|  |  |
| --- | --- |
| **Name** | Adwait S Purao |
| **UID no.** | 2021300101 |
| **Experiment No.** | 3 |

|  |  |
| --- | --- |
| **AIM:** | DML Commands Database Manipulation |
| **Program 1** | |
| **PROBLEM STATEMENT :** | . Writing queries on created tables using various DML commands in MySQL. |
| **Theory :** | DML is an abbreviation for Data Manipulation Language.  • Data Manipulation Language or DML represents a collection of programming languages explicitly used to make changes in the database, such as:  • CRUD operations to create, read, update, and delete data using the INSERT, SELECT, UPDATE and Delete commands.  • DML commands are often part of a more extensive database language, for instance, SQL (Structure Query Language). These DML commands may have a specific syntax to manage data in that language.  • DML Commands provide a way to read, update, delete, or merge data precisely. In the beginning, DML commands were part of computer programs only, but with the popularity of SQL, they have now become a part of database management.  • DML Commands provide a way to read, update, delete, or merge data precisely. In the beginning, DML commands were part of computer programs only, but with the popularity of SQL, they have now become a part of database management.  • Data Manipulation Languages (DML) have two primary classifications: Procedural and Non-procedural programming (declarative programming).  **List of DML Commands in SQL**:  Here is a shortlist of all DML commands and their specific functions in the SQL programming language.  **1)SELECT**: Command to fetch data or values from the database  **2)INSERT**: Command to add new or fresh value to the database  **3)UPDATE:** Command to change or update the present/existing data  to a newer value inside the database  **4)DELETE**: Command to remove or delete the values or data  information from the database’s current table  **SELECT DML Command**  SELECT is the most important data manipulation command in Structured Query Language. The SELECT command shows the records of the specified table. It also shows the particular record of a particular column by using the WHERE clause.  **Syntax of SELECT DML command:**  SELECT column\_Name\_1, column\_Name\_2, ….,columnName\_  FROM Name\_of\_table;  SELECT \* FROM table\_name; -- To select all columns in table  **INSERT DML Command**  INSERT is another important data manipulation command in Structured Query Language, allowing users to insert data in database tables.  **Syntax of INSERT Command:**  INSERT INTO TABLE\_NAME ( column\_Name1 , column\_NameN )  VALUES (value\_1, value\_2, value\_3, value\_N ) ;  **UPDATE DML Command**  UPDATE is another important data manipulation command in Structured Query Language, which allows users to update or modify the existing data in database tables.  **Syntax of UPDATE Command:**  UPDATE Table\_name  SET [column\_name1= value\_1, …,column\_name value\_]  WHERE CONDITION;  Here, 'UPDATE', 'SET', and 'WHERE' are the SQL keywords, and  'Table\_name' is the name of the table whose values you want to update.  **DELETE DML Command**  DELETE is a DML command which allows SQL users to remove single or multiple existing records from the database tables. This command of Data Manipulation Language does not delete the stored data permanently from the database. We use the WHERE clause with the DELETE command to select specific rows from the table.  **Syntax of DELETE Command:**  DELETE FROM Table\_Name  WHERE condition;  **DML Statements and Transactions**  DML Statements:  • Database table data can be added, changed, or deleted using Data Manipulation Language (DML) statements.  • DML Statements access the data and process/change the existing tables.  • In the SQL environment, DML statements are entered after the SQL> prompt  • DML statements are entered in the Worksheet in the SQL Developer environment. To access and manipulate data, the SQL Developer Connections frame and tools can be used.  • The effect of a DML statement is not permanent until the transaction that includes it is committed.  **Transaction control statements:**  • A transaction is a set of one or multiple SQL statements that the DBMS treats as one unit (single command): either all of the statements are executed or none of them are.  • Transactions are required when writing code for business processes that require multiple operations to be performed as a unit simultaneously.  For example, when a team leader (TL) quits the company, a row has to be inserted into the JOB\_ HISTORY table to show when the team leader left, and the value of TL\_ID in the has to be updated against each of his team members in the EMPLOYEES table. To execute this process in a business application, the ‘INSERT’ and ‘UPDATE’ DML commands must be combined into a single transaction. |
| **Queries** | Code:  CREATE DATABASE Hotel;  use Hotel  CREATE TABLE Hotel (  H\_Name Varchar(120) Not Null,  H\_ID int Primary key,  H\_Address Varchar(200) Not Null,  H\_Num\_Emp int,  H\_vacancies int  ) ;  CREATE TABLE Employee (  E\_Name Varchar(70),  E\_Type Varchar(50),  E\_ID int primary key,  H\_ID int,  foreign key(H\_ID) references Hotel(H\_ID),  LastName varchar(255),  FirstName varchar(255),  Address varchar(255),  City varchar(255),  E\_Contact int,  E\_Salary int not null check(E\_salary>0)  );  create table Room(  R\_no int primary key,  R\_vacany boolean default true,  R\_price int not null,  R\_type varchar(30),  H\_ID int references Hotel(H\_ID)  );  create table Reservation(  Reservation\_no int primary key,  R\_intime datetime not null,  R\_outtime datetime,  Amount int not null check(Amount>0),  R\_no int references Room(R\_no),  