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: Harsh Vardhan Saini
SAP ID: 500120369
Roll No: R2142230056
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:

```sql
-- Ayush Vashishth
-- 500119331

CREATE DATABASE company;
USE company;
CREATE TABLE EMPLOYEE (
    Fname VARCHAR(15) NOT NULL,
    Minit CHAR,
    Lname VARCHAR(15) NOT NULL,
    Ssn CHAR(9) NOT NULL,
    Bdate DATE,
    Address VARCHAR(30),
    Sex CHAR,
    Salary DECIMAL(10,2),
    Super_ssn CHAR(9),
    Dno INT NOT NULL
);

INSERT INTO EMPLOYEE (Fname, Minit, Lname, Ssn, Bdate, Address, Sex, Salary, Super_ssn, Dno) VALUES
('John', 'B', 'Smith', '123456789', '1965-01-09', '731 Fondren, Houston TX', 'M', 30000, '333445555', 5),
('Franklin', 'T', 'Wong', '333445555', '1965-12-08', '638 Voss, Houston TX', 'M', 40000, '888665555', 5),
('Alicia', 'J', 'Zelaya', '999887777', '1968-01-19', '3321 Castle, Spring TX', 'F', 25000, '987654321', 4),
('Jennifer', 'S', 'Wallace', '987654321', '1941-06-20', '291 Berry, Bellaire TX', 'F', 43000, '888665555', 4),
('Ramesh', 'K', 'Narayan', '666884444', '1962-09-15', '975 Fire Oak, Humble TX', 'M', 38000, '333445555', 5),
('Joyce', 'A', 'English', '453453453', '1972-07-31', '5631 Rice, Houston TX', 'F', 25000, '333445555', 5),
('Ahmad', 'V', 'Jabbar', '987987987', '1969-03-29', '980 Dallas, Houston TX', 'M', 25000, '987654321', 4),
('James', 'E', 'Borg', '888665555', '1937-11-10', '450 Stone, Houston TX', 'M', 55000, NULL, 1);

CREATE TABLE DEPARTMENT (
    Dname VARCHAR(15) NOT NULL,
    Dnumber INT NOT NULL,
    Mgr_ssn CHAR(9) NOT NULL,
    Mgr_start_date DATE
);
```

```
35
36 ●    INSERT INTO DEPARTMENT (Dname, Dnumber, Mgr_ssn, Mgr_start_date) VALUES
37      ('Research', 5, '333445555', '1988-05-22'),
38      ('Administration', 4, '987654321', '1995-01-01'),
39      ('Headquarters', 1, '888665555', '1981-06-19');
40
41 ●  CREATE TABLE PROJECT (
42          Pname VARCHAR(15) NOT NULL,
43          Pnumber INT NOT NULL,
44          Plocation VARCHAR(15),
45          Dnum INT NOT NULL
46      );
47
48 ●    INSERT INTO PROJECT (Pname, Pnumber, Plocation, Dnum) VALUES
49      ('ProductX', 1, 'Bellaire', 5),
50      ('ProductY', 2, 'Sugarland', 5),
51      ('ProductZ', 3, 'Houston', 5),
52      ('Computerization', 10, 'Stafford', 4),
53      ('Reorganization', 20, 'Houston', 1),
54      ('Newbenefits', 30, 'Stafford', 4);
55
56
57 ●  CREATE TABLE WORKS_ON (
58          Essn CHAR(9) NOT NULL,
59          Pno INT NOT NULL,
60          Hours DECIMAL(3,1) NOT NULL
61      );
62
63 ●    INSERT INTO WORKS_ON (Essn, Pno, Hours) VALUES
64      ('123456789', 1, 32.5),
65      ('123456789', 2, 7.5),
66      ('666884444', 3, 40.0),
67      ('453453453', 1, 20.0);
68
```

```sql
CREATE TABLE DEPENDENT (
    Essn CHAR(9) NOT NULL,
    Dependent_name VARCHAR(15) NOT NULL,
    Relationship VARCHAR(15)
);

-- Insert data into DEPENDENT
INSERT INTO DEPENDENT (Essn, Dependent_name, Relationship) VALUES
('123456789', 'Alice', 'Daughter'),
('333445555', 'Mike', 'Son'),
('999887777', 'Sara', 'Daughter');

 SELECT E.Fname, E.Lname
 FROM Employee E
 JOIN Works_On W ON E.Ssn = W.Essn
 JOIN Project P ON W.Pno = P.Pnumber
 WHERE E.Dno = 5 AND P.Pname = 'ProductX' AND W.Hours > 10;

 SELECT E.Fname, E.Lname
 FROM Employee E
 JOIN Dependent D ON E.Ssn = D.Essn
 WHERE E.Fname = D.Dependent_name;

SELECT E.Fname, E.Lname
FROM Employee E
JOIN Employee S ON E.Super_ssn = S.Ssn
WHERE S.Fname = 'Franklin' AND S.Lname = 'Wong';

SELECT E.Fname, E.Lname
FROM Employee E
WHERE NOT EXISTS (
SELECT P.Pnumber
FROM Project P
WHERE NOT EXISTS (
    SELECT W.Pno
    FROM Works_On W
    WHERE W.Essn = E.Ssn AND W.Pno = P.Pnumber
    )
    );

    SELECT E.Fname, E.Lname
    FROM Employee E
    LEFT JOIN Works_On W ON E.Ssn = W.Essn
    WHERE W.Pno IS NULL;

    SELECT E.Lname
    FROM Employee E
    JOIN Department D ON E.Ssn = D.Mgr_ssn
    LEFT JOIN Dependent Dep ON E.Ssn = Dep.Essn
    WHERE Dep.Essn IS NULL;
```

```
1 • SELECT * FROM company.employee;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston TX	M	30000.00	333445555	5
Franklin	T	Wong	333445555	1965-12-08	638 Voss, Houston TX	M	40000.00	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring TX	F	25000.00	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire TX	F	43000.00	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble TX	M	38000.00	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston TX	F	25000.00	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston TX	M	25000.00	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston TX	M	55000.00	NULL	1

