Task NO:-3.1	DML commands using clauses, operators and functions in Queries
Date :- 19/08/25	

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

Data Manipulation language (DML):

The DML is used to retrieve, insert and modify database information. These commands will be used by all database users during the routine operation of the database.

DML commands:

1. Insert into: This is used to add records into a relation.

Syntax: insert into <table name> (field1, field2 ... feild_n)
values (data_1, data_2, ... data_n);

Example:- SQL> insert into customer values (238, 'Ram', 'chennai', '986264090'); customer4.

SQL> insert into Customer values (409, 'Rocky', 'vizag', '844180892');

SQL> insert into customer values (112, 'Virat', 'Hyderabad', '704986929');

After inserting:

Customer_ID	Name	address	Ph_no
238	Ram	chennai	986264090
409	Rocky	vizag	844118092
112	Virat	Hyderabad	704986929

2. Update - Set - Where

This is used to update the content of a record in a relation.

Syntax :- SQL>Update relation name Set Field_name1 = data, field-name2 = data, where field_name = data;

Example:- SQL > Update customer Set name = 'kumar' where customer_ID = 409

After updating:

Customer_ID	name	address	Ph_NO
238	Ram	chennai	98626409D
409	kumar	vizag	844118092
112	Virat	Hyderabad	7049869 29

3. Delete - from:

This is used to delete ~~all~~ the records of a relation but it will retain the structure of that relation.

a) Delete - from: This is used to delete all the records of relation.

Syntax: SQL > Delete from ~~customer~~ table_name;

Example: SQL > Delete from customer;

After deleting:

customer_ID	name	address	Ph_NO

b) Delete - from - where: This is used to delete a selected record from a relation.

Syntax: SQL > Delete from relation_name where condition;

Example:- SQL > Delete from customer where name = 'Ram';

After deleting:

Customer_ID	name	address	Ph_NO
409	kumar	vizag	844118092
112	Virat	Hyderabad	7049869 29

5. Truncate

This command will remove the data permanently. But structure will not be removed.

Syntax :- Truncate Table <Table Name>

Example:- Truncate Table Customer;

After truncate:

Customer_ID	name	address	ph_no.

Queries

1. Retrieve a member name starts with letter 'V'.

Query: Select name from bank_account where name like '%V%';

Output:-

Name
Vijay
Vikram
Virat

2. List of Accounts where balance between 10000 and 20000;

Query: select * from bank_account where balance between 10000 and 20000;

output:-

Name	Account_number	Balance	category
Vijay	2345	10000	Savings
Vikram	7890	20000	savings

3. Finding records who has Minimum Balance

Query:- Select min(balance) from bank_account;

output:-

Min (Balance)
10000

4. Finding records who has Balance >= 20000;

Query:- select * from bank_account where balance >=20000;

output:-

Name	Account_number	Balance	Category
Vikram	7890	20000	Savings
Virat	4567	35000	Salary
akash	8987	50000	RD

5. Distinct

Query:- Select distinct Category from Bank_account;

output:-

Category

Savings
Salary
RD

6. Union

Query:- Select name from customer union select name from bank_account;

output:-

Name

Rocky
Virat
Vijay
Vikram
Akash

VEL TECH	
EX NO.	3.1
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	5
RECORD (5)	–
TOTAL (20)	15
SIGN WITH DATE	

29/8/25

Result:- The implementation of DML Commands using clauses, operators and functions in Queries executed successfully

Task No:-3-2

Date:-26/08/25

Aggregate Functions

Aim:- To study and implement aggregate functions (count(), Sum(), Avg(), min(), max() .

Procedure:-

1. Create a table named Bank-Account
2. Insert Sample records
3. Write queries using aggregate functions.
4. Observe and record the output.

Commands with explanation.

1. Count the total number of students

```
Select Count * AS Total_amount from Bank_Account;
```

Output:-

Total_amount
4

2. Find the highest amount in the account.

```
Select max(balance) AS highest_amount from Bank_Account;
```

Output:-

Highest_amount
50000

3. Find the Average amount Marks of ~~students~~ Accounts.

```
select Avg(balance) AS Average_amount from Bank-Account;
```

Output:-

Average_amount
28750

4) Find minimum Amount of the Account

Query: select min(balance) as min_amount from Bank_account;

Output:

min_account
10000

5) Find the total amount in the Bank Account in each category.

Query: Select category, sum(balance) as total_amount from bank_account group by category;

output:-

Category	Total_amount
Savings	30000
Salary	35000
RD	50000

6). Find the average Balance per category ordered by average Balance descending

Query:- Select category, avg(balance) as avg_balance from bank_account group by category order by avg_balance desc;

output:

category	Avg_Balance
RD	50000
Salary	35000
Savings	15000

VEL TECH	
EX NO.	32
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	5
RECORD (5)	—
TOTAL (20)	15
SIGN WITH DATE	26/4/24

Result:- The implementation of Aggregate functions are executed successfully.