C\_ID int references Customer(C\_ID)  );  CREATE TABLE Customer(  C\_Id int primary key,  C\_Name Varchar(50) Not Null,  Reservation\_no int,  C\_Age int ,  C\_Address Varchar(70) Not Null,  C\_contact int,  C\_cin\_time int,  C\_cout\_t int,  foreign key(Reservation\_no) references Reservation(Reservation\_no)  );  alter table hotel rename hotel\_info;  insert into hotel\_info values("marriot",1234,"Pune",3456,5);  insert into hotel\_info values("The Plaza",2345,"New York ",4567,7);  insert into hotel\_info values("Claridge's",3456,"London",5678,7);  insert into hotel\_info values("Raffles",5678,"Singapore",6789,8);  insert into hotel\_info values("Taj Mahal Palace",6789,"Mumbai ",7890 ,9);  insert into hotel\_info values("Beverly Hills Hotel",8970,"Los Angeles",8907,2);  insert into employee values("Adwait Purao","Permanent",1,1234,"Purao","Adwait","Kurla","Mumbai",12345 ,10000);  insert into employee values("Ram Kumar","Permanent",2,1234,"Kumar","Ram","Kalina","Mumbai",12346,20000);  insert into employee values("Akshay Kumar","Temporary",3,3456,"Kumar","Akshay","Ram chowk","Ramgad",12347,30000);  insert into employee values("Ranbir Kapoor","Permanent",4,2345,"Kapoor","Ranbir","Roopnagar","Agra",12348,40000);  insert into employee values("Angelina Jolie","Permanent",5,8970,"Jolie","Angelina","Beverly Hills","Los Angeles",12349,50000);  alter table customer modify C\_cin\_time time ;  alter table customer modify C\_cout\_t time ;  alter table reservation modify R\_intime time ;  alter table reservation modify R\_outtime time ;  insert into reservation values(1,"12:56:23","16:56:23",1000,12,1234);  insert into reservation values(2,"13:54:43","19:26:13",2000,13,1235);  insert into reservation values(3,"11:24:41","20:55:53",1500,14,1236);  insert into reservation values(4,"22:21:45","16:25:33",2500,15,1237);  insert into customer values(1234,"Sam Vaz",1,34,"Ghatkopar",123456,"12:56:23","16:56:23");  insert into customer values(1235,"Ram Sharma",2,44,"Ghansoli",123457,"13:54:43","19:26:13");  insert into customer values(1236,"Sachin Tendulkar",3,50,"Colaba",123458,"11:24:41","20:55:53");  insert into customer values(1237,"Virat Kohli",4,30,"Dadar",123459,"22:21:45","16:25:33");  insert into room values(12,1,1000,"Basic",1234);  insert into room values(13,0,2000,"Deluxe",2345);  insert into room values(14,1,1500,"Suite",5678);  insert into room values(15,0,2500," Luxury Suite",6789);  select \* from hotel\_info;  select \* from employee;  select \* from room;  select \* from reservation;  select \* from customer;  Original tables  **1)Table hotel\_info**    **2)Table Employee**    **3)Table Room**    **4)Table Reservation**    **5)Table Customer**    **Query 1(Delete command)**  **delete from employee where E\_ID=2;**  **1)Before execution**    **2)After execution**    **Query 2(Where command)**  **select \* from customer**  **where c\_id=1234;**  **1)Before execution**    **2)After execution**    **Query 3(Or command)**  **use hotel;**  **select \* from employee**  **where E\_ID=1 or E\_ID=2 ;**  **1)Before execution**    **2)After execution**    **Query 4(Update command)**  **update customer**  **set C\_Name="Ramesh Verma",C\_Address="Dharavi"**  **where C\_ID=1234;**  **1)Before execution**    **2)After execution**    **Query 5(Not command)**  **use hotel;**  **select \* from reservation**  **where not Reservation\_no=2;**  **1)Before execution**    **2)After execution**    **Query 6(Wild card command)**  **select\* from customer**  **where c\_name like 'ram%';**  **1)Before execution**    **2)After execution**    **use hotel;**  **SELECT \* FROM employee**  **WHERE e\_name LIKE '%ha%';**  **1)Before execution**    **2)After execution**    **Query 7(Alter command command)**  **alter table hotel\_info rename Info\_Hotel;**  **1)Before execution**    **2)After execution**    **Query 8(Alter command to add a column)**  **use hotel;**  **alter table employee**  **add district varchar(20);**  **1)Before execution**    **2)After execution**    **Query 9(And command)**  **use hotel;**  **SELECT \* FROM room**  **WHERE r\_vacany=1 AND r\_price=1000;**  **1)Before execution**    **2)After execution**    **Query 10(Order by command)**  **use hotel;**  **SELECT \* FROM room**  **where r\_price>1200**  **order by r\_no;**  **1)Before execution**    **2)After execution**    **Query 11(Drop command)**  **use hotel;**  **alter table employee**  **drop district**  **1)Before execution**    **2)After execution**    **Query 12(Selecting a particular column)**  **use hotel;**  **select e\_type from employee;**  **1)Before execution**    **2)After execution**    **Query 12(Multiple OR and AND commands)**  **use hotel;**  **select\*from employee**  **where (e\_type='Permanent' and e\_name like 'A%') or e\_id=5;**  **1)Before execution**    **2)After execution**    **Query 13(Wild card command)**  **use hotel;**  **select\*from employee**  **where e\_name like '\_k%'**  **1)Before execution**    **2)After execution**    **Query 14(Wild card command)**  **use hotel;**  **select\*from employee**  **where e\_name not like 'a%'**  **1)Before execution**    **2)After execution**    **Query 15(Delete a row )**  **delete from employee where e\_id=4;**  **1)Before execution**    **2)After execution** |
| **Conclusion**  **This experiment helped me learn various commands of DML(Data Manipulation language) like select \* from, set, where, update, alter, and, or, not, add, delete, like, drop,% etc. Using these commands database for Hotel system is prepared. The experiment helped to learn about the handling the data and to filter and retrieve the data according to the need. I also learned how to operate MySQL Workbench software.** | |