15/8/25

TASK (3-1)

USING CLAUSES, OPERATORS AND FUNCTIONS IN QUERIES.

Aim:- To implemented 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 DEPARTMENT1(
        DEPTID INT PRIMARY.
        DEPTNAME VARCHAR(50) UNIQUE
                             NOT NULL,
        LOCATION VARCHAR(50) NOTNULL;

CREATE TABLE STUDENT1(
        STUDENTID INT PRIMARY KEY,
        NAME VARCHAR(50) NOTNULL,
        AGE INT CHECK (AGE>=18).
        DEPTID INT FOREIGN KEY REFERENCES
                DEPARTMENT1(DEPTID)
        CITY VARCHAR(50) DEFAULT 'UN
                                  KNOWN',
        JOINDATE DATETIME 'DEFAULT
                GETDATE),
```

clearly state the

```
INSERT INTO DEPARTMENT1 VALUES.
(1, 'CSE', 'HYDERABAD'),
(2, 'EEE', 'MUMBAI');
(3, 'MECH', 'DELHI'),

INSERT INTO' STUDENT1 VALUES
101, UPPER('chandu'), 20 1, 'HYDERABAD'),
INSERT INTO STUDENT1 VALUES.
102, 'ANJALI', 22, 2 'MUMBAI'),
INSERT INTO STUDENT1 VALUES.
103, 'IRFAN', 19, 1, 'PUNE'),
INSERT INTO STUDENT, VALUES}
(104, 'mohith.
INSERT INTO STUDENT VALUES.
105, 'SARALKHAN', 21, 1, 'HYDERABAD'),
SELECT * FROM STUDENTS;
```

	STUDENT ID	NAME	AGE	DEPT ID	CITY	JOINDATE 26/8/25
1.	101	Chandu	20	1	Hyderabad	
2.	102	Anjali	22	2	Mumbai	26/8/25
3.	103	Irfan	19	1	Pune	26/8/25
4.	104	mohith	23	3	Delhi	26/8/25
5.	105	saralkhan	21	1	Hyderabad	26/8/25

```
SELECT * FROM DEPARTMENT1;
```

	DEPTID	DEPTNAME	LOCATION1
1	1	CSE	HYD
2	2	EEE	MUMBAI
3	3	MECH	DELHI

```
Select NAME, AGE
FROM STUDENT
WHERE AGE BETWEEN
```

Clearly state the pr

	name	age
1	Rahul	20
2	Anjali	22
3	Imran	19
4	Salakhan	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	Salakhan	1
3	Rahul	1
4	Imran	1

```
UPDATE STUDENT
SET AGE = AGE+1
WHERE DEPT ID = AND AGE < 24;
```

	STUDENT	NAME	AGE	DEPT ID	CITY	JOIN DATE
1	101	Rahul	21	1	Hydera-bad	26/8/25
2	102	Anjali	22	2	Mumbai	26/8/25
3	103	Imran	20	1	Pune	26/8/25
4	104	Mohith	23	3	Delhi	26/8/25
5	105	Salakhan	21	1	Hyderabad	26/8/25

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

	CITY
1	Delhi
2	Hyderabad
3	Mumbai
4	Pune

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

	DEPTID	TOTAL_STUDENTS
1	1	3
2	2	1
3	3	1

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

DEPTID	TOTAL_STUDENTS
1	3

VEL TECH	
EX No.	3.1
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	5
RECORD (5)	1
TOTAL (20)	16
SIGN WITH DATE	

18/3/25

Result: The implementation of the clauses, operators & functions in the Queries (DDL and DML commands).

25/8/25. TASK-3-2 AGGREGATE FUNCTIONS

Aim: TO study & 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()
-> MAX()

CREATE TABLE STUDENT2(
ROLLNO INT PRIMARY KEY,
NAME VARCHAR(50),
AGE INT,
DEPTID INT,
MARKS INT);

INSERT INTO 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	Given	20	101	60
6	6	Anita	23	103	88

SELECT DEPTID, 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
GROUP BY DEPTID;

	DEPTID	TOP_MARK
1.	101	90
2.	102	95
3	103	88

SELECT DEPTID, MIN(MARKS) AS LEAST MARK
FROM STUDENT2
GROUP BY DEPTID.

	DEPTID	LEAST_MARK
1.	101	60
2.	102	70
3.	103	88

SELECT DEPTID, COUNT(*) AS STU_COUNT.
FROM STUDENT2
GROUP BY DEPTID;

	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
VIVAVOCE (3)	4
RECORD (4)	—
TOTAL (15)	14
SIGN WITH DATE	8/09/25

Result: Implementation of all aggregate functions has been performed successfully on a table.