BDD - Allan Duval M1 MITIC 2025

Rapport TP – Réseaux Sociaux

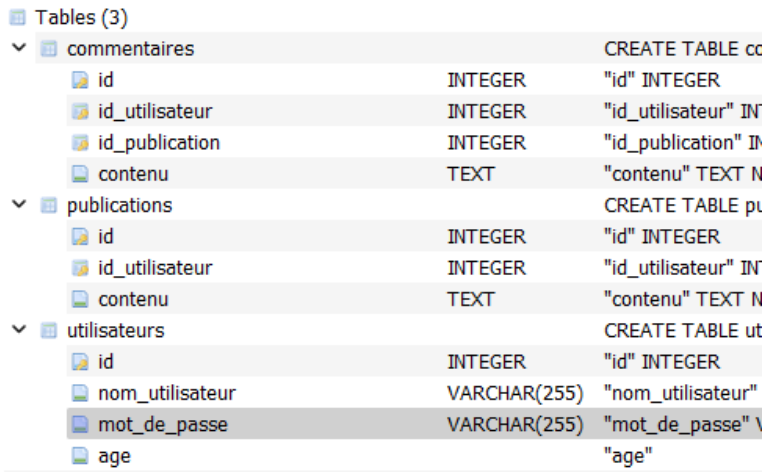
Question 1 : Donner le MLD à partir des requêtes de création de table

UTILISATEURS(
id PK,
nom_utilisateur,
mot_de_passe
)

PUBLICATIONS(
id PK,
contenu,
id_utilisateur FK → **UTILISATEURS**(id)
)

COMMENTAIRES(
id PK,
contenu,
id_utilisateur FK → **UTILISATEURS**(id),
id_publication FK → **PUBLICATIONS**(id)
)

Question 2 : Donner les requêtes SQL pour ajouter les valeurs suivantes :
Utilisateur "Harry", id 51, mot de passe "password"
Publication 51 de Harry, contenu "Le SQL c'est super !"
Commentaire 101 de l'utilisateur 12 sur la publication de Harry, contenu "Tout à fait !"



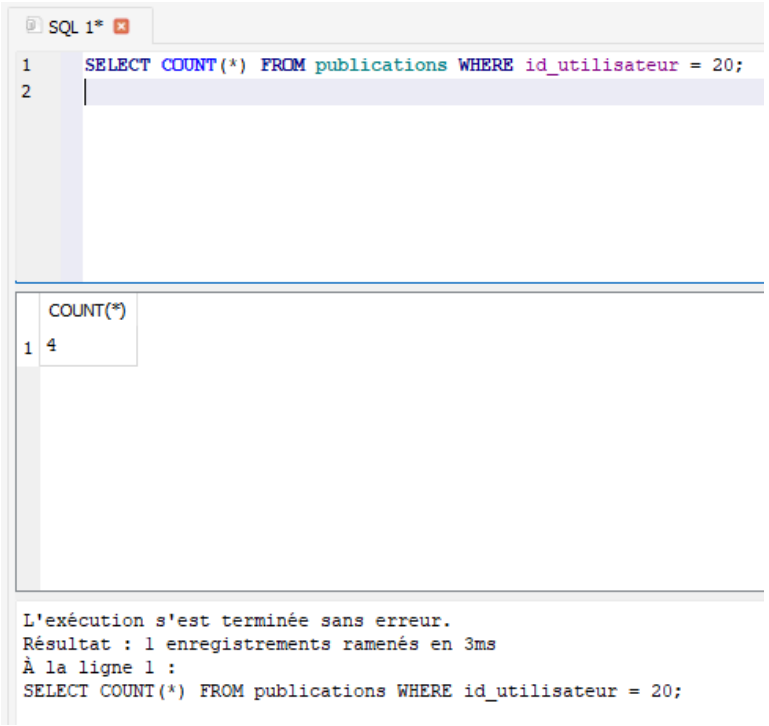
INSERT INTO utilisateurs (id, nom_utilisateur, mot_de_passe)
VALUES (51, 'Harry', 'password');

INSERT INTO publications (id, id_utilisateur, contenu)
VALUES (51, 51, 'Le SQL c'est super !');

INSERT INTO commentaires (id, id_utilisateur, id_publication, contenu)
VALUES (101, 12, 51, 'Tout à fait !');

