Lab 4
DML – Insert, Update, Delete, Transactions - Commit, Rollback, Savepoint

DML: A data manipulation language is a language that enables users to access or manipulate data as organized by the appropriate data model. DML is used to manipulate the existing data in the database objects (insert, select, update, delete).

DML Commands:

1. INSERT
2. SELECT
3. UPDATE
4. DELETE

INSERT Statement

The INSERT INTO statement of SQL is used to insert a new row in a table. There are two ways of using INSERT INTO statement for inserting rows:

1. **Only values:** First method is to specify only the value of data to be inserted without the column names.

Syntax:

INSERT INTO table_name VALUES (value1, value2, value3,……,valueN);

2. **Column names and values both:** In the second method we will specify both the columns which we want to fill and their corresponding values as shown below:

Syntax:

INSERT INTO table_name (column1, column2, column3,..) VALUES (value1, value2, value3,..);

3. **To insert multiple rows in a table:**

   INSERT INTO table_name(Column1,Column2,Column3,.......) VALUES
   (Value1, Value2, Value3,.....),
   (Value1, Value2, Value3,.....),

(Value1, Value2, Value3,.....), ... ;

Using SELECT in INSERT INTO Statement

SELECT statement with INSERT INTO statement to copy rows from one table and insert them into another table.

1. **Inserting all columns of a table:** We can copy all the data of a table and insert into in a different table.
   Syntax:
   INSERT INTO first_table SELECT * FROM second_table;
2. **Inserting specific columns of a table:** We can copy only those columns of a table which we want to insert into in a different table.
   Syntax:
   INSERT INTO first_table(names_of_columns1) SELECT names_of_columns2 FROM second_table;
3. **Copying specific rows from a table:** We can copy specific rows from a table to insert into another table by using WHERE clause with the SELECT statement. We have to provide appropriate condition in the WHERE clause to select specific rows.
   Syntax:
   INSERT INTO table1 SELECT * FROM table2 WHERE condition;

SELECT Statement

SELECT Command is used to retrieve the records from the table. It is one of the most commonly used commands while working with the databases because at the end-user wants to retrieve the results after any operation be it update, delete or modify.

Syntax:

SELECT [column1, column2, column3, ……., columnN] from [table_name]
[where condition1 [AND [OR]] condition2..........;

Comparison Operators used in SELECT statement:

Operator	Description
=	Equal
<=>	Equal (Safe to compare NULL values)

<>	Not equal
!=	Not Equal
>	Greater Than
>=	Greater Than or Equal
<	Less than
<=	Less Than or Equal
IN ( )	Matches a value in a list
NOT	Negates a condition
BETWEEN	Within a range (inclusive)
IS NULL	NULL value
IS NOT NULL	Non-NULL value
LIKE	Pattern matching with % and _
EXISTS	Condition is met if subquery returns at least one row

Logical Operators:

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td>AND, &&</td><td>Logical AND</td></tr>
<tr><td>NOT, !</td><td>Negates value</td></tr>
<tr><td>OR, ||</td><td>Logical OR</td></tr>
<tr><td>XOR</td><td>Logical XOR</td></tr>
</table>

UPDATE Statement

The UPDATE statement is used to modify data in a table.

Syntax:

UPDATE table_name SET column=value, column1=value1,... WHERE someColumn=someValue;

DELETE Statement

The DELETE FROM statement is used to delete existing records from a database table.

Syntax:

DELETE FROM tableName WHERE someColumn = someValue;

To delete all the records:

Syntax:

- DELETE FROM tablename;
- DELETE * FROM tablename;

DCL (Data Control Language)

DCL includes commands such as GRANT and mostly concerned with rights, permissions and other controls of the database system.

