

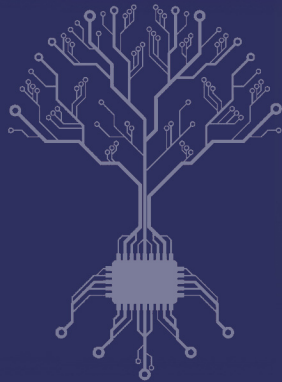


وزارة التربية الوطنية
والتعليم الأولي والرياضة

+٠L٠L٠O+ | %OXC٤ ٠٠L%O
^ %OOHC^ ٠CJL٠O% ^ +%H%+

BASE DE DONNÉES (2)

INTERROGATION



Pour interroger une base de données on utilise la commande SELECT. Elle permet d'effectuer, sur les tables, des opérations de l'algèbre relationnel :

- Récupérer certaines colonnes d'une table (projection)
- Récupérer certaines lignes d'une table en fonction de leur contenu (sélection)
- Combiner des informations venant de plusieurs tables (jointure)

■ La commande SELECT permet de :

- ▶ récupérer certaines colonnes (SELECT) de certaines tables (FROM)
- ▶ récupérer certaines lignes (WHERE)
- ▶ regrouper certaines lignes (GROUP BY)
- ▶ filtrer après le regroupement (HAVING)
- ▶ trier les résultats (ORDER BY)
- ▶ limiter le nombre d'enregistrements renvoyés (LIMIT)

Syntaxe

```
SELECT colonne1, colonne2, ... FROM table
```

ou bien

```
SELECT DISTINCT colonne1, colonne2, ... FROM table
```

Le mot clé DISTINCT permet de supprimer les doublons.

Exemple

Soit le modèle relationnel :

student(idstudent, firstname, lastname, field, average, city)

<i>idstudent</i>	<i>firstname</i>	<i>lastname</i>	<i>field</i>	<i>average</i>	<i>city</i>
1	Ali	Hassan	Mathematics	17.5	Rabat
2	Sarah	Mohammad	Economics	18.3	Rabat
3	Omar	Yussuf	Mathematics	16.0	Kénitra
4	Laila	Ahmad	Economics	14.4	Rabat
5	Khalid	Ali	Mathematics	19.2	Salé
6	Fatima	Salem	Economics	15.7	Kénitra
7	Noor	Hussein	Mathematics	18.1	Salé
8	Ibrahim	Zaid	Economics	19.8	Salé
9	Aya	Nasser	Mathematics	14.5	Salé
10	Huda	Fahim	Economics	13.0	Rabat

Table – Extrait de la relation *student*

Exemples

- Pour afficher les prénoms et les noms des élèves :
 - ▶ `SELECT firstname, lastname FROM student ;`
- Pour afficher les villes (sans doublons) des élèves :
 - ▶ `SELECT DISTINCT city FROM student ;`
- Pour afficher toutes les informations des élèves :
 - ▶ `SELECT * FROM student ;`

Syntaxe

```
SELECT colonne1, colonne2, ... FROM table WHERE conditions
```


Exemples

- Afficher les prénoms et les noms des élèves qui habitent à Rabat :
 - ▶ `SELECT firstname, lastname FROM student WHERE city = 'Rabat' ;`
- Afficher les prénoms, les noms et les moyennes des élèves qui habitent à 'Salé' ou à 'Kénitra' :
 - ▶ `SELECT firstname, lastname, average FROM student WHERE city = 'Salé' OR city = 'Kénitra' ;`
 - ▶ `SELECT firstname, lastname, average FROM student WHERE city IN ('Salé', 'Kénitra') ;`

Exemple

- Afficher les prénoms et les noms des élèves dont la moyenne est supérieure à 12 :

► `SELECT firstname, lastname FROM student WHERE average >= 12 ;`

Il est possible de trier les données sur une ou plusieurs colonnes par ordre ascendant (par défaut) ou descendant.

Syntaxe

```
... ORDER BY colonne1 [ASC ou DESC], colonne2 [ASC ou DESC], ... ;
```

Exemple

- Afficher les prénoms, les noms et les moyennes des élèves triées par ordre ascendant :
 - ▶ `SELECT firstname, lastname, average FROM student ORDER BY average ;`

Pour limiter le nombre de lignes renvoyées par une requête, on utilise la commande :

Syntaxe

```
... LIMIT nombreLigne OFFSET nombreSaut ;
```

qui permet de limiter le nombre de résultats à *nombreLigne*, et de sauter *nombreSaut* lignes avant de renvoyer le résultat.

Exemple

- Afficher les prénoms et les noms des trois premiers élèves :

```
SELECT firstname, lastname FROM student ORDER BY average DESC  
LIMIT 3;
```

- SQL dispose aussi d'un ensemble de fonctions d'agrégation comme :
 - ▶ SUM : pour calculer la somme des valeurs.
 - ▶ AVG : pour calculer la moyenne des valeurs.
 - ▶ COUNT : pour compter le nombre de lignes concernées.
 - ▶ MAX : pour récupérer la plus grande valeur.
 - ▶ MIN : pour récupérer la plus petite valeur.

Exemples

- Afficher la note maximale et minimale et la moyenne des notes des élèves du centre :

- ▶ `SELECT MAX(average), MIN(average), AVG(average) FROM student ;`

- Afficher le nombre des étudiants du centre :

- ▶ `SELECT COUNT(*) FROM student ;`

Renvoie le nombre de lignes de la table.

- ▶ ou `SELECT COUNT(idstudent) FROM student ;`

Pour s'assurer qu'on compte un étudiant une seule fois.

En utilisant une agrégation, on peut regrouper les lignes d'une table en fonction des valeurs d'une ou plusieurs colonnes à l'aide du mot-clé GROUP BY.

Exemples

- Afficher la note maximale de chaque filière :
 - ▶ `SELECT field, MAX(average) FROM student GROUP BY field ;`
- Afficher la note maximale de chaque filière de chaque ville :
 - ▶ `SELECT field, city, MAX(average) FROM student GROUP BY field, city ;`

Points importants :

- Toutes les colonnes non agrégées dans le SELECT doivent être incluses dans le GROUP BY.
- Les colonnes listées après GROUP BY définissent les groupes. Chaque groupe contiendra les lignes ayant les mêmes valeurs dans ces colonnes.

Il est possible d'effectuer une sélection après une agrégation, en ajoutant une condition à laquelle chaque groupe doit répondre.

Syntaxe

```
SELECT colonne1, ..., fonction1(colonne2), ... FROM table GROUP BY  
colonne3, ... HAVING condition ;
```

Exemple

- On ajoute à l'exemple précédent la condition de n'afficher que les notes maximales ≥ 10 :
 - ▶ `SELECT field, MAX(average) FROM student GROUP BY field HAVING MAX(average) ≥ 10 ;`