

## Table Creation – users & calculations

- The first screenshot shows creation of two related tables.
- users stores unique usernames and emails with a timestamp for account creation.
- calculations records arithmetic operations performed by users, including operands, results, and timestamps.
- A foreign key (user\_id) links each calculation to a corresponding user, demonstrating a one-to-many relationship.
- Result: “Query returned successfully” confirms both tables were created without errors.

A)

The screenshot shows the pgAdmin interface with the following details:

- Object Explorer:** Shows the database structure under "fastapi\_db".
- Properties:** Shows the connection details: fastapi\_db/postgres@Local Postgres.
- SQL:** The query window contains the SQL code for creating the "users" and "calculations" tables.

```
1 ✓ CREATE TABLE users (
2     id SERIAL PRIMARY KEY,
3     username VARCHAR(50) NOT NULL UNIQUE,
4     email VARCHAR(100) NOT NULL UNIQUE,
5     created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
6 );
7
8 ✓ CREATE TABLE calculations (
9     id SERIAL PRIMARY KEY,
10    operation VARCHAR(20) NOT NULL,
11    operand_a FLOAT NOT NULL,
12    operand_b FLOAT NOT NULL,
13    result FLOAT NOT NULL,
14    timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
15    user_id INTEGER NOT NULL,
16    FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE
17 );
```

- Data Output:** Shows the message: "Query returned successfully in 75 msec."

## Data Insertion

- Two users (alice, bob) and three calculations were inserted into the database.
- This verifies correct use of INSERT INTO for both parent and child tables.
- The foreign key relationship ensures that each calculation references a valid user ID.
- Result: “INSERT 0 3” confirms that three rows were successfully added to calculations.

B)

The screenshot shows the PgAdmin 4 interface. In the Object Explorer, under the 'fastapi\_db' database, the 'Tables' node is selected. Two queries are run in the main pane:

```

1 ✓ INSERT INTO users (username, email)
2   VALUES
3     ('alice', 'alice@example.com'),
4     ('bob', 'bob@example.com');
5
6 ✓ INSERT INTO calculations (operation, operand_a, operand_b, result, user_id)
7   VALUES
8     ('add', 2, 3, 5, 1),
9     ('divide', 10, 2, 5, 1),
10    ('multiply', 4, 5, 20, 2);

```

The 'Data Output' tab shows the results of the 'INSERT' statements:

```

INSERT 0 3
Query returned successfully in 46 msec.

```

## Data Retrieval (SELECT Queries & JOIN)

- Individual queries retrieved data from users and calculations tables.
- The JOIN query combines both tables using user\_id, clearly showing which user performed each operation.
- The output displays rows for alice and bob with their respective operations (add, divide, multiply).
- Result: Confirms relational integrity and demonstrates successful data linking between tables.

C)

The screenshot shows the pgAdmin interface. In the Object Explorer, the database 'fastapi\_db' is selected. The Query tab contains the following SQL code:

```
-- Retrieve all users
SELECT * FROM users;

-- Retrieve all calculations
SELECT * FROM calculations;

-- Join users and calculations
SELECT u.username, c.operation, c.operand_a, c.operand_b, c.result
FROM calculations c
JOIN users u ON c.user_id = u.id;
```

The Data Output tab shows the results of the last query:

username	operation	operand_a	operand_b	result
alice	add	2	3	5
alice	divide	10	2	5
bob	multiply	4	5	20

## Updating Records

- The UPDATE statement modifies an existing record in calculations, setting result = 6 for the record with id = 1.
- This demonstrates how to modify specific values while preserving other columns.
- Result: “UPDATE 1” confirms one record was successfully updated.

D)

The screenshot shows the pgAdmin interface. In the Object Explorer, the database 'fastapi\_db' is selected. The Query tab contains the following SQL code:

```
UPDATE calculations
SET result = 6
WHERE id = 1; -- or whichever row you want to update
```

The Data Output tab shows the message:

UPDATE 1  
Query returned successfully in 58 msec.

## Deleting Records

- The DELETE statement removes the calculation with id = 2.
- This verifies cascade and referential behavior—deleting a calculation doesn't affect the parent user record.
- *Result:* “DELETE 1” confirms the record was successfully removed.

E)

The screenshot shows the pgAdmin interface. The left pane is the Object Explorer, displaying a tree structure of database objects under 'fastapi\_db'. The 'calculations' table is selected. The right pane contains a query editor window with the following content:

```
1 <--> DELETE FROM calculations
2 WHERE id = 2; -- example record to remove
```

Below the query editor, the 'Messages' tab is active, showing the output:

```
DELETE 1
Query returned successfully in 78 msec.
```

## Overall Outcome

- All Docker containers (FastAPI, PostgreSQL, pgAdmin) are running correctly.
- The database operations confirm understanding of CRUD principles and relational modeling (one-to-many relationships using foreign keys).
- Queries executed successfully with expected results, fulfilling the assignment requirements.

