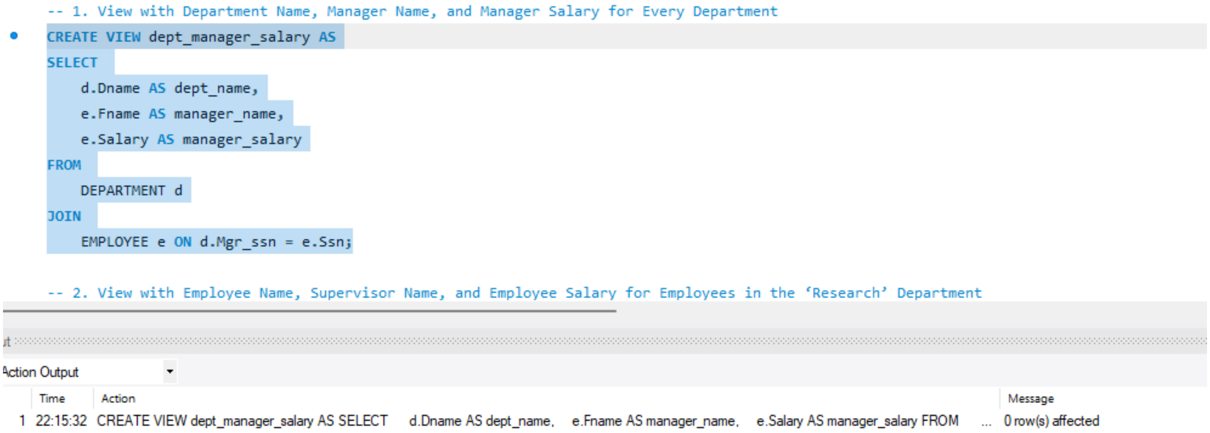
EXPERIMENT - 10

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.

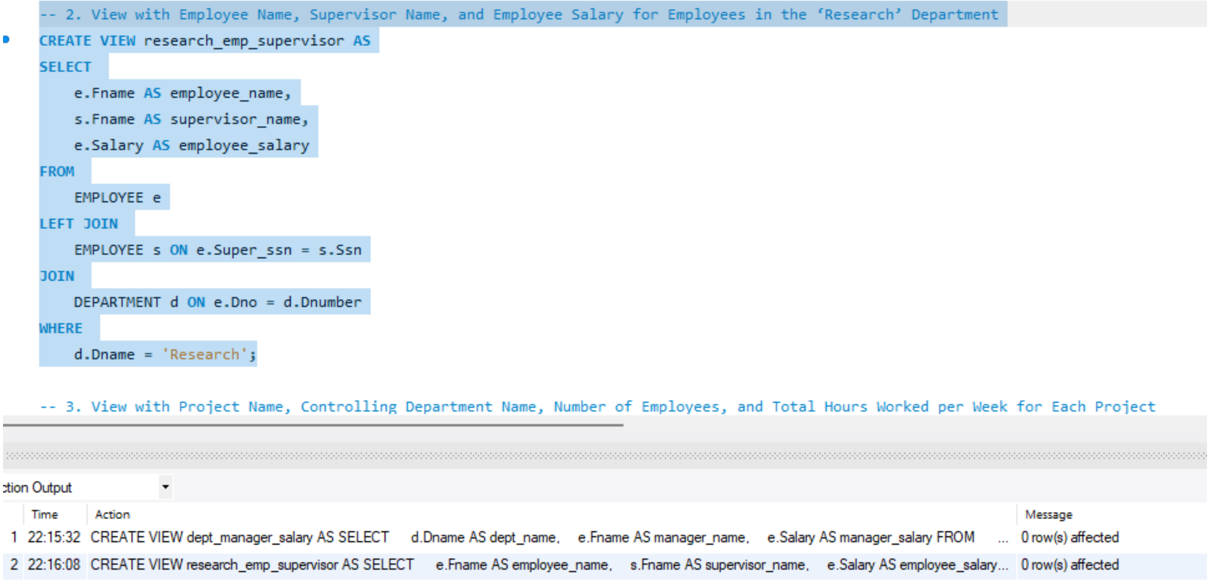


```
-- 1. View with Department Name, Manager Name, and Manager Salary for Every Department
CREATE VIEW dept_manager_salary AS
SELECT
    d.Dname AS dept_name,
    e.Fname AS manager_name,
    e.Salary AS manager_salary
FROM
    DEPARTMENT d
JOIN
    EMPLOYEE e ON d.Mgr_ssn = e.Ssn;

-- 2. View with Employee Name, Supervisor Name, and Employee Salary for Employees in the 'Research' Department
```

```
   Time      Action                                                                                                       Message
1  22:15:32  CREATE VIEW dept_manager_salary AS SELECT   d.Dname AS dept_name,   e.Fname AS manager_name,   e.Salary AS manager_salary FROM   ...  0 row(s) affected
```

