

School of Computer Science, UPES, Dehradun.

A

LABORATORY FILE

On

DATABASE MANAGEMENT SYSTEM (DBMS) LAB

B.TECH. -III Semester

AUG. – NOV.- 2024.

Submitted by:

Name: Mehul Sangwan SAP ID: 500120255 Roll No: R2142230033

Batch: 2

DBMS Lab

B.Tech. III Sem.

Experiment 11

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

Execute the following index related queries:

- 1. Create an index of name employee_idx on EMPLOYEES with column Last_Name, Department_id
- 2. Find the ROWID for the above table and create a unique index on employee_id column of the EMPLOYEES.
- 3. Create a reverse index on employee_id column of the EMPLOYEES.
- 4. Create a unique and composite index on employee_id and check whether there is duplicity of tuples or not.
- 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).
- 6. Drop the function based index on column Last_Name.

DBMS Lab B.Tech. III Sem.

Results:

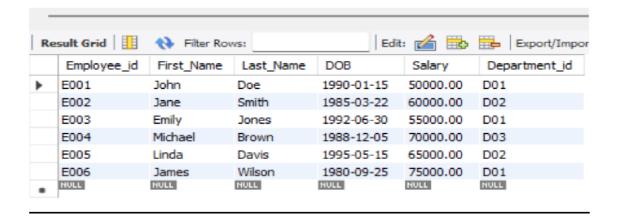
```
1
       -- Ayush Vashishth
       -- 500119331
 2
 3
4 • CREATE DATABASE exp10;
 5 • USE exp10;
7
        -- Creating the EMPLOYEES Table
8 • G CREATE TABLE EMPLOYEES (
        Employee_id CHAR(10) PRIMARY KEY,
        First_Name CHAR(30) NOT NULL,
10
        Last_Name CHAR(30) NOT NULL,
11
        DOB DATE,
12
        Salary DECIMAL(10, 2) NOT NULL, -- Using DECIMAL to handle salaries with two decimal places
13
14
        Department id CHAR(10)
     ();
15
16
       -- Inserting values into the EMPLOYEES table
17
18 • INSERT INTO EMPLOYEES (Employee_id, First_Name, Last_Name, DOB, Salary, Department_id) VALUES
       ('E001', 'John', 'Doe', '1990-01-15', 50000.00, 'D01'),
19
       ('E002', 'Jane', 'Smith', '1985-03-22', 60000.00, 'D02'),
20
       ('E003', 'Emily', 'Jones', '1992-06-30', 55000.00, 'D01'),
21
       ('E004', 'Michael', 'Brown', '1988-12-05', 70000.00, 'D03'),
22
23
       ('E005', 'Linda', 'Davis', '1995-05-15', 65000.00, 'D02'),
       ('E006', 'James', 'Wilson', '1980-09-25', 75000.00, 'D01');
24
25
        -- Creating a View named emp_view
26
27 • CREATE VIEW emp_view AS
        SELECT Employee_id, Last_Name, Salary, Department_id
28
```

DBMS Lab

B.Tech. III Sem.

```
FROM EMPLOYEES;
30
       -- You cannot directly insert into a view like this unless you are inserting into an updatable view that maps directly to a base table.
31
32
       -- Remove the insert into view since it will cause errors.
33
34
       -- If you need to modify the Salary column to allow NULL values, you'd do the following:
      ALTER TABLE EMPLOYEES MODIFY Salary DECIMAL(10, 2) NULL;
35 .
37
       -- Now, you can insert a row with a NULL salary
38 •
      INSERT INTO EMPLOYEES (Employee_id, First_Name, Last_Name, DOB, Salary, Department_id) VALUES
39
       ('E007', 'Chris', 'Anderson', NULL, NULL, 'D01');
40
41
       -- Update operations on the View (affects the base table EMPLOYEES)
42 0
      UPDATE emp_view
43
       SET Salary = 80000.00
44
       WHERE Employee_id = 'E001';
45
       -- Delete an employee from the view (and consequently from the EMPLOYEES table)
45
       DELETE FROM emp_view
48
       WHERE Employee_id = 'E003';
49
50 .
       SELECT * FROM emp view;
51
       -- Dropping the emp_view
52 •
       DROP VIEW emp_view;
53
54
       -- Create a View named salary_view to show annual salary for employees in Department D02
      CREATE VIEW salary_view AS
55 •
56
       SELECT Employee_id, Last_Name, Salary * 12 AS Annual_Salary
                SELECT Employee_id, Last_Name, Salary * 12 AS Annual_Salary
  56
                 FROM EMPLOYEES
  57
                WHERE Department id = 'D02';
  58
  59
                -- View the salary_view
  60
                SELECT * FROM salary view;
  61 •
```

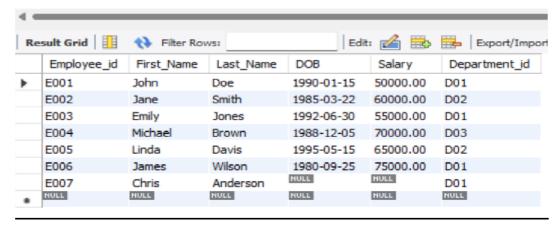
1 • SELECT * FROM exp10.employees;



DBMS Lab

B.Tech. III Sem.

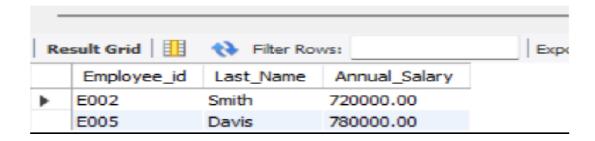
SELECT * FROM exp10.employees;



5ELECT * FROM exp10.emp_view;



SELECT * FROM exp10.salary_view;



DBMS Lab B.Tech. III Sem.

Conclusion:

The SQL code effectively sets up a relational database to manage employee information, including functionalities for inserting, updating, and deleting records. The creation of views enhances data accessibility by allowing users to easily retrieve specific employee data without dealing with the underlying table directly.

The use of a view for annual salaries provides a clear example of how to present calculated data, facilitating reporting and analysis. Overall, this database design supports efficient employee management and can be expanded further with additional features such as more complex views or stored procedures for automated reporting.

Future enhancements could include adding indexes for faster querying, more detailed employee attributes, or implementing stored procedures for common operations to streamline data management tasks.