### **ASSIGNMENT1**

### Q1: ****Create a Database****

Create a database called CompanyDB to store company-related information.

### Q2: ****Create Tables with Relationships****

Create the following three tables: **Departments**, **Employees**, and **Projects**, which are interlinked with foreign keys:

**Departments**: Stores department information.

* 1. DeptID: INT, Primary Key, Auto Increment.
  2. DeptName: VARCHAR(50), NOT NULL, UNIQUE.
  3. Location: VARCHAR(100), NOT NULL.

**Employees**: Stores employee information.

* 1. EmpID: INT, Primary Key, Auto Increment.
  2. FirstName: VARCHAR(50), NOT NULL.
  3. LastName: VARCHAR(50), NOT NULL.
  4. DOB: DATE.
  5. Email: VARCHAR(100), UNIQUE, NOT NULL.
  6. IsManager: BOOLEAN (either TRUE or FALSE).
  7. Salary: DECIMAL(10, 2), NOT NULL, CHECK (Salary > 0).
  8. DeptID: INT, Foreign Key referencing Departments(DeptID).

**Projects**: Stores project information.

* 1. ProjectID: INT, Primary Key, Auto Increment.
  2. ProjectName: VARCHAR(100), NOT NULL.
  3. StartDate: DATE, NOT NULL.
  4. EndDate: DATE.
  5. Budget: DECIMAL(15, 2), CHECK (Budget > 1000).
  6. DeptID: INT, Foreign Key referencing Departments(DeptID).

### Q3: ****Insert Data into Departments****

Insert three records into the Departments table:

1. DeptName: "HR", Location: "New York".
2. DeptName: "IT", Location: "San Francisco".
3. DeptName: "Finance", Location: "Chicago".
4. DeptName: "Marketing", Location: "Los Angeles".
5. DeptName: "Sales", Location: "New York"
6. DeptName: "R&D", Location: "San Francisco".

### Q4: ****Insert Data into Employees****

Insert the following employees:

1. John Doe, DOB: 1988-07-10, Email: john.doe@email.com, IsManager: TRUE, Salary: 80000, DeptID: 2 (IT).
2. Alice Brown, DOB: 1992-03-15, Email: alice.brown@email.com, IsManager: FALSE, Salary: 55000, DeptID: 1 (HR).
3. Sam Wilson, DOB: 1980-12-23, Email: sam.wilson@email.com, IsManager: FALSE, Salary: 40000, DeptID: 3 (Finance).
4. Michael Scott, DOB: 1964-03-15, Email: michael.scott@email.com, IsManager: TRUE, Salary: 95000, DeptID: 5 (Sales).
5. Pam Beesly, DOB: 1980-03-25, Email: pam.beesly@email.com, IsManager: FALSE, Salary: 45000, DeptID: 1 (HR).
6. Dwight Schrute, DOB: 1976-01-20, Email: dwight.schrute@email.com, IsManager: TRUE, Salary: 75000, DeptID: 5 (Sales).
7. Jim Halpert, DOB: 1978-10-01, Email: jim.halpert@email.com, IsManager: FALSE, Salary: 65000, DeptID: 5 (Sales).
8. Stanley Hudson, DOB: 1958-04-23, Email: stanley.hudson@email.com, IsManager: FALSE, Salary: 60000, DeptID: 5 (Sales).
9. Phyllis Vance, DOB: 1962-07-14, Email: phyllis.vance@email.com, IsManager: FALSE, Salary: 58000, DeptID: 5 (Sales).
10. Kelly Kapoor, DOB: 1982-11-13, Email: kelly.kapoor@email.com, IsManager: FALSE, Salary: 52000, DeptID: 4 (Marketing).
11. Ryan Howard, DOB: 1984-05-05, Email: ryan.howard@email.com, IsManager: FALSE, Salary: 50000, DeptID: 4 (Marketing).
12. Oscar Martinez, DOB: 1975-11-16, Email: oscar.martinez@email.com, IsManager: FALSE, Salary: 70000, DeptID: 3 (Finance).
13. Angela Martin, DOB: 1972-06-11, Email: angela.martin@email.com, IsManager: FALSE, Salary: 68000, DeptID: 1 (HR).
14. Creed Bratton, DOB: 1943-02-08, Email: creed.bratton@email.com, IsManager: FALSE, Salary: 40000, DeptID: 2 (IT).

