**Lab 07**

Student Name: **John Okon Ansa**  
Student ID: **619952**

**3. Query Exercises**

Write the SQL query that answers each of the following questions.

You got it! I'll add four more questions to the **Basic Selection** section of the lab assignment, focusing on DISTINCT, filtering (WHERE clause), and ordering (ORDER BY).

Here are the four new questions integrated into the lab assignment:

**3. Query Exercises:**

Write the SQL query that answers each of the following questions.

**A. Selection:**

1. List the **name** and **salary** of all employees.

**SELECT** name, salary **FROM** employee

1. Find the names of all **projects** located in **Florida (FL)**.

**SELECT** project\_name

**FROM** project

**WHERE** location = 'FL'

1. Retrieve the emp\_id and project\_id for employees working on **Project 1**.

**SELECT** emp\_id, project\_id

**FROM** employee\_project

**WHERE** project\_id = 1

1. **Find all unique (distinct) states** where employee addresses are located.

**SELECT** **DISTINCT** state

**FROM** address

1. List the **names** and **salaries** of all employees who earn a salary **less than $150,000**.

**SELECT** name, salary

**FROM** employee

**WHERE** salary < 150000

1. List the **project names** and their **estimated days**, ordered from the **longest duration to the shortest**.

**SELECT** project\_name, estimated\_days

**FROM** project

**ORDER** **BY** estimated\_days **DESC**

1. Find the emp\_ids of employees who are assigned to a project, listing each emp\_id only **once**.

**SELECT** **DISTINCT** emp\_id

**FROM** employee\_project

**B. Aggregates and Grouping:**

1. Calculate the **average salary** of all employees.

**SELECT** **AVG**(salary) **AS** AverageSalary

**FROM** employee

1. Find the **maximum** estimated\_days for any single project.

**SELECT** project\_id, **MAX**(estimated\_days) **As** Maximium

**FROM** project

**GROUP BY** project\_id;

1. For each **department**, report the dept\_id and the **total salary** expenditure.

**SELECT** dept\_id, **SUM**(salary) **AS** TotalSalary

**FROM** employee

**GROUP** **BY** dept\_id;

1. Find the dept\_id of departments that have an **average employee salary greater than $150,000**.

**SELECT** dept\_id

**FROM** employee

**GROUP BY** dept\_id

**HAVING** **AVG**(salary) > 150000;

**C. Joins:**

1. List the **employee name** and the **city** where they live. (Join Employee and Address).

**SELECT** e.name **AS** EmployeeName, a.city

**FROM** employee e

**INNER JOIN** address a **ON** e.address\_id = a.address\_id;

1. List **all departments** and the **names** of the employees who belong to them. Include departments that may not currently have any employees. (Join Department and Employee). *Note: Based on the sample data, all departments have employees, but this query structure is key for future scenarios.*

**SELECT** d.name **AS** DepartmentName, e.name **AS** EmployeeName

**FROM** department d

**LEFT JOIN** employee e **ON** d.dept\_id = e.dept\_id;

1. Find the **employee name** and the **name of the projects** they are working on. (Join Employee, Employee\_Project, and Project).

**SELECT** e.name **AS** employee\_name, p.project\_name

**FROM** employee e

**JOIN** employee\_project ep **ON** e.emp\_id = ep.emp\_id

**JOIN** project p **ON** ep.project\_id = p.project\_id;

**D. Subqueries**

1. Find the **name** of the employee who has the **highest salary** (Use a subquery in the WHERE clause).

**SELECT** name

**FROM** employee

**WHERE** salary = (**SELECT** **MAX**(salary) **FROM** employee);

1. List the **names** of employees who work on a project that has an estimated\_days of **180** (Use an IN or EXISTS subquery).

**SELECT** e.name

**FROM** employee e

**WHERE** e.emp\_id **IN** (

**SELECT** ep.emp\_id

**FROM** employee\_Project ep

**JOIN** project p ON ep.project\_id = p.project\_id

**WHERE** p.estimated\_days = 180

);

1. Find the project\_id of all projects that have an **estimated duration greater than the average estimated duration** of all projects (Use a subquery in the WHERE clause).

**SELECT** project\_id

**FROM** project

**WHERE** estimated\_days > (**SELECT** **AVG**(estimated\_days) **FROM** project);

-- \*\*\* 1. Insert Data into Address Table \*\*\*

INSERT INTO Address (address\_id, city, state, zipcode) VALUES

(1, 'Fairfield', 'IA', '52556'),

(2, 'Iowa City', 'IA', '52440'),

(3, 'Morrison', 'IL', '61270'),

(4, 'Orlando', 'FL', '34565'),

(5, 'Tampa', 'FL', '31765');

-- \*\*\* 2. Insert Data into Department Table \*\*\*

INSERT INTO Department (dept\_id, name) VALUES

(1, 'Tech'),

(2, 'HR'),

(3, 'Finance'),

(4, 'Marketing');

-- \*\*\* 3. Insert Data into Project Table \*\*\*

INSERT INTO Project (project\_id, project\_name, estimated\_days, location) VALUES

(1, 'X', 180, 'FL'),

(2, 'Y', 60, 'FL'),

(3, 'Z', 80, 'IA');

-- \*\*\* 4. Insert Data into Employee Table \*\*\*

-- NOTE: This depends on Address (address\_id) and Department (dept\_id) being populated first.

INSERT INTO Employee (emp\_id, name, salary, address\_id, dept\_id) VALUES

(111, 'Zaineh', 100000, 1, 1),

(112, 'Yasmeen', 160000, 2, 4),

(113, 'Mira', 140000, 3, 3),

(114, 'Shimaa', 200000, 4, 2),

(115, 'Dean', 150000, 5, 1);

-- \*\*\* 5. Insert Data into Employee\_Project Table \*\*\*

-- NOTE: This depends on Employee (emp\_id) and Project (project\_id) being populated first.

INSERT INTO Employee\_Project (emp\_id, project\_id) VALUES

(115, 1),

(115, 2),

(115, 3),

(114, 1),

(114, 3),

(111, 1),

(111, 2);