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

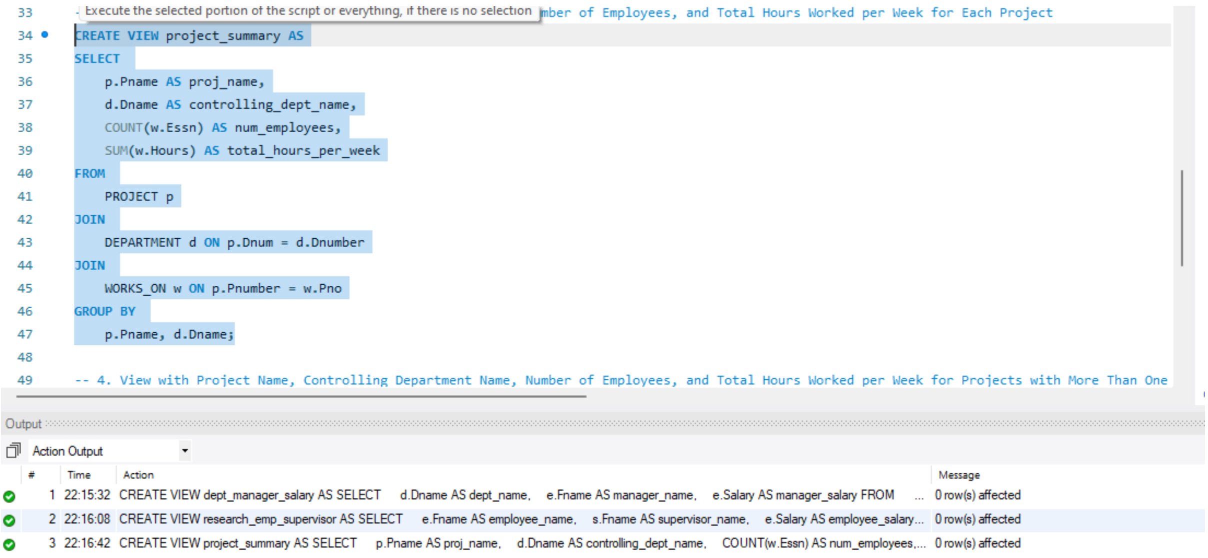


```
-- 2. View with Employee Name, Supervisor Name, and Employee Salary for Employees in the 'Research' Department
CREATE VIEW research_emp_supervisor AS
SELECT
    e.Fname AS employee_name,
    s.Fname AS supervisor_name,
    e.Salary AS employee_salary
FROM
    EMPLOYEE e
LEFT JOIN
    EMPLOYEE s ON e.Super_ssn = s.Ssn
JOIN
    DEPARTMENT d ON e.Dno = d.Dnumber
WHERE
    d.Dname = 'Research';

-- 3. View with Project Name, Controlling Department Name, Number of Employees, and Total Hours Worked per Week for Each Project
```

```
   Time      Action                                                                                                          Message
1  22:15:32  CREATE VIEW dept_manager_salary AS SELECT   d.Dname AS dept_name,   e.Fname AS manager_name,   e.Salary AS manager_salary FROM   ...  0 row(s) affected
2  22:16:08  CREATE VIEW research_emp_supervisor AS SELECT   e.Fname AS employee_name,   s.Fname AS supervisor_name,   e.Salary AS employee_salary...  0 row(s) affected
```

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.

