Langage SQL
Requêtes d'interrogation d'une base de données

SQL (pour Structured Query Language) est un langage informatique développé depuis 1974, permettant d'exploiter les bases de données relationnelles pour y rechercher de l'information, y ajouter de l'information ou y supprimer de l'information.

MySQL est un SGBD utilisant SQL.

Les SGBD comme Oracle Database ou Microsoft SQL Server utilisent aussi SQL.

Nous allons découvrir par l'exemple (en ligne) quelques requêtes d'interrogation en travaillant ici `https://www.semwebtech.org/sqlfrontend/`

Il est conseillé d'écrire la requête dans le formulaire puis de cliquer sur **send** pour se familiariser avec SQL

On voit ci-dessous l'écriture d'une requête

Sur ce site **les requêtes sont insensibles à la casse** ce qui signifie que l'on peut écrire aussi bien SELECT * FROM COUNTRY que select *from country

Database Unit at the Institute for Informatics, University of Göttingen

Playground for SQL Queries

With this form, you can state SQL queries (SELECT and DESCRIBE) against the Mondial Database.
The database is used in the lectures

- Databases
- Introduction to Databases and Database Programming in SQL/Oracle

```
select name, capital from country
```

send reset ☐ show query plan

Results: 244

NAME	CAPITAL
Albania	Tirana
Greece	Athina
North Macedonia	Skopje

1. **Obtenir la totalité d'une table**

   ```
   SELECT * FROM COUNTRY
   ```

2. **Obtenir des colonnes particulières (attributs) d'une table (par leur nom)**

   ```
   SELECT NAME,CAPITAL FROM COUNTRY
   ```

3. La requête

```
SELECT INFLATION FROM ECONOMY
```

crée des doublons (entourés)

Results: 244

INFLATION
1.7
-0.8
2.8
2.2
4
1.8
1.1
0.9
1.8
1.4
1.7
1.6
1.9

Pour ne pas avoir de doublons on utilise le mot clé DISTINCT (mais cela a un coût algorithmique)

```
SELECT DISTINCT INFLATION FROM ECONOMY
```

Results: 100

INFLATION
2.2
0.9
1.9
1
2.3
null
4.4
-0.5
4.1
5.2
25
7.8
0.6

4. **Obtenir des lignes particulières (n-uplets) d'une table sous condition**

```
SELECT * FROM COUNTRY WHERE POPULATION > 60000000
```

On veut afficher les n-uplets des pays dont la population est strictement supérieur à 60 millions.

5. **Obtenir des colonnes et des lignes particulières d'une table sous condition**

```
SELECT NAME,CAPITAL FROM COUNTRY WHERE POPULATION > 60000000
```

6. **Variable Objet et attribut** On considère une table comme un **objet** (voir programmation objet)

   Ainsi dans notre exemple, on peut **déclarer** une variable c de type COUNTRY et on peut considérer que l'objet de type COUNTRY a comme attributs name et population d'où l'écriture

   c.name et c.population

```
SELECT c.name, c.population FROM COUNTRY c WHERE c.population > 60000000
```

Results: 22

NAME	POPULATION
France	64300821
Germany	82521653
Russia	143666931
Turkey	75627384
United Kingdom	64105654
China	1427647786
Iran	79926270
Pakistan	207776954
Bangladesh	149772364
India	1210854977
Thailand	65981659
Vietnam	88772900
Egypt	94798827
Indonesia	252124458
Japan	127094745
Philippines	100981437
Mexico	112336538
United States	318857056
Brazil	202768562
Nigeria	193392500
Zaire	86026000
Ethiopia	84320987

7. Si on veut ordonner les n-uplets dans l'ordre croissant de la population

```
SELECT NAME,CAPITAL FROM COUNTRY WHERE POPULATION > 60000000 ORDER BY POPULATION ASC
```

NAME	POPULATION
Tanzania	61741120
France	64300821
United Kingdom	66980000
Thailand	67993000
Iran	79926270
Germany	82521653
Turkey	84680273
Congo, Dem.Rep.	86026000
Egypt	94798827
Vietnam	96208984
Ethiopia	105164000
Philippines	109035343
Mexico	126014024
Japan	126146099
Russia	144699673
Bangladesh	165158616
Nigeria	193392500
Brazil	202768562
Pakistan	207776954
Indonesia	270203917
United States	331449281
India	1210854977
China	1411778724

Si on veut ordonner les n-uplets dans l'ordre décroissant de la population

```
SELECT NAME,CAPITAL FROM COUNTRY WHERE POPULATION > 60000000 ORDER BY POPULATION DESC
```

