

Data types in MySQL-

1) Numeric data types-

- TINYINT (1 byte)
- SMALLINT (2 bytes)
- MEDIUMINT (3 bytes)
- INTEGER (4 bytes)
- INT (4 bytes)
- BIGINT (8 bytes)
- DOUBLE (8 bytes)
- FLOAT (4 bytes)

2) Date and time types-

- DATE (3 bytes)
- DATETIME (8 bytes - YYYY-MM-DD hh:mm:ss)
- TIMESTAMP (4 bytes - YYYY-MM-DD hh:mm:ss)
- TIME (3 bytes)
- YEAR (1 byte - YYYY)

3) String types-

- CHAR (user defined length between 0 to 255)
- VARCHAR (user defined length between 0 to 65532)

Data type mappings from MySQL to Oracle-

- BIGINT -> NUMBER(19, 0)
- CHAR -> CHAR
- DATE -> DATE
- DATETIME -> DATE
- DOUBLE -> FLOAT (24)
- FLOAT -> FLOAT
- INT -> NUMBER(10, 0)

- MEDIUMINT-> NUMBER(7, 0)
- SMALLINT -> NUMBER(5, 0)
- TIME -> DATE
- TIMESTAMP-> DATE
- TINYINT -> NUMBER(3, 0)
- VARCHAR -> VARCHAR2
- YEAR -> NUMBER

Queries in MySQL

The **CREATE DATABASE** statement is used to create a new SQL database.

Syntax: CREATE DATABASE databasename;

Example: CREATE DATABASE testDB;

The **DROP DATABASE** statement is used to drop an existing SQL database.

Syntax: DROP DATABASE databasename;

Example: DROP DATABASE testDB;

The **CREATE TABLE** statement is used to create a new table in a database.

Syntax: CREATE TABLE table_name (column1 datatype, column2 datatype,);

Example: CREATE TABLE Persons (PersonID int, Name varchar(20), Address varchar(50));

The **DROP TABLE** statement is used to drop an existing table in a database.

Syntax: DROP TABLE table_name;

Example: `DROP TABLE Shippers;`

The **TRUNCATE TABLE** statement is used to delete the data inside a table, but not the table itself.

Syntax: `TRUNCATE TABLE table_name;`

Example: `TRUNCATE TABLE Shippers;`

The **ALTER TABLE** statement is used to add, delete, or modify columns in an existing table. It is also used to add and drop various constraints on an existing table.

To add a column in a table, use the following syntax:

Syntax: `ALTER TABLE table_name ADD column_name datatype;`

Example: `ALTER TABLE Customers ADD Email varchar(25);`

To delete a column in a table, use the following syntax:

Syntax: `ALTER TABLE table_name DROP COLUMN column_name;`

Example: `ALTER TABLE Customers DROP COLUMN Email;`

To rename a column in a table, use the following syntax:

Syntax: `ALTER TABLE table_name RENAME COLUMN old_name to new_name;`

To change the data type of a column in a table, use the following syntax:

Syntax: `ALTER TABLE table_name MODIFY COLUMN column_name datatype;`

Example: `ALTER TABLE Persons MODIFY COLUMN DateOfBirth year;`

To add a new column in the table, use the following syntax:

Syntax: `ALTER TABLE table_name ADD column_name datatype;`

Example: `ALTER TABLE Persons ADD DateOfBirth date;`

SQL constraints are used to specify rules for data in a table. Constraints can be specified when the table is created with the CREATE TABLE statement, or after the table is created with the ALTER TABLE statement.

NOT NULL - Ensures that a column cannot have a NULL value

SQL NOT NULL on CREATE TABLE

Example: CREATE TABLE Persons (ID int NOT NULL, LastName varchar(255) NOT NULL, FirstName varchar(255) NOT NULL, Age int);

SQL NOT NULL on ALTER TABLE

Example: ALTER TABLE Persons MODIFY COLUMN Age int NOT NULL;

UNIQUE - Ensures that all values in a column are different

SQL UNIQUE Constraint on CREATE TABLE

Example: CREATE TABLE Persons (ID int NOT NULL UNIQUE, LastName varchar(255) NOT NULL, FirstName varchar(255), Age int);

OR

Example: CREATE TABLE Persons (ID int NOT NULL, LastName varchar(255) NOT NULL, FirstName varchar(255), Age int, UNIQUE (ID));

UNIQUE constraint on multiple columns

Example: CREATE TABLE Persons (ID int NOT NULL, LastName varchar(255) NOT NULL, FirstName varchar(255), Age int, CONSTRAINT UC_Person UNIQUE (ID,LastName));

SQL UNIQUE Constraint on ALTER TABLE

Example: ALTER TABLE Persons ADD UNIQUE (ID);

OR

Example: ALTER TABLE Persons ADD CONSTRAINT UC_Person UNIQUE (ID,LastName);

DROP a UNIQUE Constraint

ALTER TABLE Persons DROP CONSTRAINT UC_Person;

PRIMARY KEY - A combination of NOT NULL and UNIQUE. Uniquely identifies each row

SQL PRIMARY KEY on CREATE TABLE

Example: CREATE TABLE Persons (ID int NOT NULL, LastName varchar(255) NOT NULL, FirstName varchar(255), Age int, PRIMARY KEY (ID));

OR

Example: CREATE TABLE Persons (ID int NOT NULL PRIMARY KEY, LastName varchar(255) NOT NULL, FirstName varchar(255), Age int);

SQL PRIMARY KEY on ALTER TABLE

Example: ALTER TABLE Persons ADD PRIMARY KEY (ID);

OR

Example: ALTER TABLE Persons ADD CONSTRAINT PK_Person PRIMARY KEY (ID,LastName);

DROP a PRIMARY KEY Constraint

Example: ALTER TABLE Persons DROP PRIMARY KEY;

OR

Example: ALTER TABLE Persons DROP CONSTRAINT PK_Person;

FOREIGN KEY - Prevents actions that would destroy links between tables

SQL FOREIGN KEY on CREATE TABLE

Example: CREATE TABLE Orders (OrderId int NOT NULL, OrderNumber int NOT NULL, PersonId int, PRIMARY KEY (OrderId), FOREIGN KEY (PersonId) REFERENCES Persons(PersonId));

OR

Example: CREATE TABLE Orders (OrderId int NOT NULL PRIMARY KEY, OrderNumber int NOT NULL, PersonId int FOREIGN KEY REFERENCES Persons(PersonId));

SQL FOREIGN KEY on ALTER TABLE

Example: ALTER TABLE Orders ADD FOREIGN KEY (PersonId) REFERENCES Persons(PersonId);