- GRANT - allow users access privileges to the database
- REVOKE - withdraw users access privileges given by using the GRANT command

GRANT:

Syntax:

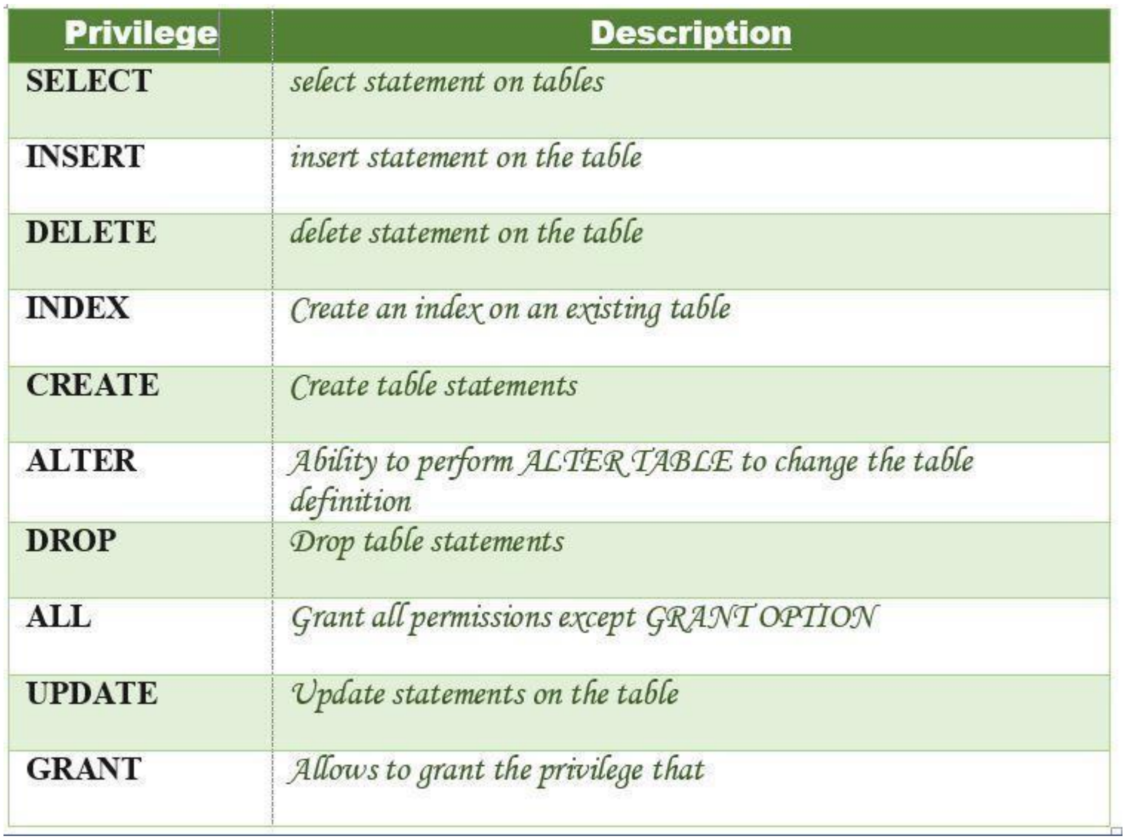
GRANT privileges_names ON object TO user;

Parameters Used:

Privileges_name: These are the access rights or privileges granted to the user.

Object: It is the name of the database object to which permissions are being granted. In the case of granting privileges on a table, this would be the table name.

User: It is the name of the user to whom the privileges would be granted.



Privilege	Description
SELECT	*select statement on tables*
INSERT	*insert statement on the table*
DELETE	*delete statement on the table*
INDEX	*Create an index on an existing table*
CREATE	*Create table statements*
ALTER	*Ability to perform ALTER TABLE to change the table definition*
DROP	*Drop table statements*
ALL	*Grant all permissions except GRANT OPTION*
UPDATE	*Update statements on the table*
GRANT	*Allows to grant the privilege that*

REVOKE:
Revoke command withdraw user privileges on database objects if any granted. When a privilege is revoked from a particular user U, then the privileges granted to all other users by user U will be revoked.
Syntax:
REVOKE privilege_name ON object_name FROM {user_name | public | role_name}

TCL (Transaction Control Language)

TCL deals with a transaction within a database.

- COMMIT - commits a Transaction
- ROLLBACK - rollback a transaction in case of any error occurs
- SAVEPOINT - to roll back the transaction making points within groups

COMMIT:
Syntax: COMMIT;

ROLLBACK:
Syntax: ROLLABACK;

SAVEPOINT:
Syntax:
SAVEPOINT SAVEPOINT_NAME;

creates points within the groups of transactions in which to ROLLBACK.
A SAVEPOINT is a point in a transaction in which you can roll the transaction back to a certain point without rolling back the entire transaction.
Syntax:
ROLLBACK TO SAVEPOINT_NAME;

Demo Problem: e-Bike Dealership

Solution:

INSERT Statements:

1.a. Using any form insert query, the above data can be inserted into respective tables.

