1 THEORY QUESTIONS (20 POINTS)

1. Describe the purpose of the following SQL commands, and write down the syntax for each command: SELECT, INSERT, UPDATE, DELETE, JOIN.

SELECT: Used to retrieve data from one or more tables.
SELECT column1, column2, ...
FROM table_name
WHERE condition;

INSERT: Used to insert new data into a table.
INSERT INTO table_name (column1, column2, ...)
VALUES (value1, value2, ...);

UPDATE: Used to modify existing data in a table.
UPDATE table_name
SET column1 = value1, column2 = value2, ...
WHERE condition;

DELETE: Used to delete data from a table.
DELETE FROM table_name
WHERE condition;

JOIN: Used to combine rows from two or more tables based on a related column.
SELECT columns
FROM table1
JOIN table2
ON table1.column = table2.column;

2. Describe the differences between CSV and JSON data formats. When might you choose one format over the other?

- CSV (Comma-Separated Values):
- A data record is any line in the file that has one or more fields separated by commas.
- Ideal for data in tables.
- Simple text editors and spreadsheet applications can handle and interpret them more easily.
- Selected for uncomplicated, flat data structures devoid of hierarchical or nested connections.

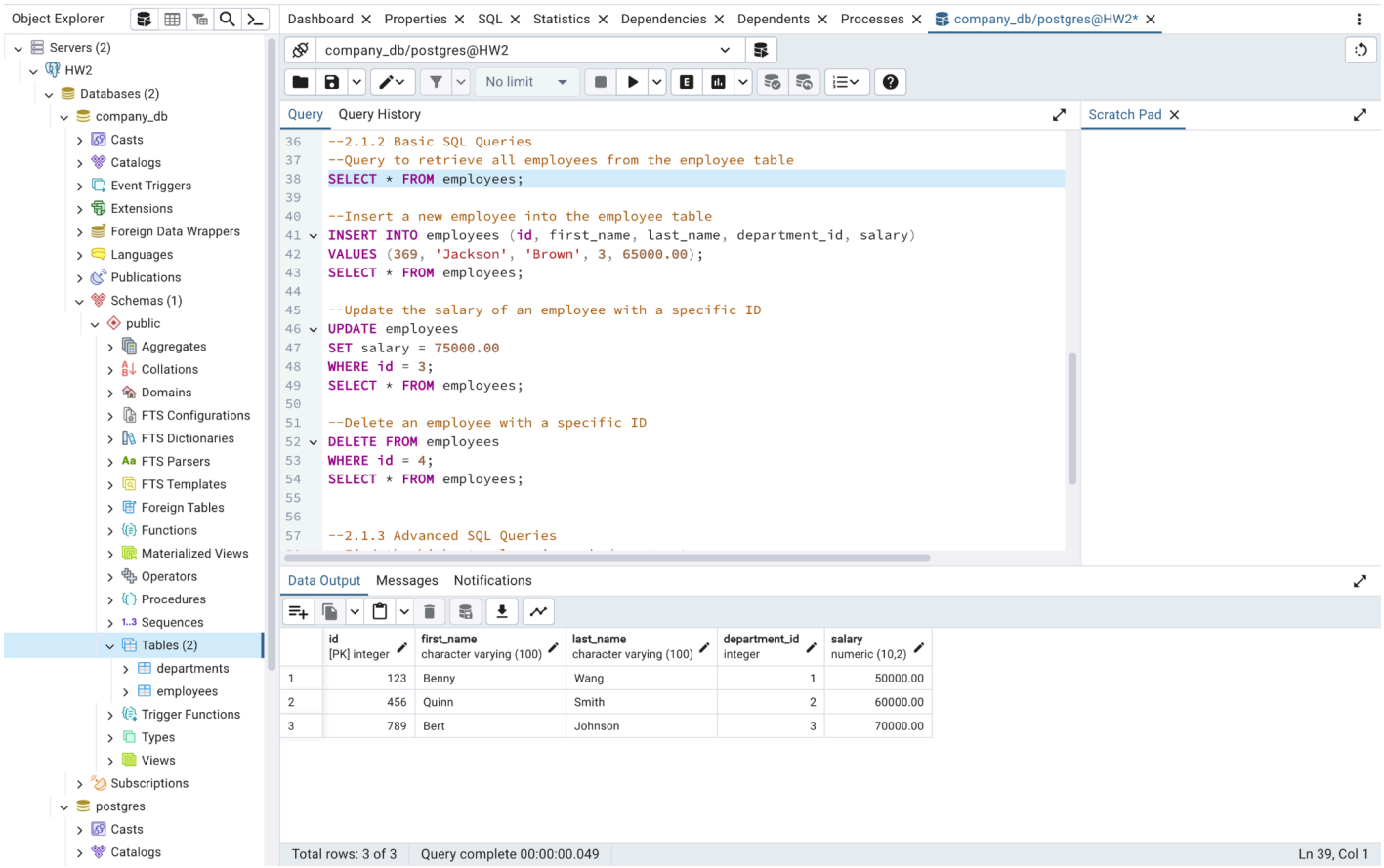
- JSON (JavaScript Object Notation):
- Text format that is simple for computers to understand and produce as well as for humans to read and write.
- Key-value pairs are used to organize data, and nested and hierarchical data structures are supported.
- Selected for intricate data structures, particularly with relation to online services or APIs.

