25-08-25

TASK 3.1

DML Commands using clauses, operators and Functions in queries

Aim:- To implement DML commands using clauses, operators and functi in queries

DML (Data Manipulation Language):-

The DML is used to retrive, insert and modify database information. These commands will be used by all database informatio

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

Syntax: insert into <table name> (field1, field2 ---- field n) values (data1, data2, ---- data n);

Example: SQL> insert into customer values (SSN, NAME, Email, Phone, country) values (CUSTOMER03, Advaita, Advaita@gmail.com, 9840079834, India)

After inserting:-

SSN	Name	Email	Phone	Country
CUSTOMER01	SELENA	Selina@gmail.com	9000001234	Korea
CUSTOMER02	Allin	Allin@gmail.com	98401132 1	China
CUSTOMER03	Advaita	Advaita@gmail.com	9840079834	India

2. update - set - where:

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

syntax:- SQL> update relation set Field_name1 = data, field_name2 data, where field_name = data;

Example:- SQL> update set name = 'Allin' where SSN = CUSTOMER02

SSN	NAME	Email	Phone	COUNTRY
CUSTOMER01	SELENA	Selina@gmail.com	9000.1234	Korea
CUSTOMER02	Allin	Allin@gmail.com	98401132 1	china
CUSTOMER03	Advaita	Advaita@gmail.com	9840079834	India

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 table_name;

Example: SQL> Delete from customer;

After deleting:

SSN	NAME	E-Mail	Phone	COUNTRY

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 = 'Albin';

After deleting:

SSN	NAME	E-Mail	Phone	COUNTRY
CUSTOMER01	SELENA	selena@gmail.com	9000001234	Korea
CUSTOMER03	Advaita	Advaita@gmail.com	8400798334	India

4.) Truncate:-

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

:- Truncate Table < Table name >

:- Truncate Table customer;

After Truncate

SSN	NAME	EMail	Phone	COUNTRY

Queries:-

1. Retrive a number name starts with 's'

query: select name from customer where name like '%s%';

output: Name
Selena

2. List of Accounts where fee between 5000 and 7000:

query: Select * from ~~branch~~ account where fee between 5000 and 7000;

output

Name	Customer_number	fee	category
Advaita	298817	6000	Twin Room

3.) Finding records who has minimum fee

Syntax:
Select min(fee) from customer_account;

output: min fee
Albin

4.) Finding Records who has fee >= 5000

Select * from customer_account where fee >= 5000;

output:

Name	Customer_number	fee	category
Advaita	298817	6000	Twin Room
Tarun	950107	4000	Quad Room

5.) Distinct :

Syntax: Select distinct category from Customer_Account;

output :-

Category
Twin Room
Quad Room
Triple Room

6.) union :

Select name from Customer union select name from Hotel_account;

output:

Name
Aldin
Advaita
Rocky

VEL TECH	
EX No.	3.
PERFORMANCE (5)	5
RESULT AND ANALYSIS (3)	5
VIVA VOCE (3)	5
RECORD (4)	
TOTAL (15)	15
SIGN WITH DATE	1818

1818

Result: The implementation of Dml commands using clauses is completed successfully.

25.05.25

TASK 3.2

Aggregate Functions

Aim: To study and implement aggregate functions (count(), sum(), Avg(), min(), Max(),

Procedure:

1) Create a table named Customer_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~~ Customers

Select count * As total_amount from Customer_account;

output: Total_Amount
4

2.) Find the highest fee in the account

Select max(fee) As highest_amount from Customer_Account;

Highest Fee

output: 9500

3.) Find the average fee of the customer

Select Avg(fee) As Average_Amount from Customer_Account;

output: 5000

4.) Find Minimum Amount of Account

Select min(fu) as Min_amount from customer_Account;

output: Min account

2500

5.) Find the total amount of fu in the Bankkner_Account;

select category, sum(fu) as total_amount from bank_account group by category;

output:

Category	Total_amount
Twin Room	4000
Triple Room	4500
quad Room	9500

6.) Find the average fu per category ordered by average fu

select category, avg(fu) as Avg-fu from customer_account group by category order by avg_fu;

output:

Category	Avg_amount
Twin Room	3000
Triple Room	4000
quad Room	6000

VELTECH	
EX No.	3.2
PERFORMANCE (5)	5
RESULT AND ANALYSIS (3)	5
VIVA VOCE (3)	5
RECORD (4)	—
	15
	25/8

25/8

Result: The implementation of Aggregate functions is executed successfully.