
Manipulations de base

Faisons ensemble les manipulations données en exemple et essayez de bien comprendre la structure des requêtes car après cette découverte vous aurez une activité évaluée.

1. SELECT / FROM :

1.a. Pour obtenir la totalité d'une table, par exemple la table country, on utilise la commande :
`SELECT * FROM Country`

1.b. Pour obtenir des colonnes particulières (attributs) d'une table :
`SELECT Name, Capital FROM Country`

2. DISTINCT

Voici deux exemples côte à côte :

1	<code>SELECT Name FROM Language</code>	1	<code>SELECT DISTINCT Name FROM Language</code>
1	Afghan Persian	1	Afghan Persian
2	Afrikaans	2	Afrikaans
3	Albanian	3	Albanian
4	Albanian	4	Amerindian
5	Albanian	5	Arabic
6	Albanian	6	Armenian
7	Albanian	7	Aymara
8	Amerindian	8	Azeri
9	Arabic	9	Balochi
10	Arabic	10	Bambara

L'exécution s'est terminée sans erreur. Résultat : 283 enregistrements ramenés en 9ms À la ligne 1 : SELECT Name FROM Language	L'exécution s'est terminée sans erreur. Résultat : 104 enregistrements ramenés en 7ms À la ligne 1 : SELECT DISTINCT Name FROM Language
---	--

Que permet de faire la commande **DISTINCT** ?

.....

3. WHERE

Pour obtenir des lignes (t-uplets) vérifiant une / plusieurs condition(s) on utilise **WHERE** :
`SELECT * FROM Country WHERE Population > 60000000`

4. ORDER BY

Pour obtenir des lignes triées on utilise **ORDER BY** suivi de **ASC** (croissant) ou **DESC** (décroissant)
`SELECT * FROM Country ORDER BY Area ASC`

Quelques exemples de requêtes SQL : <https://www.youtube.com/watch?v=PTxpVhyMg60>

Exercice 1 :

Rédiger une **requête SQL** pour obtenir :

1. la liste des pays dont la population excède 1 000 000 000 d'habitants
2. la même liste triée par ordre alphabétique
3. la liste des pays et de leurs populations respectives, triée par ordre décroissant de leur superficie
4. la liste des pays ayant moins de 10 habitants au km²

Jointures

L'intérêt d'une base de données réside en particulier dans la possibilité de **croiser des informations** présentes dans plusieurs tables par l'intermédiaire d'une **jointure**.

```
SELECT * FROM country
JOIN encompasses ON country.code = encompasses.country;
```

On remarquera qu'il s'agit d'une seule requête (on pourrait tout mettre sur la même ligne) :

1^{ère} ligne on indique que l'on veut récupérer **tous les attributs (*)** de la jointure

2^{ème} ligne on voit apparaître la façon dont on réunit les tables (ici *country est une clé étrangère de encompasses et correspond à la clé primaire code de country*)

On peut ensuite rajouter des critères comme par exemple obtenir les pays qui se situent sur le continent européen :

```
SELECT * FROM country
JOIN encompasses ON country.code = encompasses.country
WHERE encompasses.continent = 'Europe';
```

On remarquera également qu'une jointure peut être réalisée sur 3 tables : **encompasses signifie englobe**

```
SELECT Country.Name, Country.Capital FROM Country
JOIN City ON Country.Code = City.Country
JOIN encompasses ON Country.Code = encompasses.Country
WHERE encompasses.continent = 'Europe'
AND City.Name = Country.Capital
AND CITY.Latitude > 60;
```

D'autres exemples ici : https://www.youtube.com/watch?v=cHRq_WuFdB8&t=2s

Exercice 2 :

Rédiger une **requête SQL** pour obtenir :

1. le nom des pays qui sont à cheval sur plusieurs continents
2. les pays du continent américain qui comptent moins de 10 habitants par km²

Pour les 3 questions suivantes, on pourra se servir de la relation language

3. les pays du monde où on parle français (French)
4. les langues parlées par moins de 30 000 personnes dans le monde et leur pays respectif
5. la liste ordonnée pour le continent africain des langues (et du pays concerné), on précisera pour chacune d'entre elles le nombre de personnes qui la parlent.

Enfin une dernière vidéo sur les requêtes SQL :

<https://www.youtube.com/watch?v=C80KcsdzywM>

1. Déclarer une variable.

L'intérêt de cette manipulation est de raccourcir les notations.

En reprenant l'exemple précédent, on peut par exemple écrire :

```
SELECT c.Name, c.Capital FROM Country c
JOIN City y ON c.Code = y.Country
JOIN encompasses e ON c.Code = e.Country
WHERE e.continent = 'Europe'
AND y.Name = c.Capital
AND y.Latitude > 60;
```

2. Fonctions d'agrégation.

Les fonctions d'agrégation permettent d'obtenir des informations sur un ensemble de lignes en travaillant sur les colonnes et non pas sur les lignes comme avec **WHERE**. Par exemple :

- **MAX** ou **MIN** pour trouver un maximum ou minimum
- **SUM** pour faire la somme d'une colonne
- **AVG** pour calculer une moyenne (average en anglais)
- **COUNT** pour compter le nombre de répétitions

```
select count(*) as nbLangues from Language l where l.country = 'F'
```

3. Requêtes imbriquées.

On peut parfois avoir recours à des **SELECT de SELECT** etc ... :

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

Sur le site qui est donné en lien, vous pourrez vous entraîner et vous perfectionner dans la recherche d'informations sur les bases de données et les requêtes SQL :

<https://isn-icn-ljm.pagesperso-orange.fr/SQL/>