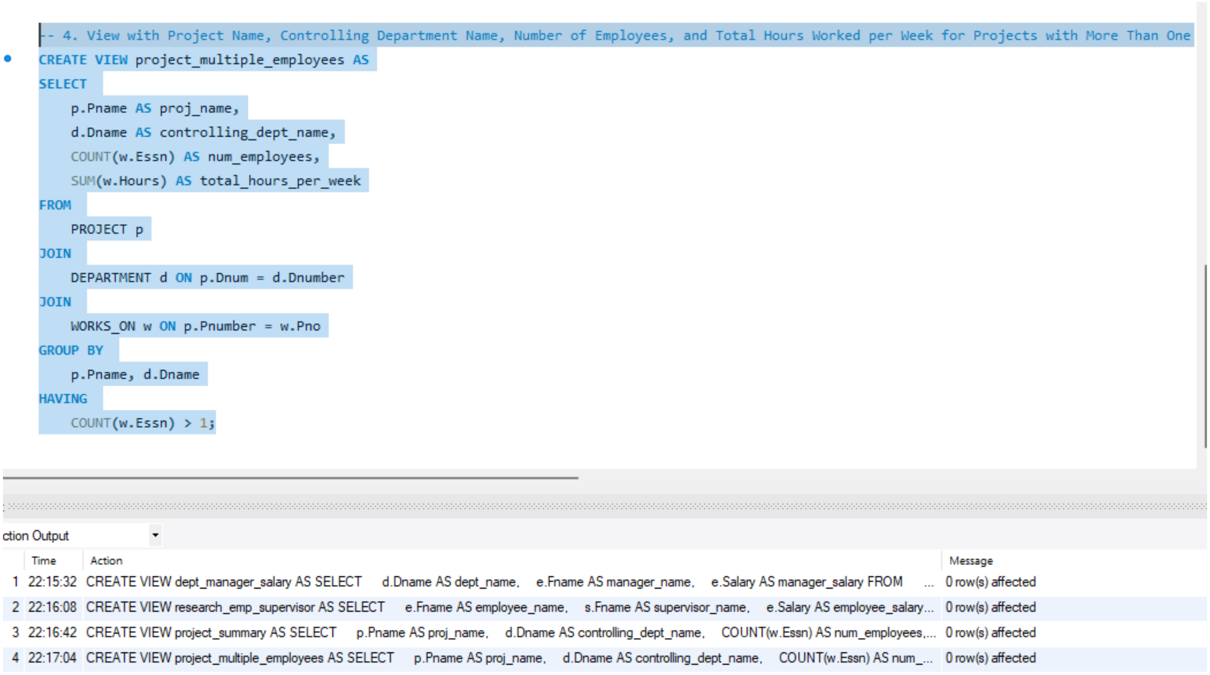

Execute the selected portion of the script or everything, if there is no selection
Output
Action Output

```
-                                                         mber of Employees, and Total Hours Worked per Week for Each Project
CREATE VIEW project_summary AS
SELECT
    p.Pname AS proj_name,
    d.Dname AS controlling_dept_name,
    COUNT(w.Essn) AS num_employees,
    SUM(w.Hours) AS total_hours_per_week
FROM
    PROJECT p
JOIN
    DEPARTMENT d ON p.Dnum = d.Dnumber
JOIN
    WORKS_ON w ON p.Pnumber = w.Pno
GROUP BY
    p.Pname, d.Dname;

-- 4. View with Project Name, Controlling Department Name, Number of Employees, and Total Hours Worked per Week for Projects with More Than One
```

```
#   Time      Action                                                                                                          Message
1   22:15:32  CREATE VIEW dept_manager_salary AS SELECT   d.Dname AS dept_name,   e.Fname AS manager_name,   e.Salary AS manager_salary FROM   ...  0 row(s) affected
2   22:16:08  CREATE VIEW research_emp_supervisor AS SELECT   e.Fname AS employee_name,   s.Fname AS supervisor_name,   e.Salary AS employee_salary...  0 row(s) affected
3   22:16:42  CREATE VIEW project_summary AS SELECT   p.Pname AS proj_name,   d.Dname AS controlling_dept_name,   COUNT(w.Essn) AS num_employees,...  0 row(s) affected
```

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.



```
-- 4. View with Project Name, Controlling Department Name, Number of Employees, and Total Hours Worked per Week for Projects with More Than One
CREATE VIEW project_multiple_employees AS
SELECT
    p.Pname AS proj_name,
    d.Dname AS controlling_dept_name,
    COUNT(w.Essn) AS num_employees,
    SUM(w.Hours) AS total_hours_per_week
FROM
    PROJECT p
JOIN
    DEPARTMENT d ON p.Dnum = d.Dnumber
JOIN
    WORKS_ON w ON p.Pnumber = w.Pno
GROUP BY
    p.Pname, d.Dname
HAVING
    COUNT(w.Essn) > 1;
```

