3.1	18/08/25	Using Clauses, Operators And Functions In Queries	13	m 18/8/25 ✓
3.2	25/08/25	Aggregate Functions	13	m 25/8/25 ✓

3/8/25 Task-3.1 : Using Clauses , Operators And Functions In Queries

Aim:-

To implement DML commands using clauses, operators and functions in queries.

Clauses:

→ WHERE, ORDER BY, GROUP BY, HAVING, DISTINCT.

Operators:

→ Equal (=)

→ BETWEEN

→ AND

→ OR

→ IN

```
CREATE TABLE DEPARTMENT1(
        DEPT ID INT PRIMARY KEY;
        DEPT NAME VARCHAR(50) UNIQUE NOT NULL;
        LOCATION VARCHAR(50) NOT NULL;
CREATE TABLE STUDENT1(
        STUDENT ID INT PRIMARY KEY;
        NAME VARCHAR(50) NOT NULL;
        AGE INT CHECK (AGE >= 18);
        DEPT ID INT FOREIGN KEY REFERENCES
                                 DEPARTMENT1(DEPT ID);
        CITY VARCHAR(50) DEFAULT 'UNKNOWN';
        JOINDATE DATETIME DEFAULT GET DATE ();

INSERT INT DEPARTMENT1 VALUES
(1, 'CSE', 'HYDERABAD');
(2, 'ECE', 'MUMBAI');
(3, 'MECH', 'DELHI');

INSERT INTO STUDENT1 VALUES
(101, UPPER('Sushant'), 20, 1, 'HYDERABAD');

INSERT INTO STUDENT1 VALUES
(102, 'KARTHIK', 22, 2, 'MUMBAI');
```

```
INSERT INTO STUDENT1 VALUES
(103, 'MAHI', 19,1, 'PUNE');
INSERT INTO STUDENT1 VALUES
(104, 'VIRAT', 23, 3, 'DELHI');
INSERT INTO STUDENT1 VALUES
(105, 'SARA', 21, 1, 'HYDERABAD');
SELECT *FROM STUDENT1;
```

S. No.	STUDENT ID	NAME	AGE	DEPT ID	CITY	JOIN DATE
1.	101	SUSHANT	20	1	HYDERABAD	2025-08-26
2.	102	KARTHIK	22	2	MUMBAI	2025-08-26
3.	103	MAHI	19	1	PUNE	2025-08-26
4.	104	VIRAT	23	3	DELHI	2025-08-26
5	105	SARA	21	1	HYDERABAD	2025-08-26

```
SELECT *FROM DEPARTMENT 1;
```

S. No	DEPT ID	DEPT NAME	LOCATION
1	1	CSE	HYDERABAD
2	2	ECE	MUMBAI
3	3	MECH	DELHI

```
SELECT NAME , AGE
FROM STUDENT 1
WHERE AGE BETWEEN 19 AND 22;
```

S. No.	NAME	AGE
1	SUSHANT	20
2	KARTHIK	22
3	MAHI	19
4	SARA	21

```
SELECT NAME, DEPT ID
FROM STUDENT1
WHERE DEPT ID IN (1, 3)
ORDER BY DEPT ID DESC;
```

Sl. No.	NAME	DEPT ID
1	VIRAT	3
2	SARA	1
3	SUSHANT	1
4	MAHI	1

```
UPDATE  STUDENT 1
SET  AGE  = AGE + 1
WHERE  DEPTID = 1  AND  AGE < 21;
```

S. No.	STUDENT ID	NAME	AGE	DEPT ID	CITY	JOIN DATE
1	101	SUSHANT	21	1	HYDERABAD	2025-08-26
2	102	KARTHIK	22	2	MUMBAI	2025-08-26
3	103	MAHI	20	1	PUNE	2025-08-26
4	104	VIRAT	23	3	DELHI	2025-08-26
5	105	SARA	21	1	HYDERABAD	2025-08-26

```
SELECT DISTINCT CITY
FROM  STUDENT 1;
```

S. No.	CITY
1	DELHI
2	HYDERABAD
3	MUMBAI
4	PUNE

```
SELECT  DEPT ID , COUNT (*)  AS  TOTAL_ STUDENTS
FROM  STUDENT 1
GROUP  BY  DEPT ID;
```

S. No.	DEPT ID	TOTAL - STUDENTS
1	1	3
2	2	1
3	3	1

```
SELECT  DEPT ID, COUNT(*)  AS  TOTAL_STUDENTS
FROM  STUDENT 1
GROUP BY DEPT ID
HAVING  COUNT(*) >= 2;
```

S. No.	DEPT-ID	TOTAL - STUDENTS
1	1	3

VELTECH	
EX No.	3.1
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	3
RECORD (5)	
TOTAL (20)	13
SIGN WITH DATE	[illegible] 18/8/25

Result:-

Thus, the implementation of clauses, operators and functions in the queries (DDL and DML commands) was successfully executed.

25/8/25

Task-3.2 : Aggregate Functions

Aim:- To implement aggregate functions (count (), sum (), avg (), min (), max ()) on a sample database in my sql.

Aggregate Functions:-

They're mostly used with grouped by to group rows.

→ count ()

→ sum ( )

→ AVG ()

→ MIN ()

→ MAX ()

```
CREATE TABLE STUDENT 2 ( )
            ROLL NO  INT  PRIMARY KEY,
            NAME  VARCHAR (50),
            AGE  INT
            DEPT ID.  INT,
            MARKS INT;
INSERT INT STUDENT 2 VALUES
(1, 'Abhay', 20, 101, 85),
(2, 'Sharvari', 21, 101, 90),
(3, 'Rajkumar', 19, 102, 95),
(4, 'Shraddha', 22, 102, 95),
(5, 'Varun', 20, 101, 60),
(6, 'Kriti', 23, 103, 88),
SELECT * FROM STUDENT 2;
```

Sl. No.	ROLL NO	NAME	AGE	DEPT ID	MARKS
1	1	Abhay	20	101	85
2	2	Sharvari	21	101	90
3	3	Rajkumar	19	102	70
4	4	Shraddha	22	102	95
5	5	Varun	20	101	60
6	6	Kriti	23	103	88

SELECT DEPT ID, AVG(MARKS) AS AVG_ MARKS
FROM STUDENT2
GROUPED BY DEPT ID;

	DEPT ID	AVG_ MARKS
1	101	78
2	102	82
3	103	88

SELECT DEPT ID ,MAX (MARKS) AS TOP_ MARK
FROM STUDENT 2
GROUP BY DEPT ID;

	DEPT ID	TOP_ MARK
1	101	90
2	102	95
3	103	88

SELECT DEPT ID, MIN (MARKS) AS LEAST MARK
FROM STUDENT 2
GROUP BY DEPT ID

	DEPT ID	LEAST_MARK
1	101	60
2	102	70
3	103	88

SELECT DEPT ID, COUNT(*) AS STU_ COUNT
FROM STUDENT 2
GROUP BY DEPT ID;

	DEPT ID	STU_ COUNT
1	101	3
2	102	2
3	103	1

VELTECH	
EX No.	3.2
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	3
RECORD (5)	—
TOTAL (20)	13
SIGN WITH DATE	[illegible] 25/8/25

Result:- Thus, the task to implement all aggregate functions has been successfully executed.