### Q5: ****Insert Data into Projects****

Insert the following projects:

1. Project Alpha, StartDate: 2023-01-10, EndDate: 2023-06-30, Budget: 50000, DeptID: 2 (IT).
2. Project Beta, StartDate: 2023-03-15, EndDate: NULL, Budget: 20000, DeptID: 1 (HR).
3. Project Gamma, StartDate: 2022-01-15, EndDate: 2022-12-31, Budget: 75000, DeptID: 5 (Sales).
4. Project Delta, StartDate: 2023-05-01, EndDate: NULL, Budget: 45000, DeptID: 4 (Marketing).
5. Project Epsilon, StartDate: 2022-09-20, EndDate: 2023-03-15, Budget: 90000, DeptID: 3 (Finance).
6. Project Zeta, StartDate: 2023-01-01, EndDate: 2023-09-30, Budget: 30000, DeptID: 1 (HR).

### Q6: ****SELECT with WHERE and LIKE****

Retrieve the details of all employees whose first name starts with 'J' and who work in the IT department.

### Q7: ****ORDER BY Clause****

List all employees along with their department names, sorted by their salary in descending order.

### Q8: ****GROUP BY and HAVING Clause****

Retrieve the total budget allocated for each department but only include departments where the total budget exceeds 20,000.

### Q9: ****Aggregate Function****

Find department wise average salary of employees .

### Q10: ****Using LIMIT****

Retrieve the details of the top 2 highest-paid employees.

### Q11: ****Logical Operators and NULL Handling****

Retrieve all projects that either have no end date (EndDate is NULL) or have a budget greater than 30,000.

### Q12: ****IF Statement****

Display the first name of each employee and whether they are a manager or not, using an IF statement.

### Q13: ****CASE Statement****

Classify employees based on their salary:

* High Salary if more than 70,000.
* Medium Salary if between 50,000 and 70,000.
* Low Salary otherwise.

### Q14: ****Deleting Data****

Delete all employees who earn less than 40,000 from the Employees table.

### Q15: ****Update Data****

Update John Doe’s department to HR.

**ASSIGNMENT2**

1. Display the first name, last name, and department name for each employee
2. Retrieve the names of employees and the projects they are associated with, including the department name. Show all employees, even if they are not assigned to any project
3. Retrieve the total salary of all employees for each department. Use GROUP BY to group by department and include departments with no employees
4. Retrieve departments that have more than 3 employees, along with the number of employees in each department.
5. Retrieve the employees whose salary is greater than the average salary of their respective department.
6. Retrieve the total budget allocated to each department by summing the budgets of all projects in that department.
7. Display the first name, last name, and salary of the highest-paid employee in each department.
8. Retrieve the names of employees who are working in departments where there is a project with a budget greater than 50,000.
9. Display each department along with the count of projects associated with it. Use GROUP BY to group by department name.
10. Retrieve the names of departments that do not have any projects assigned. Hint: Use LEFT JOIN to include all departments.
11. Display the names of employees who are managers (IsManager = TRUE) along with their department names.
12. Retrieve the average budget of all projects grouped by department. Only show departments where the average budget exceeds 40,000.
13. Retrieve each department's total salary expense for its employees and the total project budget allocated to that department.

14. Display each project, its department name, and the total number of employees in that department.

**ASSIGNMENT3**

1. Display project names after removing the leading 'P' from those that start with the letter 'P' using SUBSTRING() and IF().
2. Retrieve the names of employees whose email address contains a specific domain, e.g., 'email.com', using the LIKE operator.
3. Display the first three characters of each department's name using the SUBSTRING() function.
4. Modify the project names by replacing the word 'Project' with 'Task' using the REPLACE() function.
5. Retrieve the names of employees whose first name has more than 5 characters using the LENGTH() function.
6. Display project names padded with spaces on the right side to ensure a total length of 20 characters using the RPAD() function.
7. Retrieve the names of employees whose last name contains the string 'son' using the LIKE operator.
8. Display the domain part of each employee's email address (the part after '@') using the SUBSTRING() and LOCATE() functions.
9. Replace the letter 'e' with the letter 'a' in each employee's last name using the REPLACE() function.
10. Display department names after removing any leading and trailing spaces using the TRIM() function.
11. Find the number of characters in each employee's first name using the LENGTH() function.
12. Display the full names of all employees by concatenating their first and last names using the CONCAT() function.

13. Show the last names of all employees in lowercase using the LOWER() function.

14. Display the first names of all employees in uppercase using the UPPER() function.