```
1 • SELECT * FROM company.department;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Cont

Dname	Dnumber	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

```
1 •  SELECT * FROM company.dependent;
```

Result Grid | Filter Rows: | Export:

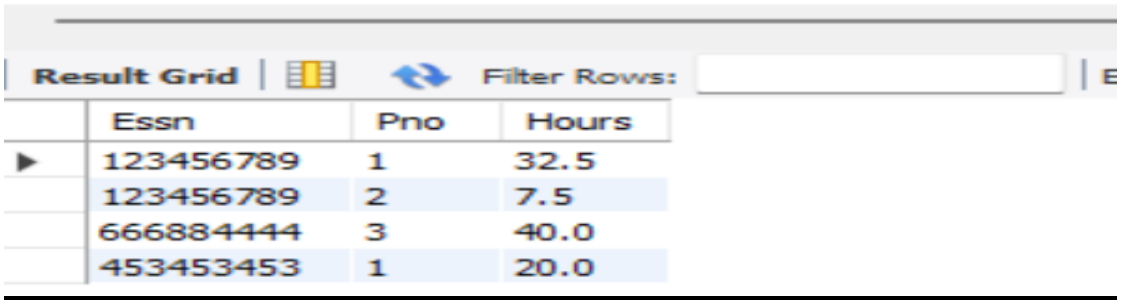
Essn	Dependent_name	Relationship
123456789	Alice	Daughter
333445555	Mike	Son
999887777	Sara	Daughter

```
1 •  SELECT * FROM company.project;
```

Result Grid | Filter Rows: | Export:

Pname	Pnumber	Plocation	Dnum
ProductX	1	Bellaire	5
ProductY	2	Sugarland	5
ProductZ	3	Houston	5
Computerization	10	Stafford	4
Reorganization	20	Houston	1
Newbenefits	30	Stafford	4

```
1 •  SELECT * FROM company.works_on;
```



Result Grid | Filter Rows:

	Essn	Pno	Hours
▶	123456789	1	32.5
	123456789	2	7.5
	666884444	3	40.0
	453453453	1	20.0

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.