

Backend Web Dev — Practice

CSCI 40

Fall 2026

LLM evaluation

2026-08-26

Re-evaluation pending after quiz-bank corrections.

Total Score: _____/4 **Quiz rules:**

1. You MAY use any printed or handwritten notes.
2. You MAY NOT use a computer or any other electronic device.

For each program below, write the output of all print statements. You MAY NOT use a computer. **Initial database state used by every problem**

Every problem starts with a fresh copy of this SQLite database.
NULL denotes a SQL NULL value. Rows are shown in id order.

Schema:

```
CREATE TABLE users (  
  id INTEGER PRIMARY KEY,  
  username TEXT NOT NULL UNIQUE,  
  password TEXT NOT NULL,  
  age INTEGER  
);  
  
CREATE TABLE messages (  
  id integer primary key,  
  sender_id integer not null,  
  message text not null,  
  created_at timestamp not null default current_timestamp  
);
```

Initial rows:

users(id, username, password, age):

```
1 | Trump | Trump | 78  
2 | Biden | Biden | 81  
3 | Evan | correct horse battery staple | 7  
4 | Isaac | soccer | 4  
5 | Aaron | guaguagua | 3  
6 | Aurelia | | 1  
7 | Teal | 524euTjrWm6uK2C5iw8mC6aNgX1JI78o | 35  
8 | Kristen | Possible-Rich-Absolute-Battle | NULL
```

messages(id, sender_id, message, created_at):

```
1 | 1 | I'm a baby | 2021-11-14 14:30:00  
2 | 2 | I'm a baby | 2021-11-14 14:30:00  
3 | 3 | I'm a baby | 2021-11-14 14:33:01  
4 | 4 | I'm a baby | 2021-11-15 14:35:45  
5 | 3 | I'm actually a toddler | 2021-11-16 14:35:45  
6 | 6 | Today in 1918, the Armistice that effectively ended WWI came into effect. | 2021-11-11 11:00:00  
7 | 6 | I'm an adult | 2021-11-17 14:35:45  
8 | 6 | SQL is the best!! | 2021-11-17 14:35:45  
9 | 7 | I'm an adult | 2021-11-17 14:35:45  
10 | 7 | WTF is SQL?! I thought you liked the snake thing. | 2021-11-17 14:35:45
```

problem1.py

```
1 # Problem 1
2
3 import sqlite3
4
5 con = sqlite3.connect('twitter_clone.db')
6 cur = con.cursor()
7 sql = """
8 SELECT id,password FROM users WHERE username='Trump';
9 """
10 cur.execute(sql)
11 for row in cur.fetchall():
12     print('row[0]=', row[0])
13     print('row[1]=', row[1])
```

Answer:

```
row[0]= 1
row[1]= Trump
```

problem1b.py

```
1 # Problem 1b
2
3 import sqlite3
4
5 con = sqlite3.connect('twitter_clone.db')
6 cur = con.cursor()
7 sql = """
8 SELECT count(*) FROM messages WHERE message LIKE '%a%';
9 """
10 cur.execute(sql)
11 for row in cur.fetchall():
12     print('row[0]=', row[0])
```

Answer:

```
row[0]= 9
```

problem2.py

```
1 # Problem 2
2
3 import sqlite3
4
5 con = sqlite3.connect('twitter_clone.db')
6 cur = con.cursor()
7
8 sql = """
9 INSERT INTO users (username, password) VALUES ('example', 'password');
10 """
11 cur.execute(sql)
12 con.commit()
13
14 sql = """
15 SELECT id,password FROM users WHERE username='example';
16 """
17 cur.execute(sql)
18 for row in cur.fetchall():
19     print('row[0]=', row[0])
20     print('row[1]=', row[1])
```

Answer:

```
row[0]= 9
row[1]= password
```

problem2b.py

```
1 # Problem 2b
2
3 import sqlite3
4
5 con = sqlite3.connect('twitter_clone.db')
6 cur = con.cursor()
7
8 sql = """
9 INSERT INTO users (username, password, age) VALUES ('example', 'password', 22);
10 """
11 cur.execute(sql)
12 con.commit()
13
14 sql = """
15 SELECT password,id FROM users WHERE username='example';
16 """
17 cur.execute(sql)
18 for row in cur.fetchall():
19     print('row[0]=', row[0])
20     print('row[1]=', row[1])
```