OR

Example: ALTER TABLE Orders ADD CONSTRAINT FK_PersonOrder FOREIGN KEY (PersonId) REFERENCES Persons(PersonId);

DROP a FOREIGN KEY Constraint

Example: ALTER TABLE Orders DROP FOREIGN KEY FK_PersonOrder;

CHECK - Ensures that the values in a column satisfies a specific condition

SQL CHECK on CREATE TABLE

Example: CREATE TABLE Persons (ID int NOT NULL, LastName varchar(255) NOT NULL, FirstName varchar(255), Age int, CHECK (Age>=18));

OR

Example: CREATE TABLE Persons (ID int NOT NULL, LastName varchar(255) NOT NULL, FirstName varchar(255), Age int CHECK (Age>=18));

SQL CHECK on ALTER TABLE

Example: ALTER TABLE Persons ADD CHECK (Age>=18);

DROP a CHECK Constraint

Example: ALTER TABLE Persons DROP CONSTRAINT CHK_PersonAge;

DEFAULT - Sets a default value for a column if no value is specified

SQL DEFAULT on CREATE TABLE

Example: CREATE TABLE Persons (ID int NOT NULL, LastName varchar(255) NOT NULL, FirstName varchar(255), Age int, City varchar(255) DEFAULT 'Sandnes');

SQL DEFAULT on ALTER TABLE

Example: ALTER TABLE Persons ALTER City SET DEFAULT 'Sandnes';

OR

Example: ALTER TABLE Persons ADD CONSTRAINT df_City DEFAULT 'Sandnes' FOR City;

DROP a DEFAULT Constraint

Example: ALTER TABLE Persons ALTER City DROP DEFAULT;

The **SELECT** statement is used to select data from a database.

Syntax: `SELECT column1, column2, ... FROM table_name;`

If you want to return all columns, without specifying every column name, you can use the `SELECT *` syntax:

Example: `SELECT * FROM Customers;`

The **SELECT DISTINCT** statement is used to return only distinct values.

Syntax: `SELECT DISTINCT column1, column2, ... FROM table_name;`

Example: `SELECT DISTINCT Country FROM Customers;`

The **WHERE** clause is used to filter records, i.e. extract only those records that fulfil a specified condition.

Syntax: `SELECT column1, column2, ... FROM table_name WHERE condition;`

Example: `SELECT * FROM Customers WHERE Country='Mexico';`

The **INSERT INTO** statement is used to insert new records in a table.

Specify both the column names and the values to be inserted:

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

Example: `INSERT INTO Customers (CustomerName, City, Country) VALUES ('Cardinal', 'Stavanger', 'Norway');`

OR

Syntax: `INSERT INTO table_name VALUES (value1, value2, value3, ...);`

Insert Multiple Rows

Example: `INSERT INTO Customers (CustomerName, ContactName, Address, City, PostalCode, Country) VALUES ('Cardinal', 'Tom B. Erichsen', 'Skagen 21', 'Stavanger', '4006', 'Norway'), ('Greasy Burger', 'Per Olsen', 'Gate Veien 15', 'Sandnes', '4306', 'Norway'), ('Tasty Tee', 'Finn Egan', 'Street Road 19B', 'Liverpool', 'L1 0AA', 'UK');`

SQL Lab 1

Q1. Create the tables described below:

Table Name: CLIENT_MASTER

Name	Type
CLIENTNO	VARCHAR2(6)
NAME	VARCHAR2(20)
CITY	VARCHAR2(15)
PINCODE	NUMBER(8)
STATE	VARCHAR2(15)
BALDUE	NUMBER(10,2)

Create database

```
mysql> create database du;
Query OK, 1 row affected (0.00 sec)

mysql> use du;
Database changed
```

Table name: CLIENT_MASTER

```
mysql> create table CLIENT_MASTER(clientno varchar(6),name varchar(20),city varchar(15),pincode int(8),state varchar(15),baldue int(10));
Query OK, 0 rows affected (0.08 sec)
```

```
mysql> describe CLIENT_MASTER;
+-----+-----+-----+-----+-----+-----+
| Field | Type  | Null | Key  | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| clientno | varchar(6) | YES |   | NULL |   |
| name | varchar(20) | YES |   | NULL |   |
| city | varchar(15) | YES |   | NULL |   |
| pincode | int(8) | YES |   | NULL |   |
| state | varchar(15) | YES |   | NULL |   |
| baldue | int(10) | YES |   | NULL |   |
+-----+-----+-----+-----+-----+
6 rows in set (0.04 sec)
```

2. Table Product_master

Table Name: PRODUCT_MASTER

Name	Type
PRODUCTNO	VARCHAR2(6)
DESCRIPTION	VARCHAR2(15)
PROFITPERCENT	NUMBER(4,2)
UNITMEASURE	VARCHAR2(10)
QTYONHAND	NUMBER(8)
REORDERLVL	NUMBER(8)
SELLPRICE	NUMBER(8,2)
COSTPRICE	NUMBER(8,2)

Table name :PRODUCT_MASTER

```
mysql> create table Product_Master(Productno varchar(6),Description varchar(15),Profitpercent float(4,2),Unitmeasure varchar(10),Qtyonhand int(8),Reorderlvl int(8),Sellprice float(8,2),Costprice float(8,2));
Query OK, 0 rows affected (0.06 sec)

mysql> describe Product_Master;
+-----+-----+-----+-----+-----+-----+
| Field | Type  | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| Productno | varchar(6) | YES | NULL | NULL |       |
| Description | varchar(15) | YES | NULL | NULL |       |
| Profitpercent | float(4,2) | YES | NULL | NULL |       |
| Unitmeasure | varchar(10) | YES | NULL | NULL |       |
| Qtyonhand | int(8) | YES | NULL | NULL |       |
| Reorderlvl | int(8) | YES | NULL | NULL |       |
| Sellprice | float(8,2) | YES | NULL | NULL |       |
| Costprice | float(8,2) | YES | NULL | NULL |       |
+-----+-----+-----+-----+-----+-----+
8 rows in set (0.00 sec)
```

Table Name: SALESMAN_MASTER

Name	Type
SALESMANNO	VARCHAR2(6)
SALESMANNAME	VARCHAR2(20)
ADDRESS1	VARCHAR2(30)
ADDRESS2	VARCHAR2(30)
CITY	VARCHAR2(20)
PINCODE	NUMBER(8)
STATE	VARCHAR2(20)
SALAMT	NUMBER(8,2)
TGTTOGET	NUMBER(6,2)
YTDSALES	NUMBER(6,2)
REMARKS	VARCHAR2(60)