1.b. Mass insert can be done as below

i. Create a .csv file containing the data (for example create ‘example.csv’ containing the data for table ‘Example’

ii. When table ‘Example’ is created within in a database ‘My_DB’. Then, we see a corresponding directory ‘My_DB’ under directory ‘/opt/lampp/var/mysql/’ (For Linux) or ‘\xampp\mysql\data’ (for Windows) or ‘XAMPP/xamppfiles/var/mysql’ (for Mac)

iii. Place example.csv to the directory ‘My_DB’

iv. Run the command

```
LOAD DATA INFILE "example.csv" INTO TABLE Example
COLUMNS TERMINATED BY ','
OPTIONALLY ENCLOSED BY '"'
ESCAPED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 LINES;
```

Note: When the csv file contains the column names on top, the top 1 line will be ignored as per the above command

1.c Easier method would be to

1. In phpMyAdmin dashboard, select the table you want to insert
2. Select Import from the top menu and follow instructions to import the respective .csv file into the table.

Dealer Table:

1.a. Using any form of insert query

```
Insert into Dealer values(001,’Ajit’,’Brigade
Road’,’Bangalore’,’Karnataka’,560001);

Insert into Dealer (Dealer_ID,Name,Street,City,State,Pin) values
(002,’Rudra’,’Bhashyam Circle’,’Bangalore’,’Karnataka’,560010);

Insert into Dealer (Dealer_ID,Name,Street,City,State,Pin) values
(003,’Arjun’,’Bapuji Nagar’,’Bangalore’, ’Karnataka’,560026),
(004,’Mohamad’,’Chickpet’,’Bangalore’, ’Karnataka’,560053),
(005,’Nirmala’,’Domlur’,’Bangalore’, ’Karnataka’,560071),
(006,’Raghu’,’HSR Layout’,’Bangalore’, ’Karnataka’,560102),
(007,’Likith’,’Hosur Road’,’Bangalore’, ’Karnataka’,560030),
(008,’Kiran’,’Indira Nagar’, ’Bangalore’, ’Karnataka’,560038),
(009,’Sherif’,’JP Nagar’, ’Bangalore’, ’Karnataka’,560078),
(010,’Tanish’,’’Kannur’, ’Bangalore’, ’Karnataka’,562149),
(011,’Mahesh’,’Adyar’,’Chennai’, ’Karnataka’,600020),
(012,’Margaret’,’Anna Nagar’,’Chennai’,’Tamil Nadu’,600040),
(013,’Nadish’,’Broadway’, ‘Chennai’, ’Tamil Nadu’,600108),
(014, ‘Virat’,’Chepak’, ’Chennai’, ’Tamil Nadu’,600005),
(015,’Vikram’,’Defence Colony’, ’Chennai’, ’Tamil Nadu’,600032),
(016,’Deepak’,’Ambewadi’,’Mumbai’,’Maharashtra’,400004),
(017,’Subash’,’’Cumballa Hill’, ’Mumbai’, ’Maharashtra’,400026),
(018,’Sudeep’,’Goregaon’, ’Mumbai’, ’Maharashtra’,400062),
(019,’Farahn’,’Jacob Circle’, ’Mumbai’, ’Maharashtra’,400011),
(020,’Ganish’,’Mandvi’, ’Mumbai’, ’Maharashtra’,400003);
```

1.b. Mass insert can be done as below

i. Create a .csv file containing the data (for example create ‘Dealer.csv’ containing the data for table ‘Dealer’

ii. When table ‘Dealer’ is created within in a database ‘db1’. Then, we see a corresponding directory ‘db1’ under directory