Answer:

```
row[0]= password
row[1]= 9
```

problem3.py

```
1 # Problem 3
2
3 import sqlite3
4
5 con = sqlite3.connect('twitter_clone.db')
6 cur = con.cursor()
7
8 username = 'example'
9 password = 'password'
10 sql = """
11 INSERT INTO users (username, password) VALUES ('"+username+"', '"+password+"');
12 """
13 cur.execute(sql)
14 con.commit()
15
16 sql = """
17 SELECT id,password FROM users WHERE username='example';
18 """
19 cur.execute(sql)
20 for row in cur.fetchall():
21     print('row[0]=', row[0])
22     print('row[1]=', row[1])
```

Answer:

```
row[0]= 9
row[1]= password
```

problem3b.py

```
1 # Problem 3b
2
3 import sqlite3
4
5 con = sqlite3.connect('twitter_clone.db')
6 cur = con.cursor()
7
8 username = "example", 'password'), ('example2"
9 password = 'password'
10 sql = """
11 INSERT INTO users (username, password) VALUES ('"+username+"', '"+password+"');
12 """
13 cur.execute(sql)
14 con.commit()
15
16 sql = """
17 SELECT id,password FROM users WHERE username LIKE 'example%' ORDER BY id;
18 """
19 cur.execute(sql)
20 for row in cur.fetchall():
21     print('row[0]=', row[0])
22     print('row[1]=', row[1])
```

Answer:

```
row[0]= 9
row[1]= password
row[0]= 10
row[1]= password
```

problem4.py

```
1 # Problem 4
2
3 import sqlite3
4
5 con = sqlite3.connect('twitter_clone.db')
6 cur = con.cursor()
7
8 username = 'example'
9 password = 'password'
10 sql = """
11 INSERT INTO users (username, password) VALUES (?, ?);
12 """
13 cur.execute(sql, [username, password])
14 con.commit()
15
16 sql = """
17 SELECT id,password FROM users WHERE username='example';
18 """
19 cur.execute(sql)
20 for row in cur.fetchall():
21     print('row[0]=', row[0])
22     print('row[1]=', row[1])
```

Answer:

```
row[0]= 9
row[1]= password
```

problem4b.py

```
1 # Problem 4b
2
3 import sqlite3
4
5 con = sqlite3.connect('twitter_clone.db')
6 cur = con.cursor()
7
8 username = "example", 'password', ('example2"
9 password = 'password'
10 sql = """
11 INSERT INTO users (username, password) VALUES (?, ?);
12 """
13 cur.execute(sql, [username, password])
14 con.commit()
15
16 sql = """
17 SELECT id,password FROM users WHERE username='example';
18 """
19 cur.execute(sql)
20 for row in cur.fetchall():
21     print('row[0]=', row[0])
22     print('row[1]=', row[1])
```

Answer:

```
(no output)
```

problem4c.py

```
1 # Problem 4c
2
3 import sqlite3
4
5 con = sqlite3.connect('twitter_clone.db')
6 cur = con.cursor()
7
8 username = "example"
9 password = 'password'
10 sql = """
11 INSERT INTO users (username, password) VALUES ('?', ?);
12 """
13 cur.execute(sql, [username])
14 con.commit()
15
16 sql = """
17 SELECT id,password FROM users WHERE username='?';
18 """
19 cur.execute(sql)
20 for row in cur.fetchall():
21     print('row[0]=', row[0])
22     print('row[1]=', row[1])
```

Answer:

```
row[0]= 9
row[1]= example
```

problem5.py

```
1 # Problem 5
2
3 import sqlite3
4
5 con = sqlite3.connect('twitter_clone.db')
6 cur = con.cursor()
7
8 username = 'Teal'
9 sql = """
10 SELECT id,password FROM users WHERE username=?;
11 """
12 cur.execute(sql, [username])
13 user_ids = []
14 for row in cur.fetchall():
15     user_ids.append(row[0])
16
17 for user_id in user_ids:
18     sql = """
19     SELECT count(*) FROM messages WHERE sender_id=?;
20     """
21     cur.execute(sql, [user_id])
22     for row in cur.fetchall():
23         print('row[0]=', row[0])
```

Answer:

```
row[0]= 2
```

problem5b.py

```
1 # Problem 5b
2
3 import sqlite3
4
5 con = sqlite3.connect('twitter_clone.db')
6 cur = con.cursor()
7
8 username = "Teal' OR username='Trump"
9 sql = """
10 SELECT id,password FROM users WHERE username='\"'+ username +\"'\" ORDER BY username;
11 """
12 cur.execute(sql)
13 user_ids = []
14 for row in cur.fetchall():
15     user_ids.append(row[0])
16
17 for user_id in user_ids:
18     sql = """
19     SELECT count(*) FROM messages WHERE sender_id=?;
20     """
21     cur.execute(sql, [user_id])
22     for row in cur.fetchall():
23         print('row[0]=', row[0])
```

