**6-2: Join Clauses**

**1.Create departments and locations Tables for First Query**

-- Create the locations table

CREATE TABLE locations (

location\_id INT PRIMARY KEY,

city VARCHAR(50),

country VARCHAR(50)

);

-- Insert data into the locations table

INSERT INTO locations (location\_id, city, country) VALUES (1000, 'New York', 'USA');

INSERT INTO locations (location\_id, city, country) VALUES (1400, 'Seattle', 'USA');

INSERT INTO locations (location\_id, city, country) VALUES (1500, 'Toronto', 'Canada');

-- Create the departments table

CREATE TABLE departments (

department\_id INT PRIMARY KEY,

department\_name VARCHAR(50),

location\_id INT,

FOREIGN KEY (location\_id) REFERENCES locations(location\_id)

);

-- Insert data into the departments table

INSERT INTO departments (department\_id, department\_name, location\_id) VALUES (10, 'HR', 1000);

INSERT INTO departments (department\_id, department\_name, location\_id) VALUES (20, 'IT', 1400);

INSERT INTO departments (department\_id, department\_name, location\_id) VALUES (30, 'Finance', 1400);

INSERT INTO departments (department\_id, department\_name, location\_id) VALUES (50, 'Sales', 1500);

**2. Create d\_play\_list\_items, d\_track\_listings, and d\_cds Tables for Second Query**

-- Create d\_play\_list\_items table

CREATE TABLE d\_play\_list\_items (

song\_id INT PRIMARY KEY,

cd\_number INT,

comments VARCHAR(100)

);

-- Insert data into d\_play\_list\_items table

INSERT INTO d\_play\_list\_items (song\_id, cd\_number, comments) VALUES (1, 101, 'Great Song');

INSERT INTO d\_play\_list\_items (song\_id, cd\_number, comments) VALUES (2, 102, 'Classic Hit');

-- Create d\_track\_listings table

CREATE TABLE d\_track\_listings (

song\_id INT PRIMARY KEY,

title VARCHAR(100),

FOREIGN KEY (song\_id) REFERENCES d\_play\_list\_items(song\_id)

);

-- Insert data into d\_track\_listings table

INSERT INTO d\_track\_listings (song\_id, title) VALUES (1, 'Song A');

INSERT INTO d\_track\_listings (song\_id, title) VALUES (2, 'Song B');

-- Create d\_cds table

CREATE TABLE d\_cds (

cd\_number INT PRIMARY KEY,

cd\_name VARCHAR(100)

);

-- Insert data into d\_cds table

INSERT INTO d\_cds (cd\_number, cd\_name) VALUES (101, 'CD 1');

INSERT INTO d\_cds (cd\_number, cd\_name) VALUES (102, 'CD 2');

**3. Create countries and regions Tables for Fourth Query**

-- Create the regions table

CREATE TABLE regions (

region\_id INT PRIMARY KEY,

region\_name VARCHAR(50)

);

-- Insert data into the regions table

INSERT INTO regions (region\_id, region\_name) VALUES (1, 'Americas');

INSERT INTO regions (region\_id, region\_name) VALUES (2, 'Europe');

-- Create the countries table

CREATE TABLE countries (

country\_id INT PRIMARY KEY,

country\_name VARCHAR(50),

region\_id INT,

FOREIGN KEY (region\_id) REFERENCES regions(region\_id)

);

-- Insert data into the countries table

INSERT INTO countries (country\_id, country\_name, region\_id) VALUES (1, 'USA', 1);

INSERT INTO countries (country\_id, country\_name, region\_id) VALUES (2, 'Canada', 1);

INSERT INTO countries (country\_id, country\_name, region\_id) VALUES (3, 'UK', 2);

**4. Create employees and jobs Tables for Fifth, Sixth, and Seventh Queries**

-- Create the jobs table

CREATE TABLE jobs (

job\_id INT PRIMARY KEY,

job\_title VARCHAR(50),

max\_salary INT

);

-- Insert data into the jobs table

INSERT INTO jobs (job\_id, job\_title, max\_salary) VALUES (1, 'Stock Clerk', 12000);

INSERT INTO jobs (job\_id, job\_title, max\_salary) VALUES (2, 'Manager', 15000);

INSERT INTO jobs (job\_id, job\_title, max\_salary) VALUES (3, 'Developer', 13000);