2 EXPERIMENT (30 POINTS)

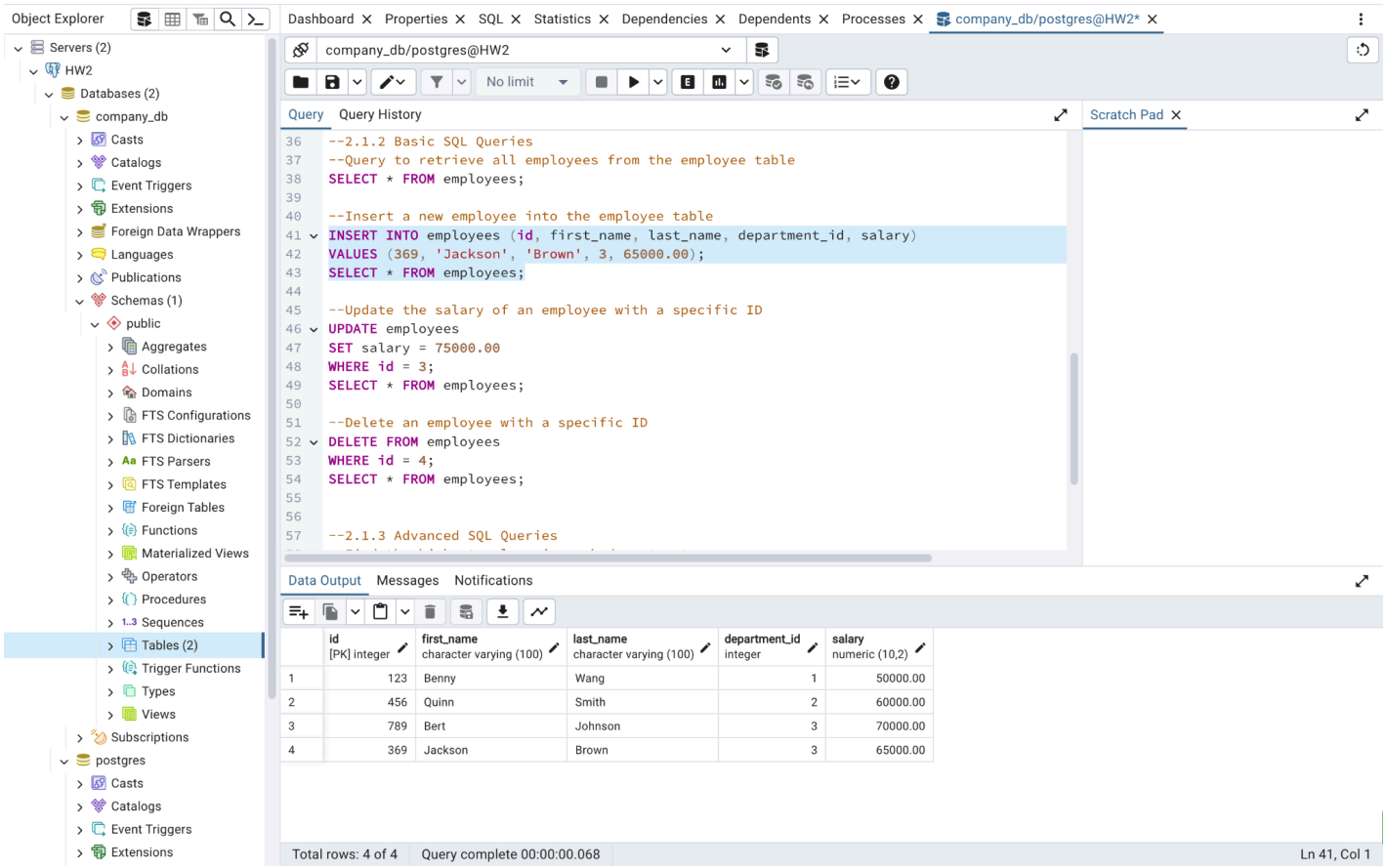
```
-- Create the database
CREATE DATABASE company_db;

-- Use the database
USE company_db;

-- Create departments table
CREATE TABLE departments (
    id INT PRIMARY KEY,
    department_name VARCHAR(100) NOT NULL
);

-- Create employees table
CREATE TABLE employees (
    id INT PRIMARY KEY,
    first_name VARCHAR(100) NOT NULL,
    last_name VARCHAR(100) NOT NULL,
    department_id INT,
    salary DECIMAL(10, 2),
    FOREIGN KEY (department_id) REFERENCES departments(id)
);

-- Insert data into departments table
INSERT INTO departments (id, department_name) VALUES
(1, 'Technology'),
(2, 'Finance'),
(3, 'Operations');

-- Insert data into employees table
INSERT INTO employees (id, first_name, last_name, department_id, salary) VALUES
(123, 'Benny', 'Wang', 1, 50000.00),
```