Table name:SALESMAN_MASTER

```
mysql> create table Salesman_Master(Salesmanno varchar(6),salesmanname varchar(20),Address1 varchar(30),address2 varchar(30),city varchar(20),Pincode int(8),State varchar(20),s
lant float(8,2),Tgtnoget float(6,2),Ytdsales float(6,2),Remarks varchar(60));
Query OK, 0 rows affected (0.05 sec)

mysql> describe Salesman_Master;
+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| Salesmanno | varchar(6) | YES | NULL |
| salesmanname | varchar(20) | YES | NULL |
| Address1 | varchar(30) | YES | NULL |
| address2 | varchar(30) | YES | NULL |
| city | varchar(20) | YES | NULL |
| Pincode | int(8) | YES | NULL |
| State | varchar(20) | YES | NULL |
| salamt | float(8,2) | YES | NULL |
| Tgtnoget | float(6,2) | YES | NULL |
| Ytdsales | float(6,2) | YES | NULL |
| Remarks | varchar(60) | YES | NULL |
+-----+-----+-----+-----+-----+
11 rows in set (0.00 sec)
```

3. Insert the following data into their respective tables:

Data for CLIENT_MASTER table:

CLIENT	NAME	CITY	PINCODE	STATE	BALDUE
C00001	Ivan Bayross	Mumbai	400054	Maharashtra	15000
C00003	Chhaya Bankar	Mumbai	400057	Maharashtra	5000
C00004	Ashwini Joshi	Bangalore	560001	Karnataka	0
C00005	Hansel Colaco	Mumbai	400060	Maharashtra	2000
C00006	Deepak Sharma	Mangalore	560050	Karnataka	0

Table name:CLIENT_MASTER

```
mysql> insert into CLIENT_MASTER values("C0001","Ivan bayross","Mumbai",40054,"Maharashtra",15000),("C0003","Chhaya Bankar","Mumbai",40057,"Maharashtra",500),("C0004","Ash
oshi","Bangalore",56001,"Karnataka",0),("C0005","Hansel colaco","Mumbai",40060,"Maharashtra",200),("C0006","Deepak Sharma","Mangalore",560050,"Karnataka",0);
Query OK, 5 rows affected (0.06 sec)
Records: 5  Duplicates: 0  Warnings: 0
```

```

mysql> select * from CLIENT_MASTER;
+-----+-----+-----+-----+-----+-----+
| clientno | name      | city     | pincode | state    | baldue |
+-----+-----+-----+-----+-----+-----+
| C0001    | Ivan bayross | Mumbai   | 40054   | Maharashtra | 15000  |
| C0003    | Chhaya Bankar | Mumbai   | 40057   | Maharashtra | 500    |
| C0004    | Ashwini joshi | Bangalore | 56001   | Karnataka  | 0      |
| C0005    | Hansel colaco | Mumbai   | 40060   | Maharashtra | 200    |
| C0006    | Deepak sharma | Mangalore | 560050  | Karnataka  | 0      |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

```

Data for PRODUCT_MASTER table

PRODUC	DESCRIPTION	PROFITPERCENT	UNITMEASUR	QTYONHAND	REORDERLVL	SELLPRICE	COSTPRICE
P00001	T-Shirts	5	Piece	200	50	350	250
P03453	Shirts	6	Piece	150	50	500	350
P06734	Cotton Jeans	5	Piece	100	20	600	450
P07865	Jeans	5	Piece	100	20	750	500
P07868	Trousers	2	Piece	150	50	850	550
P07885	Pull Overs	2.5	Piece	80	30	700	450
P07965	Denim Shirts	4	Piece	100	40	350	250
P07975	Lycra Tops	5	Piece	70	30	300	175
P08865	Skirts	5	Piece	75	30	450	300

Data for SALESMAN_MASTER table

SALESM	SALESMANNAME	ADDRESS1	ADDRESS2	CITY	PINCODE	STATE	SALAMT	TGTTOGET	YTDSALES	REMARKS
S00001	Aman	A/14	Worli	Mumbai	400002	Maharashtra	3000	100	50	Good
S00002	Omkar	65	Nariman	Mumbai	400001	Maharashtra	3000	200	100	Good
S00003	Raj	P-7	Bandra	Mumbai	400032	Maharashtra	3000	200	100	Good
S00004	Ashish	A/5	Juhu	Bombay	400044	Maharashtra	3500	200	150	Good

Table name : PRODUCT_MASTER

```

mysql> Insert into PRODUCT_MASTER values ("P00001", "T-Shirts", 5, "Piece", 200, 50, 350, 250);
Query OK, 1 row affected (0.03 sec)

mysql> Insert into PRODUCT_MASTER values ("P03453", "Shirts", 6, "Piece", 150, 50, 500, 350);
Query OK, 1 row affected (0.06 sec)

mysql> Insert into PRODUCT_MASTER values ("P06734", "Cotton Jeans", 5, "Piece", 100, 20, 600, 450);
Query OK, 1 row affected (0.05 sec)

mysql> Insert into PRODUCT_MASTER values ("P07865", "Jeans", 5, "Piece", 100, 20, 750, 500);
Query OK, 1 row affected (0.03 sec)

mysql> Insert into PRODUCT_MASTER values ("P07868", "Trousers", 2, "Piece", 150, 50, 850, 550);
Query OK, 1 row affected (0.06 sec)

mysql> Insert into PRODUCT_MASTER values ("P07885", "Pull Overs", 2.5, "Piece", 80, 30, 700, 450);
Query OK, 1 row affected (0.06 sec)

mysql> Insert into PRODUCT_MASTER values ("P07965", "Denim Shirts", 4, "Piece", 100, 40, 350, 250);
Query OK, 1 row affected (0.06 sec)

mysql> Insert into PRODUCT_MASTER values ("P07975", "Lycra Tops", 5, "Piece", 70, 30, 300, 175);
Query OK, 1 row affected (0.07 sec)

mysql> Insert into PRODUCT_MASTER values ("P08865", "Skirts", 5, "Piece", 75, 30, 450, 300);
Query OK, 1 row affected (0.06 sec)

mysql> Select * from PRODUCT_MASTER;
+-----+-----+-----+-----+-----+-----+-----+-----+
| Product_No | Description | Profit_Percentage | Unit_Measure | Qty_onhand | Recorder_LVL | Sell_Price | Cost_price |
+-----+-----+-----+-----+-----+-----+-----+-----+
| P00001 | T-Shirts | 5.00 | Piece | 200 | 50 | 350.00 | 250.00 |
| P03453 | Shirts | 6.00 | Piece | 150 | 50 | 500.00 | 350.00 |
| P06734 | Cotton Jeans | 5.00 | Piece | 100 | 20 | 600.00 | 450.00 |
| P07865 | Jeans | 5.00 | Piece | 100 | 20 | 750.00 | 500.00 |
| P07868 | Trousers | 2.00 | Piece | 150 | 50 | 850.00 | 550.00 |
| P07885 | Pull Overs | 2.50 | Piece | 80 | 30 | 700.00 | 450.00 |
| P07965 | Denim Shirts | 4.00 | Piece | 100 | 40 | 350.00 | 250.00 |
| P07975 | Lycra Tops | 5.00 | Piece | 70 | 30 | 300.00 | 175.00 |
| P08865 | Skirts | 5.00 | Piece | 75 | 30 | 450.00 | 300.00 |
+-----+-----+-----+-----+-----+-----+-----+-----+
9 rows in set (0.00 sec)