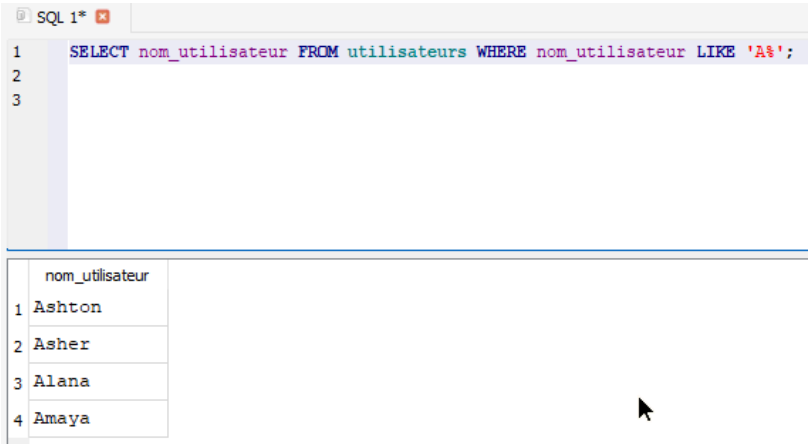
Question 3 : Donner la requête pour compter le nombre de publications de l'utilisateur 20

SELECT COUNT(*) FROM publications WHERE id_utilisateur = 20;



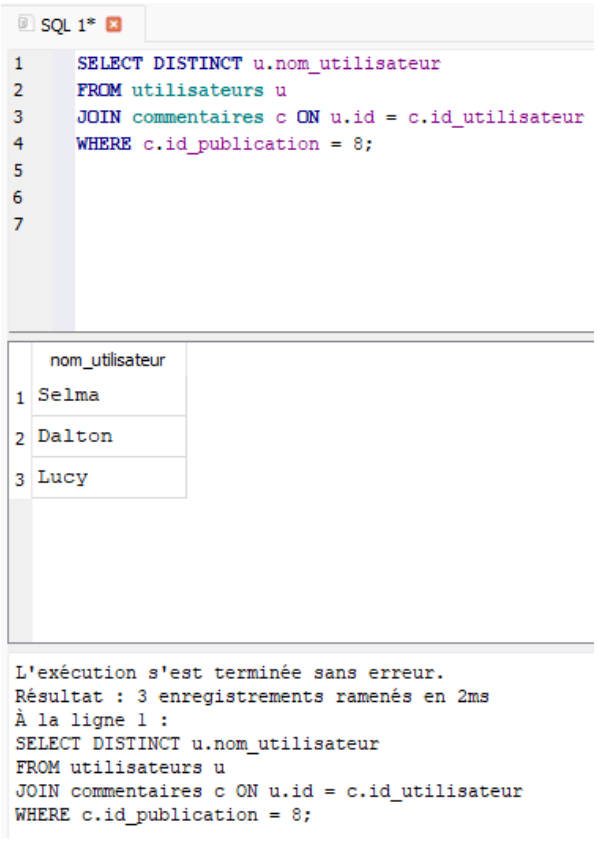
Question 4 : Donner la requête pour lister les noms d'utilisateurs commençant par la lettre "A"

SELECT nom_utilisateur FROM utilisateurs WHERE nom_utilisateur LIKE 'A%';



Question 5 : Donner le nom de tous les utilisateurs qui ont commenté la publication 8

Selma, Dalton, Lucy



Question 6 : Donner la requête pour supprimer l'utilisateur 51

DELETE FROM utilisateurs WHERE id = 51;

