EXPERIMENT NO: 4
DML COMMANDS

AIM:

To solve queries using DML Commands

THEORETICAL BACKGROUND:

Data manipulation languages are used to query and manipulate existing objects like tables.
The DML commands are:

1. Insert

    Syntax: insert into <table name> values (a list of data values)

2. Select

    Syntax: Select <column _name> from <table name>;

    a. Selecting distinct rows

        Syntax: Select distinct <column name> from <table name>;

    b. Select with where clause

        Syntax: Select <columns> from <table name> where <conditions> [order by <column>]

3. Update

    Syntax: update <table name> set field = value [where <condition>];

4. Delete

    Syntax: Delete from <table name> [where <conditions>];

Questions:

1. Insert a single record in the following tables

```
Customer_db, item_db, order_db
insert into customer_db values(1000, 'Ajith', 'Karthika', 'N C C Road', 2437189);
insert into order_db values(500, 1000, 143, '9-mar-09', '31-dec-99', '9-mar-09',
'R');
insert into item_db values(143, 'RSS', 1, 10000);
```

2. Insert multiple records in the following tables

```
Customer_db, item_db, order_db
insert into customer_db values(&cust_id, '&cust_name', '&housename', '&street',
&phoneno);

Enter value for cust_id: 1006
Enter value for cust_name: Devi
Enter value for housename: Karthika
        Enter value for street: N C C Road
Enter value for phoneno: 2438959

old 1: insert into customer_db values(&cust_id, '&cust_name', '&housename',
'&street',&phoneno)

new 1: insert into customer_db values(1006, 'Devi', 'Karthika', 'N C C Road',
2438959)
```

3. update the delivery date in the field order_db as delivery_date= date of order +10;

```
update order_db set delivery_date=order_date+10;
```

4. Display the records in ascending order by delivery date in order_db

```
select * from order_db order by delivery_date asc;
```

5. Display the records in descending order by unit price in item_db

```
select * from item_db order by unit_price desc;
```

6.Display the records in order_db where order_id >1 and <7

```
select * from order_db where order_id between 1 and 7;
```

7. Delete the order from order_db where expiry date < current date

```
delete from order_db where expiry_date<'11-aug-09';
```

8.Duplicate the table of item_db as test_db

```
create table test_db as select * from item_db;
```

9. Select customer name from customer_db whose anme starts with alphabet A. select

```
cust_name from customer_db where cust_name like 'A%';
```

10. Display order_id and cust_id from order_db

```
select order_id, cust_id from order_db;
```

11. Truncate the table item_db

```
truncate table item_db;
```

12. Display order_id , customer_id from order_id whose month of delivery is current month.

```
select order_id, cust_id from order_db where delivery_date like '%-AUG-%';
```

13 .Display orderid and customerid from order database

```
select orderid,custid from order1;
```

14. Truncate the table

```
truncate table item_db;

select * from item_db;
```

15.Display the orderid,customerid from orders where the month of delivery is the current month.

```
update order1 set deliverydate='01-JUL-2015' where deliverydate='17-jan-15'

select * from order1;

select * from order1 where extract(month from
        deliverydate)=to_char(SYSDATE,'MM');
```

16. Find the average of order quantity with item code =256.

```
insert into order1 values(150,4677,256,102,'29-may-2015','R','2-jul-2015');

insert into order1 values(151,4299,256,6,'31-may-2015','C','2-jul-2015');

select * from order1;

select avg(orderquantity) from order1 where itemcode=256;
```

17. What is the item with the highest unit price?

```
select * from item_db where unitprice=(select max(unitprice) from item_db);
```

18. What is the cheapest item?

```
select * from item_db where unitprice=(select min(unitprice) from item_db);
```

19. How many orders were made for item with itemcode 250

```
select count(*) from order1 where itemcode=256
```

20. How many items have unit price between 100 and 200.

```
select count(*) from item_db where unitprice>100 and unitprice<200;
```

21. What is the average unit price?

```
select avg(unitprice) from item_db;
```

22. Display the orderid and delivery date with heading prdercode and date of delivery.

```
select orderid as order_code,deliverydate as dateofdelivery from order1;
```

23. Display the name of customers which contain occurances of a and j in the same name.

```
insert into customer values(4618,'arjun','abc','xyz',9447455934)

insert into customer values(4186,'Jasmine','abc','xyz',123456789);

select cust_name from customer where cust_name like '%a%j%' or cust_name like '%J%a%'
```

24 .What is the length of shortest name?

```
select(length(min(CUST_NAME))) from customer;
```

25. Create table deliverdb with same structure as orderdb.

```
create table deliver_db as select * from order1;

select * from deliver_db;
```

26. Display the records of tables orderdb deliverydb using union operator

```
insert into deliver_db values(176,4911,1001,5,'05-JUL-15','C','29-JUL-15');

select * from deliver_db;

select * from deliver_db union select * from order1
```

27. Display the records having order id common for both tables orderdb and deliverydb using intersect operator

```
select orderid from deliver_db intersect select orderid from order1;
```

28. Display the orderid of the order that is not delivered yet.

```
insert into deliver_db values(192,4299,1001,2,'08-JUL-15','C','12-JUL-15');

select * from deliver_db;

select orderid from deliver_db where deliverydate>SYSDATE
```

RESULT:

The query was executed successfully and output was verified.