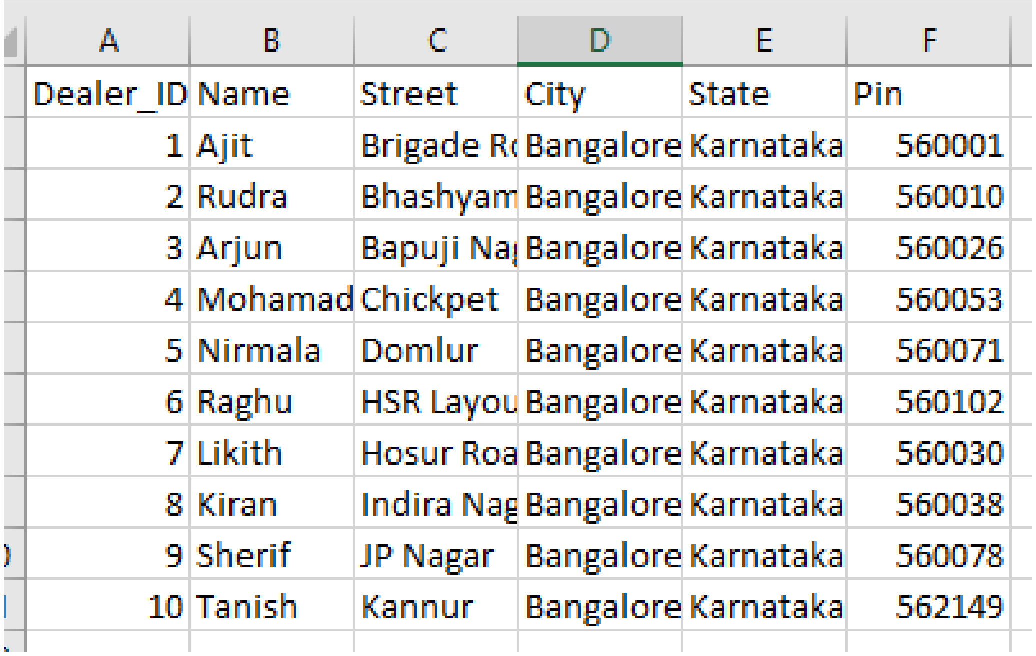
‘/opt/lampp/var/mysql/’ (For Linux) or ‘\xampp\mysql\data’ (for Windows) or ‘XAMPP/xamppfiles/var/mysql’ (for Mac)

iii. Place Dealer.csv to the directory ‘db1’

iv. Run the command in PHPMYADMINSss

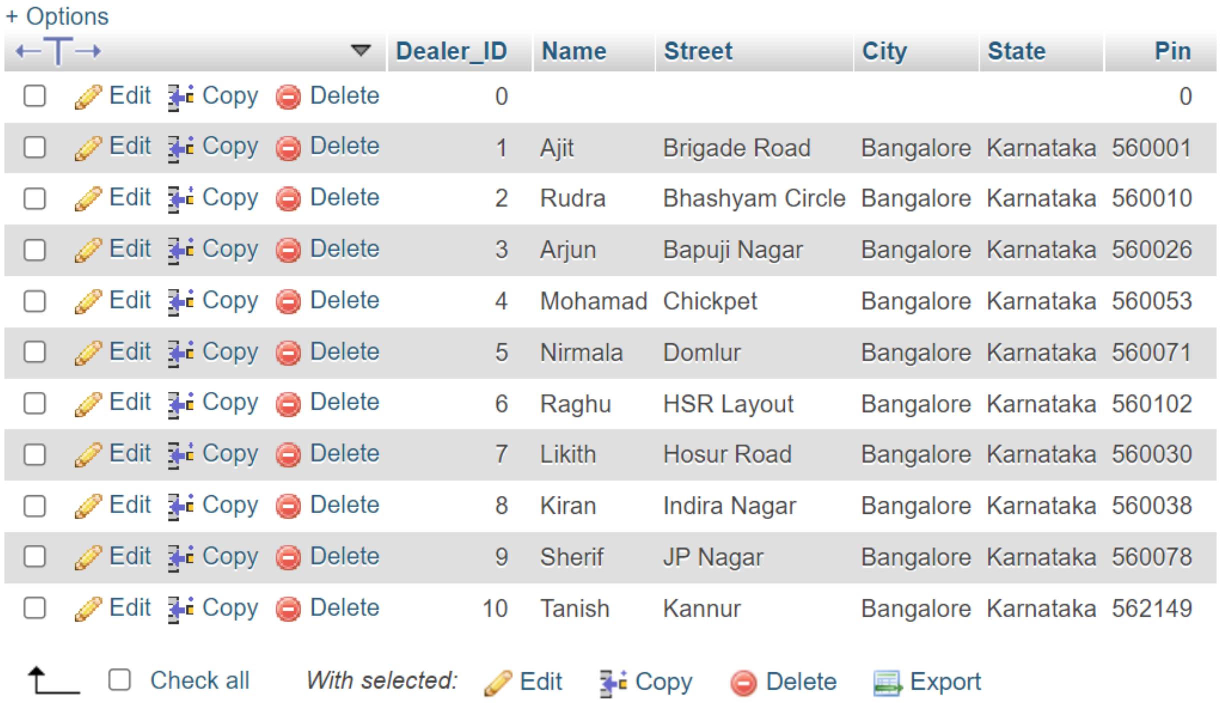
```
LOAD DATA INFILE "Dealer.csv" INTO TABLE Dealer
COLUMNS TERMINATED BY ','
OPTIONALLY ENCLOSED BY '"'
ESCAPED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 LINES;
```