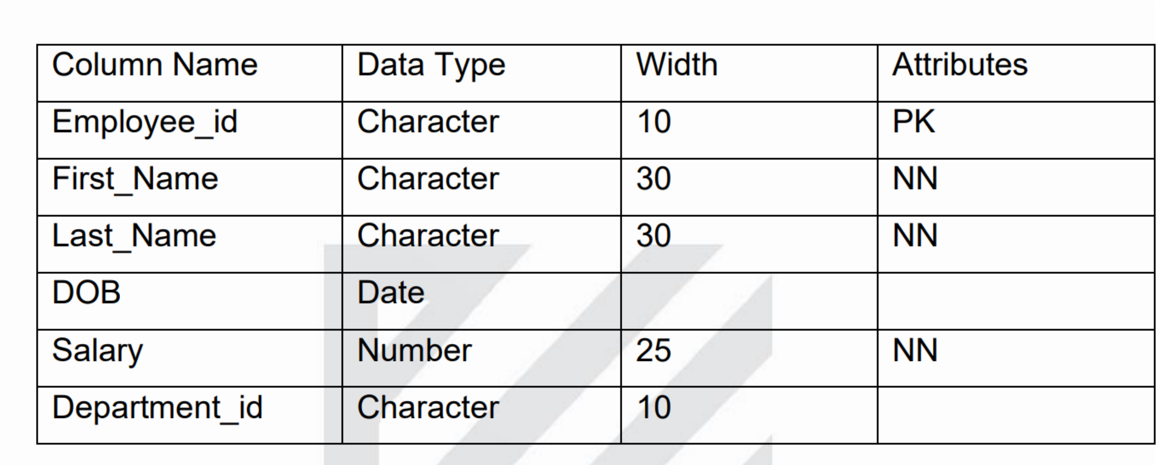
```
   Time      Action                                                                                                          Message
1  22:15:32  CREATE VIEW dept_manager_salary AS SELECT   d.Dname AS dept_name,   e.Fname AS manager_name,   e.Salary AS manager_salary FROM   ...  0 row(s) affected
2  22:16:08  CREATE VIEW research_emp_supervisor AS SELECT   e.Fname AS employee_name,   s.Fname AS supervisor_name,   e.Salary AS employee_salary...  0 row(s) affected
3  22:16:42  CREATE VIEW project_summary AS SELECT   p.Pname AS proj_name,   d.Dname AS controlling_dept_name,   COUNT(w.Essn) AS num_employees,...  0 row(s) affected
4  22:17:04  CREATE VIEW project_multiple_employees AS SELECT   p.Pname AS proj_name,   d.Dname AS controlling_dept_name,   COUNT(w.Essn) AS num_...  0 row(s) affected
```

EXPERIMENT - 11

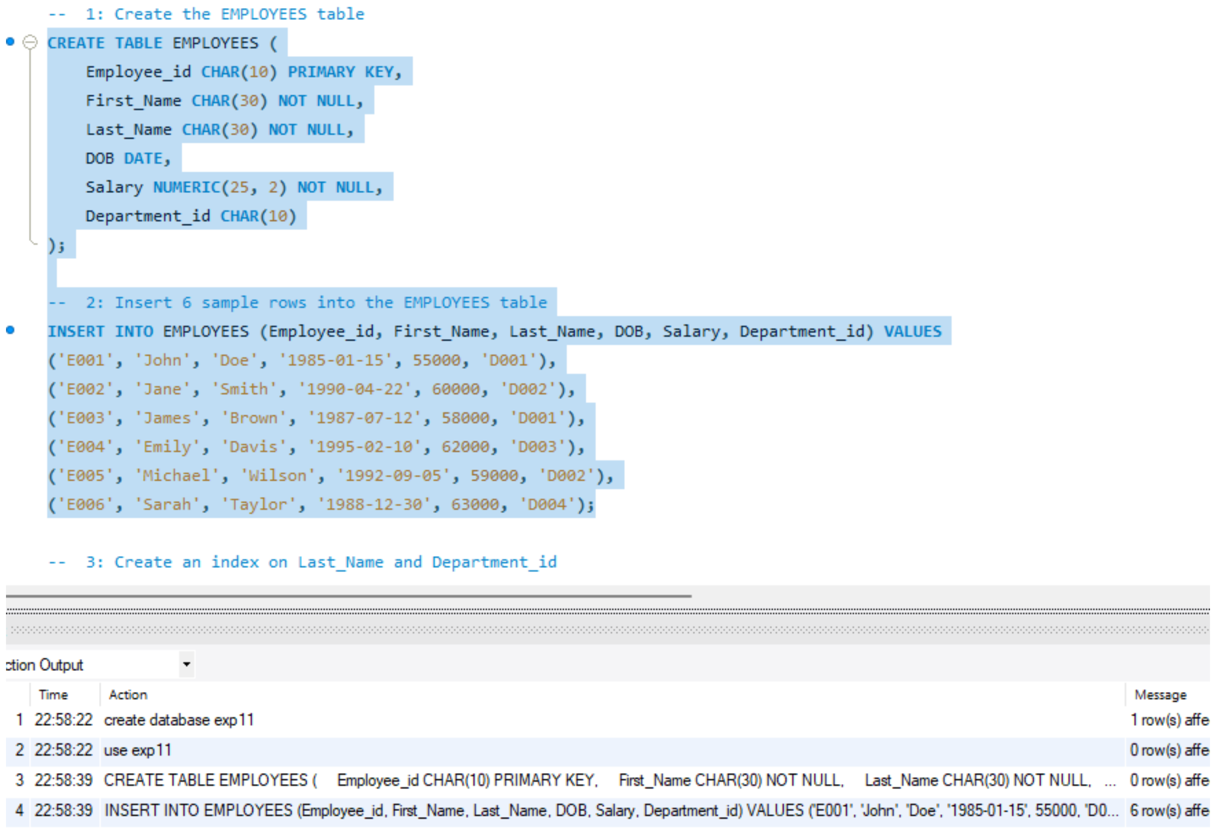
Title: To understand the concepts of Index.

