Analyse et Conception de Base de Données (Piste de Documentation sur SQL)

Algèbre relationnelle

La Sélection

La sélection permet de récupérer des lignes qui satisfont une condition donnée. Elle est notée **σ**. Par exemple, pour récupérer tous les Ouvrage dont le titre est "Les Misérables", on peut utiliser la commande suivante:

```
mysql> SELECT * FROM Ouvrage WHERE titre = "Les Misérables";
+------------+-----------------+----------+-------+
| id_ouvrage | titre           | id_rayon | id_ad |
+------------+-----------------+----------+-------+
|          2 | Les Misérables  |        1 |     1 |
|          3 | Les Misérables  |        1 |     1 |
+------------+-----------------+----------+-------+
2 rows in set (0,00 sec)

mysql>
```

La Projection

La projection permet de récupérer une sous-ensemble de colonnes d'une table.Elle est notée **∏**. Par exemple, pour récupérer uniquement les colonnes "prenom_ad" et "adresse_ad" de la table "Adherent", vous pouvez utiliser la comm

MODIFY Modifier une colonne ou une contraande suivante:

```
ERROR 1054 (42S22): Unknown column 'prenom_ad' in 'field list
mysql> SELECT prenom_ad, adresse_ad FROM Adherent;
+---------------+------------+
| prenom_ad     | adresse_ad |
+---------------+------------+
| Djily         | Parcelle   |
| Daour         | 26         |
| Serigne Modou | Yoff       |
+---------------+------------+
3 rows in set (0,00 sec)

mysql>
```

Le Produit Cartésien

Le produit cartésien combine chaque ligne d'une table avec chaque ligne d'une autre table. . Elle est notée **X**. Par exemple, pour récupérer toutes les combinaisons possibles de Ouvrage et de Auteur, on peut utiliser la commande suivante:

```
mysql> SELECT * FROM Ouvrage, Auteur;
+------------+-----------------+----------+-------+--------+---------+------------+
| id_ouvrage | titre           | id_rayon | id_ad | id_aut | nom_aut | prenom_aut |
+------------+-----------------+----------+-------+--------+---------+------------+
|          2 | Les Misérables  |        1 |     1 |      3 | Zola    | Emile      |
|          2 | Les Misérables  |        1 |     1 |      2 | Camus   | Albert     |
|          2 | Les Misérables  |        1 |     1 |      1 | Hugo    | Victor     |
|          3 | Les Misérables  |        1 |     1 |      3 | Zola    | Emile      |
|          3 | Les Misérables  |        1 |     1 |      2 | Camus   | Albert     |
|          3 | Les Misérables  |        1 |     1 |      1 | Hugo    | Victor     |
|          4 | L'Étranger      |        2 |     2 |      3 | Zola    | Emile      |
|          4 | L'Étranger      |        2 |     2 |      2 | Camus   | Albert     |
|          4 | L'Étranger      |        2 |     2 |      1 | Hugo    | Victor     |
|          5 | Germinal        |        1 |     3 |      3 | Zola    | Emile      |
|          5 | Germinal        |        1 |     3 |      2 | Camus   | Albert     |
|          5 | Germinal        |        1 |     3 |      1 | Hugo    | Victor     |
+------------+-----------------+----------+-------+--------+---------+------------+
12 rows in set (0,00 sec)

mysql>
```

Union:

L'union combine deux tables en une seule table, en éliminant les doublons. elle est notée **U.** Par exemple, pour récupérer les Titres, l'ID de Rayon , l'ID de Adhérent dans Ouvrage ou pour recuperer tout les attribut de auteur vous pouvez utiliser la commande suivante:

```
mysql> select * from Ouvrage UNION SELECT * FROM Auteur;
ERROR 1222 (21000): The used SELECT statements have a different number of columns
mysql> select titre, id_rayon,id_ad from Ouvrage UNION SELECT * FROM Auteur;
+-----------------+----------+--------+
| titre           | id_rayon | id_ad  |
+-----------------+----------+--------+
| Les Misérables  | 1        | 1      |
| L'Étranger      | 2        | 2      |
| Germinal        | 1        | 3      |
| 1               | Hugo     | Victor |
| 2               | Camus    | Albert |
| 3               | Zola     | Emile  |
+-----------------+----------+--------+
6 rows in set (0,01 sec)

mysql> select * from Ouvrage WHERE id_rayon = 1  UNION SELECT * FROM Ouvrage WHERE id_rayon = 3;
```

Remarque: Pour utiliser l'union il faut que les deux relation est le même schéma.