-- Create the employees table

CREATE TABLE employees (

employee\_id INT PRIMARY KEY,

first\_name VARCHAR(50),

last\_name VARCHAR(50),

hire\_date DATE,

job\_id INT,

manager\_id INT,

department\_id INT,

email VARCHAR(50),

FOREIGN KEY (job\_id) REFERENCES jobs(job\_id),

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

FOREIGN KEY (manager\_id) REFERENCES employees(employee\_id)

);

-- Insert data into the employees table

INSERT INTO employees (employee\_id, first\_name, last\_name, hire\_date, job\_id, manager\_id, department\_id, email)

VALUES (1, 'John', 'Smith', '1994-06-07', 1, NULL, 10, 'john.smith@example.com');

INSERT INTO employees (employee\_id, first\_name, last\_name, hire\_date, job\_id, manager\_id, department\_id, email)

VALUES (2, 'Jane', 'Doe', '1995-07-10', 2, 1, 20, 'jane.doe@example.com');

INSERT INTO employees (employee\_id, first\_name, last\_name, hire\_date, job\_id, manager\_id, department\_id, email)

VALUES (3, 'Bob', 'Johnson', '1994-06-07', 3, 1, 30, 'bob.johnson@example.com');

**1. Join locations and departments on location\_id and limit the results to location\_id = 1400:**

SELECT departments.department\_id, departments.department\_name, locations.location\_id, locations.city

FROM departments

JOIN locations ON departments.location\_id = locations.location\_id

WHERE locations.location\_id = 1400;

**2. Join d\_play\_list\_items, d\_track\_listings, and d\_cds tables using the JOIN USING syntax:**

SELECT d\_play\_list\_items.song\_id, d\_cds.cd\_number, d\_track\_listings.title, d\_play\_list\_items.comments

FROM d\_play\_list\_items

JOIN d\_track\_listings USING (song\_id)

JOIN d\_cds USING (cd\_number);

**3. Display city, department name, location ID, and department ID for departments 10, 20, and 30 in Seattle:**

SELECT locations.city, departments.department\_name, departments.location\_id, departments.department\_id

FROM departments

JOIN locations ON departments.location\_id = locations.location\_id

WHERE departments.department\_id IN (10, 20, 30)

AND locations.city = 'Seattle';

**4. Display country name, region ID, and region name for Americas:**

SELECT countries.country\_name, regions.region\_id, regions.region\_name

FROM countries

JOIN regions ON countries.region\_id = regions.region\_id

WHERE regions.region\_name = 'Americas';

**5. Join employees and jobs tables and display employees in jobs with max salary > 12,000:**

SELECT employees.first\_name, employees.last\_name, employees.hire\_date, jobs.job\_id, jobs.job\_title, jobs.max\_salary

FROM employees

JOIN jobs ON employees.job\_id = jobs.job\_id

WHERE jobs.max\_salary > 12000;

**6. Display job title, first name, last name, and email for employees who are stock clerks:**

SELECT jobs.job\_title, employees.first\_name, employees.last\_name, employees.email

FROM employees

JOIN jobs ON employees.job\_id = jobs.job\_id

WHERE jobs.job\_title = 'Stock Clerk';

**7. Self-join employees to display employee and manager details (employee ID, first name, last name, manager ID, manager's first name, and last name):**

SELECT e.employee\_id, e.first\_name, e.last\_name, e.manager\_id, m.first\_name AS manager\_first\_name, m.last\_name AS manager\_last\_name

FROM employees e

JOIN employees m ON e.manager\_id = m.employee\_id;

**8. Display location ID, city, and department name for all Canadian locations:**

SELECT locations.location\_id, locations.city, departments.department\_name

FROM locations

JOIN departments ON locations.location\_id = departments.location\_id

WHERE locations.country = 'Canada';

**9. Display manager ID, department ID, department name, first name, and last name for employees in departments 80, 90, 110, and 190:**

SELECT employees.manager\_id, departments.department\_id, departments.department\_name, employees.first\_name, employees.last\_name FROM employees

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

WHERE departments.department\_id IN (80, 90, 110, 190);

**10. Display employee ID, last name, department ID, department name, and hire date for employees hired on June 7, 1994:**

SELECT employees.employee\_id, employees.last\_name, employees.department\_id, departments.department\_name, employees.hire\_date

FROM employees

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

WHERE employees.hire\_date = '1994-06-07';