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	



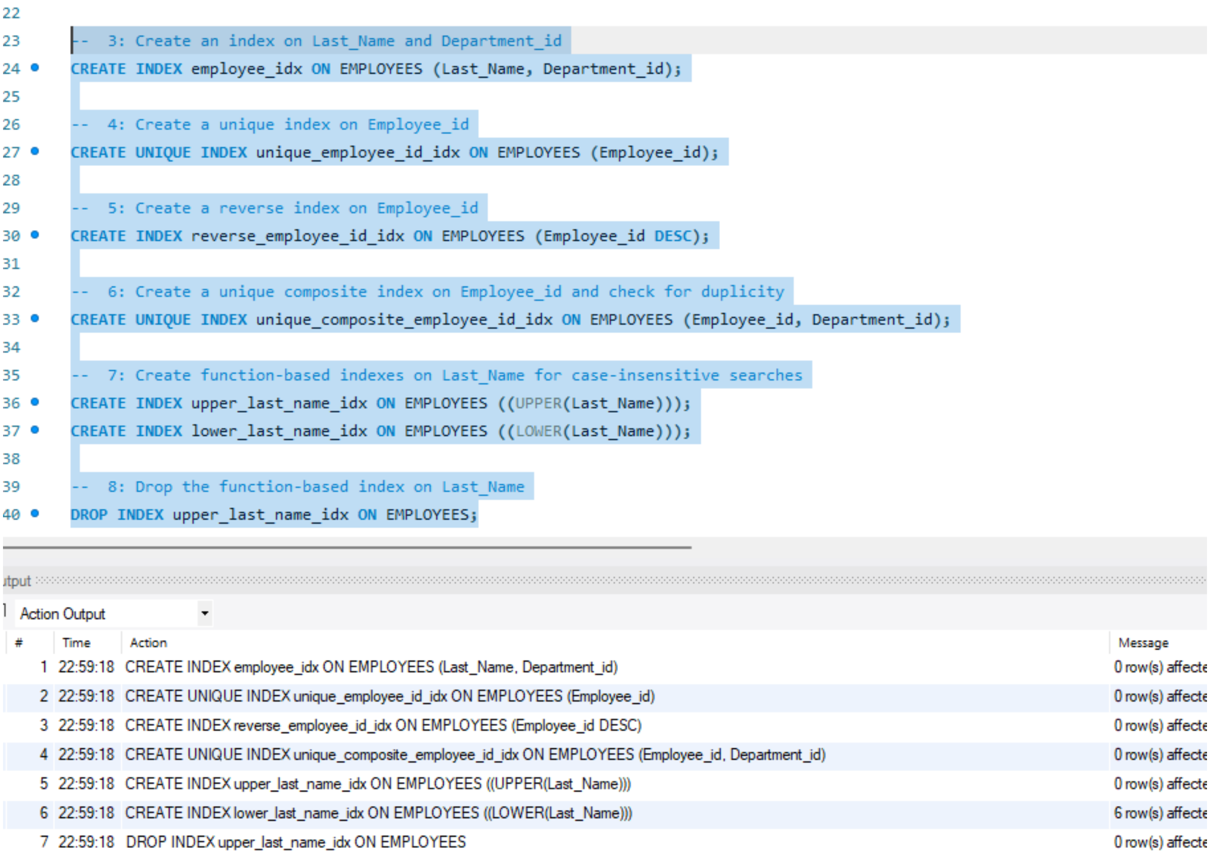
```
-- 1: Create the EMPLOYEES table
CREATE TABLE EMPLOYEES (
    Employee_id CHAR(10) PRIMARY KEY,
    First_Name CHAR(30) NOT NULL,
    Last_Name CHAR(30) NOT NULL,
    DOB DATE,
    Salary NUMERIC(25, 2) NOT NULL,
    Department_id CHAR(10)
);

-- 2: Insert 6 sample rows into the EMPLOYEES table
INSERT INTO EMPLOYEES (Employee_id, First_Name, Last_Name, DOB, Salary, Department_id) VALUES
('E001', 'John', 'Doe', '1985-01-15', 55000, 'D001'),
('E002', 'Jane', 'Smith', '1990-04-22', 60000, 'D002'),
('E003', 'James', 'Brown', '1987-07-12', 58000, 'D001'),
('E004', 'Emily', 'Davis', '1995-02-10', 62000, 'D003'),
('E005', 'Michael', 'Wilson', '1992-09-05', 59000, 'D002'),
('E006', 'Sarah', 'Taylor', '1988-12-30', 63000, 'D004');

-- 3: Create an index on Last_Name and Department_id
```

```
   Time      Action                                                                                                                          Message
1  22:58:22  create database exp11                                                                                                           1 row(s) affe
2  22:58:22  use exp11                                                                                                                       0 row(s) affe
3  22:58:39  CREATE TABLE EMPLOYEES (    Employee_id CHAR(10) PRIMARY KEY,    First_Name CHAR(30) NOT NULL,    Last_Name CHAR(30) NOT NULL,  ...  0 row(s) affe
4  22:58:39  INSERT INTO EMPLOYEES (Employee_id, First_Name, Last_Name, DOB, Salary, Department_id) VALUES ('E001', 'John', 'Doe', '1985-01-15', 55000, 'D0...  6 row(s) affe
```

