18/2/25

TASK (3.1)

Using CLAUSES, OPERATES

AND FUNCTIONS IN QUERS

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

Clauses:-

→ where, ORDER BY, GROUP BY, HAVING, DISTINT

OPERATORS

→ equal (=)

→ BETWEEN

→ AND

→ OR

→ IN

```
CREATE TABLE DEPARTMENT1(
    DEPTID INT PRIMARY KEY,
    DEPTNAME VARCHAR(50) UNIQUE NOTNLK
    LOCTION VARCH AR(50) NOT NULL;
CREATE TABLE STUDENT (
    STUDENT ID INT PRIMARY KEY,
    NAME : VARCHAR(50)NOT NULL,
    AGE INT. CHECK (AGE)=18),
    DEPT ID INT PORBIGN KEY Referene.
    CITY VARCHA R(50) DEFAULT (UNIKOWN)
    JOINDATE DATE TIME DEFAULT DATE);
```

```
INSERT INTO DEPARTMENT VALUES
    (1,'CSE' HYDERABAD');
    (2,'EEE','MUMBAI');
    (3,'MECH','DELHI');
INSERT INTO STUDENT VALUES
   (101, UPPER('rahul'),20,1,'HYDERBAD');
INSERT INTO STUDENT VALUES
  (102,'ANITHA',22,2,MUMBAI);
INSERT INTO STUDENT VALUES
  (103,'KIRAN',19,1,'PUNE');
INSERT INTO STUDENT VALUES
   (104,(MOHITH),23,3,'DELHI'
   INSERT INTO STUDENT VALUES.
   (105,'SARAICHNA),21,1,HYBDRBAD);
```

	student ID	NAME	AGE	DEPTID	CITY	JOIN DATE
1	101	RAHUL	20	1	HYDERABAD	2025-08-26
2	102	ANIA·	22	2	MUMBAI	2025-08-26
3	103	kiraj	19	1	PUNE	2025-08-26
4	104	MOHITH	23	3	DELHI	2025-08-26
5	105	SARAICHAN	21	1	HYBDERBAD	2025-08-26

```
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
```

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

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

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

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

	STUDENT ID	NAME	AGE	DEPT ID	CITY	JOIN DATE
1	101	RAHUL	21	1	HYDERBAD	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	HYDERBAD	2025-8-25

```
SELECT DISTINCT CITY
FROM STUDENT1;
```

	CITY
1	DEHLI
2	HYDERBAD
3	MUMBAI
4.	PUNE

```
SELECT DEPT ID, COUNT(*) AS TOTAL_STUDENTS
 FROM STUDENT1
GROUP BY DEPT ID;
```

	DEPT ID	TOTAL_STUDENTS
1	1	3
2	2	1
3	3	1

```
SELECT DEPT ID, COUNT(*) AS TOTAL_STUDENTS
FROM STUDENT1
GROUP BY DEPT ID
HAVING COUNT(*)>=2;
```

```
DEPTID   TOTAL_STUDENTS
1   1         3
```

VEL TECH	
EX No.	8
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	1
RECORD (5)	
TOTAL (20)	11
SIGN WITH DATE	18/9/20

Result!- The implementation of the clauses, operators and function in the qorie (DDL and DML Commands)

25/8/25

TASK (3.2) AGGREGATE FUNCTIONS

Aim:- To study and implement aggregate functions (count(), sum(), Avg(), Min() Max()) on a sample database

AGGREGATE FUNCTIONS

They're mostly used with GROUPED BY to group the rows

- → COUNT()
- → SUM()
- → AVG()
- → MIN()
- → MAV()

```
CRETE TABLE STUDENT2(
        ROLL NO INT PRIMARY KEY,
         NAME VARCHAR(50),
         AGE INT,
         DEPT ID INT,
         MARKS INT,);

INSERT INTO STUDENT 2 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	DEPT ID	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 DEPT ID, AVG(MARKS) AS AVG-MARKS
FROM STUDENT2
GROUPED BY DEPT ID;
```

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

```
SELECT DEPT ID, MAX(MARKS) AS TOP_MARK
 FROM STUDENT 2
 GROUP BOY 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)	1
RECORD (5)	
TOTAL	11
SIGN WITH DATE	25/8/23

Result:- Implemenation of all aggregate function has been performed sucessfully on a table.