

# TD Informatique TSI2

## DATABASE : SQL (Structured Query Language)

### PARTIE I : PARAGRAPHES 1,2,3 language SQL

#### SELECT FROM, WHERE, AND OR, AS, MAX, COUNT, ORDER BY

## SOLUTION

### EXERCICE 01

On dispose d'une table **Note** (**id**, **maths**, **physique**, **informatique**) de type entier pour **id** et flottant pour les autres champs. Parmi les assertions suivantes, lesquelles sont vraies ?

**A.** La requête **SELECT maths FROM Note WHERE 5 < maths < 15** ; est correcte ?

Elle est correcte et donc l'**assertion A est True**. Mais le résultat obtenu n'est probablement pas celui qui est attendu. En effet, **5 < maths < 15** est une expression booléenne a pour valeur soit **0 < 15** soit **1 < 15** car **5 < maths** est elle-même une expression booléenne. Comme **0 < 15** ou **1 < 15** renvoie **True**, on obtient en fait toutes les notes de maths.

**B.** La requête **SELECT maths FROM Note WHERE maths < 5 OR > 15** ; est correcte ?

Elle n'est pas correcte, l'**assertion B est False** et pour qu'elle devienne correcte, il faut écrire **WHERE maths < 5 OR maths > 15** à la place de **WHERE maths < 5 OR > 15**

**C.** La requête **SELECT AVG(maths) > 10 FROM Note** ; est correcte ?

L'**assertion C est True**. On récupère la moyenne des notes de maths si supérieure à 10.

**D.** La requête **SELECT "vrai" FROM Note WHERE maths > 10** ; est correcte ?

L'**assertion D est True**.

**E.** La requête **SELECT "vrai" FROM Note WHERE AVG(maths) > 10** ; est correcte ?

L'**assertion E est False** car **AVG** ne s'utilise pas après **WHERE** donc sur les lignes mais juste après **SELECT** donc sur les colonnes.

**F.** La requête **SELECT AVG(maths), MAX(Physique) FROM Note** ; est correcte ?

L'**assertion F est True**.

### EXERCICE 02

Les chambres d'un hôtel sont gérées avec une base de données. La relation **Chambre** permet d'enregistrer : le numéro de la chambre, la date (le jour réservé), l'heure d'arrivée, le nombre de lits, la présence d'un balcon. Les attributs **numéro** et **lits** sont au format entier, l'attribut **date** est au format chaîne de caractères et l'attribut **balcon** au format booléen.

Le tableau ci-dessous précise le contenu d'une ligne.

numéro	date	heure	lits	balcon
108	"2022-04-18"	17	3	True

Écrivons les requêtes suivantes en SQL.

1. Les numéros des chambres réservées à la date du 19/04/22

**SELECT numéro FROM Chambre WHERE date = "2022-04-19" ;**

2. Les numéros des chambres réservées à la date du 19/04/22 qui ont un balcon

**SELECT numéro FROM Chambre WHERE date = "2022-04-19" AND balcon = True ;**

3. Le nombre de chambres sans balcon réservées à la date du 23/04/22

**SELECT COUNT(numéros) FROM Chambre WHERE date = "2022-04-23" AND balcon = False ;** On peut prendre aussi **COUNT(\*)**

4. Le nombre de chambres réservées le 25/04/22 avec deux ou trois lits

```
SELECT COUNT(numéro) FROM Chambre WHERE date = "2022-04-25"  
AND (lits = 2 OR lits = 3);
```

5. Tous les enregistrements du 18/04/22 avec arrivée après 19 heures

```
SELECT * FROM Chambre WHERE date = "2022-04-18" AND heure > 19;
```

6. L'heure de la première chambre de prise le 18/04/2022

```
SELECT MIN(heure) FROM Chambre WHERE date = "2022-04-18";
```

### EXERCICE 03

On dispose d'une base de données constituée d'une seule table. Le schéma relationnel est le suivant :

**Meteo** (Id, Lieu, Année, Mois, Tmin, Tmax, Précipitations)

Écrire les requêtes suivantes en SQL.