1. Execute the following index related queries:



```sql
--  3: Create an index on Last_Name and Department_id
CREATE INDEX employee_idx ON EMPLOYEES (Last_Name, Department_id);

--  4: Create a unique index on Employee_id
CREATE UNIQUE INDEX unique_employee_id_idx ON EMPLOYEES (Employee_id);

--  5: Create a reverse index on Employee_id
CREATE INDEX reverse_employee_id_idx ON EMPLOYEES (Employee_id DESC);

--  6: Create a unique composite index on Employee_id and check for duplicity
CREATE UNIQUE INDEX unique_composite_employee_id_idx ON EMPLOYEES (Employee_id, Department_id);

--  7: Create function-based indexes on Last_Name for case-insensitive searches
CREATE INDEX upper_last_name_idx ON EMPLOYEES ((UPPER(Last_Name)));
CREATE INDEX lower_last_name_idx ON EMPLOYEES ((LOWER(Last_Name)));

--  8: Drop the function-based index on Last_Name
DROP INDEX upper_last_name_idx ON EMPLOYEES;
```

```
#   Time      Action                                                                                    Message
1   22:59:18  CREATE INDEX employee_idx ON EMPLOYEES (Last_Name, Department_id)                         0 row(s) affecte
2   22:59:18  CREATE UNIQUE INDEX unique_employee_id_idx ON EMPLOYEES (Employee_id)                     0 row(s) affecte
3   22:59:18  CREATE INDEX reverse_employee_id_idx ON EMPLOYEES (Employee_id DESC)                      0 row(s) affecte
4   22:59:18  CREATE UNIQUE INDEX unique_composite_employee_id_idx ON EMPLOYEES (Employee_id, Department_id)  0 row(s) affecte
5   22:59:18  CREATE INDEX upper_last_name_idx ON EMPLOYEES ((UPPER(Last_Name)))                        0 row(s) affecte
6   22:59:18  CREATE INDEX lower_last_name_idx ON EMPLOYEES ((LOWER(Last_Name)))                        6 row(s) affecte
7   22:59:18  DROP INDEX upper_last_name_idx ON EMPLOYEES                                               0 row(s) affecte
```