```

Table name : SALESMAN_MASTER

```

mysql> insert into Salesman_Master values("S00001","Aman","A/14","Worli","Mumbai",400002,"Maharashtra",3000,100,50,"Good");
Query OK, 1 row affected (0.03 sec)

mysql> insert into Salesman_Master values("S00002","Omkar","65","Nariman","Mumbai",400001,"Maharashtra",3000,200,100,"Good");
Query OK, 1 row affected (0.03 sec)

mysql> insert into Salesman_Master values("S00003","Raj","P-7","Bandra","Mumbai",4000032,"Maharashtra",3000,200,100,"Good");
Query OK, 1 row affected (0.03 sec)

mysql> insert into Salesman_Master values("S00004","Ashish","A/5","Juhu","Bombay",4000044,"Maharashtra",3500,200,150,"Good");
Query OK, 1 row affected (0.07 sec)

mysql> select * from Salesman_Master;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Salesmanno | salesmannname | Address1 | address2 | city | Pincode | State | salamt | Tgttoget | Ytdsales | Remarks |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| S00001 | Aman | A/14 | Worli | Mumbai | 400002 | Maharashtra | 3000.00 | 100.00 | 50.00 | Good |
| S00002 | Omkar | 65 | Nariman | Mumbai | 400001 | Maharashtra | 3000.00 | 200.00 | 100.00 | Good |
| S00003 | Raj | P-7 | Bandra | Mumbai | 4000032 | Maharashtra | 3000.00 | 200.00 | 100.00 | Good |
| S00004 | Ashish | A/5 | Juhu | Bombay | 4000044 | Maharashtra | 3500.00 | 200.00 | 150.00 | Good |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql>

```

SQL Lab 2

4. Create the table and insert rows described below:

Regdno	Name	Branch
0001	Ram	CSE
0002	Hari	MECH
0003	Pradeep	EEE
0004	Deepak	ETC

```
mysql> create table student(
    -> regdno int(4) primary key,
    -> name varchar(15),
    -> branch varchar(15));
Query OK, 0 rows affected (0.09 sec)
```

```
mysql> insert into student values
    -> (0001,"ram","cse"),
    -> (0002,"hari","mech"),
    -> (0003,"pradeep","eee"),
    -> (0004,"deepak","etc");
Query OK, 4 rows affected (0.02 sec)
```

- Find regdno of Pradeep

```
mysql> select regdno from student where name = "pradeep";
+-----+
| regdno |
+-----+
|      3 |
+-----+
1 row in set (0.02 sec)
```

- Find name, branch of table Student

```
mysql> select name,branch from student;
+-----+-----+
| name   | branch |
+-----+-----+
| ram    | cse   |
| hari   | mech  |
| pradeep | eee   |
| deepak | etc   |
+-----+-----+
4 rows in set (0.00 sec)
```

- Add another column address in the student table

```
mysql> alter table student add address varchar(20);
Query OK, 4 rows affected (0.16 sec)
Records: 4  Duplicates: 0  Warnings: 0
```

- Count the number of rows in the table

```
mysql> select count(regdno) from student;
+-----+
| count(regdno) |
+-----+
|          4   |
+-----+
1 row in set (0.00 sec)
```

SQL Lab 3

5. Create the tables described below:

Table name: CLIENT_MASTER

Column Name	Data type	Size	Attributes
CLIENTNO	Varchar2	6	Primary Key / First letter must start with 'C'
NAME	Varchar2	20	Not Null
CITY	Varchar2	15	
PINCODE	Number	8	
STATE	Varchar2	15	
BALDUE	Number	10,2	

```
mysql> create table client_master(
-> clientno varchar(6) primary key,
-> name varchar(20) not null,
-> city varchar(15),
-> pincode int(8),
-> state varchar(15),
-> baldue double(10,2),
-> constraint clientno check (clientno like "C%"));
Query OK, 0 rows affected (0.14 sec)
```

Table Name: PRODUCT_MASTER

Column Name	Data Type	Size	Attributes
PRODUCTNO	Varchar2	6	Primary Key / First letter must start with 'P'
DESCRIPTION	Varchar2	15	Not Null
PROFITPERCENT	Number	4,2	Not Null
UNITMEASURE	Varchar2	10	Not Null
QTYONHAND	Number	8	Not Null
REORDERLVL	Number	8	Not Null
SELLPRICE	Number	8,2	Not Null, cannot be 0
COSTPRICE	Number	8,2	Not Null, cannot be 0

```

mysql> create table product_master(
-> productno varchar(6) primary key,
-> description varchar(15) not null,
-> profitpercent double(4,2) not null,
-> unitmeasure varchar(10) not null,
-> qtyonhand int(8) not null,
-> reorderlvl int(8) not null,
-> sellprice double(8,2) not null,
-> costprice double(8,2) not null,
-> constraint productno check (productno like "P%"),
-> constraint sellprice check (sellprice > 0.00),
-> constraint costprice check (costprice > 0.00));
Query OK, 0 rows affected (0.11 sec)

```

Table Name: SALESMAN_MASTER

Column Name	Data Type	Size	Attributes
SALESMANNO	Varchar2	6	Primary Key / First letter must start with 'S'
SALESMANNAME	Varchar2	20	Not Null
ADDRESS1	Varchar2	30	Not Null
ADDRESS2	Varchar2	30	
CITY	Varchar2	20	
PINCODE	Number	8	
STATE	Varchar2	20	
SALAMT	Number	8,2	Not Null, cannot be 0
TGTTOGET	Number	6,2	Not Null, cannot be 0
YTDSALES	Number	6,2	Not Null
REMARKS	Varchar2	60	

```

mysql> create table salesman_master(
-> salesmanno varchar(6) primary key,
-> name varchar(20) not null,
-> address1 varchar(30) not null,
-> address2 varchar(30) not null,
-> city varchar(20) ,
-> pincode int(8),
-> state varchar(20),
-> salamt double(8,2) not null,
-> tgttoget double(6,2) not null,
-> ytdsales double(6,2) not null,
-> remarks varchar(60),
-> constraint salesmanno check (salesmanno like "S%"),
-> constraint salamt check (salamt > 0.00),
-> constraint tgttoget check (tgttoget > 0.00));
Query OK, 0 rows affected (0.13 sec)