(456, 'Quinn', 'Smith', 2, 60000.00),
(789, 'Bert', 'Johnson', 3, 70000.00);

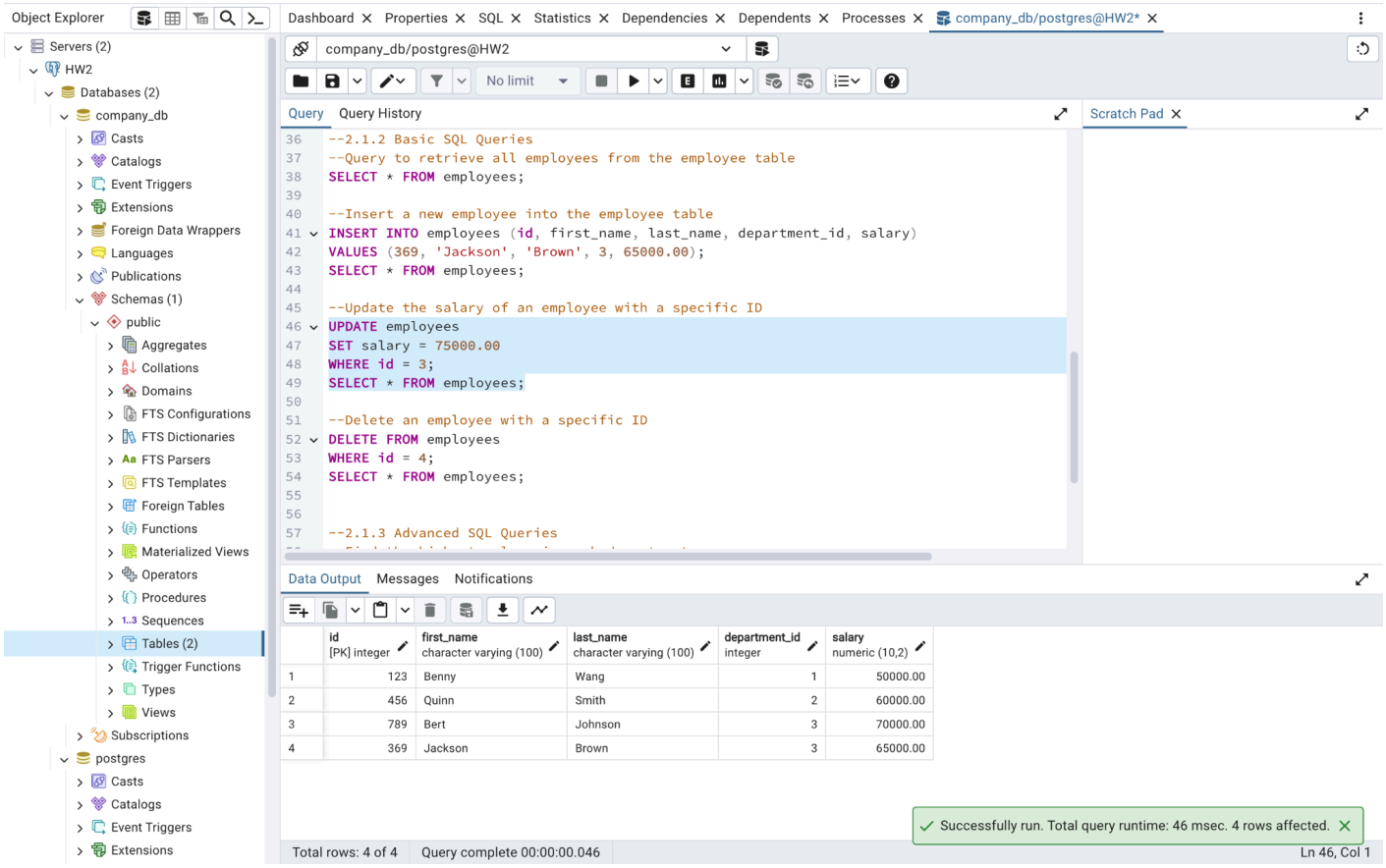
--2.1.2 Basic SQL Queries
--Query to retrieve all employees from the employee table
SELECT * FROM employees;



