TASK (3.1) 18/8/25.

Using clauses, operators and functions in Queries

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

Clauses

→ Where, order BY, GROUPBY, HAVING, DISTINCT

Operators

→ equal (=)
→ BETWEEN
→ AND
→ AND
→ OR
→ IN

```
CREATE   TABLE  DEPARTMENT(
         DEPTID  INT  PRIMARY KEY,
         DEPTNAME  VARCHAR (50) UNIQUE  NOT NULL,
         LOCATION , VARCHAR (50) NOT NULL;

CREATE  TABLE  STUDENT(
      STUDENTID  INT   PRIMARY KEY,
      NAME_VARCHAR (50)  NOTNULL,
      AGE  INT  CHECK  (AGE>==18),
      DEPT ID INT  FOREIGN  KEY  REFERENCES
                          DEPARTMENT (DEPT ID)
    CITY  VARCHAR(50) DEFAULT 'UNKNOWN',

    JOINDATE  DATETIME  DEFAULT  GET DATE();
```

```
INSERT INTO DEPARTMENT VALUES
    (1, 'CSE', 'HYDERABAD'),
    (2, 'EEE', 'MUMBAI');
    (3, 'MECH', 'DELHI');
INSERT INTO STUDENT VALUES
(101, UPPER ('radesh'), 20, 1, 'HYDERABAD');
INSERT INTO STUDENT I VALUES
(102, 'ANJALI', 22, 2, 'MUMBA');
INSERT INTO STUDENT VALUES
(103, ('KIRAN', 19, 1, 'PUNE');
INSERT INTO STUDENT VALUES;
~~INSERT INTO~~ STUDENT VALUES
(104, 'MOITH', 23, 3, 'DELHI');
INSERT INTO STUDENT VALUES.
(105, 'SARAKHAN', 21, 1, 'HYDERABA');
SELECT * FROM STUDENT;
```

	STUDENT NAME ID	STUDENT ID NAME	AGE	DEPT ID	CITY	JOIN DATE
1	Radesh	101	20	1	HYDERABAD	2025-8-26
2	ANJALI	102	22	2	MUMBAI	2025-8-26
3	KIRAN	103	19	1	PUNE	2025-8-26
4	MOHITH	104	23	3	DELHI	2025-8-26
5	SARAKHAN	105	21	1	HYDERABAD	2025-8-26.

```
SELECT * FROM DEPARTMENT;
```

	DEPT ID	DEPT NAME	LOCATION
1	1	CSE	HYD
2	2	EEE	MUMBAI
3	3	MECH	DELHI

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

	NAME	AGE
1	RAHUL	20
2	ANJALI	22
3	KIRAN	19
4	SARAKHAN	21

```
SELECT NAME, DEPT ID
from STUDENT
WHERE DEPT ID IN(1,3)
ORDER BY DEPTID DESC;
```

	NAME	DEPT ID
1	MOHITH	3
2	SARAKHAN	1
3	RAHUL	1
4	KIRAN	1

```
UPDATE STUDENT1
SET AGE=AGE+1
WHERE DEPTID =1 AND AGE <21;
```

	STUDENT ID	NAME	Age	DEPT ID	CITY	JOINDATE
1	101	RAHUL	21	1	HYDARABAD	2025-8-26
2	102	ANJALI	22	2	MUMBAI	2025-8-26
3	103	KIRAN	20	1	PUNE	2025-8-26
4	104	MOHITH	23	3	DELHI	2025-8-26
5	105	SARAKHAN	21	1	HYDERABAD	2025-8-26.

```
SELECT DISTINCTCITY
FROM STUDENT;
```

	CITY
1	DEHLI
2	HYDERABAD
3	MUMBAI
4	PUNE

```
SELECT DEPTID, COUNT(*) AS TOTAL - STUDENTS
FROM STUDENT
GROUP BY DEPTID;
```

	DEPTID	TOTAL-STUDENTS
1	1	3
2	2	1
3	3	1

```
SELECT DEPTID, COUNT(*) AS TOTAL - STUDENTS
FROM STUDENT!
GROUP BY DEPT ID
HAVING COUNT(*) >= 2;
```

	DEPTID	TOTAL-STUDENTS.
1	1	3

VELTECH	
EX No.	3.1
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	5
RECORD (5)	–
TOTAL (20)	15
SIGN WITH DATE	18/8/[illegible]

Result:- The implementation of the clauses, operators & function in the queue (DDL & DML commands) are exicuted sucessfully.

25/8/25

Task (3.2). Aggregate functions.

Aim:- To ~~study~~ & implement aggregate functions (Count(), sum (), Avg (), Min (), Max ()) on a sample database. in my SQL.

Aggregate functions

They're mostly used with groupedby to group rows.

→ COUNT()
→ SUM()
→ AVG()
→ MIN()
→ MAX()

```
CREATE TABLE STUDENT2(
              RollNO  INT PRIMARY KEY,
              NAME   VARCHAR(50),
              AGE INT,
              DEPTID  INT,
              MARKS INT;
INSERT INT STUDENT2  VALUES
(1, 'Arjun', 20, 101, 85),
(2, 'sneha', 21, 101, 90),
(3, 'Ravi'. 19, 102, 95),
(4, 'Priya'. 22, 102, 95),
(5, 'kiran', 20, 101, 60),
(6, 'Anita', 23, 103, 88),
```

SELECT* FROM STUDENT2;

	RollNO	NAME	AGE	DEPTID	MARKS
1	1	Arjun	20	101	85
2	2	Sneha	21	101	90
3	3	Ravi	19	102	70
4	4	Priya	22	102	95
5	5	kiran	20	101	60
6	6	Anita	23	103	88

SELECT·DETID, AVG (MARKS) AS AVG-MARKS
FROM STUDENT2
GROUPED BY DEPTID;

	DEPTID	AVG-MARKS
1	101	78
2	102	82
3	103	88

SELECT DEPTID, MAX (MARKS) AS TOP_MARK
FROM STUDENT2
GROUPBDY DEPTID;

	DEPTID	TOP_MARK
1	101	90
2	102	95
3	103	88

SELECT DEPID, MIN (MARKS) AS LEASTMARK
FROM STUDENT2
GROUP BY DEPTID

	DEPTID	LEAST_MARK
1	101	60
2	102	70
3	103	88

SELCT DEPTID, COUNT (*) AS STO-COUNT
FROM STUDENT2
GROUP BY DEPTID;

	DEPTID	STU-COUNT
1	101	3
2	102	2
3	103	1

VELTECH	
No.	3-2
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	5
RECORD (5)	-
TOTAL (20)	15
SIGN WITH DATE	25/8/25

Result:- Implementation of all aggregate function has been performed successfully ~~on a ta~~ ~~Table~~. excicuted.