```

Table Name: SALES_ORDER

Column Name	Data Type	Size	Attributes
ORDERNO	Varchar2	6	Primary Key / First letter must start with 'O'
CLIENTNO	Varchar2	6	Foreign Key references ClientNo of Client_Master table
ORDERDATE	Date		Not Null
DELYADDR	Varchar2	25	
SALESMANNO	Varchar2	6	Foreign Key references SalesmanNo of Salesman_Master table
DELYTYPE	Char	1	
BILLYN	Char	1	
DELYDATE	Date		
ORDERSTATUS	Varchar2	10	

```

mysql> create table sales_order(
-> orderno varchar(6) primary key,
-> clientno varchar(6),
-> orderdate date,
-> delayaddr varchar(25),
-> salesmanno varchar(6),
-> delytype char(1),
-> billyn char(1),
-> delydate date,
-> orderstatus varchar(10),
-> constraint orderno check (orderno like "0%"),
-> constraint foreign key(clientno) references client_master(clientno),
-> constraint foreign key(salesmanno) references salesman_master(salesmanno));
Query OK, 0 rows affected (0.19 sec)

```

Table Name: SALES_ORDER_DETAILS

Column Name	Data Type	Size	Attributes
ORDERNO	Varchar2	6	Foreign Key references OrderNo of Sales_Order table
PRODUCTNO	Varchar2	6	Foreign Key references ProductNo of Product_Master table
QTYORDERED	Number	8	
QTYDISP	Number	8	
PRODUCTRATE	Number	10,2	

```

mysql> create table sales_order_details(
-> orderno varchar(6),
-> productno varchar(6),
-> qtyorderno int(8),
-> qtydisp int(8),
-> productrate double(10,2),
-> constraint foreign key(orderno) references sales_order(orderno),
-> constraint foreign key(productno) references product_master(productno));
Query OK, 0 rows affected (0.09 sec)

```

2. Insert the following data into their respective tables:

Data for CLIENT_MASTER table:

CLIENTNO	NAME	CITY	PINCODE	STATE	BALDUE
C00001	Ivan Bayross	Mumbai	400054	Maharashtra	15000
C00002	Mamta Muzumdar	Madras	780001	Tamil Nadu	0
C00003	Chhaya Bankar	Mumbai	400057	Maharashtra	5000
C00004	Ashwini Joshi	Bangalore	560001	Karnataka	0
C00005	Hansel Colaco	Mumbai	400060	Maharashtra	2000
C00006	Deepak Sharma	Mangalore	560050	Karnataka	0

```
mysql> insert into client_master values
-> ("C00001","Ivan Bayross","Mumbai",400054,"Maharashtra",15000.00),
-> ("C00003","Chhaya Bankar","Mumbai",400057,"Maharashtra",5000.00),
-> ("C00004","Ashwini Joshi","Bangalore",560001,"Karnataka",0.00),
-> ("C00005","Hansel Colaco","Mumbai",400060,"Maharashtra",2000.00),
-> ("C00006","Deepak Sharma","Mangalore",560050,"Karnataka",0.00);
Query OK, 5 rows affected (0.01 sec)
Records: 5  Duplicates: 0  Warnings: 0
```

Data for PRODUCT_MASTER table

PRODUC	DESCRIPTION	PROFITPERCENT	UNITMEASUR	QTYONHAND	REORDERLVL	SELLPRICE	COSTPRICE
P00001	T-Shirts	5	Piece	200	50	350	250
P03453	Shirts	6	Piece	150	50	500	350
P06734	Cotton Jeans	5	Piece	100	20	600	450
P07865	Jeans	5	Piece	100	20	750	500
P07868	Trousers	2	Piece	150	50	850	550
P07885	Pull Overs	2.5	Piece	80	30	700	450
P07965	Denim Shirts	4	Piece	100	40	350	250
P07975	Lycra Tops	5	Piece	70	30	300	175
P08865	Skirts	5	Piece	75	30	450	300

```
mysql> Insert into product_master values
-> ("P00001","T-Shirts",5.0,"piece",200,50,350.00,250.00),
-> ("P03453","Shirts",6.0,"piece",150,50,500.00,350.00),
-> ("P06734","Cotton jeans",5.0,"piece",100,20,600.00,450.00),
-> ("P07865","Jeans",5.0,"piece",100,20,750.00,500.00),
-> ("P07868","Trousers",2.0,"piece",150,50,850.00,550.00),
-> ("P07885","Pull overs",2.5,"piece",80,30,700.00,450.00),
-> ("P07965","Denim shirts",4.0,"piece",100,40,350.00,250.00),
-> ("P07975","Lycra tops",5.0,"piece",70,30,300.00,175.00),
-> ("P08865","Skirts",5.0,"piece",75,30,450.00,300.00);
Query OK, 9 rows affected (0.01 sec)
Records: 9  Duplicates: 0  Warnings: 0
```

Data for SALESMAN_MASTER table

SALESM	SALESMANNAME	ADDRESS1	ADDRESS2	CITY	PINCODE	STATE	SALAMT	TGTTOGET	YTDSALES	REMARKS
S00001	Aman	A/14	Worli	Mumbai	400002	Maharashtra	3000	100	50	Good
S00002	Omkar	65	Nariman	Mumbai	400001	Maharashtra	3000	200	100	Good
S00003	Raj	P-7	Bandra	Mumbai	400032	Maharashtra	3000	200	100	Good
S00004	Ashish	A/5	Juhu	Bombay	400044	Maharashtra	3500	200	150	Good

```
mysql> insert into salesman_master values
    -> ("S00001","Aman","A/14","Worli","Mumbai",400002,"Maharashtra",3000,100,50,"Good"),
    -> ("S00002","Omkar","65","Nariman","Mumbai",400001,"Maharashtra",3000,200,108,"Good"),
    -> ("S00003","Raj","P-7","Bandra","Mumbai",400032,"Maharashtra",3000,200,100,"Good"),
    -> ("S00004","Ashish","A/5","Juhu","Bombay",4000844,"Maharashtra",3500,200,150,"Good");
```

