Lab Assignment 01:

Task 1: Create Tables

1. Create three tables: Employees, Departments, and Projects to track employees, departments, and projects, respectively.

o Ensure each table has a Primary Key for uniquely identifying records.

o Set up Foreign Key constraints to link employees to departments and projects.

o Use appropriate constraints (e.g., NOT NULL, UNIQUE, etc.) to maintain data integrity.

```
mysql> CREATE TABLE Departments (
    -> department_id INT PRIMARY KEY,
    ->  department_name VARCHAR(50) NOT NULL
    -> );

mysql>  CREATE TABLE Employees (
    ->  employee_id INT PRIMARY KEY,
    ->  first_name VARCHAR(50) NOT NULL,
    ->  last_name VARCHAR(50) NOT NULL,
    ->  email VARCHAR(100) UNIQUE,
    ->  hire_date DATETIME NOT NULL,
    ->  salary DECIMAL(10,2) NOT NULL,
    ->  department_id INT,
    ->  FOREIGN KEY (department_id) REFERENCES
Departments(department_id)
    ->  );

mysql> CREATE TABLE Projects (
    ->  project_id INT PRIMARY KEY,
    ->  project_name VARCHAR(100) NOT NULL,
    ->  start_date DATETIME NOT NULL,
    ->  end_date DATETIME,
    ->  department_id INT,
```

```
    -> FOREIGN KEY (department_id) REFERENCES
Departments(department_id)
    -> );
```

Task 2: Insert Data (Given in excel sheet)
Once you have created the tables, insert the provided data into the respective tables. The data contains details about employees, departments, and projects.

```
mysql>  INSERT INTO Departments (department_id, department_name)
VALUES
    -> (1, 'IT'),
    ->  (2, 'HR'),
    ->  (3, 'Sales'),
    ->  (4, 'Finance'),
    ->  (5, 'Marketing');
```

```
mysql> INSERT INTO Employees (employee_id, first_name, last_name,
email, hire_date, salary,
    -> department_id) VALUES
    -> (101, 'Ravi', 'Sharma', 'ravi.sharma@specialforce.com', '2017-05-15 ',
55000.00, 1),
    ->  (102, 'Neha', 'Kapoor', 'neha.kapoor@specialforce.com', '2019-03-23
', 48000.00, 2),
    ->  (103, 'Jyoti', 'Verma', 'jyoti.verma@specialforce.com', '2020-11-02 ',
60000.00, 1),
    ->  (104, 'Anil', 'Patil', 'anil.patil@specialforce.com', '2018-09-18 ',
70000.00, 3),
    ->  (105, 'Pooja', 'Singh', 'pooja.singh@specialforce.com', '2021-06-10',
40000.00, 4),
    ->  (106, 'Sanjay', 'Iyer', 'sanjay.iyer@specialforce.com', '2018-01-22 ',
75000.00, 3),
    ->  (107, 'Jatin', 'Reddy', 'jatin.reddy@specialforce.com', '2021-12-12 ',
85000.00, 2),
```

```
   ->  (108, 'Shreya', 'Mehta', 'shreya.mehta@specialforce.com',
'2022-04-19 ', 30000.00, 5),
   ->  (109, 'Rajesh', 'Gupta', 'rajesh.gupta@specialforce.com', '2020-08-11
', 90000.00, 1),
   ->  (110, 'Kavita', 'Nair', 'kavita.nair@specialforce.com', '2021-02-07 ',
50000.00, 2);
mysql>  INSERT INTO Projects (project_id, project_name, start_date,
end_date, department_id) VALUES
   -> (201, 'Project Phoenix', '2021-01-15 ', '2022-07-30 00:00:00', 1),
   ->  (202, 'Client Onboarding', '2020-06-20 ', NULL, 3),
   ->  (203, 'Financial Overhaul', '2019-03-10 ', '2021-12-15 00:00:00', 4),
   ->  (204, 'Marketing Revamp', '2022-03-01 ', NULL, 5),
   ->  (205, 'Internal System Audit', '2023-02-15 ', NULL, 2);

mysql>  SELECT * FROM Departments;

mysql> select * from projects;
```

Query 1: Write a query to retrieve the first name, last name, and department name of all employees. If an employee does not belong to any department, the department name should be NULL.

```
mysql> select employees.first_name, employees.last_name,
department_name from employees,departments where
employees.department_id = departments.department_id;
```

Query 2: Write a query to find all employees in the IT department who earn more than ₹50,000.

```
mysql> select * from employees where salary > 50000 and department_id
=(select department_id from departments where department_name='IT');
```

Query 3: Write a query to list the first name, last name, and email of all employees whose first name starts with 'J' and whose email contains specialforce.com.

```
mysql> select first_name, last_name, email from employees where
First_name like 'J%'and email like '%specialforce.com';
```