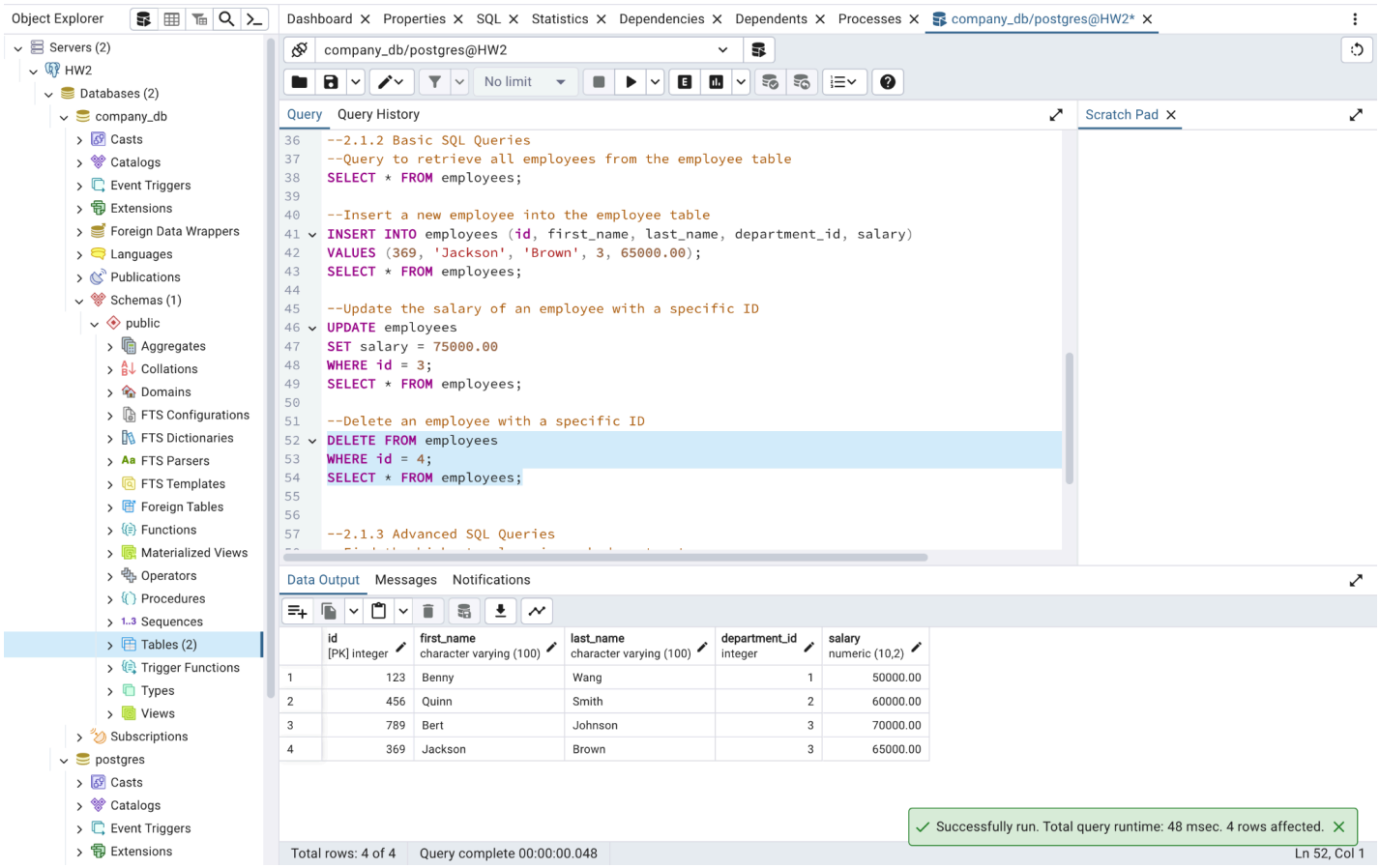
--Insert a new employee into the employee table
INSERT INTO employees (id, first_name, last_name, department_id, salary)
VALUES (369, 'Jackson', 'Brown', 3, 65000.00);
SELECT * FROM employees;



