

# Experiment – 10

**Title: Create the following views in SQL on the COMPANY database schema presented in Experiment 2.**

```
CREATE TABLE DEPARTMENT (
```

```
    Dept_ID INT PRIMARY KEY,
```

```
    Dept_Name VARCHAR(50) UNIQUE,
```

```
    Mgr_ID INT
```

```
);
```

```
CREATE TABLE EMPLOYEE (
```

```
    Emp_ID INT PRIMARY KEY,
```

```
    Emp_Name VARCHAR(50),
```

```
    Salary DECIMAL(10, 2),
```

```
    Supervisor_ID INT,
```

```
    Dept_ID INT,
```

```
    FOREIGN KEY (Dept_ID) REFERENCES DEPARTMENT(Dept_ID)
```

```
);
```

```
CREATE TABLE PROJECT (
```

```
    Project_ID INT PRIMARY KEY,
```

```
    Project_Name VARCHAR(50),
```

```
    Controlling_Dept_ID INT,
```

```
    FOREIGN KEY (Controlling_Dept_ID) REFERENCES DEPARTMENT(Dept_ID)
```

```
);
```

```
CREATE TABLE WORKS_ON (  
    Emp_ID INT,  
    Project_ID INT,  
    Hours DECIMAL(5, 2),  
    FOREIGN KEY (Emp_ID) REFERENCES EMPLOYEE(Emp_ID),  
    FOREIGN KEY (Project_ID) REFERENCES PROJECT(Project_ID)  
);  
  
INSERT INTO DEPARTMENT (Dept_ID, Dept_Name, Mgr_ID) VALUES  
(1, 'Human Resources', 101),  
(2, 'Finance', 102),  
(3, 'Engineering', 103),  
(4, 'Marketing', 104);
```

```
INSERT INTO EMPLOYEE (Emp_ID, Emp_Name, Salary, Supervisor_ID, Dept_ID) VALUES  
(101, 'Alice Johnson', 80000, NULL, 1),  
(102, 'Bob Smith', 95000, NULL, 2),  
(103, 'Carol White', 120000, NULL, 3),  
(104, 'David Brown', 85000, NULL, 4),  
(105, 'Eva Green', 75000, 101, 1),  
(106, 'Frank Black', 70000, 101, 1),  
(107, 'Grace King', 90000, 102, 2),  
(108, 'Henry Miller', 95000, 103, 3),  
(109, 'Ivy Wilson', 99000, 104, 4),  
(110, 'Jake Taylor', 85000, 103, 3);
```

```
INSERT INTO PROJECT (Project_ID, Project_Name, Controlling_Dept_ID) VALUES
```

```
(201, 'Project Alpha', 3),
```

```
(202, 'Project Beta', 2),
```

```
(203, 'Project Gamma', 4),
```

```
(204, 'Project Delta', 1);
```

```
INSERT INTO WORKS_ON (Emp_ID, Project_ID, Hours) VALUES
```

```
(105, 201, 35.5),
```

```
(106, 201, 20.0),
```

```
(107, 202, 40.0),
```

```
(108, 203, 45.0),
```

```
(109, 204, 38.0),
```

```
(110, 201, 25.5),
```

```
(101, 204, 10.0),
```

```
(102, 202, 12.0),
```

```
(103, 201, 15.5),
```

```
(104, 203, 30.0);
```

```
mysql> select * from department;
```

Dept_ID	Dept_Name	Mgr_ID
1	Human Resources	101
2	Finance	102
3	Engineering	103
4	Marketing	104

```
4 rows in set (0.00 sec)
```

```
mysql> select * from employee;
```