Query 4: Write a query to find all the distinct department names in the Departments table.

```
mysql> select distinct department_name from departments;
```

Query 5: Write a query to calculate the total salary expenditure of each department.

```
mysql> select department_name, sum(salary) from
departments,employees where employees.department_id=
departments.department_id group by department_name;
```

Query 6: Write a query to find the average salary of employees in the Finance department.

```
mysql> select avg(salary) from employees where department_id = (select
department_id from departments where department_name ='Finance');
```

Query 7: Write a query to find the minimum and maximum salaries of employees in the Sales department.

```
mysql> select max(salary) from employees where department_id = (select
department_id from departments where department_name = 'sales');
```

Query 8: Write a query to count the number of employees in each department.

```
mysql> select departments.department_name, count(*)
count_of_employees from employees, departments where
departments.department_id = employees.department_id group by
departments.department_name;
```

Query 9: Write a query to find all employees who were hired between January 1, 2018, and December 31, 2020. Sort the result by hire date in ascending order.

```
mysql> select * from employees where hire_date between '2018-01-01' and '2020-12-31' order by hire_date;
```

Query 10: Write a query to list all employees who do not have an email address.

```
mysql> select * from employees where email is null;
```

Query 11: Write a query to find all employees who work in HR, Finance, or IT departments.

```
mysql> select * from employees where department_id in (select department_id from departments where department_name in ('HR','Finance','IT'));
```

Query 12: Write a query to list the first name, last name, and salary of employees earning between ₹30,000 and ₹70,000. Sort the results by salary in descending order.

```
mysql> select first_name, Last_name, salary from employees where salary between 30000 and 70000;
```

Transaction Management Tasks:

Task 1: Increase HR Salaries:
Write a query to increase the salaries of all employees in the HR department by 5%. Start a transaction before applying the changes.

```
mysql> update employees set salary = salary*1.05 where department_id =
(select department_id from departments where department_name = 'HR');
```

Task 2: Savepoint Before Sales Increase:
Set a savepoint before increasing the salaries of employees in the Sales department by 3%.

```
mysql> start transaction;

mysql> savepoint before1;

mysql> update employees set salary = salary*1.03 where department_id =
(select department_id from departments where department_name =
'Sales');

mysql> select * from employees;

mysql> ROLLBACK TO SAVEPOINT before1;

mysql> select * from employees;
```

Task 3: Rollback Sales Salary Increase:
Rollback to the savepoint created before the Sales salary increase.

Already done above !!

Task 4: Commit the Transaction:
After rolling back the Sales increase, commit the changes made to the HR department salaries.

Already done above !!

Query 13: Write a query to join the Employees and Departments tables to list employees and their department names. Make sure all employees are included, even if they don’t belong to any department.

```
mysql> select first_name, last_name, department_name from
employees,departments where
    -> employees.department_id = departments.department_id;
```

Query 14: Write a query to list employees who are working on projects that started after January 1, 2023.

```
mysql> select * from employees where department_id = (select
department_id from projects where start_date > '2023-01-01');
```

Query 15: Write a query to list all departments, even those without any employees assigned.

```
mysql> select d.department_id, d.department_name, e.employee_id,
e.first_name from departments d left join employees e on d.department_id
= e.department_id;
```

Query 16: Write a query to find the employee with the highest salary in each department.

```
mysql> SELECT e.employee_id, e.first_name, e.salary, e.department_id
FROM employees e WHERE salary = (SELECT MAX(salary) FROM
employees WHERE department_id = e.department_id);
```

Query 17: Write a query to remove all data from the Employees table but keep the structure intact.

```
mysql> TRUNCATE TABLE Employees;
```

Query 18: Write a query to drop the Projects table from the database.

```
mysql>  drop table projects;
```

Query 19: SpecialForce Private Limited realized they need to store the phone numbers of employees. Write a query to add a new column phone_number (VARCHAR(15)) to the Employees table using the ALTER statement.

mysql>alter table employees add column Phone_number varchar(15);

Query 20: The company also decided to track the budget for each project. Write a query to add a column budget (DECIMAL(10,2)) to the Projects table.

mysql> alter table projects add column budget DECIMAL(10,2) ;

Query 21: Write a query to find the 2nd largest salary from the Employees table using: A subquery. The LIMIT clause.

mysql> select max(salary) from employees where salary <(select max(salary) from employees);

mysql> select salary from employee order by salary desc limit 1 offset 1;

Query 22: Write a query to find the 3rd largest salary from the Employees table using: A subquery. The LIMIT clause.

mysql> select max(salary) from employee where salary <(select max(salary) from employee where salary < (select Max(salary) from employee));

mysql> select salary from employee order by salary desc limit 1 offset 2;

Query 23: Write a query to drop the Projects table.

```
mysql>  drop table projects;
```

Query 24: Write a query to truncate the Employees table.

```
mysql>  TRUNCATE TABLE Employees;
```