	ordinateur	informatique	5	950
3	souris	informatique	16	30

2. Le langage de manipulation de données (LMD)

2.2.1 La commande WHERE

Exemple Opérateur OR:

id	nom	categorie	stock	prix
1	ordinateur	informatique	5	950
2	clavier	informatique	32	35
3	souris	informatique	16	30
4	crayon	fourniture	147	2

```
SELECT * FROM produit  
WHERE nom = 'ordinateur' OR nom = 'clavier'
```

id	nom	categorie	stock	prix
1	ordinateur	informatique	5	950
2	clavier	informatique	32	35

2. Le langage de manipulation de données (LMD)

2.2.1 La commande WHERE

Exemple Opérateurs combinés (OR, AND) :

id	nom	categorie	stock	prix
1	ordinateur	informatique	5	950
2	clavier	informatique	32	35
3	souris	informatique	16	30
4	crayon	fourniture	147	2

```
SELECT * FROM produit  
WHERE ( categorie = 'informatique' AND prix >= 35 )  
OR ( categorie = 'fourniture' AND stock < 200 )
```

id	nom	categorie	stock	prix
1	ordinateur	informatique	5	950
2	clavier	informatique	32	35
4	crayon	fourniture	147	2

2. Le langage de manipulation de données (LMD)

2.2.1 La commande WHERE

L'Opérateur IN:

L'opérateur logique IN dans SQL s'utilise avec la commande WHERE pour vérifier si une colonne est égale à une des valeurs comprise dans un ensemble déterminé de valeurs. C'est une méthode simple pour vérifier si une colonne est égale à une valeur OU une autre valeur OU une autre valeur et ainsi de suite, sans avoir à utiliser de multiple fois l'opérateur OR.

```
SELECT nom_Champ  
FROM nom_Table  
WHERE nom_Champ IN ( valeur1, valeur2, valeur3, ... )
```

A savoir : entre les parenthèses il n'y a pas de limite du nombre d'arguments. Il est possible d'ajouter encore d'autres valeurs.

Cette syntaxe peut être associée à l'opérateur NOT pour rechercher toutes les lignes qui ne sont pas égales à l'une des valeurs stipulées.

2. Le langage de manipulation de données (LMD)

2.2.1 La commande WHERE

L'Opérateur IN:

La syntaxe utilisée avec l'opérateur est plus simple qu'une succession d'opérateur OR. Pour le montrer concrètement avec un exemple, voici 2 requêtes qui retourneront les mêmes résultats, l'une utilise l'opérateur IN, tandis que l'autre utilise plusieurs OR.

Requête avec plusieurs OR

```
SELECT prenom  
FROM utilisateur  
WHERE prenom = 'Maurice' OR  
prenom = 'Marie' OR prenom =  
'Thimote'
```

Requête équivalent avec l'opérateur IN

```
SELECT prenom  
FROM utilisateur  
WHERE prenom IN ( 'Maurice',  
'Marie', 'Thimote' )
```

2. Le langage de manipulation de données (LMD)

2.2.1 La commande WHERE

Exemple Opérateur IN: On considère la table « Adresse » qui contient une liste d'adresse associée à des utilisateurs d'une application.

id	id_utilisateur	addr_rue	addr_code_postal	addr_ville
1	23	35 Rue Madeleine Pelletier	25250	Bournois
2	43	21 Rue du Moulin Collet	75006	Paris
3	65	28 Avenue de Cornouaille	27220	Mousseaux-Neuville
4	67	41 Rue Marcel de la Provoté	76430	Graimbouville
5	68	18 Avenue de Navarre	75009	Paris

Si on souhaite obtenir les enregistrements des adresses de Paris et de Graimbouville:

```
SELECT * FROM adresse WHERE addr_ville IN ( 'Paris', 'Graimbouville' )
```

id	id_utilisateur	addr_rue	addr_code_postal	addr_ville
2	43	21 Rue du Moulin Collet	75006	Paris
4	67	41 Rue Marcel de la Provoté	76430	Graimbouville
5	68	18 Avenue de Navarre	75009	Paris

2. Le langage de manipulation de données (LMD)

2.2.1 La commande WHERE

L'Opérateur BETWEEN:

L'opérateur **BETWEEN** est utilisé dans une requête SQL pour sélectionner un intervalle de données dans une requête utilisant **WHERE**. L'intervalle peut être constitué de chaînes de caractères, de nombres ou de dates.

Syntaxe :

```
SELECT *  
FROM Table  
WHERE nom_colonne BETWEEN 'valeur1' AND 'valeur2'
```

La requête suivante retournera toutes les lignes dont la valeur de la colonne « nom_colonne » est comprise entre valeur1 et valeur2.