Data for Sales_Order Table

ORDERN	CLIENT	ORDERDATE	DELYADDR	SALESM	D	B	DELYDATE	ORDERSTATU
O19003	C00001	03-APR-02	Delhi	S00001	F	Y	07-APR-02	Fulfilled
046866	C00004	20-MAY-02	Delhi	S00002	P	N	22-MAY-02	Cancelled
O19008	C00005	24-MAY-02	Delhi	S00004	F	N	26-JUL-96	In Process
O19001	C00001	12-JUN-02	Delhi	S00001	F	N	20-JUL-02	In Process
O19002	C00002	25-JUN-02	Delhi	S00002	P	N	27-JUL-02	Cancelled

```
mysql> insert into sales_order values
    -> ("O19003","C00001","2002-04-03","Delhi","S00001",'F','Y',"2002-04-07","Fulfilled"),
    -> ("046866","C00004","2002-05-20","Delhi","S00002",'P','N","2002-05-22","Cancelled"),
    -> ("O19008","C00005","2002-05-24","Delhi","S00004",'F','N',"1996-07-26","In process"),
    -> ("O19001","C00001","2002-06-12","Delhi","S00001",'F','N',"2002-07-02","In process"),
    -> ("O19002","C00002","2002-06-25","Delhi","S00002",'P','N',"2002-07-27","Cancelled");
Query OK, 5 rows affected (0.03 sec)
Records: 5  Duplicates: 0  Warnings: 0
```

Data for Sales_Order_Details Table

OrderNo	ProductNo	Qtyordered	QtyDisp	ProductRate
O19001	P00001	4	4	525
O19001	P07965	2	1	8400
O19001	P07885	2	1	5250
O19002	P00001	10	0	525
046865	P07868	3	3	3150
046865	P07885	3	1	5250
046865	P00001	10	10	525
046865	P03453	4	4	1050
O19003	P03453	2	2	1050
O19003	P06734	1	1	12000
046866	P07965	1	0	8400
046866	P07975	1	0	1050
O19008	P00001	10	5	525
O19008	P07975	5	3	1050

```

mysql> insert into sales_order_details values
-> ("019001", "P00001", 4, 4, 525),
-> ("019001", "P07965", 2, 1, 8400),
-> ("019001", "P07885", 2, 1, 5250),
-> ("019002", "P00001", 10, 0, 525),
-> ("046866", "P07868", 3, 3, 3150),
-> ("046866", "P07885", 3, 1, 5250),
-> ("046866", "P00001", 10, 10, 525),
-> ("046866", "P03453", 4, 4, 1050),
-> ("019003", "P03453", 2, 2, 1050),
-> ("019003", "P06734", 1, 1, 12000),
-> ("046866", "P07965", 1, 0, 8400),
-> ("046866", "P07975", 1, 0, 1050),
-> ("019008", "P00001", 10, 5, 525),
-> ("019008", "P07975", 5, 3, 1050);

```

Query OK, 14 rows affected (0.01 sec)
Records: 14 Duplicates: 0 Warnings: 0

SQL Lab 4

6.Exercises on Using Having and Group By Clauses:

- a. Print the description and total Qty sold for each product.

```

mysql> SELECT DESCRIPTION, SUM(QTYDISP) FROM PRODUCT_MASTER, SALES_ORDER_DETAILS WHERE PRODUCT_MASTER.PRODUCT_NO = SALES_ORDER_DETAILS.PRODUCTNO GROUP BY DESCRIPTION;
+-----+-----+
| DESCRIPTION | SUM(QTYDISP) |
+-----+-----+
| T-shirts     |      19 |
| denim shirts |       1 |
| pull overs   |       2 |
| trousers     |       3 |
| shirts        |       6 |
| cotton jeans |       1 |
| lycra tops    |       3 |
+-----+-----+
7 rows in set (0.00 sec)

```

- b. Find the value of each product sold.

```

mysql> SELECT SALES_ORDER_DETAILS.PRODUCTNO, PRODUCT_MASTER.DESCRIPTION, SUM(SALES_ORDER_DETAILS.QTYDISP * SALES_ORDER_DETAILS.PRODUCTRATE) SALESPPERPRODUCT FROM SALES_ORDER_DETAILS, PRODUCT_MASTER WHERE PRODUCT_MASTER.PRODUCT_NO = SALES_ORDER_DETAILS.PRODUCTNO GROUP BY PRODUCT_MASTER.PRODUCT_NO, PRODUCT_MASTER.DESCRIPTION;
+-----+-----+-----+
| PRODUCTNO | DESCRIPTION | SALESPPERPRODUCT |
+-----+-----+-----+
| P00001    | T-shirts     |      9975.00 |
| P07965    | denim shirts |      8400.00 |
| P07885    | pull overs   |      10500.00 |
| P07868    | trousers     |      9450.00 |
| P03453    | shirts        |      6300.00 |
| P06734    | cotton jeans |     12000.00 |
| P07975    | lycra tops    |      3150.00 |
+-----+-----+-----+
7 rows in set (0.00 sec)

```

- C. Calculate the average qty sold for each client that has a maximum order value of 15000.00

```
mysql> SELECT CLIENT_MASTER.CLIENT_NO, AVG(SALES_ORDER_DETAILS.QTYDISP) AVGSALES FROM SALES_ORDER_DETAILS, CLIENT_MASTER, SALES_ORDER WHERE CLIENT_MASTER.CLIENT_NO = SALES_ORDER.CLIENTNO AND SALES_ORDER.ORDER_NO = SALES_ORDER_DETAILS.ORDERNO GROUP BY CLIENT_MASTER.CLIENT_NO HAVING MAX(SALES_ORDER_DETAILS.QTYORDERED * SALES_ORDER_DETAILS.PRODUCTRATE) > 15000;
+-----+-----+
| CLIENT_NO | AVGSALES |
+-----+-----+
| C00001    |   1.8000 |
| C00004    |   3.0000 |
+-----+-----+
2 rows in set (0.00 sec)
```

d. Find out the total of all the billed orders for the month of June.

```
mysql> SELECT SALES_ORDER.ORDER_NO, SALES_ORDER.ORDERDATE, SUM(SALES_ORDER_DETAILS.QTYORDERED * SALES_ORDER_DETAILS.PRODUCTRATE) ORDERBILLED FROM SALES_ORDER, SALES_ORDER_DETAILS WHERE SALES_ORDER_DETAILS.ORDERNO = SALES_ORDER.ORDER_NO AND SALES_ORDER.BILLYN = "Y" AND TO_CHAR(ORDERDATE,"MON") = "06" GROUP BY SALES_ORDER.ORDER_NO, SALES_ORDER.ORDERDATE;
```