Emp_ID	Emp_Name	Salary	Supervisor_ID	Dept_ID
101	Alice Johnson	80000.00	NULL	1
102	Bob Smith	95000.00	NULL	2
103	Carol White	120000.00	NULL	3
104	David Brown	85000.00	NULL	4
105	Eva Green	75000.00	101	1
106	Frank Black	70000.00	101	1
107	Grace King	90000.00	102	2
108	Henry Miller	95000.00	103	3
109	Ivy Wilson	99000.00	104	4
110	Jake Taylor	85000.00	103	3

```
10 rows in set (0.00 sec)
```

```
mysql> select * from project;
```

Project_ID	Project_Name	Controlling_Dept_ID
201	Project Alpha	3
202	Project Beta	2
203	Project Gamma	4
204	Project Delta	1

```
4 rows in set (0.00 sec)
```

```
mysql> select * from works_on;
```

Emp_ID	Project_ID	Hours
105	201	35.50
106	201	20.00
107	202	40.00
108	203	45.00
109	204	38.00
110	201	25.50
101	204	10.00
102	202	12.00
103	201	15.50
104	203	30.00

```
10 rows in set (0.00 sec)
```

1. A view that has the department name, manager name, and manager salary for every department.

```
CREATE VIEW Research_Emp_Supervisor_View AS
SELECT
    E1.Emp_Name AS Employee_Name,
    E2.Emp_Name AS Supervisor_Name,
    E1.Salary AS Employee_Salary
FROM
    EMPLOYEE E1
JOIN
    EMPLOYEE E2 ON E1.Supervisor_ID = E2.Emp_ID
JOIN
    DEPARTMENT D ON E1.Dept_ID = D.Dept_ID
```

WHERE

D.Dept\_Name = 'Research';

```
mysql> select * from Dept_Manager_View;
+-----+-----+-----+
| Department_Name | Manager_Name | Manager_Salary |
+-----+-----+-----+
| Human Resources | Alice Johnson | 80000.00 |
| Finance         | Bob Smith    | 95000.00 |
| Engineering     | Carol White  | 120000.00 |
| Marketing       | David Brown  | 85000.00 |
+-----+-----+-----+
4 rows in set (0.00 sec)
```

2. A view that has the employee name, supervisor name, and employee salary for each employee who works in the 'Research' department.

CREATE VIEW Research\_Emp\_Supervisor\_View AS

SELECT

E1.Emp\_Name AS Employee\_Name,

E2.Emp\_Name AS Supervisor\_Name,

E1.Salary AS Employee\_Salary

FROM

EMPLOYEE E1

JOIN

EMPLOYEE E2 ON E1.Supervisor\_ID = E2.Emp\_ID

JOIN

DEPARTMENT D ON E1.Dept\_ID = D.Dept\_ID

WHERE

```
D.Dept_Name = 'Research';
```

```
mysql> select * from Research_Emp_Supervisor_View;  
Empty set (0.00 sec)
```

3. A view that has the project name, controlling department name, number of employees, and total hours worked per week on the project for each project.

```
CREATE VIEW Project_Work_View AS
```

```
SELECT
```

```
    P.Project_Name,
```

```
    D.Dept_Name AS Controlling_Dept_Name,
```

```
    COUNT(W.Emp_ID) AS Number_of_Employees,
```

```
    SUM(W.Hours) AS Total_Hours_Worked
```

```
FROM
```

```
    PROJECT P
```

```
JOIN
```

```
    DEPARTMENT D ON P.Controlling_Dept_ID = D.Dept_ID
```

```
JOIN
```

```
    WORKS_ON W ON P.Project_ID = W.Project_ID
```

```
GROUP BY
```

```
    P.Project_ID;
```

```
mysql> select * from Project_Work_View;  
+-----+-----+-----+-----+  
| Project_Name | Controlling_Dept_Name | Number_of_Employees | Total_Hours_Worked |  
+-----+-----+-----+-----+  
| Project Alpha | Engineering           | 4 | 96.50 |  
| Project Beta  | Finance               | 2 | 52.00 |  
| Project Gamma | Marketing              | 2 | 75.00 |  
| Project Delta | Human Resources        | 2 | 48.00 |  
+-----+-----+-----+-----+  
4 rows in set (0.00 sec)
```

