**6.3- Inner versus Outer Joins:**

**1. Create employees and departments Tables (for first three queries)**

-- Create the departments table

CREATE TABLE departments (

department\_id INT PRIMARY KEY,

department\_name VARCHAR(50)

);

INSERT INTO departments (department\_id, department\_name) VALUES (10, 'HR');

INSERT INTO departments (department\_id, department\_name) VALUES (20, 'IT');

INSERT INTO departments (department\_id, department\_name) VALUES (30, 'Finance');

INSERT INTO departments (department\_id, department\_name) VALUES (40, 'Sales');

CREATE TABLE employees (

employee\_id INT PRIMARY KEY,

first\_name VARCHAR(50),

last\_name VARCHAR(50),

department\_id INT,

FOREIGN KEY (department\_id) REFERENCES departments(department\_id)

);

INSERT INTO employees (employee\_id, first\_name, last\_name, department\_id)

VALUES (1, 'John', 'Smith', 10);

INSERT INTO employees (employee\_id, first\_name, last\_name, department\_id)

VALUES (2, 'Jane', 'Doe', 20);

INSERT INTO employees (employee\_id, first\_name, last\_name, department\_id)

VALUES (3, 'Alice', 'Johnson', NULL); -- No department assigned

INSERT INTO employees (employee\_id, first\_name, last\_name, department\_id)

VALUES (4, 'Bob', 'Brown', 30);

**2. Create clients and events Tables (for fourth query)**

-- Create the clients table

CREATE TABLE clients (

client\_id INT PRIMARY KEY,

first\_name VARCHAR(50),

last\_name VARCHAR(50)

);

INSERT INTO clients (client\_id, first\_name, last\_name)

VALUES (1, 'John', 'Smith');

INSERT INTO clients (client\_id, first\_name, last\_name)

VALUES (2, 'Jane', 'Doe');

INSERT INTO clients (client\_id, first\_name, last\_name)

VALUES (3, 'Alice', 'Johnson');

-- Create the events table

CREATE TABLE events (

event\_id INT PRIMARY KEY,

event\_date DATE,

description VARCHAR(100),

client\_id INT,

FOREIGN KEY (client\_id) REFERENCES clients(client\_id)

);

-- Insert data into the events table

INSERT INTO events (event\_id, event\_date, description, client\_id)

VALUES (1, '2024-09-05', 'Birthday Party', 1);

INSERT INTO events (event\_id, event\_date, description, client\_id)

VALUES (2, '2024-09-10', 'Wedding', 2);

**3. Create shifts and shift\_assignments Tables (for fifth query)**

sql

Copy code

-- Create the shifts table

CREATE TABLE shifts (

shift\_id INT PRIMARY KEY,

shift\_description VARCHAR(100)

);

-- Insert data into the shifts table

INSERT INTO shifts (shift\_id, shift\_description)

VALUES (1, 'Morning Shift');

INSERT INTO shifts (shift\_id, shift\_description)

VALUES (2, 'Evening Shift');

INSERT INTO shifts (shift\_id, shift\_description)

VALUES (3, 'Night Shift');

-- Create the shift\_assignments table

CREATE TABLE shift\_assignments (

assignment\_id INT PRIMARY KEY,

shift\_id INT,

assignment\_date DATE,

FOREIGN KEY (shift\_id) REFERENCES shifts(shift\_id)

);

-- Insert data into the shift\_assignments table

INSERT INTO shift\_assignments (assignment\_id, shift\_id, assignment\_date)

VALUES (1, 1, '2024-09-05'); -- Morning shift assigned

-- Evening shift has no assignment

INSERT INTO shift\_assignments (assignment\_id, shift\_id, assignment\_date)

VALUES (2, 3, '2024-09-06');

ANSWRES:

1.**Employees including those not assigned to a department:**

SELECT employees.first\_name, employees.last\_name, departments.department\_name

FROM employees

LEFT JOIN departments ON employees.department\_id = departments.department\_id;

2.**Employees and departments even if no employees are assigned:**

SELECT employees.first\_name, employees.last\_name, departments.department\_name

FROM employees

RIGHT JOIN departments ON employees.department\_id = departments.department\_id;

3.**Employees not assigned to a department and departments with no employees:**

SELECT employees.first\_name, employees.last\_name, departments.department\_name

FROM employees

FULL OUTER JOIN departments ON employees.department\_id = departments.department\_id;

4.**Clients and their events, including clients without scheduled events:**

SELECT clients.first\_name, clients.last\_name, events.event\_date, events.description

FROM clients

LEFT JOIN events ON clients.client\_id = events.client\_id;

5.**Shift descriptions and assignment dates, even if no date is assigned:**

SELECT shifts.shift\_description, shift\_assignments.assignment\_date

FROM shifts

LEFT JOIN shift\_assignments ON shifts.shift\_id = shift\_assignments.shift\_id;