

# SQL : requêtes simples

Session BRE02 - 21/05/2024

# MySQL et SQL

MySQL est un SGBD ( Système de Gestion de Base de Données), plus exactement c'est un SGBDR (Système de Gestion de Bases de Données Relationnelles).

Pourquoi on utilise celui-ci ? Parce que c'est le plus connu et le plus courant 😊.

Le SQL (Structured Query Language) est un langage qui nous permet de discuter avec MySQL. Le SQL fonctionne au moyen de requêtes.

On envoie une requête à la base de données qui nous retourne un résultat.

# Tables et colonnes

Dans MySQL nos informations sont séparées en tables dont les champs sont appelés des colonnes.

<b>id</b>	<b>first_name</b>	<b>last_name</b>
1	Alain	Pératif
2	Sarah	Toustra

# SELECT : Lire dans la base de données

SELECT c'est le mot clé qui nous permet d'envoyer une requête pour rechercher quelque chose dans une base.

# La table d'exemple

id	title	year	director
1	Alien, le 8ème passager	1979	Ridley Scott
2	Aliens, le retour	1986	James Cameron
3	Alien 3	1992	David Fincher
4	Alien, la résurrection	1997	Jean-Pierre Jeunet
5	Titanic	1998	James Cameron
6	Gladiator	2000	Ridley Scott
7	Le fabuleux destin d'Amélie Poulain	2001	Jean-Pierre Jeunet

# Sélectionner une colonne

```
SELECT nom_de_la_colonne FROM nom_de_la_table;
```

```
SELECT title FROM movies;
```

title
Alien, le 8ème passager
Aliens, le retour
Alien 3
Alien, la résurrection
Titanic
Gladiator
Le fabuleux destin d'Amélie Poulain

# Sélectionner plusieurs colonnes

```
SELECT nom_de_la_colonne1, nom_de_la_colonne2 FROM nom_de_la_table;
```

```
SELECT id, title FROM movies;
```

id	title
1	Alien, le 8ème passager
2	Aliens, le retour
3	Alien 3
4	Alien, la résurrection
5	Titanic
6	Gladiator
7	Le fabuleux destin d'Amélie Poulain

# Sélectionner toutes les colonnes

```
SELECT * FROM nom_de_la_table;
```

```
SELECT * FROM movies;
```

id	title	year	director
1	Alien, le 8ème passager	1979	Ridley Scott
2	Aliens, le retour	1986	James Cameron
3	Alien 3	1992	David Fincher
4	Alien, la résurrection	1997	Jean-Pierre Jeunet
5	Titanic	1998	James Cameron
6	Gladiator	2000	Ridley Scott
7	Le fabuleux destin d'Amélie Poulain	2001	Jean-Pierre Jeunet



# SELECT WHERE : lire avec des conditions

Si on veut sélectionner des colonnes sous certaines conditions, il faut utiliser une clause `WHERE` dans notre `SELECT`.

```
SELECT nom_de_la_colonne WHERE nom_de_la_colonne[condition];
```

## Exemple : les films de James Cameron

```
SELECT * FROM movies WHERE director = "James Cameron";
```

id	title	year	director
2	Aliens, le retour	1986	James Cameron
5	Titanic	1998	James Cameron

## Exemples 2 : les films après 1999

```
SELECT * FROM movies WHERE year > 1999;
```

id	title	year	director
6	Gladiator	2000	Ridley Scott
7	Le fabuleux destin d'Amélie Poulain	2001	Jean-Pierre Jeunet

# WHERE ... AND pour utiliser plusieurs conditions

Tous les films de James Cameron après 1990 :

```
SELECT * FROM movies WHERE director = "James Cameron" AND year > 1990;
```

id	title	year	director
5	Titanic	1998	James Cameron

**SELECT ... ORDER BY** pour modifier l'ordre

SELECT colonne FROM table ORDER BY colonne ASC | DESC;

# Exemple : les films du plus récent au plus ancien

```
SELECT * FROM movies ORDER BY year DESC;
```

id	title	year	director
7	Le fabuleux destin d'Amélie Poulain	2001	Jean-Pierre Jeunet
6	Gladiator	2000	Ridley Scott
5	Titanic	1998	James Cameron
4	Alien, la résurrection	1997	Jean-Pierre Jeunet
3	Alien 3	1992	David Fincher
2	Aliens, le retour	1986	James Cameron
1	Alien, le 8ème passager	1979	Ridley Scott

# LIMIT : pour obtenir un nombre de résultats maximum

SELECT colonne FROM table LIMIT nombre;

SELECT \* FROM movies LIMIT 3;

id	title	year	director
1	Alien, le 8ème passager	1979	Ridley Scott
2	Aliens, le retour	1986	James Cameron
3	Alien 3	1992	David Fincher

# Et on peut mélanger tout ça

Les films de Ridley Scott du plus récent au plus ancien :

```
SELECT * from movies WHERE director = "Ridley Scott" ORDER BY year DESC;
```

id	title	year	director
6	Gladiator	2000	Ridley Scott
1	Alien, le 8ème passager	1979	Ridley Scott



## Exemple : 3 films de James Cameron ou Ridley Scott

```
SELECT * FROM movies WHERE director = "Ridley Scott" OR director = "James Cameron" LIMIT 3;
```

id	title	year	director
1	Alien, le 8ème passager	1979	Ridley Scott
2	Aliens, le retour	1986	James Cameron
5	Titanic	1998	James Cameron

# Exercice 1

Que va donner cette requête :

```
SELECT * FROM movies WHERE director = "Jean-Pierre Jeunet";
```

# Réponse exercice 1

id	title	year	director
4	Alien, la résurrection	1997	Jean-Pierre Jeunet
7	Le fabuleux destin d'Amélie Poulain	2001	Jean-Pierre Jeunet

## Exercice 2

Que va donner cette requête ?

```
SELECT id, title FROM movies WHERE year > 1990 AND year < 2000;
```

## Réponse de l'exercice 2

id	title
3	Alien 3
4	Alien, la résurrection
5	Titanic

## Exercice 3

Ecrire une requête pour récupérer les titres des 5 films les plus récents de la liste.

## Réponse Exercice 3

```
SELECT title FROM movies ORDER BY year DESC LIMIT 5
```

# INSERT : ajouter des données

Pour ajouter une ligne à une table il faut utiliser une requête INSERT.

```
INSERT INTO nom_de_la_table (nom_colonne1, nom_colonne2, ...)
```

```
VALUES(valeur_colonne1, valeur_colonne2, ...);
```

```
INSERT INTO movies (id, title, year, director)
```

```
VALUES(NULL, "Avatar", 2010, "James Cameron");
```



# UPDATE : modifier des données

Pour modifier des données, il faut utiliser une requête UPDATE.

```
UPDATE nom_de_la_table  
SET nom_de_la_colonne = nouvelle_valeur  
WHERE colonne = valeur;
```

```
UPDATE movies  
SET year = 2009  
WHERE title = "Avatar";
```

# DELETE : supprimer des données

Pour supprimer des données il faut utiliser une requête DELETE.

```
DELETE FROM nom_de_la_table WHERE colonne = valeur;
```

```
DELETE FROM movies WHERE title = "Avatar";
```

Attention à mettre une clause WHERE précise pour ne pas effacer des données par erreur.