NAME	POPULATION
China	1411778724
India	1210854977
United States	331449281
Indonesia	270203917
Pakistan	207776954
Brazil	202768562
Nigeria	193392500
Bangladesh	165158616
Russia	144699673
Japan	126146099
Mexico	126014024
Philippines	109035343
Ethiopia	105164000
Vietnam	96208984
Egypt	94798827
Congo, Dem.Rep.	86026000
Turkey	84680273
Germany	82521653
Iran	79926270
Thailand	67993000
United Kingdom	66980000
France	64300821
Tanzania	61741120

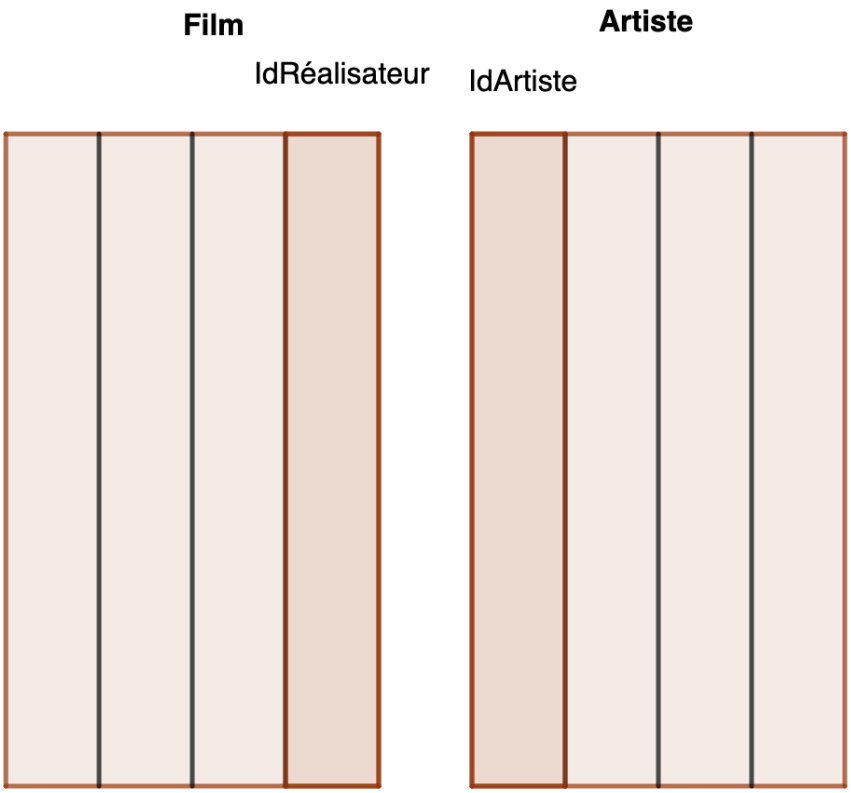
On verra plus loin que les requêtes peuvent être imbriquées, ce qui signifie qu'une requête peut utiliser le résultat d'une autre requête, **tant que le résultat n'est pas ordonné**

8. **Retenir** qu'une requête sql doit être comprise dans l'ordre :
   FROM objets **WHERE** conditions **SELECT** affichage

9. **Jointure**

   On peut se poser la question de l'intérêt de fragmenter l'information en tables. Cette séparation n'est pas définitive car on peut faire un lien entre les tables par l'opération de **jointure**

   Dans la base de données Films on souhaite joindre à chaque titre de Film le prénom et le nom du réalisateur



On réalise l'opération de jointure entre les tables Film et Artiste par :

FROM Film join Artiste

et en rajoutant comme condition (la clé étrangère d'une table est égale à la clé primaire de l'autre)

ON idRéalisateur = idArtiste

ensuite on précise ce que l'on veut comme colonnes

SELECT titre, prénom, nom

ce qui donne en tout

```
SELECT titre, prénom, nom FROM Film JOIN Artiste ON idRéalisateur = idArtiste
```

Tester cette requête ici -> `http://deptfod.cnam.fr/bd/tp/requetes/`

On obtient pour les premières lignes

Résultat

titre	prénom	nom
La Guerre des étoiles	George	Lucas
Kill Bill : Volume 1	Quentin	Tarantino
Apocalypse Now	Francis	Ford Coppola
Impitoyable	Clint	Eastwood
Eternal Sunshine of the Spotless Mind	Michel	Gondry
A History of Violence	David	Cronenberg
2001 l'Odyssée de l'espace	Stanley	Kubrick
La Guerre des Mondes	Steven	Spielberg
Memento	Christopher	Nolan
Blade Runner	Ridley	Scott
Les Aventuriers de l'arche perdue	Steven	Spielberg
Indiana Jones et le temple maudit	Steven	Spielberg
Indiana Jones et la dernière croisade	Steven	Spielberg
Gladiator	Ridley	Scott