7. Exercises on Joins:

a. Find out the products, which have been sold to 'Ivan Bayross'.

```
mysql> SELECT Sales_order_details.PRODUCTNO,Product_master.DESCRIPTION FROM SALES_ORDER_DETAILS,PRODUCT_MASTER,SALES_ORDER,CLIENT_MASTER WHERE product_master.PRODUCT_NO = Sales_order_details.PRODUCTNO AND Sales_order.ORDER_NO = sales_order_details.ORDERNO AND client_master.CLIENT_NO = sales_order.CLIENTNO AND client_master.NAME = "Ivan bayross";
+-----+-----+
| PRODUCTNO | DESCRIPTION |
+-----+-----+
| P00001    | T-shirts      |
| P07965    | denim shirts  |
| P07885    | pull overs   |
| P03453    | shirts        |
| P06734    | cotton jeans  |
+-----+-----+
5 rows in set (0.01 sec)

mysql>
```

b. Find out the products and their quantities that will have to be delivered in the current month.

```
mysql> SELECT Sales_order_details.PRODUCTNO,Product_master.DESCRIPTION, SUM(Sales_order_details.QTYORDERED) FROM SALES_ORDER_DETAILS, SALES_ORDER, PRODUCT_MASTER WHERE Product_master.PRODUCT_NO = Sales_order_details.PRODUCTNO AND Sales_order.ORDER_NO = Sales_order_details.ORDERNO AND TO_CHAR(DELIVDATE, 'MON-YY') = TO_CHAR(SYSDATE,'MON-YY') GROUP BY Sales_order_details.PRODUCTNO,Product_master.DESCRIPTION;
```

c. List the product no. and description of constantly sold (i.e. rapidly moving) products.

```
mysql> SELECT DISTINCT PRODUCT_MASTER.PRODUCT_NO, DESCRIPTION FROM SALES_ORDER_DETAILS, PRODUCT_MASTER WHERE PRODUCT_MASTER.PRODUCT_NO = SALES_ORDER_DETAILS.PRODUCTNO;
+-----+-----+
| PRODUCT_NO | DESCRIPTION |
+-----+-----+
| P00001    | T-shirts      |
| P03453    | shirts        |
| P06734    | cotton jeans  |
| P07868    | trousers      |
| P07885    | pull overs   |
| P07965    | denim shirts  |
| P07975    | lycra tops   |
+-----+-----+
7 rows in set (0.01 sec)
```

d. Finding the names of clients who have purchased 'Trousers'.

```
mysql> SELECT DISTINCT SALES_ORDER.CLIENTNO,CLIENT_MASTER.NAME FROM SALES_ORDER_DETAILS,SALES_ORDER,PRODUCT_MASTER,CLIENT_MASTER WHERE PRODUCT_MASTER.PRODUCT_NO = SALES_ORDER_DETAILS.PRODUCTNO AND SALES_ORDER.ORDER_NO = SALES_ORDER_DETAILS.ORDERNO AND CLIENT_MASTER.CLIENT_NO = SALES_ORDER.CLIENTNO AND DESCRIPTION = "trousers";
+-----+-----+
| CLIENTNO | NAME      |
+-----+-----+
| C00004    | ashwini joshi |
+-----+-----+
1 row in set (0.00 sec)
```

e. Listing the products and orders from customers who have ordered less than 5 units of 'Pull Overs'.

```

mysql> SELECT SALES_ORDER_DETAILS.PRODUCTNO,SALES_ORDER_DETAILS.ORDERNO FROM SALES_ORDER_DETAILS,SALES_ORDER,PRODUCT_MASTER WHERE SALES_ORDER.ORDER_NO = SALES_ORDER_DETAILS.ORDERNO AND PRODUCT_MASTER.PRODUCT_NO = SALES_ORDER_DETAILS.PRODUCTNO AND SALES_ORDER_DETAILS.QTYORDERED < description ="pull overs";
+-----+-----+
| PRODUCTNO | ORDERNO |
+-----+-----+
| P00001 | 019801 |
| P07965 | 019801 |
| P07885 | 019801 |
| P03453 | 019803 |
| P06734 | 019803 |
| P00001 | 019802 |
| P07868 | 046866 |
| P07885 | 046866 |
| P00001 | 046866 |
| P03453 | 046866 |
| P07965 | 046866 |
| P07975 | 046866 |
| P00001 | 019808 |
| P07975 | 019808 |
+-----+-----+
14 rows in set, 15 warnings (0.00 sec)

```

f. Finding the products and their quantities for the orders placed by 'Ivan Bayross' and 'Mamta Muzumdar'.

```

mysql> SELECT SALES_ORDER_DETAILS.PRODUCTNO, PRODUCT_MASTER.DESCRIPTION, SUM(QTYORDERED) UNITSORDERED FROM SALES_ORDER_DETAILS , SALES_ORDER , PRODUCT_MASTER , CLIENT_MASTER WHERE SALES_ORDER.ORDER_NO = SALES_ORDER_DETAILS.ORDERNO AND PRODUCT_MASTER.PRODUCT_NO = SALES_ORDER_DETAILS.PRODUCTNO AND CLIENT_MASTER.CLIENT_NO = SALES_ORDER.CLIENTNO AND (CLIENT_MASTER.NAME = 'ivan bayross' OR CLIENT_MASTER.NAME = 'mamta muzumdar') GROUP BY SALES_ORDER_DETAILS.PRODUCTNO, PRODUCT_MASTER.DESCRIPTION;
+-----+-----+-----+
| PRODUCTNO | DESCRIPTION | UNITSORDERED |
+-----+-----+-----+
| P00001 | T-shirts | 14 |
| P07965 | denim shirts | 2 |
| P07885 | pull overs | 2 |
| P03453 | shirts | 2 |
| P06734 | cotton jeans | 1 |
+-----+-----+-----+
5 rows in set (0.00 sec)

```

g. Finding the products and their quantities for the orders placed by ClientNo 'C00001' and 'C00003'.

```

mysql> SELECT SALES_ORDER.CLIENTNO, SALES_ORDER_DETAILS.PRODUCTNO, PRODUCT_MASTER.DESCRIPTION, SUM(QTYORDERED) UNITSORDERED FROM SALES_ORDER , SALES_ORDER_DETAILED,PRODUCT_MASTER , CLIENT_MASTER WHERE SALES_ORDER.ORDER_NO = SALES_ORDER_DETAILS.ORDERNO AND SALES_ORDER_DETAILS.PRODUCTNO = PRODUCT_MASTER.PRODUCT_NO AND SALES_ORDER.CLIENTNO = CLIENT_MASTER.CLIENT_NO GROUP BY SALES_ORDER.CLIENTNO, SALES_ORDER_DETAILS.PRODUCTNO, PRODUCT_MASTER.DESCRIPTION HAVING SALES_ORDER.CLIENTNO = "C00001" OR SALES_ORDER.CLIENTNO = "C00002";
+-----+-----+-----+
| CLIENTNO | PRODUCTNO | DESCRIPTION | UNITSORDERED |
+-----+-----+-----+
| C00001 | P00001 | T-shirts | 4 |
| C00001 | P07965 | denim shirts | 2 |
| C00001 | P07885 | pull overs | 2 |
| C00001 | P03453 | shirts | 2 |
| C00001 | P06734 | cotton jeans | 1 |
| C00002 | P00001 | T-shirts | 10 |
+-----+-----+-----+
6 rows in set (0.00 sec)

