

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

Experiment 07

Use of Inbuilt functions and relational algebra operation

AIM:

To understand the use of inbuilt function and relational algebra with SQL query.

Problem Statement:

- 1. Create the company database and tables as Exp. 4
- 2. Perform some SQL Queries on database

THEORY:

SQL (Structured Query Language) is a programming language used for managing and manipulating data in relational databases. It allows users to query, update, and organize data stored in tables using commands like SELECT, JOIN, WHERE, and GROUP BY. Joins combine rows from different tables based on a common column, while the WHERE clause filters data based on specified conditions. Subqueries and operators like NOT EXISTS allow for more complex queries, checking for the presence or absence of data. SQL also incorporates relational algebra operations, which are fundamental in querying and managing relational databases.

COMMAND USED:

- 1. DROP TABLE: Deletes a specified table and all its data.
- 2. PRIMARY KEY: Defines a column or set of columns as the unique identifier for rows in a table.
- 3. FOREIGN KEY: Establishes a relationship between columns in different tables.
- 4. ORDER BY: Sorts the result set by one or more specified columns.
- 5. WHERE: Filters records based on specific conditions.
- 6. GROUP BY: Groups rows that have the same values into summary rows.
- 7. HAVING: Filters records after the GROUP BY clause.

8. JOIN: Combines rows from two or more tables based on a related column.

9. CREATE INDEX: Creates an index on a table to speed up data retrieval operations.

RESULTS:

```
-- Ayush Vashishth
         -- 500119331
         CREATE DATABASE company;
         USE company;
 6 • @ CREATE TABLE EMPLOYEE (
             Fname VARCHAR(15) NOT NULL,
             Minit CHAR,
 8
 9
             Lname VARCHAR(15) NOT NULL,
10
            Ssn CHAR(9) NOT NULL,
            Bdate DATE,
11
            Address VARCHAR(30),
12
             Sex CHAR,
13
14
            Salary DECIMAL(10,2),
             Super_ssn CHAR(9),
15
             Dno INT NOT NULL
16
       - );
17
18
19 0
         INSERT INTO EMPLOYEE (Fname, Minit, Lname, Ssn, Bdate, Address, Sex, Salary, Super_ssn, Dno) VALUES
         ('John', '8', 'Smith', '123456789', '1965-01-09', '731 Fondren, Houston TX', 'M', 30000, '333445555', 5),
20
         ('Franklin', 'T', 'Wong', '333445555', '1965-12-08', '638 Voss, Houston TX', 'M', 40000, '888665555', 5),
21
         ('Alicia', 'J', 'Zelaya', '999887777', '1968-01-19', '3321 Castle, Spring TX', 'F', 25000, '987654321', 4),
22
         ('Jennifer', '5', 'Wallace', '987654321', '1941-86-20', '291 Berry, Bellaire TX', 'F', 43000, '888665555', 4),
23
         ('Ramesh', 'K', 'Narayan', '666884444', '1962-89-15', '975 Fire Oak, Humble TX', 'M', 38000, '333445555', 5),
24
         ('Joyce', 'A', 'English', '453453453', '1972-07-31', '5631 Rice, Houston TX', 'F', 25000, '333445555', 5),
25
         ('Ahmad', 'V', 'Jabbar', '987987987', '1969-03-29', '980 Dallas, Houston TX', 'M', 25000, '987654321', 4),
26
         ('James', 'E', 'Borg', '888665555', '1937-11-10', '450 Stone, Houston TX', 'M', 55000, NULL, 1);
27
28
29 • CREATE TABLE DEPARTMENT (
30
             Dname VARCHAR(15) NOT NULL,
31
             Dnumber INT NOT MULL,
32
             Mgr_ssn CHAR(9) NOT NULL,
             Mgr_start_date DATE
33
34
        );
```