On souhaite afficher pour chaque pays le taux de chômage

Le taux de chômage se trouve dans la relation economy et dans cette dernière il n'y a pas un nom de pays mais un code constitué de un, deux ou trois lettres

Ce code se trouve aussi dans la table country qui a pour attribut le nom du pays

On va donc faire une jointure

```
select name, unemployment from economy join country on country = code
```

NAME	UNEMPLOYMENT
Austria	4.9
Akrotiri and Dhekelia	null
Afghanistan	35
Antigua and Barbuda	11
Albania	16.9
Andorra	4
Angola	null
Armenia	17.3
Aruba	6.9
American Samoa	29.8
Australia	5.7
Anguilla	8
Azerbaijan	6
Belgium	8.8
Bangladesh	5
Barbados	11.4
Benin	null
Burkina Faso	77
Bulgaria	11.6
Bhutan	2.1
Burundi	null
Bosnia and Herzegovina	44.3

Remarques

(a) Dans certaines situations la clé étrangère et la clé primaire peuvent être identiques (ce qu'on essaie d'éviter)

Par exemple dans la base Films dont le schéma relationnel est :

Film(idFilm,titre,année,genre,résumé,#idRéalisateur,#codePays)
Pays(code,nom,langue)
Artiste(idArtiste,nom,prénom,annéeNaiss)
Rôle(#idFilm,#idActeur,nomRôle)
Internaute(email,nom,prénom,région)
Notation(#email, #idFilm,note)

La clé primaire de la relation Film s'écrit comme une clé étrangère de la relation Rôle.

Comment va-t-on écrire en SQL la requête :

Dans quels films Bruce Willis a-t-il joué le rôle de John McClane ?

Procédons par étapes

i. On peut commencer par faire une jointure entre les tables Film et Rôle

pour *afficher tous les titres de films de la base de données dans lesquels il y a le rôle 'John McClane', ainsi que les identifiants de films.*

Pour différencier la clé primaire idFilm de la relation Film de la clé étrangère idFilm de la relation Rôle on va écrire en notation objet

Film.idFilm = Rôle.idFilm

ce qui donne

```
select idFilm, titre from Rôle join Film on Rôle.idFilm = Film.idFilm where nomRôle = 'John McClane'
```

idFilm	titre
562	Piège de cristal
1571	Die Hard 4 : Retour en enfer
1572	Une Journée en enfer
1573	58 minutes pour vivre

ii. Ensuite on cherche parmi ces films uniquement ceux dans lesquels le rôle de 'John McClane' a été joué par Bruce Willis.

Ce qui s'écrit ainsi

```
select titre from Rôle join Film on Rôle.idFilm = Film.idFilm join Artiste on idActeur = idArtiste where nomRôle = 'John McClane' and nom = 'Willis' and prénom = 'Bruce'
```

On a ajouté en gras la jointure entre Rôle et Artiste et on obtient

idFilm	titre
562	Piège de cristal
1571	Die Hard 4 : Retour en enfer
1572	Une Journée en enfer
1573	58 minutes pour vivre

La requête précédente peut s'écrire de différentes manières (Voir la solution proposée sur le site du CNAM)

On dit que SQL est un langage **déclaratif** dans le sens où il permet d'écrire des requêtes mais la façon d'écrire ces requêtes n'ont aucun impact sur la performance des requêtes

Ce qui n'est pas le cas du langage Python où un programme peut être plus performant qu'un autre pour résoudre le même problème

10. **Fonctions d'agrégation**

Il existe des fonctions permettant de faire des statistiques.

