NAME-LEPAKSHI JAGMOHAN

BATCH-2

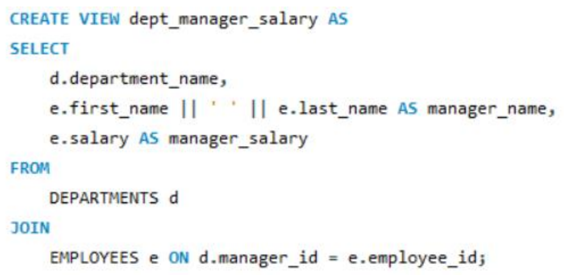
SAP-500119039

DATABASE MANAGEMENT SYSTEMS LAB

EXPERIMENT-10

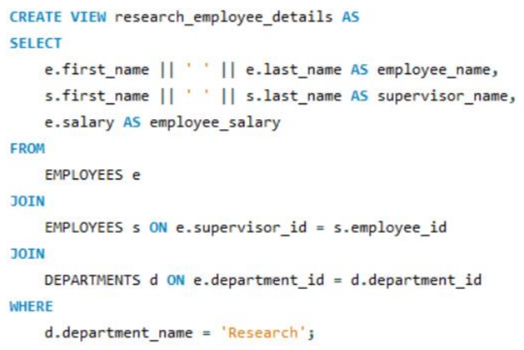
Title: Create the following views in SQL on the COMPANY database schema presented in Experiment 2.

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



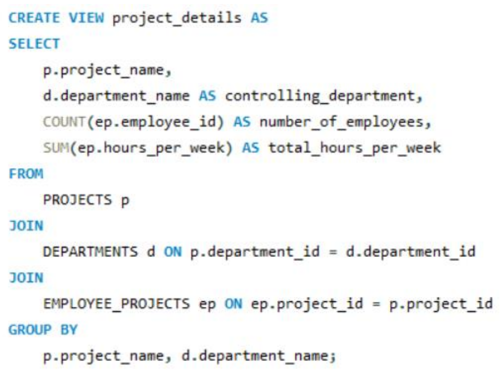
```sql
CREATE VIEW dept_manager_salary AS
SELECT
    d.department_name,
    e.first_name || ' ' || e.last_name AS manager_name,
    e.salary AS manager_salary
FROM
    DEPARTMENTS d
JOIN
    EMPLOYEES e ON d.manager_id = e.employee_id;
```

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



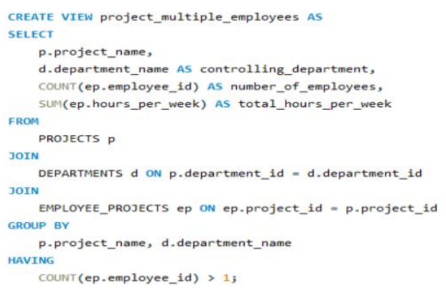
```sql
CREATE VIEW research_employee_details AS
SELECT
    e.first_name || ' ' || e.last_name AS employee_name,
    s.first_name || ' ' || s.last_name AS supervisor_name,
    e.salary AS employee_salary
FROM
    EMPLOYEES e
JOIN
    EMPLOYEES s ON e.supervisor_id = s.employee_id
JOIN
    DEPARTMENTS d ON e.department_id = d.department_id
WHERE
    d.department_name = 'Research';
```

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.



```sql
CREATE VIEW project_details AS
SELECT
    p.project_name,
    d.department_name AS controlling_department,
    COUNT(ep.employee_id) AS number_of_employees,
    SUM(ep.hours_per_week) AS total_hours_per_week
FROM
    PROJECTS p
JOIN
    DEPARTMENTS d ON p.department_id = d.department_id
JOIN
    EMPLOYEE_PROJECTS ep ON ep.project_id = p.project_id
GROUP BY
    p.project_name, d.department_name;
```

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.



```sql
CREATE VIEW project_multiple_employees AS
SELECT
    p.project_name,
    d.department_name AS controlling_department,
    COUNT(ep.employee_id) AS number_of_employees,
    SUM(ep.hours_per_week) AS total_hours_per_week
FROM
    PROJECTS p
JOIN
    DEPARTMENTS d ON p.department_id = d.department_id
JOIN
    EMPLOYEE_PROJECTS ep ON ep.project_id = p.project_id
GROUP BY
    p.project_name, d.department_name
HAVING
    COUNT(ep.employee_id) > 1;
```

NAME-LEPAKSHI JAGMOHAN

BATCH-2

SAP-500119039

DATABASE MANAGEMENT SYSTEMS LAB

EXPERIMENT-11

Title: To understand the concepts of Index.

Objective: Students will be able to implement the concept of index.

Create table of table name: EMPLOYEES and add 6 rows

Column Name	Data Type	Width	Attributes
Employee_id	Character	10	PK
First_Name	Character	30	NN
Last_Name	Character	30	NN
DOB	Date		
Salary	Number	25	NN
Department_id	Character	10	

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_idx
ON EMPLOYEES (employee_id);
```

3. Create a reverse index on employee_id column of the EMPLOYEES.

```
ALTER TABLE EMPLOYEES
ADD COLUMN employee_id_reverse CHAR(10) AS (REVERSE(employee_id)) STORED;
CREATE INDEX employee_reverse_idx
ON EMPLOYEES (employee_id_reverse);
```

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

```
CREATE UNIQUE INDEX employee_composite_idx
ON EMPLOYEES (employee_id, department_id);
```

5. Create Function-based indexes defined on the SQL functions UPPER(column_name) or LOWER(column_name) to facilitate case-insensitive searches(on column Last_Name).

6. Drop the function based index on column Last_Name.

```
DROP INDEX last_name_upper_idx ON EMPLOYEES;
```