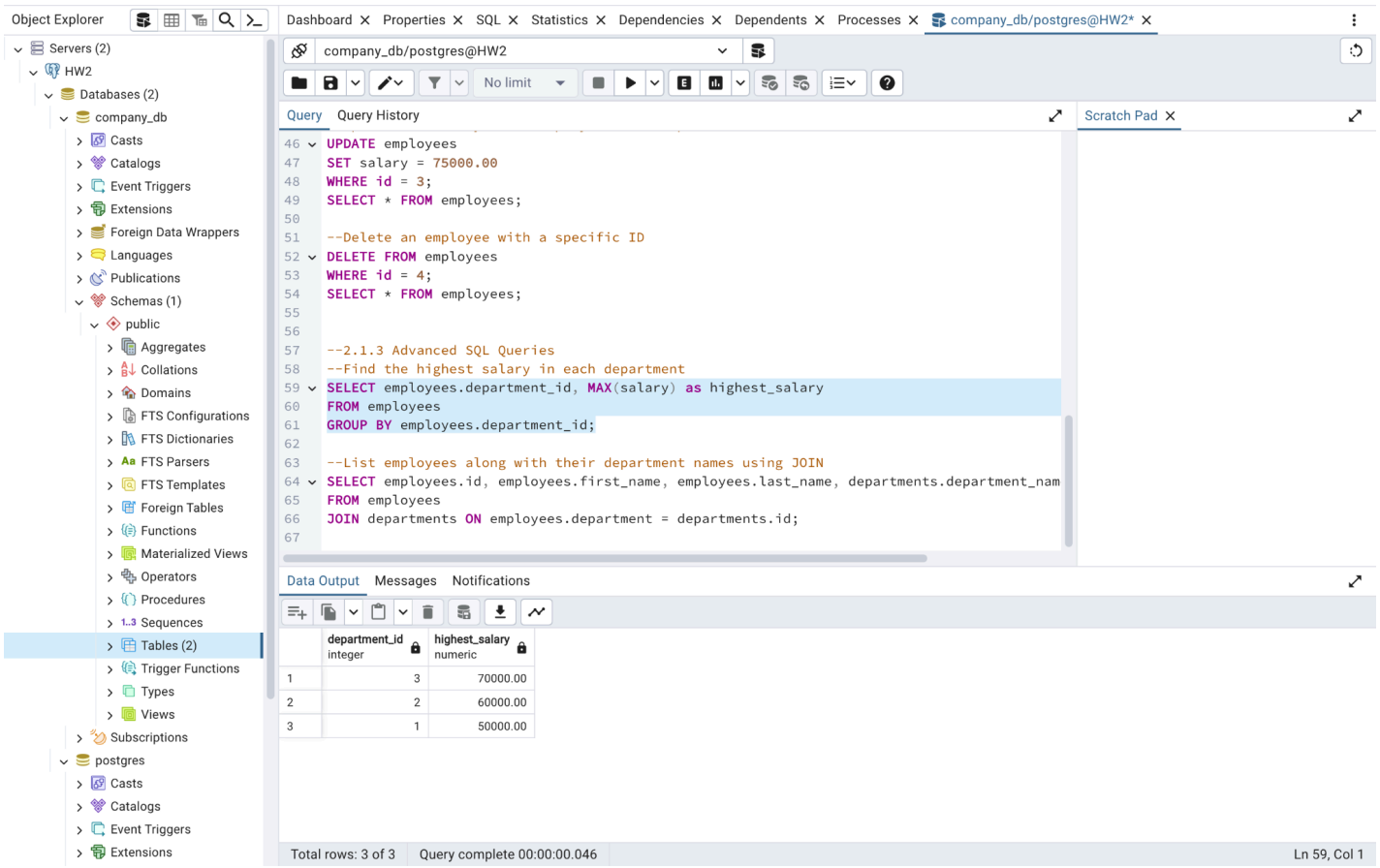
--Update the salary of an employee with a specific ID
UPDATE employees
SET salary = 75000.00
WHERE id = 3;
SELECT * FROM employees;



--Delete an employee with a specific ID
DELETE FROM employees
WHERE id = 4;
SELECT * FROM employees;



--2.1.3 Advanced SQL Queries
--Find the highest salary in each department
SELECT employees.department_id, MAX(salary) as highest_salary
FROM employees
GROUP BY employees.department_id;



--List employees along with their department names using JOIN
SELECT employees.id, employees.first_name, employees.last_name,
departments.department_name, employees.salary
FROM employees
JOIN departments ON employees.department_id = departments.id;