La Différence

La différence récupère toutes les lignes d'une table qui ne se trouvent pas dans une autre table. Elle est notée — . Par exemple, pour adhérent qui n'ont pas emprunter de livre, on peut utiliser la commande suivante:

```
ERROR 1054 (42S22): Unknown column 'titre' in 'field list'
mysql> SELECT * FROM Adherent WHERE id_ad NOT IN ( SELECT id_ad FROM Ouvrage);
Empty set (0,01 sec)

mysql>
```

L'intersection

L'intersection récupère toutes les lignes qui se trouvent à la fois dans une table et dans une autre table. Par exemple, pour récupérer toutes les rayon qui contient un Ouvrage, on peut utiliser la commande suivante:

SELECT * FROM Rayon WHERE id_rayon in (SELECT id_rayon from Ouvrage);

```
mysql> SELECT * FROM Rayon WHERE id_rayon in (SELECT id_rayon from Ouvrage);
+----------+-----------------+
| id_rayon | nom_rayon       |
+----------+-----------------+
|        1 | Policier        |
|        2 | Science-fiction |
+----------+-----------------+
2 rows in set (0,00 sec)

mysql>
```

La Division

La division permet de récupérer toutes les lignes d'une table qui correspondent à un ensemble de conditions dans une autre table. Par exemple, pour récupérer toutes les ouvrage qui sont référencer par les mot clés Roman et Thriller on peut utiliser la commande suivante:

La Jointure

Une jointure combine des colonnes de deux tables en fonction d'une condition de jointure. Par exemple, pour récupérer les oeuvres de chaque auteur , on peut utiliser la commande suivante:

SELECT Ouvrage.titre, Auteur.nom_aut FROM Ouvrage JOIN Auteur JOIN Rediger ON Rediger.id_aut = Auteur.id_aut AND Rediger.id_ouvrage = Ouvrage.id_ouvrage ;

```
mysql> SELECT Ouvrage.titre, Auteur.nom_aut from Ouvrage JOIN Auteur JOIN Rediger on Rediger.id_aut = Aut
eur.id_aut AND Rediger.id_ouvrage = Ouvrage.id_ouvrage ;
+-----------------+---------+
| titre           | nom_aut |
+-----------------+---------+
| L'Étranger      | Hugo    |
| Les Misérables  | Camus   |
| Les Misérables  | Zola    |
| Germinal        | Zola    |
+-----------------+---------+
4 rows in set (0,00 sec)

mysql>
```

Langage de définition des données (LDD)

CREATE DATABASE nom_de_la_base: créer une base de données.

```
mysql> CREATE DATABASE bibliotheque2;
Query OK, 1 row affected (0,02 sec)

mysql> =
```

```
mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| Bibliotheque1      |
| bibliotheque2      |
| information_schema |
| mysql              |
| performance_schema |
| phpmyadmin         |
| sys                |
+--------------------+
7 rows in set (0,00 sec)
```

DROP DATABASE nom_de_la_base : supprimer la base de données.

```
mysql> DROP DATABASE bibliotheque2;
Query OK, 0 rows affected (0,02 sec)

mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| Bibliotheque1      |
| information_schema |
| mysql              |
| performance_schema |
| phpmyadmin         |
| sys                |
+--------------------+
6 rows in set (0,00 sec)

mysql>
```

CREATE TABLE : créer une table (une relation).

ALTER

Modifier la définition d'une table, les mot cles :

RENAME: Changer le nom de la table, renommer une colonne ou une contrainte.

```
mysql> ALTER TABLE Auteur RENAME TO auteur;
Query OK, 0 rows affected (0,02 sec)

mysql>
```

ADD: Ajouter une colonne ou une contrainte.

```
mysql> ALTER TABLE Auteur ADD adr_aut VARCHAR(15);
Query OK, 0 rows affected (0,02 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

MODIFY: Modifier une colonne ou une contrainte.

```
mysql>
mysql> ALTER TABLE Auteur CHANGE nom_aut nom_auteur VARCHAR(15);
Query OK, 0 rows affected (0,02 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

```
mysql> ALTER TABLE Auteur MODIFY nom_auteur VARCHAR(50);
Query OK, 0 rows affected (0,01 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

DROP: Supprimer une colonne ou une contrainte.

