

Chapitre 2 : Requêtes SQL de base

● **Durée** : 4 heures 20 minutes

● **Mots clés** : corrélation, imbrication, projection, sélection, sous-requête.

● **Objectif** :

Ce chapitre est dédié aux requêtes SQL de base, à l'aide de la commande `SELECT`. Quand leur contenu n'est pas spécifié, les tables utilisées dans les questions des pré-requis sont celles créées au premier chapitre.

Pré-requis

Pour valider les pré-requis nécessaires avant d'aborder le TP, répondez aux questions ci-après :

1. Le nombre de colonnes spécifiées dans la clause `SELECT` est limité à :
 - a. 50
 - b. 100
 - c. 255
 - d. infini
 - e. aucune des réponses précédentes
2. Pour sélectionner toutes les colonnes d'une table, il faut utiliser la commande :
 - a. `SELECT ALL COLUMNS FROM ...`
 - b. `SELECT * FROM ...`
 - c. `SELECT '*' FROM ...`
3. Les deux requêtes suivantes sont équivalentes :

```
SELECT ville
FROM editions
WHERE ville != 'Paris';
SELECT ville
FROM editions
WHERE ville <> 'Paris';
```

 - a. VRAI
 - b. FAUX

4. La clause `LIMIT` dans une requête `SELECT` permet de préciser le nombre maximal de lignes que l'utilisateur veut obtenir.
- a. VRAI
 - b. FAUX
5. La clause `FROM` d'une requête `SELECT` peut être omise.
- a. VRAI
 - b. FAUX

6. La requête suivante va renvoyer la liste des maisons d'édition dont le nom de la ville n'est pas `NULL` :

```
SELECT nom
FROM editions
WHERE ville != NULL;
```

- a. VRAI
 - b. FAUX
7. Que calcule la requête suivante sur notre table `editions` ?

```
SELECT max(code_postal)
FROM editions;
```

8. La requête suivante va provoquer une erreur :

```
SELECT ville, max(employes)
FROM entrepots;
```

- a. VRAI
 - b. FAUX
9. Les questions 9 à 12 portent sur la table `personnes` suivante (une case vide contient la valeur `NULL`) :

NOM	PRENOM
Dupond	Jean
Robert	Anaïs
Dupont	Pierre
Burrel	André
Dupond	Jean
Dupond	

Quels seront les couples de valeurs renvoyés par la requête suivante ?

```
SELECT DISTINCT *  
FROM personnes;
```

10. Quelles seront les valeurs renvoyées par la requête suivante ?

```
SELECT DISTINCT prenom  
FROM personnes;
```

11. Quelle sera la valeur renvoyée par la requête suivante ?

```
SELECT count(DISTINCT prenom)  
FROM personnes;
```

- a.** 4
- b.** 5
- c.** NULL

12. Exécutez la commande :

```
INSERT INTO personnes VALUES (NULL, NULL);
```

Quelle sera la valeur renvoyée par la requête suivante ?

```
SELECT count(*)  
FROM personnes;
```

- a.** 4
- b.** 5
- c.** 7
- d.** NULL

13. Dans la liste suivante, quelles sont les requêtes équivalentes ?

- a.**

```
SELECT titre FROM livres  
WHERE prix > 5.50 AND prix < 9.50;
```
- b.**

```
SELECT titre FROM livres  
WHERE prix BETWEEN 5.50 AND 9.50;
```
- c.**

```
SELECT titre FROM livres  
WHERE prix >= 5.50 AND prix <= 9.50;
```
- d.**

```
SELECT titre FROM livres  
WHERE prix NOT BETWEEN 5.50 AND 9.50;
```

- e.** `SELECT titre FROM livres
WHERE prix < 5.50 AND prix < 9.50;`
- f.** `SELECT titre FROM livres
WHERE prix <= 5.50 AND prix >= 9.50;`
- g.** `SELECT titre FROM livres
WHERE NOT prix >= 5.50;`
-
-

14. Les questions 14, 15 et 16 portent sur la table `personnes` ci-dessous :

NOM	PRENOM	AGE
Dupond	Jean	26
Robert	Anaïs	30
Dupont	Pierre	45
Burrel	André	
Dupond	Jean	60

Quelle sera la valeur renvoyée par la requête suivante ?

```
SELECT round(avg(age), 2)  
FROM personnes  
WHERE nom LIKE 'd%';
```

- a.** NULL
- b.** 0
- c.** 40
- d.** 43.67
- 15.** Sur la table `personnes`, la requête suivante va-t-elle renvoyer un résultat ?

```
SELECT nom, age  
FROM personnes  
WHERE age >= ALL (SELECT age FROM personnes);
```

- a.** VRAI
- b.** FAUX

16. Sur la table `personnes`, la requête suivante va-t-elle renvoyer un résultat ?

```
SELECT nom, age
FROM personnes
WHERE age >= ANY (SELECT age FROM personnes);
```

- a. VRAI
b. FAUX

17. La requête suivante affiche le nom des auteurs dont la date de décès et le prénom sont connus :

```
SELECT nom
FROM auteurs
WHERE date_deces, prenom IS NOT NULL;
```

- a. VRAI
b. FAUX

18. Que va calculer la requête suivante ?

```
SELECT count(*) FROM auteurs
WHERE EXTRACT(year FROM age(date_naissance)) > 50
AND date_deces IS NULL;
```

19. Soit la requête suivante :

```
SELECT titre, prix, date_impression
FROM livres
ORDER BY titre DESC, prix ASC, date_impression ASC;
```

Le résultat ci-dessous a-t-il pu être produit par cette requête ?

TITRE	PRIX	DATE_IMPRESSION
La violence urbaine en Europe	18.20	2000-12-11
La paix du ménage	8.80	2006-05-09
La paix du ménage	8.80	2007-05-08
La paix du ménage	8.80	
La paix du ménage	8.80	
La paix du ménage	8.80	
La paix du ménage	8.80	
La paix du ménage	7.80	
L'œuvre d'Oscar Niemeyer	72.10	2007-09-15

- a. VRAI
- b. FAUX

20. Soit la requête suivante :

```
SELECT titre, prix, date_impression
FROM livres
ORDER BY titre DESC, prix DESC, date_impression ASC NULLS
FIRST;
```

Les lignes suivantes font partie de la réponse, réordonnez-les :

1

La violence urbaine en Europe	18.20	2000-12-11
-------------------------------	-------	------------

2

La paix du ménage	8.80	2006-05-09
-------------------	------	------------

3

La paix du ménage	8.80	2007-05-08
-------------------	------	------------

4

La paix du ménage	8.80	
-------------------	------	--

5

La paix du ménage	7.80	
-------------------	------	--

6

L'Œuvre d'Oscar Niemeyer	72.10	2007-09-15
--------------------------	-------	------------

21. La requête ci-dessous va provoquer une erreur :

```
SELECT 'Le livre ' || titre || ' coûte ' || prix || ' euros.'
FROM livres;
```

- a. VRAI
- b. FAUX