Dealer.csv file:



A	B	C	D	E	F
Dealer_ID	Name	Street	City	State	Pin
1	Ajit	Brigade R	Bangalore	Karnataka	560001
2	Rudra	Bhashyam	Bangalore	Karnataka	560010
3	Arjun	Bapuji Na	Bangalore	Karnataka	560026
4	Mohamad	Chickpet	Bangalore	Karnataka	560053
5	Nirmala	Domlur	Bangalore	Karnataka	560071
6	Raghu	HSR Layou	Bangalore	Karnataka	560102
7	Likith	Hosur Roa	Bangalore	Karnataka	560030
8	Kiran	Indira Nag	Bangalore	Karnataka	560038
9	Sherif	JP Nagar	Bangalore	Karnataka	560078
10	Tanish	Kannur	Bangalore	Karnataka	562149

Output: All the 10 tuples are inserted using LOAD DATA INFILE command



+ Options

←T→ ▼	Dealer_ID	Name	Street	City	State	Pin
□ Edit Copy Delete	0					0
□ Edit Copy Delete	1	Ajit	Brigade Road	Bangalore	Karnataka	560001
□ Edit Copy Delete	2	Rudra	Bhashyam Circle	Bangalore	Karnataka	560010
□ Edit Copy Delete	3	Arjun	Bapuji Nagar	Bangalore	Karnataka	560026
□ Edit Copy Delete	4	Mohamad	Chickpet	Bangalore	Karnataka	560053
□ Edit Copy Delete	5	Nirmala	Domlur	Bangalore	Karnataka	560071
□ Edit Copy Delete	6	Raghu	HSR Layout	Bangalore	Karnataka	560102
□ Edit Copy Delete	7	Likith	Hosur Road	Bangalore	Karnataka	560030
□ Edit Copy Delete	8	Kiran	Indira Nagar	Bangalore	Karnataka	560038
□ Edit Copy Delete	9	Sherif	JP Nagar	Bangalore	Karnataka	560078
□ Edit Copy Delete	10	Tanish	Kannur	Bangalore	Karnataka	562149

□ Check all *With selected:* Edit Copy Delete Export

1.c. Easier method: Select Import from the top menus in PHPMYADMIN

1. In phpMyAdmin dashboard, select the table you want to insert
2. Select Import from the top menu and follow instructions to import the respective .csv file into the table.

Sales Person Table:

Insert into SalesPerson(SalesPersonID,Fname,Lname,Dealer_ID) values
(100,’Raghul’,’Kanna’,001),
(101,’Akshay’,’Kumar’,002),
(102,’Anil’,’Kapoor’,015),
(103,’Barath’,’Kumar’,003),
(104,’Smiriti’,’Bhai’,004),
(105,’Rishi’,’Sunak’,005),
(106,’Srihari’,’Udupa’,011),
(107,’Pallavi’,’Sharma’,012),
(108,’Bala’,’Reddy’,013),
(109,’Sindhya’,’Kapoor’,019),
(110,’Suma’,’Sampat’,020);

Customer Table:

```
Insert into Customer (Cust_ID,FName,LName,Address,Email) values
(201,’Sai’,’Shankar’,’Chennai’,’saishankar@gmail.com’),
(202,’Apoorva’,’Kishore’,’Bangalore’,’apoorvak@gmail.com’),
(203,’Bala’,’Kumar’,’Chennai’,’balakumar@gmail.com’),
(204,’Chethan’,’Kumar’,’Kerala’,’chethankumar@gmail.com’),
(205,’Gowtham’,’Raman’,’Bangalore’,’gowtham@gmail.com’);
```