```
mysql>
mysql> ALTER TABLE Auteur DROP COLUMN adr_aut;
Query OK, 0 rows affected (0,02 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

TRUNCATE

La commande TRUNCATE permet de supprimer toutes les données d'une table sans supprimer la table en elle-même. En d'autres mots, cela permet de purger la table.

```
mysql> TRUNCATE TABLE Adherent;
ERROR 1701 (42000): Cannot truncate a table referenced in a foreign key constraint (`Bibliotheque1`.`Ouvr
age`, CONSTRAINT `Ouvrage_ibfk_2`)
mysql>
```

INSERT

L'insertion de données dans une table s'effectue à l'aide de la commande INSERT INTO. Cette commande permet au choix d'inclure une seule ligne à la base existante ou plusieurs lignes d'un coup.

Insérer une ligne en spécifiant toutes les colonnes: INSERT INTO table VALUES ('valeur 1', 'valeur 2', ...)

```
mysql> INSERT INTO Adherent (id_ad, nom_ad, prenom_ad, adresse_ad, tel_ad) VALUES (4, 'Gassama', 'Fatou',
 'Pavillon Q', '778945612');
Query OK, 1 row affected (0,01 sec)
```

Insérer une ligne en spécifiant seulement les colonnes souhaitées :

INSERT INTO table (nom_colonne_1, nom_colonne_2, ...) VALUES ('valeur 1', 'valeur 2', ...)

```
mysql> INSERT INTO Adherent (nom_ad, prenom_ad, adresse_ad, tel_ad) VALUES
    ->     ('Ndao', 'Daour', 'Rufisque', '777777777');
Query OK, 1 row affected (0,00 sec)
```

Insertion de plusieurs lignes à la fois:

```
mysql> INSERT INTO Adherent (nom_ad, prenom_ad, adresse_ad, tel_ad) VALUES     ('Thiam', 'Fatou', '12 Rue
 Dakar', '766543219'),     ('Diop', "Mara", 'UCAD', '771112374');
Query OK, 2 rows affected (0,01 sec)
Records: 2  Duplicates: 0  Warnings: 0
```

exp

UPDATE

La commande UPDATE permet d'effectuer des modifications sur des lignes existantes. Très souvent cette commande est utilisée avec WHERE pour spécifier sur quelles lignes doivent porter la ou les modifications.

UPDATE table SET nom_colonne_1 = 'nouvelle valeur' WHERE condition

exp

DELETE

La commande DELETE en SQL permet de supprimer des lignes dans une table. La syntaxe pour supprimer des lignes est la suivante :

DELETE FROM table WHERE condition

```
mysql> DELETE FROM Adherent WHERE prenom_ad='Mara';
Query OK, 1 row affected (0,01 sec)
```

```
mysql>
```

exp

Attention : s'il n'y a pas de condition WHERE alors toutes les lignes seront supprimées et la table sera alors vide.

Langage d'interrogation de données (LID) : SELECT

Jointure

- INNER JOIN :

jointure interne pour retourner les enregistrements quand la condition est vrai dans les 2 tables. C'est l'une des jointures les plus communes.

Pour recuperer tout les adherent qui on emprunter un livre

```
mysql> SELECT * FROM Ouvrage INNER JOIN Adherent on Ouvrage.id_ad =Adherent.id_ad;
+------------+-----------------+----------+-------+-------+--------+---------------+------------+-----------+
| id_ouvrage | titre           | id_rayon | id_ad | id_ad | nom_ad | prenom_ad     | adresse_ad | tel_ad    |
+------------+-----------------+----------+-------+-------+--------+---------------+------------+-----------+
|          2 | Les Misérables  |        1 |     1 |     1 | Mbaye  | Djily         | Parcelle   | 771231234 |
|          3 | Les Misérables  |        1 |     1 |     1 | Mbaye  | Djily         | Parcelle   | 771231234 |
|          4 | L'Étranger      |        2 |     2 |     2 | Top    | Daour         | 26         | 781232342 |
|          5 | Germinal        |        1 |     3 |     3 | Diop   | Serigne Modou | Yoff       | 771122941 |
+------------+-----------------+----------+-------+-------+--------+---------------+------------+-----------+
4 rows in set (0,00 sec)

mysql>
```