Answer:

```
row[0]= 2
row[0]= 1
```

problem5c.py

```
1 # Problem 5c
2
3 import sqlite3
4
5 con = sqlite3.connect('twitter_clone.db')
6 cur = con.cursor()
7
8 username = "Teal' OR username='Trump"
9 sql = """
10 SELECT id,password FROM users WHERE username=?;
11 """
12 cur.execute(sql, [username])
13 user_ids = []
14 for row in cur.fetchall():
15     user_ids.append(row[0])
16
17 for user_id in user_ids:
18     sql = """
19     SELECT count(*) FROM messages WHERE sender_id=?;
20     """
21     cur.execute(sql, [user_id])
22     for row in cur.fetchall():
23         print('row[0]=', row[0])
```

Answer:

```
(no output)
```

problem6.py

```
1 # Problem 6
2
3 import sqlite3
4
5 con = sqlite3.connect('twitter_clone.db')
6 cur = con.cursor()
7
8 username = 'Teal'
9 sql = """
10 SELECT id FROM users WHERE username='''' + username + '''';
11 """
12 cur.execute(sql)
13 user_ids = []
14 for row in cur.fetchall():
15     user_ids.append(row[0])
16
17 for user_id in user_ids:
18     sql = """
19     SELECT id, sender_id, message, created_at FROM messages WHERE sender_id='''' + str(user_id) + '''
20     "" ORDER BY id;
21     """
22     cur.execute(sql)
23     for row in cur.fetchall():
24         print('id=', row[0], 'sender_id=', row[1], 'message=', row[2], 'created_at=', row[3])
```

Answer:

```
id= 9 sender_id= 7 message= I'm an adult created_at= 2021-11-17 14:35:45
id= 10 sender_id= 7 message= WTF is SQL?! I thought you liked the snake
thing. created_at= 2021-11-17 14:35:45
```

problem6b.py

```
1 # Problem 6b
2
3 import sqlite3
4
5 con = sqlite3.connect('twitter_clone.db')
6 cur = con.cursor()
7
8 username = 'Teal'
9 sql = """
10 SELECT id FROM users WHERE username=?;
11 """
12 cur.execute(sql, [username])
13 user_ids = []
14 for row in cur.fetchall():
15     user_ids.append(row[0])
16
17 for user_id in user_ids:
18     sql = """
19     SELECT id, sender_id, message, created_at FROM messages WHERE sender_id=? ORDER BY id;
20     """
21     cur.execute(sql, [user_id])
22     for row in cur.fetchall():
23         print('id=', row[0], 'sender_id=', row[1], 'message=', row[2], 'created_at=', row[3])
```

Answer:

```
id= 9 sender_id= 7 message= I'm an adult created_at= 2021-11-17 14:35:45
id= 10 sender_id= 7 message= WTF is SQL?! I thought you liked the snake
thing. created_at= 2021-11-17 14:35:45
```

problem6c.py

```
1 # Problem 6c
2
3 import sqlite3
4
5 con = sqlite3.connect('twitter_clone.db')
6 cur = con.cursor()
7
8 username = "Teal' OR username='Trump"
9 sql = """
10 SELECT id FROM users WHERE username='"" + username + ""' ORDER BY username;
11 """
12 cur.execute(sql)
13 user_ids = []
14 for row in cur.fetchall():
15     user_ids.append(row[0])
16
17 for user_id in user_ids:
18     sql = """
19     SELECT id, sender_id, message, created_at FROM messages WHERE sender_id="" + str(user_id) + "
20     "" ORDER BY id;
21     """
22     cur.execute(sql)
23     for row in cur.fetchall():
24         print('id=', row[0], 'sender_id=', row[1], 'message=', row[2], 'created_at=', row[3])
```

Answer:

```
id= 9 sender_id= 7 message= I'm an adult created_at= 2021-11-17 14:35:45
id= 10 sender_id= 7 message= WTF is SQL?! I thought you liked the snake
    thing. created_at= 2021-11-17 14:35:45
id= 1 sender_id= 1 message= I'm a baby created_at= 2021-11-14 14:30:00
```