

Université de Carthage Institut Préparatoire aux Etudes d'ingénieur de Nabeul Département :mathématique	 المعهد التحضيري للدراسات الهندسية بنابل Institut Préparatoire aux Etudes d'Ingénieurs de Nabeul	Année universitaire : 2024/2025
		Filière : SM/SP/ST
		Niveau d'étude : 2 ^{ème} année
		Semestre : 2
		Nombre de pages :
		Date : 20/02/2025 Durée :1h30

DS Informatique N° 2 - Corrigé

Partie 1

1. $\pi_{nom_{utilisateur}, date_{inscription}} (Utilisateurs)$
2. $\pi_{nom_{utilisateur}, contenu} (Utilisateurs \bowtie_{utilisateurs.id=id_{utilisateur}} Commentaires)$
3.

$$R1 = \Pi_{id_{utilisateur}} (Publications) \cup \Pi_{id_{utilisateur}} (Commentaires)$$

$$R2 = \Pi_{id} (Utilisateurs) - R1$$

$$R3 = R2 \bowtie Utilisateurs$$
4. La requête permet de récupérer les noms d'utilisateurs qui ont effectué une interaction de type "like" sur une publication.

Partie 2: SQL (13 points)

1.


```
CREATE TABLE Utilisateurs (
  id INTEGER PRIMARY KEY,
  nom_utilisateur TEXT NOT NULL,
  email TEXT NOT NULL,
  date_inscription DATE NOT NULL
);
```
2.


```
ALTER TABLE Publications ADD COLUMN visibilité TEXT;
```
3.


```
INSERT INTO Utilisateurs (id, nom_utilisateur, email, date_inscription)
VALUES (101, 'JD123', 'jamel123@email.com', '2025-02-01');
```
4.


```
DELETE FROM Interactions WHERE id_publication = 55;
```

5.

```
SELECT      id_utilisateur, COUNT(*) AS nb_publications
FROM        Publications
GROUP BY    id_utilisateur;
```
6.

```
SELECT *
FROM        Publications
WHERE date_publication > '2025-01-01'
AND    contenu LIKE '%promotion%';
```
7.

```
SELECT p.id, p.contenu, p.date_publication, u.nom_utilisateur
FROM Publications p
JOIN Utilisateurs u ON p.id_utilisateur = u.id;
```
8.

```
SELECT u.nom_utilisateur, COUNT(p.id) AS nb_publications
FROM Utilisateurs u
JOIN Publications p ON u.id = p.id_utilisateur
GROUP BY u.nom_utilisateur;
```
9.

```
SELECT u.nom_utilisateur
FROM Utilisateurs u
JOIN Publications p ON u.id = p.id_utilisateur
GROUP BY u.nom_utilisateur
HAVING COUNT(p.id) >= 5;
```
10.

```
SELECT *
FROM Publications
ORDER BY date_publication DESC
LIMIT 5;
```

11.

```
SELECT u.*
FROM Utilisateurs u
JOIN Publications p ON u.id = p.id_utilisateur
GROUP BY u.id
ORDER BY COUNT(p.id) DESC
LIMIT 1;
```

12.

```
SELECT p.*
FROM Publications p
JOIN (
    SELECT id_utilisateur
    FROM Publications
    GROUP BY id_utilisateur
    HAVING COUNT(id) >= 3
) u ON p.id_utilisateur = u.id_utilisateur;
```

13.

```
SELECT id_publication, COUNT(*) AS nb_interactions
FROM Interactions
GROUP BY id_publication;
```

14.

Affiche les utilisateurs qui ont aimé une publication mais n'ont pas encore commenté cette publication.

Partie 3: (5 points)

1.

```
def open(nom_base):
    conn = sqlite3.connect(nom_base)
    return conn.cursor()
```

2.

```
def recuperer_publications(id_utilisateur):
    cursor.execute("SELECT * FROM Publications WHERE id_utilisateur = ?", (id_utilisateur,))
    return cursor.fetchall()
```

```

3.
def supprimer_utilisateur(id_utilisateur):

    cursor.execute("DELETE FROM Interactions WHERE id_utilisateur = ?", (id_utilisateur,))

    cursor.execute("DELETE FROM Publications WHERE id_utilisateur = ?", (id_utilisateur,))

    cursor.execute("DELETE FROM Amis WHERE id_utilisateur_1 = ? OR id_utilisateur_2 = ?", (id_utilisateur,
id_utilisateur))

    cursor.execute("DELETE FROM Utilisateurs WHERE id = ?", (id_utilisateur,))

4.
def close():
    conn.commit()
    conn.close()

5.
import sqlite3

conn = sqlite3.connect('reseau_social.db')
cursor = conn.cursor()

cursor.execute("SELECT * FROM Publications")
publications = cursor.fetchall()

for publication in publications:
    print(publication)

conn.commit()
conn.close()

```

5. Le passage du modèle relationnel au modèle objet se fait généralement par la transformation des données relationnelles en objets, chaque ligne d'une table devenant une instance d'une classe.

Par exemple, une table Utilisateurs avec les colonnes id, nom_utilisateur, et email pourrait être convertie en une classe Utilisateur avec les attributs correspondants. Chaque enregistrement dans la table serait un objet de cette classe.

```

class Utilisateur:

    def __init__(self, id, nom_utilisateur, email, date_inscription):

        self.id = id

        self.nom_utilisateur = nom_utilisateur

        self.email = email

        self.date_inscription = date_inscription

# extraction des données à partir du modèle relationnel

import sqlite3

conn = sqlite3.connect('reseau_social.db')

cur = conn.cursor()

cur.execute("SELECT * FROM utilisateurs where id = 101")

u = cur.fetchone()

# Création d'un objet Utilisateur à partir d'une ligne de la table

utilisateur = Utilisateur(u[0], u[1], u[2], u[3])

print(utilisateur.email)

```