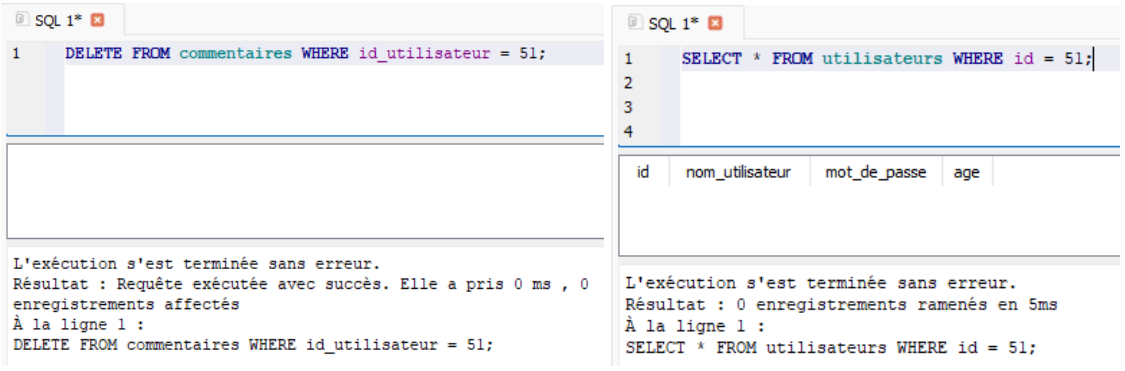
La requête a échoué initialement car des publications / commentaires sont liés à l'utilisateur 51 **(dépendances)**. Pour l'exécuter, il suffit que je supprime ses commentaires, publications et donc l'utilisateur **dans cet ordre précis (à cause des clés contraintes de clé étrangères) :**

DELETE FROM commentaires WHERE id_utilisateur = 51;

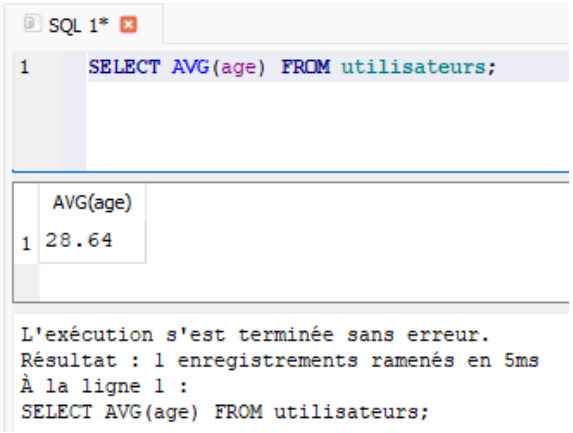
DELETE FROM commentaires
WHERE id_publication IN (
  SELECT id FROM publications WHERE id_utilisateur = 51
);

DELETE FROM publications WHERE id_utilisateur = 51;

DELETE FROM utilisateurs WHERE id = 51;



Question 7 : Donner la moyenne d'âge des utilisateurs



Ici, l'utilisateur Harry ne sera pas pris en compte puisqu'il a été supprimé au préalable. S'il était encore présent dans la db, il était le seul à ne pas avoir d'âge renseigné dans la BDD. Il n'aurait donc pas été pris en compte dans la requête SQL, puisque sa valeur était *NULL*.

Le résultat est donc de 28.64 ans d'âge moyen pour tous les utilisateurs.

Question 8 : Pour chaque publication, donner la moyenne d'âge des utilisateurs l'ayant commentée

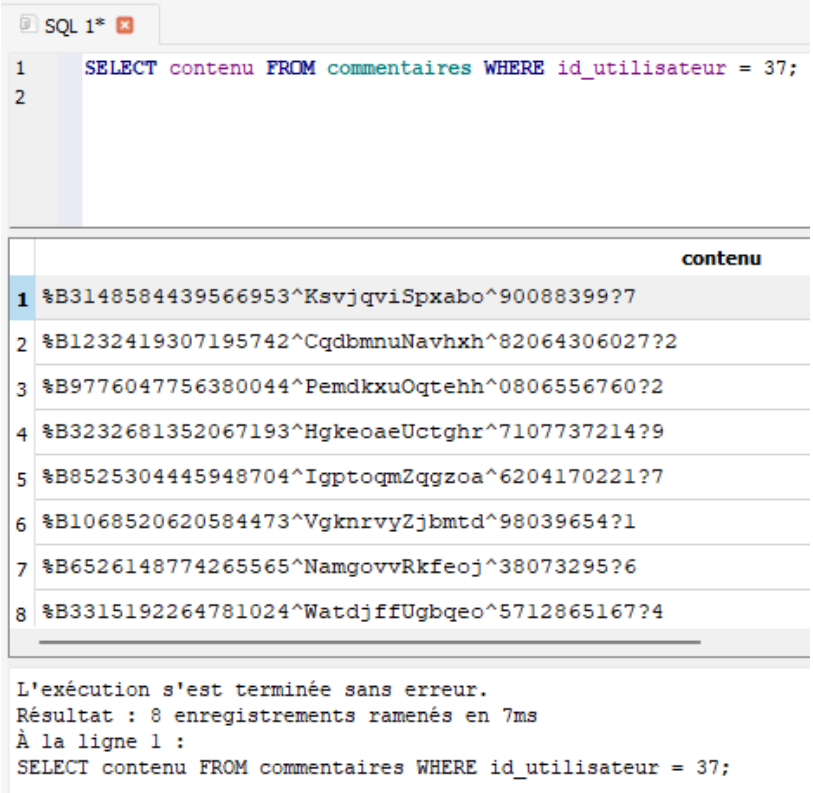
Certaines publications n'apparaissent pas (1, 33, 40, 42, 50) dans ce cas de figure, puisque personne n'a commenté ces publications. La requête ci-jointe prend *seulement en compte les publications ayant reçu au moins un commentaire.*

```
SELECT p.id AS publication_id, AVG(u.age) AS moyenne_age
FROM commentaires c
JOIN utilisateurs u ON c.id_utilisateur = u.id
JOIN publications p ON c.id_publication = p.id
GROUP BY p.id;
```

