

# DBMS Lab Work - SQL Queries and Implementation

---

## 1. Database and Tables

```
CREATE DATABASE company;  
USE company;
```

```
CREATE TABLE DEPARTMENT (  
    Department_ID INT PRIMARY KEY,  
    Name VARCHAR(50),  
    Location_ID VARCHAR(50)  
);
```

```
CREATE TABLE JOB (  
    Job_ID INT PRIMARY KEY,  
    Function VARCHAR(50)  
);
```

```
CREATE TABLE EMPLOYEE (  
    Employee_ID INT PRIMARY KEY,  
    Name VARCHAR(50),  
    DOB DATE,  
    Job_ID INT,  
    Manager_ID INT,  
    Hire_Date DATE,  
    Salary DECIMAL(10,2),  
    Department_ID INT,  
    FOREIGN KEY (Job_ID) REFERENCES JOB(Job_ID),  
    FOREIGN KEY (Department_ID) REFERENCES DEPARTMENT(Department_ID)  
);
```

Table	Action	Rows	Type	Collation	Size	Overhead
<input type="checkbox"/> department	Browse  Structure  Search  Insert  Empty  Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
<input type="checkbox"/> employee	Browse  Structure  Search  Insert  Empty  Drop	0	InnoDB	utf8mb4_general_ci	48.0 KiB	-
<input type="checkbox"/> job	Browse  Structure  Search  Insert  Empty  Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
3 tables	Sum	0	InnoDB	utf8mb4_general_ci	80.0 KiB	0 B

## 2. Insert Dummy Data

INSERT INTO DEPARTMENT VALUES

```
(1,'HR','Delhi'),  
(2,'Finance','Mumbai'),  
(3,'IT','Delhi'),  
(4,'Sales','Bangalore');
```

INSERT INTO JOB VALUES

```
(1,'Manager'),  
(2,'Analyst'),  
(3,'Developer'),  
(4,'Clerk');
```

INSERT INTO EMPLOYEE VALUES

```
(101,'Amit','1990-02-10',1,NULL,'2015-03-05',60000,1),  
(102,'Priya','1988-06-20',2,101,'2015-03-15',45000,2),  
(103,'Ravi','1992-01-12',3,101,'2016-05-11',40000,3),  
(104,'Neha','1993-08-21',4,102,'2015-03-25',30000,1),  
(105,'Karan','1987-11-30',2,101,'2014-07-14',50000,4);
```

Department_ID	Name	Location_ID
1	HR	Delhi
2	Finance	Mumbai
3	IT	Delhi
4	Sales	Bangalore

Employee_ID	Name	DOB	Job_ID	Manager_ID	Hire_Date	Salary	Department_ID
101	Amit	1990-02-10	1	<i>NULL</i>	2015-03-05	60000.00	1
102	Priya	1988-06-20	2	101	2015-03-15	45000.00	2
103	Ravi	1992-01-12	3	101	2016-05-11	40000.00	3
104	Neha	1993-08-21	4	102	2015-03-25	30000.00	1
105	Karan	1987-11-30	2	101	2014-07-14	50000.00	4

Job_ID	Function
1	Manager
2	Analyst
3	Developer
4	Clerk

### 3. Queries

#### a) Count employees who joined in March 2015

```
SELECT COUNT(*) AS NumEmployees
FROM EMPLOYEE
WHERE YEAR(Hire_Date) = 2015
AND MONTH(Hire_Date) = 3;
```

**NumEmployees**

3

### b) Display Nth highest salary employee

```
SELECT *
FROM EMPLOYEE e1
WHERE N-1 = (
    SELECT COUNT(DISTINCT e2.Salary)
    FROM EMPLOYEE e2
    WHERE e2.Salary > e1.Salary
);
-- Replace N with 2 for 2nd highest, 3 for 3rd highest, etc.
```

Employee_ID	Name	DOB	Job_ID	Manager_ID	Hire_Date	Salary	Department_ID
105	Karan	1987-11-30	2	101	2014-07-14	50000.00	4

### c) Find the budget (total salary) of each department

```
SELECT d.Name AS Department, SUM(e.Salary) AS Budget
FROM EMPLOYEE e
JOIN DEPARTMENT d ON e.Department_ID = d.Department_ID
GROUP BY d.Name;
```

Department	Budget
Finance	45000.00
HR	90000.00
IT	40000.00
Sales	50000.00

### d) Find the department with maximum budget

```
SELECT d.Name AS Department, SUM(e.Salary) AS Budget
FROM EMPLOYEE e
JOIN DEPARTMENT d ON e.Department_ID = d.Department_ID
```

```
GROUP BY d.Name
ORDER BY Budget DESC
LIMIT 1;
```

Department	Budget
HR	90000.00

#### e) Create a view to show employees in Delhi

```
CREATE VIEW Delhi_Employees AS
SELECT d.Location_ID, COUNT(e.Employee_ID) AS NumEmployees
FROM EMPLOYEE e
JOIN DEPARTMENT d ON e.Department_ID = d.Department_ID
WHERE d.Location_ID = 'Delhi'
GROUP BY d.Location_ID;
```

```
-- Check the view
SELECT * FROM Delhi_Employees;
```

Location_ID	NumEmployees
Delhi	3

#### f) Trigger to ensure no employee under 25 is inserted

```
DELIMITER //
CREATE TRIGGER check_age_before_insert
BEFORE INSERT ON EMPLOYEE
FOR EACH ROW
BEGIN
    IF TIMESTAMPDIFF(YEAR, NEW.DOB, CURDATE()) < 25 THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Employee must be at least 25 years old.';
    END IF;
END;
```

```
//  
DELIMITER ;
```