Task: 3.1
Date: 19-8-25

# DML Commands using clauses, operations and functions in Queries

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

Data Manipulation language:

insert into: This is used to add records the relation.

syntax:- INSERT INTO table_name (col1, col2,...)
Values (val1, val2,....)

Example:

```
SQL insert into customer values(1, 'John Doe',
        '123.456.789', 'New york', 100.00);

SQL insert into customer values 2, 'smith',
        '987.654.321', 'chicago', 200.00);

SQL insert into customer values 3, 'krish',
        555.123.456', 'America', 50.00)
```

After inserting:

| CUSt-Id | CUSt-Name | Phone-NO | City | Amount Paid |
|---|---|---|---|---|
| 1 | John Doe | 123.456.789 | NewYork | 100.00 |
| 2 | Smith | 987.654.321 | Chicago | 200.00 |
| 3 | krish | 555.123456 | America | 50.00 |

2. UPDATE-SET-WHERE

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

```
Syntax: SQL > update table-name
              SET column = value
              WHERE condition;
```

Example:-

```
SQL > update customer
      SET cust-Phone NO = '8919883456'
      WHERE CUSt-ID = 1;
```

After inserting:

| CUSt-Id | CUSt-Name | Phone-NO | City | Amount Paid |
|---|---|---|---|---|
| 1 | John DOe | 8919883456 | NewYork | 100.00 |
| 2 | Smith | 987654321 | Chicago | 200.00 |
| 3 | krish | 555123456 | America | 50.00 |

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 all the records of relation

Syntax SQL> Delete from table-name;

Example : SQL> Delete from customer;

After deleting.

| Cust-id | Cust-Name | Phone-No | City | Amount paid |
|---|---|---|---|---|
| | | | | |

b) Delete-from-where: This is used to delete a records select of relations.

Syntax: SQL> Delete from relation-name Whi condition;

Example:

SQL> Delete from Customer.

WHERE Cust-ID=2;

After Deleting:

| cust-Id | cust-Name | Phone-No | City | Amount Paid |
|---|---|---|---|---|
| 1 | John Doe | 89198835456 | Newyork | 100.00 |
| 3 | krish | 555.123.456 | America | 50.00 |

STRUNCATE:

This commond will remove the data peremenantly, But structure will not be removed.

Synatx: Truncate table <Table Name)

Example: Truncate table Customer;

| cust-Id | cust-Name | Phone-No | City | Amount Paid |
|---|---|---|---|---|
| | | | | |

Distinct

Query: Select Distinct cust-city
From Customer;

Output: cust-city
New York
Chicago
Newyork

Union:

Query:-
```
select cust_Name As Name from
customer union select mobile_Name
As Name from mobile;
```

Output:-
```
Name
John
Alice
Ravi
Meena.
```

| VEL TECH | |
|---|---|
| EX NO. | 3.5 |
| PERFORMANCE (5) | 5 |
| RESULT AND ANALYSIS (5) | 5 |
| VIVA VOCE (5) | 5 |
| RECORD (5) | - |
| TOTAL (20) | 15 |
| SIGN WITH DATE | 19/8/24 |

Result:

The task to implement the DML commands using clauses, operators and functions in queries executed successfully

Task :3.2
Date:26-8-25

# Aggregate Functions

Aim: To Study and implement aggregate Functions (count (), sum(), AVg(), Min(), Max()) on a Sample mobile phone database

Procedure:

1. Create a table named mobile phone
2. Insert sample records
3. Write queries using aggregate functions.
4. Observe and record output.

Commands with Explanation

1) Count the total number of mobile phones

```
SELECT COUNT (*) AS Total-mobile phones from
mobile phone.;
```

Output:- Total_mobile phones : 3

2) Find the highest purchase obtained by a mobile phone

```
SELECT MAX(Purchase) AS highest_Purchase
FROM Mobile phone;
```

Output:- Highest_Purchase : 30000

3) Find the average amount of mobile phone

```
SELECT AVG(amount) AS Average_amount
from mobile phone;
```

output: Average_amount: 15000

4) Find Minimum purchase among mobile phone in the brand

```
SELECT MIN(Purchase) AS MIN-Brand Purchase,
from Mobile phone.
WHERE Mobile phone = Redmi;
```

5) Find the total amount in the mobile phone in each category. Brand.

```
SELECT Brand, Sum(amount) as total_
                              Mobile phone
amount. from purchase group by Brand;
```

~~fro~~

output:

| Brand | Total_amount |
|---|---|
| Realme | 30,000 |
| Redmi | 15,000 |
| Vivo | 25,000 |

6) Find the average amount per brand ordered by average ~~amount~~ decending.

```
SELECT Brand ,avg (amount) as Avg_amount
From mobile phones group by brands.
ordered by avg-amount dec;
```

OUTPUT:

```
Brand      Avg_amout
vivo       251000
Redmi      151000
Realme     30,000.
```

| VEL TECH | |
|---|---|
| EX NO. | 3.2 |
| PERFORMANCE (5) | 5 |
| RESULT AND ANALYSIS (5) | 5 |
| VIVA VOCE (5) | 5 |
| RECORD (5) | - |
| TOTAL (20) | 15 |
| SIGN WITH DATE | 26/8/25 |

Result: Thus the implementation of Aggregate functions executed successfully.