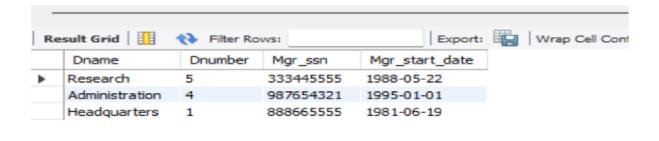
```
35
         INSERT INTO DEPARTMENT (Dname, Dnumber, Mgr_ssn, Mgr_start_date) VALUES
         ('Research', 5, '333445555', '1988-05-22'),
37
         ('Administration', 4, '987654321', '1995-01-01'),
38
         ('Headquarters', 1, '888665555', '1981-86-19');
39
48
41 • ( CREATE TABLE PROJECT (
             Pname VARCHAR(15) NOT NULL,
42
43
            Pnumber INT NOT NULL,
             Plocation VARCHAR(15),
44
             Down INT NOT NULL
45
        - );
46
47
         INSERT INTO PROJECT (Pname, Pnumber, Plocation, Dnum) VALUES
48 •
         ('ProductX', 1, 'Bellaire', 5),
49
         ('ProductY', 2, 'Sugarland', 5),
50
         ('ProductZ', 3, 'Houston', 5),
51
         ('Computerization', 10, 'Stafford', 4),
         ('Reorganization', 20, 'Houston', 1),
53
54
         ('Newbenefits', 30, 'Stafford', 4);
55
56
57 • CREATE TABLE WORKS_ON (
             Essn CHAR(9) NOT NULL,
58
59
             PRO INT NOT NULL,
             Hours DECIMAL(3,1) NOT NULL
68
61
       - ):
62
         INSERT INTO WORKS_ON (Essn, Pno, Hours) VALUES
63 •
         ('123456789', 1, 32.5),
64
         ('123456789', 2, 7.5),
65
         ('6668844444', 3, 40.0),
         ('453453453', 1, 20.0);
67
68
```

```
69 • (--) CREATE TABLE DEPENDENT (
              Essn CHAR(9) NOT NULL,
 71
              Dependent_name VARCHAR(15) NOT NULL,
              Relationship VARCHAR(15)
 72
 73
         - );
 74
 75
          -- Insert data into DEPENDENT
         INSERT INTO DEPENDENT (Essn, Dependent_name, Relationship) VALUES
 76 0
 77
          ('123456789', 'Alice', 'Daughter'),
          ('333445555', 'Mike', 'Son'),
          ('999887777', 'Sara', 'Daughter');
 719
 RIB
81 •
          SELECT E.Fname, E.Lname
           FROM Employee E
 83
          JOIN Works_On W ON E.Ssn = W.Essn
           JOIN Project P ON W.Pno = P.Pnumber
 R4
          WHERE E.Dno = 5 AND P.Pname = 'ProductX' AND W.Hours > 10;
           SELECT E.Fname, E.Lname
 B7 •
BR
           FROM Employee E
          JOIN Dependent D ON E.Ssn = D.Essn
90
          WHERE E.Fname = D.Dependent_name;
ST
92 0
          SELECT E.Fname, E.Lname
          FROM Employee E
          JOIN Employee S ON E.Super_ssn = S.Ssn
95
          WHERE S.Fname = 'Franklin' AND S.Lname = 'Wong';
96
         SELECT E.Fname, E.Lname
         FROM Employee E
SR
     WHERE NOT EXISTS (
 4040
          SELECT P.Pnumber
          FROM Project P
       WHERE NOT EXISTS (
102
```

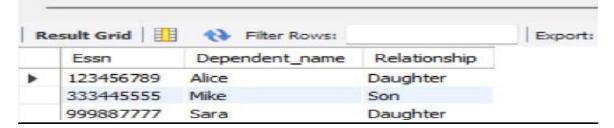
```
183
          SELECT W.Pno
184
          FROM Works_On W
          WHERE W.Essn = E.Ssn AND W.Pno = P.Pnumber
105
186
        - )
        - );
107
188
109 0
          SELECT E.Fname, E.Lname
110
          FROM Employee E
          LEFT JOIN Works_On W ON E.Ssn = W.Essn
111
          WHERE W. Pno IS NULL:
112
113
114
          SELECT E. Lname
115
          FROM Employee E
116
          JOIN Department D ON E.Ssn = D.Mgr_ssn
117
          LEFT JOIN Dependent Dep ON E.Ssn = Dep.Essn
          WHERE Dep.Essn IS NULL;
118
119
```



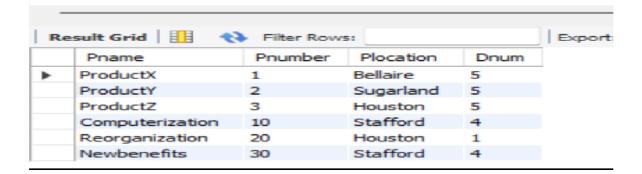
5ELECT * FROM company.department;



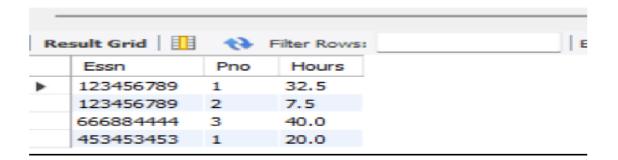
1 • SELECT * FROM company.dependent;



SELECT * FROM company.project;



1 • SELECT * FROM company.works_on;



Conclusion:

In this experiment, we successfully explored the use of SQL in relational databases, demonstrating how inbuilt functions and relational algebra operations can be applied through various queries. By utilizing commands such as JOIN, WHERE, GROUP BY, and subqueries like NOT EXISTS, we could filter, combine, and manipulate data across different tables in the company database. These operations reflect the power of SQL in managing complex data relationships efficiently. The experiment provided hands-on experience with key SQL features, including filtering conditions, relationships between entities, and summarizing data, which are essential for handling real-world database scenarios. This exercise reinforced our understanding of relational algebra principles as they apply to practical SQL query writing.