publication_id	moyenne_age
2	31.75
3	24.3333333333333
4	34.3333333333333
5	22.0
6	30.0
7	25.0
8	22.3333333333333
9	27.5
10	19.0
11	32.0
12	27.5
13	20.0
14	24.5
15	36.0
16	22.0
17	26.5
18	25.0
19	24.6666666666667
20	25.0
21	30.75
22	30.0
23	26.75
24	28.5
25	30.6666666666667
26	27.0
27	27.0
28	36.0
29	19.0
30	27.0
31	25.25
32	36.0
34	27.75
35	31.5
36	28.5
37	20.0
38	32.0
39	29.0
41	31.3333333333333
43	22.0
44	27.0
45	26.0
46	30.0
47	33.0
48	25.0
49	23.3333333333333
51	19.0

Question 9 : Donner le contenu des commentaires de l'utilisateur 37

SELECT contenu FROM commentaires WHERE id_utilisateur = 37;



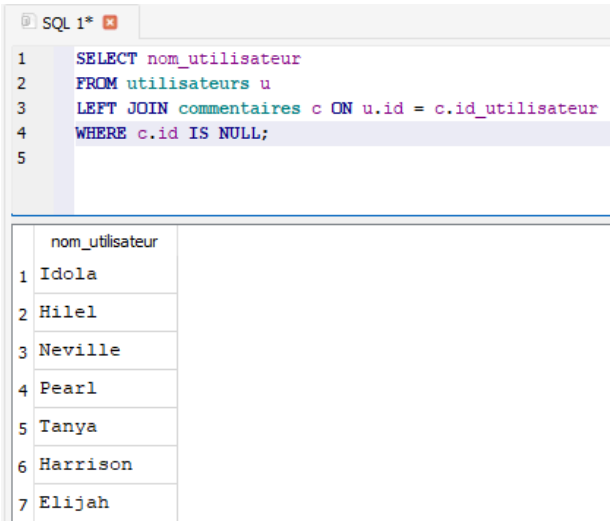
```
SQL 1*
1  SELECT contenu FROM commentaires WHERE id_utilisateur = 37;
2
```

	contenu
1	%B3148584439566953^KsvjqviSpxabo^9008839?7
2	%B1232419307195742^CqdbmnuNavhxh^8206430602?2
3	%B9776047756380044^PemdkxuOqtehh^08065567 60?2
4	%B3232681352067193^HgkeoaeUctghr^710773214?9
5	%B8525304445948704^IgptoqmZqgzoa^620417022 1?7
6	%B1068520620584473^VgknrvyZjbmtd^9803965 4?1
7	%B6526148774265565^NamgovvRkfeoj^3807329 5?6
8	%B3315192264781024^WatdjffUgbqeo^571286516 7?4

```
L'exécution s'est terminée sans erreur.
Résultat : 8 enregistrements ramenés en 7ms
À la ligne 1 :
SELECT contenu FROM commentaires WHERE id_utilisateur = 37;
```

Question 10 : Donner le nom des utilisateurs qui n'ont commenté aucune publication

