

18/12/25

TASK (3.1)

Using CLAUSES, OPERATORS
AND FUNCTIONS IN QUERIES

Aim:- To implement of 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 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) NOT NULL,

AGE INT CHECK (AGE >= 18),

DEPTID INT FOREIGN KEY REFERENCES,

CITY VARCHAR(50) DEFAULT UNKNOWN,

JOINDATE DATETIME DEFAULT DATE);

INSERT INTO DEPARTMENT VALUES

(1, 'CSE', 'HYDERABAD'),

(2, 'EEE', 'MUMBAI'),

(3, 'MECH', 'DELHI');

INSERT INTO STUDENT VALUES

(101, 'RAHUL', 20, 1, 'HYDERABAD'),
INSERT INTO STUDENT VALUES

(102, 'ANJALI', 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, 'SARAKHAN', 21, 1, 'HYDERABAD'),

student ID	NAME	AGE	DEPT ID	CITY	JOIN DATE
1	RAHUL	20	1	HYDERABAD	2025-08-26
2	ANJALI	22	2	MUMBAI	2025-08-26
3	KIRAN	19	1	PUNE	2025-08-26
4	MOHITH	23	3	DELHI	2025-08-26
5	SARAKHAN	21	1	HYDERABAD	2025-08-26

SELECT * FROM DEPARTMENT;

DEPT ID	Dept Name	Location
1	CSE	HVD
2	EEE	MUMBAI
3	MECH	DELHI

SELECT NAME, AGE FROM STUDENT

NAME	AGE
RAHUL	20
ANJALI	22
KIRAN	19
SARAKHAN	21

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

	NAME	DEPT ID
1.	MONTU	3
2.	SARAJAHAN	1
3.	RAHUL	1
4.	KIRAN	1

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

STUDENT ID	NAME	AGE	DEPT ID	CITY	JOB DATE
101	RAHUL	21	1	HYDERBAD	2025-8-26
102	ANALI	22	2	MUMBAI	2025-8-26
103	KIRAN	20	1	PUNE	2025-8-26
104	MONTU	23	3	DELHI	2025-8-26
105	SARAJAHAN	21	1	HYDERBAD	2025-8-26

SELECT DISTINCT CITY
 FROM STUDENT;

CITY
1. DELHI
2. HYDERBAD
3. MUMBAI
4. PUNE

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

DEPT ID	TOTAL_STUDENTS
1	3
2	1
3	1

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

DEPTID	TOTAL STUDENTS
--------	----------------

1	1
---	---

	3
--	---

VEL TECH	
EX NO.	81
*PERFORMANCE (5)	5
*RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	1
RECORD (1)	1
TOTAL (18)	11
DATE WITH DATE	18/9/20

Result:- The implementation of the clauses, operators and function in the query (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()
- MAX()

~~CREATE TABLE STUDENT2(~~
ROLL NO INT PRIMARY KEY,
NAME VARCHAR(50),
AGE INT,
DEPTID INT,
MARKS INT);

INSERT IN 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);

~~AT~~ 111
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	90
4	4	Priya	22	102	95
5	5	Kiran	20	101	60
6	6	Anita	23	103	88

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

	DEPTID	Avg Marks
1	101	98
2	102	82
3	103	88

SELECT DEPT ID, MAX(MARKS) AS TOP-MARK
FROM STUDENT2
GROUP BY DEPT ID;

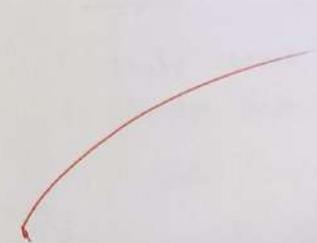
	DEPT ID	TOP-MARK
1	101	90
2	102	95
3	103	88

SELECT DEPT ID, MIN(MARKS) AS LEAST MARK FROM
STUDENT2
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 STUDENT2
 GROUP BY DEPT ID;

	DEPT ID	STU_COUNT
1	101	3
2	102	2
3	103	1



VELTECH	
EX No.	8.2
PERFORMANCE (S)	5
RESULT AND ANALYSIS (S)	15
VIVA VOCE (S)	1
RECORD(S)	
TOTAL (S)	11
SIGN WITHIN	✓ 25/8/2023

Result → Implementation of all aggregate function has
 been performed successfully on a table