Lab 07

This lab will guide you through creating a simple database based on the provided schema . You will then write SQL queries to answer various business questions.

Employee

emp_id	name	salary	address_id	dept_id
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

Address

address_id	city	state	zipcode
1	Fairfield	IA	52556
2	Iowa City	IA	52440
3	Morrison	IL	61270
4	Orlando	FL	34565
5	Tampa	FL	31765

Department

dept_id	name
1	Tech
2	HR
3	Finance
4	Marketing

Employee_Project

emp_id	project_id
115	1
115	2
115	3
114	1
114	3
111	1
111	2

Project

project_id	project_name	estimated_days	location
1	X	180	FL
2	Y	60	FL
3	Z	80	IA

1. Database Schema and Table Creation

The database consists of five tables with the following relationships:

- **Employee Address** (via `address_id`)
- **Employee Department** (via `dept_id`)
- **Employee Project** (Many-to-Many, linked via **Employee_Project** table)

Your first task is to create these five tables using SQL. You should choose appropriate data types and primary/foreign key constraints.

Table Name	Primary Key	Foreign Keys	Columns
Employee	`emp_id`	`address_id` (references Address), `dept_id` (references Department)	`emp_id` (INT), `name` (VARCHAR), `salary` (INT), `address_id` (INT), `dept_id` (INT)
Address	`address_id`	None	`address_id` (INT), `city` (VARCHAR), `state` (VARCHAR), `zipcode` (VARCHAR)
Department	`dept_id`	None	`dept_id` (INT), `name` (VARCHAR)
Employee_Project	Composite: (`emp_id`, `project_id`)	`emp_id` (references Employee), `project_id` (references Project)	`emp_id` (INT), `project_id` (INT)
Project	`project_id`	None	`project_id` (INT), `project_name` (VARCHAR), `estimated_days` (INT), `location` (VARCHAR)

Table Creation SQL Sample

Write the `CREATE TABLE` statements for all five tables. Ensure you include `NOT NULL` constraints where appropriate and define the primary and foreign keys.

SQL

```
-- Example for Address table
CREATE TABLE Address (
    address_id INT PRIMARY KEY,
    city VARCHAR(50) NOT NULL,
    state CHAR(2) NOT NULL,
    zipcode VARCHAR(10)
);

-- Complete the remaining four tables: Department, Project, Employee,
Employee_Project
```

---

2. Data Insertion

Insert the sample data [**END OF THIS DOCUMENT**] from the provided schema into the tables you created.

---

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.
2. Find the names of all **projects** located in **Florida (FL)**.
3. Retrieve the `emp_id` and `project_id` for employees working on **Project 1**.
4. **Find all unique (distinct) states** where employee addresses are located. (Uses
5. List the **names** and **salaries** of all employees who earn a salary **less than $150,000**.
6. List the **project names** and their **estimated days**, ordered from the **longest duration to the shortest**.
7. Find the `emp_id`s of employees who are assigned to a project, listing each `emp_id` only **once**.

B. Aggregates and Grouping:

1. Calculate the **average salary** of all employees.
2. Find the **maximum** `estimated_days` for any single project.
3. For each **department**, report the `dept_id` and the **total salary** expenditure.
4. Find the `dept_id` of departments that have an **average employee salary greater than $150,000**.

C. Joins:

1. List the **employee name** and the **city** where they live. (Join `Employee` and `Address`).
2. 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.*
3. Find the **employee name** and the **name of the projects** they are working on. (Join `Employee`, `Employee_Project`, and `Project`).

D. Subqueries

1. Find the **name** of the employee who has the **highest salary** (Use a subquery in the `WHERE` clause).
2. List the **names** of employees who work on a project that has an `estimated_days` of **180** (Use an `IN` or `EXISTS` subquery).
3. 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).

```
-- *** 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);
```