2. Le langage de manipulation de données (LMD)

2.2.1 La commande WHERE

L'Opérateur BETWEEN:

Exemple entre 2 dates : Considérons la table « utilisateur » qui contient les membres d'une application en ligne

id	nom	date_inscription
1	Maurice	2012-03-02
2	Simon	2012-03-05
3	Chloé	2012-04-14
4	Marie	2012-04-15
5	Clémentine	2012-04-26

```
SELECT *  
FROM utilisateur  
WHERE date_inscription  
BETWEEN '2012-04-01' AND  
'2012-04-20'
```

Si on souhaite obtenir les membres qui se sont inscrits entre le 1 avril 2012 et le 20 avril 2012

id	nom	date_inscription
3	Chloé	2012-04-14
4	Marie	2012-04-15

2. Le langage de manipulation de données (LMD)

2.2.1 La commande WHERE

L'Opérateur BETWEEN:

Exemple entre 2 entiers : Considérons la table « utilisateur » qui contient les membres d'une application en ligne

id	nom	date_inscription
1	Maurice	2012-03-02
2	Simon	2012-03-05
3	Chloé	2012-04-14
4	Marie	2012-04-15
5	Clémentine	2012-04-26

Si on souhaite obtenir tous les résultats dont l'identifiant n'est pas situé entre 4 et 10,

```
SELECT * FROM utilisateur  
WHERE id NOT BETWEEN 4  
AND 10
```

id	nom	date_inscription
1	Maurice	2012-03-02
2	Simon	2012-03-05
3	Chloé	2012-04-14

2. Le langage de manipulation de données (LMD)

2.2.1 La commande WHERE

L'Opérateur LIKE:

L'opérateur **LIKE** est utilisé dans la clause WHERE des requêtes SQL. Ce mot-clé permet d'effectuer une recherche sur un modèle particulier. Il est par exemple possible de rechercher les enregistrements dont la valeur d'une colonne commence par telle ou telle lettre.

Syntaxe :

```
SELECT *  
FROM table  
WHERE colonne LIKE modèle
```

Un modèle permet de définir un critère de filtrage des données selon sa structure.

2. Le langage de manipulation de données (LMD)

2.2.1 La commande WHERE

L'Opérateur LIKE:

Un modèle ressemble généralement à l'un des exemples suivants :

LIKE 'a%' : le caractère « % » est un caractère joker qui remplace tous les autres caractères. Ainsi, ce modèle permet de rechercher toutes les chaînes de caractère qui se terminent par un « a ».

LIKE 'a%' : ce modèle permet de rechercher toutes les lignes de « colonne » qui commencent par un « a »

LIKE '%a%' : ce modèle est utilisé pour rechercher tous les enregistrements qui utilisent le caractère « a ».

LIKE 'pa%on' : ce modèle permet de rechercher les chaînes qui commencent par « pa » et qui se terminent par « on », comme « pantalon » ou « pardon ».

LIKE 'a_c' : Le caractère « _ » (underscore) peut être remplacé par n'importe quel caractère, mais un seul caractère uniquement. Ainsi, ce modèle permet de retourner les lignes « aac », « abc » ou même « azc ».

2. Le langage de manipulation de données (LMD)

2.2.1 La commande WHERE

L'Opérateur LIKE:

Exemple : On considère la table « Client » ci-dessous

id	nom	ville
1	Léon	Lyon
2	Odette	Nice
3	Vivien	Nantes
4	Etienne	Lille

*Les clients habitant une ville qui
commence par « N »*

```
SELECT *  
FROM client  
WHERE ville LIKE 'N%'
```

*Les clients habitant une ville qui
termine par « e »*

```
SELECT *  
FROM client  
WHERE ville LIKE '%e'
```

2. Le langage de manipulation de données (LMD)

2.2.1 La commande WHERE

L'Opérateur IS NULL / IS NOT NULL:

Dans le langage SQL, l'opérateur **IS** permet de filtrer les résultats qui contiennent la valeur NULL. Cet opérateur est indispensable car la valeur NULL est une valeur inconnue et ne peut par conséquent pas être filtrée par les opérateurs de comparaison (égal, inférieur, supérieur ou différent).

Syntaxe :

```
SELECT *  
FROM `table`  
WHERE nom_colonne IS NULL
```

A l'inverse, pour filtrer les résultats et obtenir uniquement les enregistrements qui ne sont pas nuls

```
SELECT *  
FROM `table`  
WHERE nom_colonne IS NOT NULL
```

2. Le langage de manipulation de données (LMD)

2.2.1 La commande WHERE

L'Opérateur IS NULL / IS NOT NULL:

Exemple :

On considère une application qui possède une table contenant les utilisateurs. Cette table possède 2 colonnes pour associer les adresses de livraison et de facturation à un utilisateur (grâce à une clé étrangère). Si cet utilisateur n'a pas d'adresse de facturation ou de livraison, alors le champ reste à NULL.

id	nom	date_inscription	fk_adresse_livraison_id	fk_adresse_facturation_id
23	Grégoire	2013-02-12	12	12
24	Sarah	2013-02-17	NULL	NULL
25	Anne	2013-02-21	13	14
26	Frédérique	2013-03-02	NULL	NULL

utilisateurs sans adresse de livraison

utilisateurs avec une adresse de livraison

2. Le langage de manipulation de données (LMD)

2.2.1 La commande WHERE

L'Opérateur IS NULL / IS NOT NULL:

id	nom	date_inscription	fk_adresse_livraison_id	fk_adresse_facturation_id
23	Grégoire	2013-02-12	12	12
24	Sarah	2013-02-17	NULL	NULL
25	Anne	2013-02-21	13	14
26	Frédérique	2013-03-02	NULL	NULL

Utilisateurs sans adresse de livraison

```
SELECT *  
FROM `utilisateur`  
WHERE  
`fk_adresse_livraison_id` IS NULL
```

Utilisateurs avec une adresse de livraison

```
SELECT *  
FROM `utilisateur`  
WHERE  
`fk_adresse_livraison_id` IS NOT NULL
```



LE LANGUAGE SQL

**LMD : Langage de Manipulation
des Données (2/2)**