4. A view that has the project name, controlling department name, number of employees, and total hours worked per week on the project for each project

with more than one employee working on it.

```
CREATE VIEW Project_Work_MultiEmp_View AS
```

```
SELECT
```

```
    P.Project_Name,
```

```
    D.Dept_Name AS Controlling_Dept_Name,
```

```
    COUNT(W.Emp_ID) AS Number_of_Employees,
```

```
    SUM(W.Hours) AS Total_Hours_Worked
```

```
FROM
```

```
    PROJECT P
```

```
JOIN
```

```
    DEPARTMENT D ON P.Controlling_Dept_ID = D.Dept_ID
```

```
JOIN
```

```
    WORKS_ON W ON P.Project_ID = W.Project_ID
```

```
GROUP BY
```

```
    P.Project_ID
```

```
HAVING
```

```
    COUNT(W.Emp_ID) > 1;
```

```
mysql> select * from Project_Work_MultiEmp_View;
```

Project_Name	Controlling_Dept_Name	Number_of_Employees	Total_Hours_Worked
Project Alpha	Engineering	4	96.50
Project Beta	Finance	2	52.00
Project Gamma	Marketing	2	75.00
Project Delta	Human Resources	2	48.00

```
4 rows in set (0.01 sec)
```



# Experiment – 11

```
create database exp11;
```

```
use exp11;
```

```
create table employees (
```

```
    employee_id char(10) primary key,
```

```
    first_name char(30) not null,
```

```
    last_name char(30) not null,
```

```
    dob date,
```

```
    salary decimal(25,2) not null,
```

```
    department_id char(10)
```

```
);
```

```
insert into employees (employee_id, first_name, last_name, dob, salary, department_id)
values
```

```
('E001', 'John', 'Doe', '1985-06-15', 55000.00, 'D001'),
```

```
('E002', 'Jane', 'Smith', '1990-04-20', 62000.00, 'D002'),
```

```
('E003', 'Robert', 'Johnson', '1982-12-05', 75000.00, 'D003'),
```

```
('E004', 'Emily', 'Davis', '1995-11-11', 50000.00, 'D001'),
```

```
('E005', 'Michael', 'Brown', '1988-09-22', 67000.00, 'D004'),
```

```
('E006', 'Sarah', 'Miller', '1992-02-17', 72000.00, 'D003');
```

```
mysql> select * from employees;
```

employee_id	first_name	last_name	dob	salary	department_id
E001	John	Doe	1985-06-15	55000.00	D001
E002	Jane	Smith	1990-04-20	62000.00	D002
E003	Robert	Johnson	1982-12-05	75000.00	D003
E004	Emily	Davis	1995-11-11	50000.00	D001
E005	Michael	Brown	1988-09-22	67000.00	D004
E006	Sarah	Miller	1992-02-17	72000.00	D003

```
6 rows in set (0.00 sec)
```

1. Execute the following index related queries:

1. Create an index of name employee\_idx on EMPLOYEES with column Last\_Name, Department\_id

```
create index employee_idx on employees (last_name, department_id);
```

2. Find the ROWID for the above table and create a unique index on employee\_id column of the EMPLOYEES.

```
create unique index employee_id_unique_idx on employees (employee_id);
```

3. Create a reverse index on employee\_id column of the EMPLOYEES.

```
create index employee_id_reverse_idx on employees (employee_id);
```

4. Create a unique and composite index on employee\_id and check whether there is duplicity of tuples or not.

```
create unique index employee_id_composite_unique_idx on employees (employee_id);
```

5. Create Function-based indexes defined on the SQL functions

UPPER(column\_name) or LOWER(column\_name) to facilitate caseinsensitive

searches(on column Last\_Name).

```
create index last_name_upper_idx on employees (upper(last_name));
```

6. Drop the function based index on column Last\_Name.

```
drop index last_name_upper_idx on employees;
```