1. Obtenir tous les enregistrements complets concernant Paris en 2021.

```
SELECT * FROM Meteo WHERE Lieu = "Paris" AND Année = 2021;
```

2. Obtenir la quantité de pluie tombée à Paris en 2021.

```
SELECT SUM(Précipitations) FROM Meteo WHERE Lieu = "Paris" AND Année = 2021;
```

3. Obtenir pour chaque mois l'écart de température entre **Tmin** et **Tmax** à Paris en 2021 que l'on renommera **Écart**

```
SELECT Mois, Tmax - Tmin AS Écart FROM Meteo  
WHERE Lieu = "Paris" AND Année = 2021;
```

### EXERCICE 04

#### Gestion d'une librairie. Part I

Une librairie est gérée à l'aide d'une base de données. Le modèle relationnel contient les cinq relations décrites ci-dessous avec leur schéma :

**Livre** (Id, Titre, PrixHT, Année, #Id\_genre, #Id\_editeur

**Auteur** (Id, Nom, Prénom)

**Écrit** (#Id\_auteur, #Id\_titre)

**Genre** (Id, Nom)

**Éditeur** (Id, Nom)

Les champs **Id**, **Id\_auteur**, **Id\_titre** sont des clés primaires et tous les champs du genre **Id\_NomTable** sont des clés étrangères. Plus précisément :

le champ **Id\_editeur** est une clé étrangère en référence à la clé primaire **Id** de la table **Éditeur**

Le champ **Id\_genre** est une clé étrangère en référence à la clé primaire **Id** de la table **Genre**

Le champ **Id\_titre** est une clé étrangère en référence à la clé primaire **Id** de la table **Livres**

Le champ **Id\_auteur** est une clé étrangère en référence à la clé primaire **Id** de la table **Auteur**

Tous les champs **Id** ou commençant par **Id** et le champ **Année** sont du type entier. Le champ **PrixHT** est de type flottant et les autres champs sont de type chaîne de caractères.

Écrire les requêtes suivantes en SQL.

1. Les titres des livres de la base.

```
SELECT Titre FROM Livre;
```

2. Les titres et les années de parution des livres.

```
SELECT Titre, Année FROM Livre;
```

3. Les titres des livres commençant par la lettre "A"

```
SELECT Titre FROM Livre WHERE Titre < "B";
```

4. Le prix HT maximal d'un livre de la base.

```
SELECT MAX(PrixHT) FROM Livre;
```

5. Tous les champs de la table **Auteur**

```
SELECT * FROM Auteur ;
```

6. Le nombre d'auteurs contenus dans la base ;

```
SELECT COUNT(Id) FROM Auteur ; On peut prendre COUNT(*)
```

7. Le nombre de livres référencés et le prix moyen en nommant "**Nombre de livres**" la nouvelle première colonne et "**Prix moyen**" la nouvelle seconde colonne.

```
SELECT COUNT(Id) AS "Nombre de livres", AVG(PrixHT) AS "Prix moyen"  
FROM Livre ;
```

8. Les noms et les prénoms de tous les auteurs par ordre alphabétique.

```
SELECT Nom, Prénom FROM Auteur ORDER BY Nom ;
```

9. Les noms des auteurs dont le prénom est Walter

```
SELECT Nom FROM Auteur WHERE Prénom = "Walter" ;
```

10. Les titres des livres coûtant plus de 15 euros HT avec leur prix HT

```
SELECT Titre, Prix HT FROM Livre WHERE PrixHT > 15 ;
```

11. Les titres et années de parution des livres parus de 2010 à 2015 ordonnés suivant l'année de parution de manière de manière décroissante.

```
SELECT Titre, Année FROM Livre WHERE Année >= 2010 AND Année <= 2015  
ORDER BY Année DESC ;
```