AIM:	DML Commands Database Manipulation
PROBLEM STATEMENT :	Write queries on the previous database created in experiment 2.
Theory :	

DML

DML is an abbreviation of Data Manipulation Language. The DML commands in Structured Query Language change the data present in the SQL database. We can easily access, store, modify, update and delete the existing records from the database using the DML command

Following are the four main DML commands in SQL:

- SELECT Command
- INSERT Command
- UPDATE Command
- DELETE Command

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, ….., column_Name_N
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_nameN = value_N]
```

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;

Queries

Using created database. The following are tables in the previous database

Customer Table:

CustomerID	Customername	phoneno	address	Email	orderID
10	sahil	87971838	vashi	sahil.ved@gmail.com	1345
11	vansh	879718398	unkown	vansh@gmail.com	1356
12	yash	87971834	andheri	yash@gmail.com	1398
13	me	87971338	SOBO	pranay@gmail.com	1340
14	you	8797138	My heart	tellyourname@gmail.com	1349
15	and our loneliness	6969696	place not mentionables	loneliness.hatao@gmail.com	1369
NULL	NULL	NULL	NULL	NULL	NULL

Delivery Person Table:

employeeID	delname	phoneno	Rating	Shift_hr	orderID
20	ram	87247923	2	8	1345
21	sham	8727843	3	8	1356
22	kam	872479823	4	8	1398
23	rod	87247983	5	8	1340
24	red	8722387	3	8	1349
25	kum kaj	8724823	3	8	1369
NULL	NULL	NULL	NULL	NULL	NULL

Order Details Table:

orderID	CustomerID	Resturname	employeeID	price	payment_stat...
1340	13	maj hotel	23	2000	completed
1345	10	taj hotel	20	20000	completed
1349	14	Roj hotel	24	20	completed
1356	11	raj hotel	21	21	completed
1369	15	Raz hotel	25	69	fully completed
1398	12	kaj hotel	22	200	completed
NULL	NULL	NULL	NULL	NULL	NULL

Restaurants Details Table:

ResturID	Resturname	Rating	Varieties	Typ_of_food	Timing	orderID	location
1	taj hotel	5	Indian	Poha	1:00PM	1345	mumbai
2	raj hotel	4	conti	Baked Pesto Pasta.	10:00PM	1356	mumbai
3	kaj hotel	4	maxican	Machaca	11:00PM	1398	mumbai
4	maj hotel	3	russian	Kasha	12:00PM	1340	mumbai
5	Roj hotel	2	japp	Kinpira Gobo	1:00AM	1349	mumbai
6	Raz hotel	1	ghar	Chapal	Roj	1369	mumbai
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Query 1: To drop column location in restaurant table

```
-- Query 1
select * from restur;
ALTER table Restur
Drop location;
select * from restur;
```

Before executing query :

ResturID	Resturname	Rating	Varieties	Typ_of_food	Timing	orderID	location
1	taj hotel	5	Indian	Poha	1:00PM	1345	mumbai
2	raj hotel	4	conti	Baked Pesto Pasta.	10:00PM	1356	mumbai
3	kaj hotel	4	maxican	Machaca	11:00PM	1398	mumbai
4	maj hotel	3	russian	Kasha	12:00PM	1340	mumbai
5	Roj hotel	2	japp	Kinpira Gobo	1:00AM	1349	mumbai
6	Raz hotel	1	ghar	Chapal	Roj	1369	mumbai
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

After executing query :

ResturID	Resturname	Rating	Varieties	Typ_of_food	Timing	orderID
1	taj hotel	5	Indian	Poha	1:00PM	1345
2	raj hotel	4	conti	Baked Pesto Pasta.	10:00PM	1356
3	kaj hotel	4	maxican	Machaca	11:00PM	1398
4	maj hotel	3	russian	Kasha	12:00PM	1340
5	Roj hotel	2	japp	Kinpira Gobo	1:00AM	1349
6	Raz hotel	1	ghar	Chapal	Roj	1369
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Query 2: To show only that rows whose rating is above 3 and they are sorted with respect to rating

```
-- Query 2
select * from restur
where Rating>3
order by Rating ;
```

Result:

ResturID	Resturname	Rating	Varieties	Typ_of_food	Timing	orderID
2	raj hotel	4	conti	Baked Pesto Pasta.	10:00PM	1356
3	kaj hotel	4	maxican	Machaca	11:00PM	1398
1	taj hotel	5	Indian	Poha	1:00PM	1345
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Query 3: Add Location column in Restaurant table. Update the location to Mumbai wherever restaurant id is less than 6.

```
-- Query 3
ALTER table Restur
ADD location varchar(225);
Update Restur
set location='mumbai'
where ResturID<=6;
select * from Restur;
```

Result:

ResturID	Resturname	Rating	Varieties	Typ_of_food	Timing	orderID	location
1	taj hotel	5	Indian	Poha	1:00PM	1345	mumbai
2	raj hotel	4	conti	Baked Pesto Pasta.	10:00PM	1356	mumbai
3	kaj hotel	4	maxican	Machaca	11:00PM	1398	mumbai
4	maj hotel	3	russian	Kasha	12:00PM	1340	mumbai
5	Roj hotel	2	japp	Kinpira Gobo	1:00AM	1349	mumbai
6	Raz hotel	1	ghar	Chapal	Roj	1369	mumbai
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Query 4: Show the restaurant name whose last word is hotel and has 4 star

rating

```
-- Query 4
-- show the resturname whose last word is hotel and has 4 star rating
select ResturID, Resturname, Rating
from Restur
where Resturname like '%hotel' and rating=4;
```

Result:

ResturID	Resturname	Rating
2	raj hotel	4
3	kaj hotel	4
NULL	NULL	NULL

Query 5: Show customer whose customer id is more than or equal 13

```
-- Query 5
-- show customer whose customer id is more than or equal 13
select CustomerID, phoneno, orderID
FROM Customers
where CustomerID>=13
Order by orderID;
```

Result:

CustomerID	phoneno	orderID
13	87971338	1340
14	8797138	1349
15	6969696	1369
NULL	NULL	NULL

Query 6: ADD another column in Order details table which takes quantity and show total price

```sql
-- Query 6
-- ADD another column in Order details table which takes quantity and show total price
Alter Table Orderde
ADD Quantity int default 1;
update Orderde
set Quantity =2
where orderID=1369;
select orderID, CustomerID, price*Quantity as Total_price
from orderde
order by CustomerID;
```

Result:

orderID	CustomerID	Total_price
1345	10	20000
1356	11	21
1398	12	200
1340	13	2000
1349	14	20
1369	15	138

Conclusion

From this experiment we conclude that we can write query for different situation. We also learned to use keywords like ALTER , SELECT, ORDER BY, LIKE, WHERE, UPDATE, etc.