SELECT nom_utilisateur
FROM utilisateurs u
LEFT JOIN commentaires c ON u.id = c.id_utilisateur
WHERE c.id IS NULL;



```
SQL 1*
1  SELECT nom_utilisateur
2  FROM utilisateurs u
3  LEFT JOIN commentaires c ON u.id = c.id_utilisateur
4  WHERE c.id IS NULL;
5
```

	nom_utilisateur
1	Idola
2	Hilel
3	Neville
4	Pearl
5	Tanya
6	Harrison
7	Elijah

Question 11 : Donner la requête pour ajouter une colonne “Like” à publications

ALTER TABLE publications ADD COLUMN likes INTEGER;

Question 12 : Ajouter dans cette nouvelle colonne des valeurs aléatoires entre 0 et 100 likes

Pour éviter de faire manuellement la manipulation, j’ai décidé d’utiliser Python en créant un script pour automatiser l’insertion des valeurs aléatoires entre 0 et 100 dans la colonne likes de la table publications. (disponible dans le dossier)

C: > Users > Allan > Desktop > BDD_Allan > ajouter_likes.py > ...

```python
import sqlite3
import random

# Ce script met à jour la colonne "likes" de la table "publications" avec des valeurs random entre 0 et 100.

conn = sqlite3.connect("Reseau_soc.db")
cursor = conn.cursor()

cursor.execute("SELECT id FROM publications")
ids = cursor.fetchall()

for (pub_id,) in ids:
    likes = random.randint(0, 100)
    cursor.execute("UPDATE publications SET likes = ? WHERE id = ?", (likes, pub_id))

conn.commit()
conn.close()

print("Likes random insérés avec succès.")
```

Run Python > Run Python File in Terminal

```
PS C:\Users\Allan> & C:/Users/Allan/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/Allan/Desktop/BDD_Allan/ajouter_likes.py
Traceback (most recent call last):
  File "c:\Users\Allan\Desktop\BDD_Allan\ajouter_likes.py", line 9, in <module>
    cursor.execute("SELECT id FROM publications")
sqlite3.OperationalError: no such table: publications
PS C:\Users\Allan> & C:/Users/Allan/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/Allan/Desktop/BDD_Allan/ajouter_likes.py
Likes random insérés avec succès.
PS C:\Users\Allan>
```

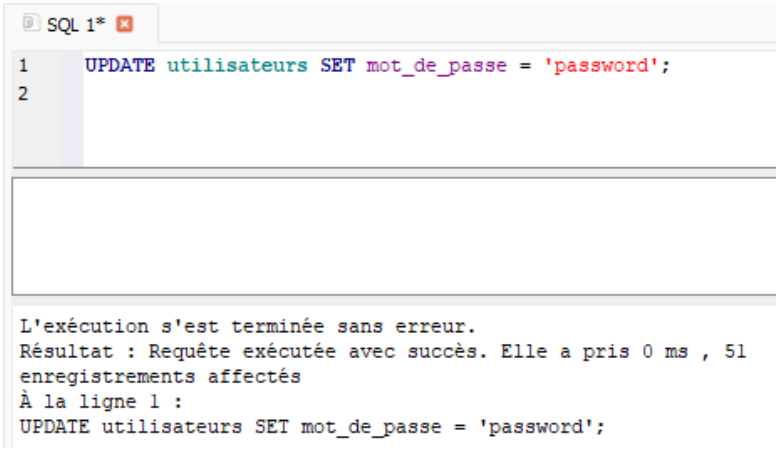
Après un premier essai, le script ne voulait pas s'exécuter car il ne trouvait pas la BDD. J’ai remplacé la ligne 6 par le chemin absolu du fichier db pour qu’il s'exécute.

Un exemple de quelques modifications :

	id	id_utilisateur	contenu	likes
	Filtre	Filtre	Filtre	Filtre
1	1	19	FFCA6015-994B-3753-EDD6-490E1B721171	9
2	2	5	70DAB49A-5575-5281-3A59-79333EC2DA37	32
3	3	34	B8D278AA-6374-8844-8489-9EBB2F87B626	83
4	4	47	B112D491-E0F2-799D-44AE-F54A4E838AD5	29
5	5	12	B9165776-E1E4-8BBD-84FD-EFEE32BD1628	14
6	6	7	1A7DBF87-F058-14D4-8A25-A1FD43E9B41A	56
7	7	23	825923E4-37D8-AA6C-A86B-28C1B7E96DC7	24
8	8	37	727DDC8E-8621-54DE-AD68-B4CD6F6D8CA8	36
9	9	11	8DBB8F36-5583-2BA4-3CC1-6B6631C297BC	12
10	10	25	3C1EAF23-B7B3-5765-2BCA-08E52B334DC3	6
11	11	48	785533DA-152D-52D8-1C64-9E4455AA4CAF	81
12	12	15	8C78B3C2-52A6-9A13-1431-3B4EDC778522	94
13	13	14	38153749-91E7-DD13-B938-B9239EAA1C64	96
14	14	42	232636C4-181A-7B5B-2C3B-B7D3383A4B73	4
15	15	29	15A42999-3D2E-E38B-C484-65AE228EA504	16
16	16	27	89104BF9-C307-A77B-BD7D-3E424986DE3E	16

