

# 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 relationnelle pour y rechercher de l'information, y ajouter de l'information ou y supprimer de l'information

Nous allons découvrir par l'exemple (en ligne) quelques requêtes d'interrogation en travaillant ici <http://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
```

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. Une des règles dans l'utilisation des bases de données est de conserver l'unicité des n-uplets et par conséquent de ne pas avoir de doublons

Par exemple 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
```

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

On veut toutes les langues parlées en France et les pourcentages associées

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

## Results: 16

NAME	PERCENTAGE
French	87.2
Occitan	1
Alsatian	0.8
Breton	0.3
Corsu	0.2
Catalan	0.1
Basque	0.1
Arabic	3.6
Berber	3.5
Spanish	1.2
Italian	1.2
Portuguese	1.1
English	0.5
Vietnamese	0.5
Turkish	0.3
Kurdish	0.3

On verra plus loin des fonctions statistiques qui permettent par exemple de calculer des sommes et ainsi vérifier si la somme des pourcentages vaut 100 etc...

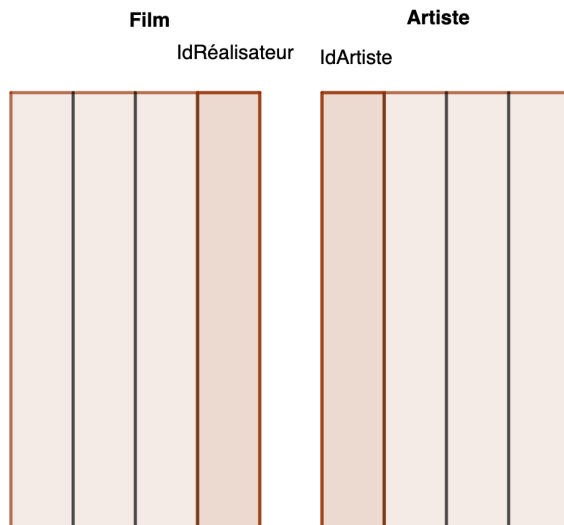
**Retenir** qu'une requête sql doit être comprise dans l'ordre :

**FROM** objets **WHERE** conditions **SELECT** affichage

### 7. 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 Film 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 cross join Artiste**

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

**WHERE 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 cross join Artiste WHERE idRéalisateur = idArtiste
```

On obtient pour les premières lignes

<b>Résultat</b>		
<b>titre</b>	<b>prénom</b>	<b>nom</b>
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 (avec un alias)

```
select name, unemployment from economy as e cross join country as c where e.country = c.code
```

ou encore

```
select name, unemployment from economy e cross join country c where e.country = c.code
```

### On souhaite trouver tous les pays dont le PIB est supérieur ou égal à celui de la France

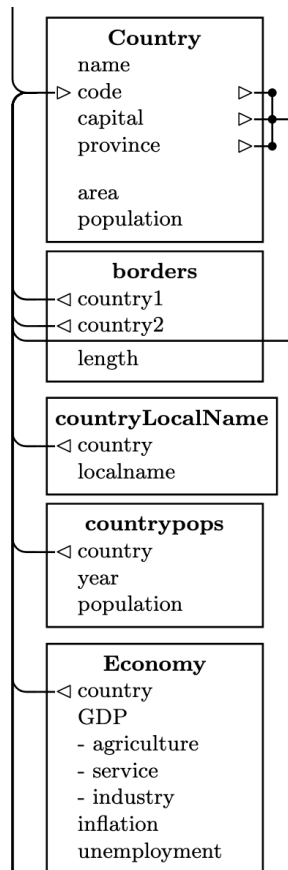
La requête suivante ne fournira les pays qu'à partir de leurs **codes** des pays or nous ne connaissons pas bien à priori les codes des pays

```
select e1.country, e1.GDP from economy e1, economy e2 where e2.country = 'F' and e1.GDP >= e2.GDP
```

## Results: 5

COUNTRY	GDP
F	2739000
D	3593000
CN	9330000
J	5007000
USA	16720000

Or le **lien code - nom** se trouve dans la table **Country** et comme le code se trouve à la fois dans la table **Country** et la table **Economy**



on va faire la "jointure" entre les deux tables par l'intermédiaire du code des pays ainsi

```
select c.name, e1.GDP from country c, economy e1, economy e2 where e2.country = 'F' and e1.GDP >= e2.GDP and c.code = e1.country
```

## Results: 5

NAME	GDP
China	9330000
France	2739000
Germany	3593000
Japan	5007000
United States	16720000

La requête suivante permet d'ordonner le PIB par ordre décroissant

```
select c.name, e1.GDP from country c, economy e1, economy e2 where e2.country = 'F' and e1.GDP >= e2.GDP and c.code = e1.country order by GDP desc
```

## Results: 5

NAME	GDP
United States	16720000
China	9330000
Japan	5007000
Germany	3593000
France	2739000

### 8. Fonctions d'agrégation

On a déjà vu la requête permettant d'avoir toutes les langues parlées en France

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

On aimerait :

- (a) compter le nombre de langues parlées
- (b) vérifier que la somme des pourcentages est bien égal à 100
- (c) afficher dans l'ordre (croissant ou décroissant) les langues en fonction des pourcentages

### Réponses

- (a) 

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

On voit ici deux requêtes imbriquées :

La requête

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

crée une table et sur cette table on exécute une autre requête voilà pourquoi il y a les parenthèses

- (b) 

```
select count(*) as nbLangues,sum(pourcentage) as somme from (select l.name,l.pourcentage from language l where l.country = 'F')
```
- (c) 

```
select l.name,l.pourcentage from language l where l.country = 'F' order by pourcentage desc
```



# 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