- CROSS JOIN :

jointure croisée permettant de faire le produit cartésien de 2 tables. En d'autres mots, permet de joindre chaque lignes d'une table avec chaque lignes d'une seconde table. Attention, le nombre de résultats est en général très élevé.

```
mysql> SELECT nom_rayon, titre FROM Rayon CROSS JOIN Ouvrage;
+-----------------+-----------------+
| nom_rayon       | titre           |
+-----------------+-----------------+
| Science-fiction | Les Misérables  |
| Policier        | Les Misérables  |
| Littérature     | Les Misérables  |
| Science-fiction | Les Misérables  |
| Policier        | Les Misérables  |
| Science-fiction | Les Misérables  |
| Policier        | Les Misérables  |
| Littérature     | Les Misérables  |
| Science-fiction | Les Misérables  |
| Policier        | Les Misérables  |
| Science-fiction | L'Étranger      |
| Policier        | L'Étranger      |
| Littérature     | L'Étranger      |
| Science-fiction | L'Étranger      |
| Policier        | L'Étranger      |
| Science-fiction | Germinal        |
| Policier        | Germinal        |
| Littérature     | Germinal        |
| Science-fiction | Germinal        |
| Policier        | Germinal        |
+-----------------+-----------------+
20 rows in set (0,00 sec)

mysql>
```

- LEFT JOIN (ou LEFT OUTER JOIN) :

jointure externe pour retourner tous les enregistrements de la table de gauche (LEFT = gauche) même si la condition n'est pas vérifié dans l'autre table.

Les Adherent qui ont emprunter un livre

```
mysql> SELECT * FROM Adherent LEFT JOIN Ouvrage on Ouvrage.id_ad =Adherent.id_ad;
+-------+---------+--------------+--------------+-----------+------------+-----------------+----------+-------+
| id_ad | nom_ad  | prenom_ad    | adresse_ad   | tel_ad    | id_ouvrage | titre           | id_rayon | id_ad |
+-------+---------+--------------+--------------+-----------+------------+-----------------+----------+-------+
|     1 | Mbaye   | Djily        | Parcelle     | 771231234 |          3 | Les Misérables  |        1 |     1 |
|     1 | Mbaye   | Djily        | Parcelle     | 771231234 |          2 | Les Misérables  |        1 |     1 |
|     2 | Top     | Daour        | 26           | 781232342 |          4 | L'Étranger      |        2 |     2 |
|     3 | Diop    | Serigne Modou | Yoff         | 771122941 |          5 | Germinal        |        1 |     3 |
|     4 | Gassama | Fatou        | Pavillon Q   | 778945612 |       NULL | NULL            |     NULL |  NULL |
|     5 | Ndao    | Daour        | Rufisque     | 777777777 |       NULL | NULL            |     NULL |  NULL |
|     6 | Thiam   | Fatou        | 12 Rue Dakar | 766543219 |       NULL | NULL            |     NULL |  NULL |
+-------+---------+--------------+--------------+-----------+------------+-----------------+----------+-------+
7 rows in set (0,00 sec)

mysql>
```

Les Adherent qui n'ont pas emprunter de livre

```
;
+-------+---------+-----------+--------------+-----------+------------+-------+----------+-------+
| id_ad | nom_ad  | prenom_ad | adresse_ad   | tel_ad    | id_ouvrage | titre | id_rayon | id_ad |
+-------+---------+-----------+--------------+-----------+------------+-------+----------+-------+
|     4 | Gassama | Fatou     | Pavillon Q   | 778945612 |       NULL | NULL  |     NULL |  NULL |
|     5 | Ndao    | Daour     | Rufisque     | 777777777 |       NULL | NULL  |     NULL |  NULL |
|     6 | Thiam   | Fatou     | 12 Rue Dakar | 766543219 |       NULL | NULL  |     NULL |  NULL |
+-------+---------+-----------+--------------+-----------+------------+-------+----------+-------+
3 rows in set (0,00 sec)

mysql>
```

- RIGHT JOIN (ou RIGHT OUTER JOIN) :

jointure externe pour retourner tous les enregistrements de la table de droite (RIGHT = droite) même si la condition n'est pas vérifié dans l'autre table.