Question 13 : Tous les mots de passe sont réinitialisés à leur valeur par défaut : “password”

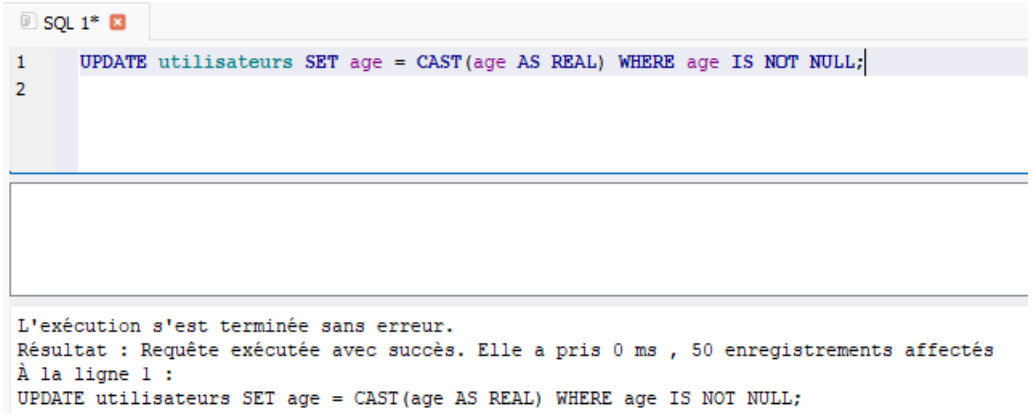
UPDATE utilisateurs SET mot_de_passe = 'password';



Un exemple de la table utilisateurs :

20	20	Giacomo	password	20
21	21	Asher	password	27
22	22	Pearl	password	39
23	23	Serina	password	29
24	24	Cheyenne	password	26
25	25	Lucy	password	21
26	26	Forrest	password	25
27	27	Chantale	password	34
28	28	Mercedes	password	35
29	29	Wilma	password	22

Question 14 : Transformer les âges (nombres entiers) en nombres réels

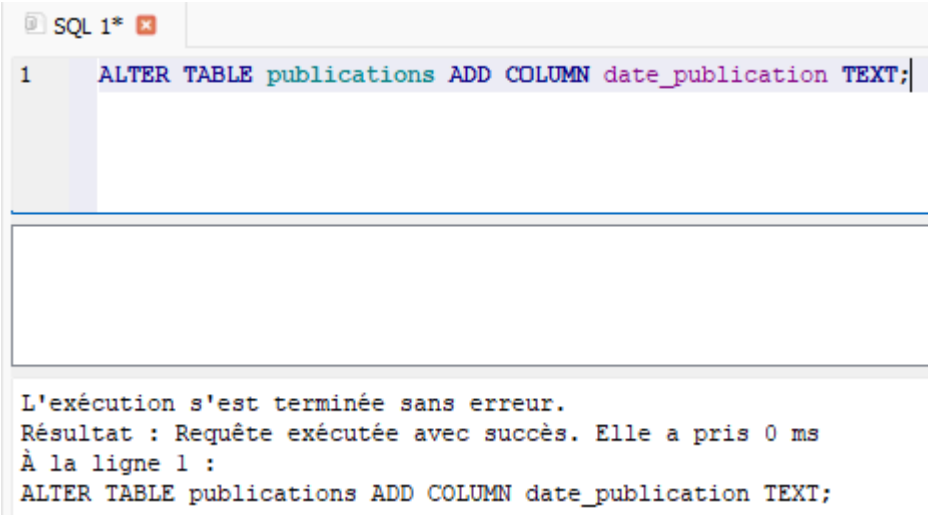


id	nom_utilisateur	mot_de_passe	age
Filtre	Filtre	Filtre	Filtre
1	Idola	password	24.0
2	Lesley	password	34.0
3	Cullen	password	36.0
4	Ashton	password	22.0
5	Samson	password	22.0
6	Lance	password	26.0
7	Colin	password	34.0
8	Hilel	password	37.0
9	MacKenzie	password	38.0
10	Ross	password	39.0
11	Neville	password	33.0