(a) **la fonction count()**

Si on veut compter le nombre de notes strictement supérieures à 9 dans la table Notation

```
select count(note) as nbNotes from Notation where note > 9
```

(b) **la fonction avg()**

Si on veut compter la note moyenne dans la table Notation

```
select avg(note) as moyenne from Notation
```

11. **Requêtes imbriquées**

Etant donné qu'une requête (sans order by) produit un ensemble de n-uplets, on peut imbriquer des requêtes

Par exemple

```
select count(*) as nbLangues from (select name,percentage from language where country = 'F')
```

On voit ici deux requêtes imbriquées :

La requête entre parenthèses

```
select name,percentage from language where l.country = 'F'
```

crée un ensemble de n-uplets E et sur cet ensemble E on exécute une autre requête voilà pourquoi il y a les parenthèses

1 Exercices : Langage SQL . Interrogation

Rédiger des requêtes en SQL pour obtenir les informations suivantes de la base de données DATABASE Mondial dont le schéma relationnel est sur Pronote

Ex 1

Il s'agit de trouver les noms des pays dont les codes sont les suivants :
COCO, XMAS, GROX, SMAR, SBAR

Ex 2

1. La liste de tous les pays et de leur population trié par ordre croissant de la population
2. La liste de tous les pays et de leur population trié par ordre décroissant de la population
3. Le nom des dix pays ayant la plus petite superficie

Ex 3

1. La liste de tous les aéroports dans le monde ? Combien y-a-t-il d'aéroports ?
2. La liste des aéroports (noms) situés en France
3. La liste des aéroports (noms) situés à plus de 2000 m d'altitude

Ex 4

1. Quels sont tous les fleuves qui se jettent dans la mer Méditerranée ? Combien sont ils ? (triés dans l'ordre décroissant de leur longueur)
2. Quel est le fleuve le plus court ? le plus long ?

Ex 5

Quelle est la mer la plus profonde ? la moins profonde ?

Ex 6

Quel est le désert le plus vaste et où se trouve-t-il ?(le moins vaste ?)

Ex 7

1. La liste de toutes les capitales dont la latitude est supérieure à 48 degrés
2. La liste de toutes les capitales dont la latitude est supérieure à celle de Paris
3. La liste de toutes les capitales européennes dont la latitude est supérieure à celle de Paris

Ex 8

L'île de la Réunion (dans la base de données le nom est Reunion) est une île volcanique ayant un sommet à plus de 3000 mètres.

1. Quelles sont toutes les montagnes situées en Islande (nom et altitude) ?
2. Quelles sont toutes les montagnes situées sur une île (nom, altitude, nom de l'île) ayant un sommet à plus de 3000 mètres ?

Ex 9

L'île Maurice , située dans l'Océan Indien à 200 km à peu près de l'île de la Réunion est une île "densément peuplée" (dans la base de données le nom est Mauritius)

1. Rédiger une requête permettant de donner du sens à cette phrase. On pourra utiliser la fonction d'agrégation AVG pour average qui permet de calculer une moyenne et calculer la densité moyenne des pays dans la table country puis comparer la densité de l'île Maurice à la densité moyenne
2. Rédiger une requête pour avoir les 10 premiers pays européens les plus densément peuplés

Ex 10

On aimerait avoir :

1. La liste de tous les pays (leurs noms) limitrophes à la France ainsi que la longueur de la frontière
2. La liste de tous les pays (leurs noms) limitrophes à la France ainsi que la longueur de la frontière, classée dans l'ordre croissante (décroissante)
3. La longueur totale des frontières terrestres de la France

Ex 11

La Loire est le plus long fleuve de France (à peu près 1000 km)

Quels sont les fleuves dans le Monde de plus de 1000 km de longueur ? Combien sont ils ?

Ex 12

La Russie est un pays qui est sur deux continents, l'Europe (pour 25 % de son territoire et l'Asie pour 75 %)

Quels sont les autres pays ? (Afficher les noms des pays (pas les codes) les continents et les pourcentages)

Ex 13

1. Afficher la liste des pays d'Europe avec leur PIB, la part de l'industrie, des services et de l'agriculture, et le taux de chômage
2. Calculer la moyenne du taux de chômage en Europe continentale. Afficher les pays européens dont le taux de chômage est inférieure à cette moyenne
3. Calculer la moyenne de la part de l'industrie pour les pays européens.Afficher les pays dont la part de l'industrie est supérieure à la moyenne en Europe
4. Faire de même avec l'agriculture et les services

Ex 14

Afficher les noms des villes et des pays que traversent les fleuves suivants :

1. La Loire (Loire dans la base de données)
2. Le Nil (Nile)
3. Le Rhin (Rhein)
4. Le Danube (Donau)

Ex 15

Le Loch Ness est un lac situé sur une île, la Grande-Bretagne

1. Afficher tous les lacs situés en Grande-Bretagne
2. Afficher dans l'ordre décroissant les lacs occupant la plus grande surface dans l'île en proportion