Les Adherent qui on fait un emprunt, NULL pour dire que l'adherent n'a pas fait d'emprunt

```
mysql> SELECT * FROM Ouvrage right JOIN Adherent on Ouvrage.id_ad =Adherent.id_ad;
+------------+-----------------+----------+-------+-------+---------+---------------+--------------+-----------+
| id_ouvrage | titre           | id_rayon | id_ad | id_ad | nom_ad  | prenom_ad     | adresse_ad   | tel_ad    |
+------------+-----------------+----------+-------+-------+---------+---------------+--------------+-----------+
|          3 | Les Misérables  |        1 |     1 |     1 | Mbaye   | Djily         | Parcelle     | 771231234 |
|          2 | Les Misérables  |        1 |     1 |     1 | Mbaye   | Djily         | Parcelle     | 771231234 |
|          4 | L'Étranger      |        2 |     2 |     2 | Top     | Daour         | 26           | 781232342 |
|          5 | Germinal        |        1 |     3 |     3 | Diop    | Serigne Modou | Yoff         | 771122941 |
|       NULL | NULL            |     NULL |  NULL |     4 | Gassama | Fatou         | Pavillon Q   | 778945612 |
|       NULL | NULL            |     NULL |  NULL |     5 | Ndao    | Daour         | Rufisque     | 777777777 |
|       NULL | NULL            |     NULL |  NULL |     6 | Thiam   | Fatou         | 12 Rue Dakar | 766543219 |
+------------+-----------------+----------+-------+-------+---------+---------------+--------------+-----------+
7 rows in set (0,00 sec)
```

- FULL JOIN (ou FULL OUTER JOIN) :

jointure externe pour retourner les résultats quand la condition est vrai dans au moins une des 2 tables.

- SELF JOIN :

permet d'effectuer une jointure d'une table avec elle-même comme si c'était une autre table.

- NATURAL JOIN :

jointure naturelle entre 2 tables s'il y a au moins une colonne qui porte le même nom entre les 2 tables SQL.

Pour recuperer les ouvrages que chaque adherent a emprunter

```
mysql> SELECT * FROM Ouvrage NATURAL JOIN Adherent;
+-------+------------+-----------------+----------+--------+---------------+------------+-----------+
| id_ad | id_ouvrage | titre           | id_rayon | nom_ad | prenom_ad     | adresse_ad | tel_ad    |
+-------+------------+-----------------+----------+--------+---------------+------------+-----------+
|     1 |          2 | Les Misérables  |        1 | Mbaye  | Djily         | Parcelle   | 771231234 |
|     1 |          3 | Les Misérables  |        1 | Mbaye  | Djily         | Parcelle   | 771231234 |
|     2 |          4 | L'Étranger      |        2 | Top    | Daour         | 26         | 781232342 |
|     3 |          5 | Germinal        |        1 | Diop   | Serigne Modou | Yoff       | 771122941 |
+-------+------------+-----------------+----------+--------+---------------+------------+-----------+
4 rows in set (0,00 sec)
```

- UNION JOIN : jointure d'union

Projection

exp

Sous requête

Dans le langage SQL une sous-requête (aussi appelé « requête imbriquée » ou « requête en cascade ») consiste à exécuter une requête à l'intérieur d'une autre requête.

- Requête imbriquée qui retourne un seul résultat

```
mysql>
mysql> SELECT * FROM Adherent WHERE  id_ad = (SELECT id_ad FROM Ouvrage LIMIT 1);
+-------+--------+-----------+------------+-----------+
| id_ad | nom_ad | prenom_ad | adresse_ad | tel_ad    |
+-------+--------+-----------+------------+-----------+
|     1 | Mbaye  | Djily     | Parcelle   | 771231234 |
+-------+--------+-----------+------------+-----------+
1 row in set (0,00 sec)

mysql>
```

- Requête imbriquée qui retourne une colonne

```
mysql> SELECT * FROM Adherent WHERE  id_ad IN  (SELECT id_ad FROM Ouvrage);
+-------+--------+---------------+------------+-----------+
| id_ad | nom_ad | prenom_ad     | adresse_ad | tel_ad    |
+-------+--------+---------------+------------+-----------+
|     1 | Mbaye  | Djily         | Parcelle   | 771231234 |
|     2 | Top    | Daour         | 26         | 781232342 |
|     3 | Diop   | Serigne Modou | Yoff       | 771122941 |
+-------+--------+---------------+------------+-----------+
3 rows in set (0,00 sec)
```

Alias (AS)

Dans le langage SQL il est possible d'utiliser des alias pour renommer temporairement une colonne ou une table dans une requête. Cette astuce est particulièrement utile pour faciliter la lecture des requêtes.