Bike Table:

```
Insert into Bike(VIN,Make,Model, Reg_No,Y_O_M,Chasis_No) values
(300,’Honda’,’CB500X’,’KA20H2010’,2015,’1GNCS18Z3M0115561’),
(301,’Kawasaki’,’KLX230’,’KA56H2344’,2017,’1HBGH41JXMN109186’),
(302,’Suzuki’,’GSX-R1000’,’TN70A1653’,2016,’JYA2UJE0X2A050036’),
(303,’Yamaha’,’Smax’,’MH04AT4351’,2014,’WYGZZZ5NZJM131396’),
(304,’TVS’,’Ntorq 125’,’TN30J5699’,2020,’JN3MS37A9PW202929’),
(305,’Mahindra’,’Duro’,’MH02BW1620’,2013,’MLHNC5110G5200064’);
```

Invoice Table:

```
Insert into Invoice(SalesPersonID,VIN,Cust_ID,Date,Total_Bill) values
(100,300,201,’2016-04-04’,120998.98),
(102,301,202,’2021-09-10’,132432.98),
(103,302,203,’2012-10-02’,908907.87),
(105,304,204,’2017-03-23’,78900.90),
(109,305,205,’2015-12-27’,88998.99),
(110,303,201,’2022-01-20’,120090.89);
```

Service Ticket Table:

```
Insert into Service_Ticket(Service_ID,VIN,Cust_ID, Date_Rec,Date_Del,
M_emp_id) Values (400,300,201,’2022-01-23’,’2022-01-24’,601),
(401,301,202,’2022-01-23’,’2022-01-24’,604)
(402,302,203,’2021-02-23’,’2022-02-27’,603),
(403,303,204,’2022-04-10’,’2022-04-13’,605),
(404,304,205,’2022-05-15’,’2022-05-24’,605),
(405,300,203,’2021-12-26’,’2021-12-27’,601);
```

Service Manager Table:

```
Insert into Service_Manager(M_emp_id,F_Name,L_Name ,No_of_bikes_serviced)
values (600,’Guru’,’Prasad’,2,),
(601,’Kalyan’,’Bhat’,3),
(602,’Jeevan’,’Kishore’,5),
(603,’Anjan’,’Jain’,4),
(604,’Madhan’,’Raj’,6),
(605,’Yashah’,’Kiran’,7);
```

Parts Table:

```
Insert into Parts(P_ID,Description,Qty,Price,Service_ID) values
(700,’Air Filter’,1,217.99,400),
(701,’Chain Set’,2,1987.99,401),
(702,’Clutch Plate’,517.99,404),
(703,’Handle Bar’,4,2519.99,405),
(704,’Horn’,5,1675.78,402),
(705,’Leg Guard’,101,6217.99,403);
```

Customer Phone Table:

```
Insert into cust_phone(Cust_ID,Phone_no) values
(201,9047223778), (201,8970654327), (202,8903562403), (203,6784523450),
(203,6547892020), (204,8760001589),(205,9988071233), (205,9080102030);
```

SELECT Statements:

Retrieving records from the table with different conditions using Comparison operators, Between, Membership operators, Pattern matching and Logical operators:

```
Select * from Dealer;
Select * from Dealer where city=’Bangalore’;
Select Dealer_ID from Dealer where City in(‘Bangalore’,’Chennai’);
Select Dealer_ID from Dealer where City not in(‘Bangalore’,’Chennai’);
Select * from Parts where Price<1000;
Select * from Customer where FName like ‘C%’;
Select * from Customer where FName like ‘_a%’;
Select * from Customer where LName is not null;
Select * from Customer where LName is null;
Select * from Bike where Price between 60000 and 150000;
Select * from Bike where Model != ’Ntorq 125’;
Select * from Service_Manager where L_name <> ‘Jain’;
Select * from Invoice where VIN=302 AND Cust_ID=203;
Select * from Invoice where VIN=302 OR Cust_ID=203;
Select * from Dealer where NOT City=’Mumbai’;
```