Question 15 (Bonus) : Ajouter une date aux publications, date aléatoire entre le 01-01-2023 à 00:00:00 et le 02-02-2023 à 00:00:00

Pour réaliser toutes ces étapes, j'ai décidé de la décomposer en plusieurs parties :

- La première, sur celle de l'ajout de la colonne date_publication



La seconde, en se basant comme pour la question 12, à travers un script Python pour insérer cette fois-ci des dates aléatoires.

C: > Users > Allan > Desktop > BDD_Allan > dates_aleatoires.py > ...

```python
import sqlite3
import random
from datetime import datetime, timedelta

conn = sqlite3.connect(r"C:\Users\Allan\Desktop\BDD_Allan\Reseau_soc.db")
cursor = conn.cursor()

start = datetime(2023, 1, 1, 0, 0, 0)
end = datetime(2023, 2, 2, 0, 0, 0)

cursor.execute("SELECT id FROM publications")
ids = cursor.fetchall()

for (pub_id,) in ids:
    delta = end - start
    random_seconds = random.randint(0, int(delta.total_seconds()))
    random_date = start + timedelta(seconds=random_seconds)
    formatted = random_date.strftime("%Y-%m-%d %H:%M:%S")
    cursor.execute("UPDATE publications SET date_publication = ? WHERE id = ?", (formatted, pub_id))

conn.commit()
conn.close()

print("Dates de publication modifiées avec succès.")
```

```
PS C:\Users\Allan> & C:/Users/Allan/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/Allan/Desktop/BDD_Allan/dates_aleatoires.py
Dates de publication modifiées avec succès.
```

id	*id_utilisateur*	contenu	likes	date_publication
Filtre	Filtre	Filtre	Filtre	Filtre
1	19	FFCA6015-994B-3753-EDD6-490E1B721171	9	2023-01-17 18:38:50
2	5	70DAB49A-5575-5281-3A59-79333EC2DA37	32	2023-01-21 17:11:19
3	34	B8D278AA-6374-8844-8489-9EBB2F87B626	83	2023-01-09 01:00:16
4	47	B112D491-E0F2-799D-44AE-F54A4E838AD5	29	2023-01-04 13:00:44
5	12	B9165776-E1E4-8BBD-84FD-EFEE32BD1628	14	2023-01-26 04:47:51
6	7	1A7DBF87-F058-14D4-8A25-A1FD43E9B41A	56	2023-01-17 16:14:11
7	23	825923E4-37D8-AA6C-A86B-28C1B7E96DC7	24	2023-01-22 08:05:23
8	37	727DDC8E-8621-54DE-AD68-B4CD6F6D8CA8	36	2023-01-05 07:59:48
9	11	8DBB8F36-5583-2BA4-3CC1-6B6631C297BC	12	2023-01-09 17:53:53
10	25	3C1EAF23-B7B3-5765-2BCA-08E52B334DC3	6	2023-01-08 18:13:24
11	48	785533DA-152D-52D8-1C64-9E4455AA4CAF	81	2023-01-17 13:54:01
12	15	8C78B3C2-52A6-9A13-1431-3B4EDC778522	94	2023-01-03 21:32:35
13	14	38153749-91E7-DD13-B938-B9239EAA1C64	96	2023-01-14 14:57:04
14	42	232636C4-181A-7B5B-2C3B-B7D3383A4B73	4	2023-01-24 07:59:56
15	29	15A42999-3D2E-E38B-C484-65AE228EA504	16	2023-01-08 04:29:14
16	27	89104BF9-C307-A77B-BD7D-3E424986DE3E	16	2023-01-31 06:09:15

On constate alors que toutes les publications ont leur date aléatoire entre le 01-01-2023 à 00:00:00 et le 02-02-2023 à 00:00:00.
On peut aussi vérifier la modification avec la commande SQL : SELECT id, date_publication FROM publications;