- Alias sur une colonne

Permet de renommer le nom d'une colonne dans les résultats d'une requête SQL.

```
mysql> SELECT prenom_ad AS prenom, nom_ad AS nom FROM Adherent;
+---------------+---------+
| prenom        | nom     |
+---------------+---------+
| Djily         | Mbaye   |
| Daour         | Top     |
| Serigne Modou | Diop    |
| Fatou         | Gassama |
| Daour         | Ndao    |
| Fatou         | Thiam   |
+---------------+---------+
6 rows in set (0,00 sec)
```

- Alias sur une table

Permet d'attribuer un autre nom à une table dans une requête SQL. Cela peut aider à avoir des noms plus court, plus simple et plus facilement compréhensible.

```
ysql> SELECT * FROM Adherent AS AD NATURAL JOIN Ouvrage as OUV;
```

Commande

UNION

GROUP BY

La commande GROUP BY est utilisée en SQL pour grouper plusieurs résultats et utiliser une fonction de totaux sur un groupe de résultat.

De façon générale, la commande GROUP BY s'utilise de la façon suivante :

SELECT colonne1, fonction(colonne2) FROM table GROUP BY colonne1

Remarque: cette commande doit toujours s'utiliser après la commande WHERE et avant la commande HAVING.

ORDER BY

La commande ORDER BY permet de trier les lignes dans un résultat d'une requête SQL. Il est

possible de trier les données sur une ou plusieurs colonnes, par ordre ascendant ou descendant.

Une requête où l'ont souhaite filtrer l'ordre des résultats utilise la commande ORDER BY de la sorte :

SELECT colonne1, colonne2 FROM table ORDER BY colonne1

```
mysql> SELECT * FROM Adherent ORDER BY nom_ad;
+-------+---------+---------------+--------------+-----------+
| id_ad | nom_ad  | prenom_ad     | adresse_ad   | tel_ad    |
+-------+---------+---------------+--------------+-----------+
|     3 | Diop    | Serigne Modou | Yoff         | 771122941 |
|     4 | Gassama | Fatou         | Pavillon Q   | 778945612 |
|     1 | Mbaye   | Djily         | Parcelle     | 771231234 |
|     5 | Ndao    | Daour         | Rufisque     | 777777777 |
|     6 | Thiam   | Fatou         | 12 Rue Dakar | 766543210 |
```

```
|      6 | Thiam    | Fatou          | 12 Rue Dakar | 786543219 |
|      2 | Top      | Daour          | 26           | 781232342 |
+--------+----------+----------------+--------------+-----------+
6 rows in set (0,00 sec)
```

Remarque: Les résultats sont toujours classé par ordre ascendant par défaut (DESC pour ordre descandant).

HAVING

La condition HAVING en SQL est presque similaire à WHERE à la seule différence que HAVING permet de filtrer en utilisant des fonctions telles que SUM(), COUNT(), AVG(), MIN() ou MAX().

L'utilisation de HAVING s'utilise de la manière suivante :

SELECT colonne1, SUM(colonne2) FROM nom_table GROUP BY colonne1 HAVING fonction(colonne2) operateur valeur

Important : HAVING est très souvent utilisé en même temps que GROUP BY bien que ce ne soit pas obligatoire.

Fonction

DISTINCT

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 simplement ajouter DISTINCT après le mot SELECT.

SELECT DISTINCT ma_colonne FROM nom_du_tableau

```
mysql> SELECT DISTINCT titre FROM Ouvrage;
+-----------------+
| titre           |
+-----------------+
| Les Misérables  |
| L'Étranger      |
| Germinal        |
+-----------------+
3 rows in set (0,01 sec)
```

COUNT

Il permet de compter le nombre de lignes dans une table qui répondent à une condition donnée.

LIMIT

Il permet de limiter le nombre de résultats retournés par une requête SELECT.

LIKE %% (B%A)

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

Le modele B%A est utilisé pour rechercher tous les enregistrement qui commence par un «B» et se termine par un «A».

SELECT * FROM table WHERE colonne LIKE modele

IN & NOT IN

Il permet de filtrer les résultats d'une requête SELECT en spécifiant une liste de valeurs ou de sous-requêtes à inclure ou à exclure.

SELECT nom_colonne FROM table WHERE nom_colonne IN (valeur1, valeur2, valeur3, ...)

BETWEEN

Permet de sélectionner les résultats d'une requête SELECT où une colonne se situe entre deux valeurs données

SELECT * FROM table WHERE nom_colonne BETWEEN 'valeur1' AND 'valeur2'

AVG

permet de calculer la moyenne des valeurs numériques dans une colonne.

MIN

permet de trouver la plus petite valeur dans une colonne.

MAX

permet de trouver la plus grande valeur dans une colonne.

SUM

permet de calculer la somme des valeurs numériques dans une colonne.