UPDATE Statements:

```
Update Customer set Address=’Mumbai’, Email=’gowthamraman@wipro.com’
where Cust_ID=205;
Update Dealer set Pin=560001 where city=’Bangalore’;
Update Dealer set Pin=560001;
 // Updates particular tuples by satisfying where clause condition
// Updates all the tuples in a relation when where clause is missing
```

DELETE Statements:

```
Delete from Sevice_Manager where M_emp_id =605;
Delete from Parts;
Delete from Dealer where Name like ‘%h’;
Delete from Bike where Y_O_M between 2012 and 2018;
Delete * from SalesPerson;
// Deletes particular tuples by satisfying where clause condition
// Deletes all the tuples in a relation when where clause is missing
```

DCL Statements:

GRANT:

```
GRANT SELECT, INSERT, UPDATE, DELETE ON Customer TO 'User_Name'@'localhost';
//Specified privileges are granted to the user
GRANT ALL ON Dealer TO 'User_Name'@'localhost';
// All the privileges are granted
GRANT SELECT ON Invoice TO '*'@'localhost';
//Read-only privilege granted
```

REVOKE:

```
REVOKE DELETE, UPDATE ON Bike FROM 'Username'@'localhost';
// Delete and Update privileges are cancelled / revoked from the user. So, the user cannot do any
write opeartion
```

```
REVOKE ALL ON Service_Manager FROM 'Username'@'localhost';
// All the privileges are revoked / cancelled. So, the user cannot perform any read / write operation
REVOKE SELECT ON Parts FROM '*'@'localhost';
//Cancelling select privilege from the user.
```

TCL Statements:

COMMIT & ROLLBACK

MySQL saves the changes done after the execution of each statement. To save changes automatically, set the autocommit option as shown below:

```
SET autocommit=0 | 1; // off | on
```

Ex1:

```
START TRANSACTION;
ALTER TABLE Service_Ticket ADD Service_Amt decimal(10,2) NOT NULL AFTER Date_Del;
SELECT @A:=SUM(Price) FROM Parts WHERE Description in(‘Air Filter’,’Horn’,’Chain Set);
UPDATE Service_Ticket SET Service_Amt=@A WHERE M_emp_id =602;
COMMIT;
```

// After issuing the **Commit,** the above changes in Service_Ticket relation has been saved.

Ex2:

```
mysql> CREATE TABLE Dealer_Address (Street_Name varchar(30),City varchar(30),Pincode int);
Query OK, 0 rows affected (0.00 sec)
mysql> -- Do a transaction with autocommit turned on.
mysql> SET autocommit=1;
mysql> START TRANSACTION;
Query OK, 0 rows affected (0.00 sec)
mysql> INSERT INTO Dealer_Address(Street_Name,City,Pincode) SELECT Street,City,Pin from Dealer;
Query OK, 1 row affected (0.00 sec)
//The above changes were saved automatically
mysql> -- Do another transaction with autocommit turned off.
mysql> SET autocommit=0;
```

```
Query OK, 0 rows affected (0.00 sec)
mysql> INSERT INTO Dealer_Address VALUES (‘Nehru Street’,’Chennai’,600010);
Query OK, 1 row affected (0.00 sec)
mysql> DELETE FROM Dealer_Address WHERE City= 'Chennai';
Query OK, 1 row affected (0.00 sec)
mysql> -- Now we undo those last 2 inserts and the delete.
mysql> ROLLBACK;
Query OK, 0 rows affected (0.00 sec)
mysql> SELECT * FROM Dealer_Address;
```

// The above code shows that, when Auto commit turned on, the changes were saved. When the Auto commit turned off, no changes were saved. And, when we issue Rollback, the changes were undone.

SAVEPOINT:

Ex:

```
SAVEPOINT SP1;
//Savepoint created.
DELETE FROM Customer WHERE Cust_ID= 203;
//delete operation is performed
SAVEPOINT SP2;
//Another Savepoint created.
```

The above Deletion is undone by the following statement:

```
ROLLBACK TO SP1;
//Rollback completed. So, it retains the customer tuple whose Cust_ID is 203
```

RELEASE SAVEPOINT:

This command is used to remove a SAVEPOINT that you have created.

Syntax:

```
RELEASE SAVEPOINT SAVEPOINT_NAME
```

Ex:

```
RELEASE SAVEPOINT SP2; // The above created Savepoint SP2 is removed
```