```

SQL Lab 5

8. Generate the SQL statements to perform the following computations on table data:

- a. List the names of all clients having 'a' as the second letter in their names.

```
mysql> select name from client_master where name like "_a%";  
+-----+  
| name |  
+-----+  
| Hansel Colaco |  
+-----+  
1 row in set (0.00 sec)
```

- b. List the clients who stay in a city whose first letter is 'M'.

```
mysql> select name, city from client_master where city like "M%";  
+-----+-----+  
| name | city |  
+-----+-----+  
| Ivan Bayross | Mumbai |  
| Chhaya Bankar | Mumbai |  
| Hansel Colaco | Mumbai |  
| Deepak Sharma | Mangalore |  
+-----+-----+  
4 rows in set (0.00 sec)
```

- c. List all clients who stay in 'Bangalore' or 'Mangalore'.

```
mysql> select name, city from client_master where city = "bangalore" or city = "mangalore";  
+-----+-----+  
| name | city |  
+-----+-----+  
| Ashwini Joshi | Bangalore |  
| Deepak Sharma | Mangalore |  
+-----+-----+  
2 rows in set (0.01 sec)
```

d. List all clients whose BalDue is greater than value 10000.

```
mysql> select name, baldue from client_master where baldue > 10000;
+-----+-----+
| name | baldue |
+-----+-----+
| Ivan Bayross | 15000.00 |
+-----+-----+
1 row in set (0.00 sec)
```

e. List all information from the Sales_order table for orders placed in the month of June.

```
mysql> select * from sales_order where orderdate like "____-06%";
+-----+-----+-----+-----+-----+-----+-----+-----+
| orderno | clientno | orderdate | delayaddr | salesmanno | delytype | billyn | delydate | orderstatus |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 019001 | C00001 | 2002-06-12 | Delhi | S00001 | F | N | 2002-07-02 | In process |
| 019002 | C00003 | 2002-06-25 | Delhi | S00002 | P | N | 2002-07-27 | Cancelled |
+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

f. List the order information for ClientNo 'C00001' and 'C00003'.

```
mysql> select * from sales_order where clientno = "C00001" or clientno = "C00003";
+-----+-----+-----+-----+-----+-----+-----+-----+
| orderno | clientno | orderdate | delayaddr | salesmanno | delytype | billyn | delydate | orderstatus |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 019001 | C00001 | 2002-06-12 | Delhi | S00001 | F | N | 2002-07-02 | In process |
| 019003 | C00001 | 2002-04-03 | Delhi | S00001 | F | Y | 2002-04-07 | Fulfilled |
| 019002 | C00003 | 2002-06-25 | Delhi | S00002 | P | N | 2002-07-27 | Cancelled |
+-----+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.01 sec)
```

g. List products whose selling price is greater than 500 and less than or equal to 750.

```
mysql> select description, sellprice from product_master where sellprice > 500 and sellprice <= 750;
+-----+-----+
| description | sellprice |
+-----+-----+
| Cotton jeans | 600.00 |
| Jeans | 750.00 |
| Pull overs | 700.00 |
+-----+-----+
3 rows in set (0.00 sec)
```

h. List products whose selling price is more than 500 with the new selling price calculated as, original selling price plus 15%. Rename the new column in the output of the above query as new_price.

```
mysql> select description, sellprice, sellprice+sellprice*0.15 as new_price from product_master where sellprice > 500 and sellprice <= 750;
+-----+-----+-----+
| description | sellprice | new_price |
+-----+-----+-----+
| Cotton jeans | 600.00 | 690.00 |
| Jeans | 750.00 | 862.50 |
| Pull overs | 700.00 | 805.00 |
+-----+-----+-----+
3 rows in set (0.00 sec)
```

i. List the names, city and state of clients who are not in the state of 'Maharashtra'.

```
mysql> select name, city, state from client_master where state not like "Maharashtra";
+-----+-----+-----+
| name      | city     | state    |
+-----+-----+-----+
| Ashwini Joshi | Bangalore | Karnataka |
| Deepak Sharma | Mangalore | Karnataka |
+-----+-----+-----+
2 rows in set (0.00 sec)
```

- j. Count the total no. of orders.

```
mysql> select count(orderno) from sales_order;
+-----+
| count(orderno) |
+-----+
|          5 |
+-----+
1 row in set (0.01 sec)
```

- k. Calculate the average price of all the products.

```
mysql> select avg(costprice) from product_master;
+-----+
| avg(costprice) |
+-----+
|   363.888889 |
+-----+
1 row in set (0.00 sec)
```

- l. Determine the maximum and minimum product prices. Rename the output as Max_Price and Min_Price respectively.

```
mysql> select max(costprice) as max_price, min(costprice) as min_price from product_master;
+-----+-----+
| max_price | min_price |
+-----+-----+
|   550.00 |    175.00 |
+-----+-----+
1 row in set (0.01 sec)
```

- m. Count the no. of products having price less than or equal to 500.

```
mysql> select count(costprice <= 500) from product_master;
+-----+
| count(costprice <= 500) |
+-----+
|          9 |
+-----+
1 row in set (0.00 sec)
```

n. List all the products whose Qtyonhand is less than reorder level.

```
mysql> select description from product_master where qtyonhand < reorderlvl;
+-----+
| description |
+-----+
| crop tops |
+-----+
1 row in set (0.00 sec)
```

9. Exercise on Date manipulation.

a. List the Order No & day on which clients placed their order.

```
mysql> select orderno, orderdate from sales_order;
+-----+-----+
| orderno | orderdate   |
+-----+-----+
| 019001  | 2002-06-12 |
| 019002  | 2002-06-25 |
| 019003  | 2002-04-03 |
| 019008  | 2002-05-24 |
| 046866  | 2002-05-20 |
+-----+-----+
5 rows in set (0.00 sec)
```

b. List the month (in alphabets) and date when the order must be delivered.

```
mysql> select orderno, monthname(delydate) from sales_order;
+-----+-----+
| orderno | monthname(delydate) |
+-----+-----+
| 019001  | July          |
| 019002  | July          |
| 019003  | April         |
| 019008  | July          |
| 046866  | May           |
+-----+-----+
5 rows in set (0.01 sec)
```

c. List the order date in the format 'DD-Month-YY' e.g. '24-February-14'.

```
mysql> select concat(right(orderdate,2),"-",monthname(orderdate),"-",year(orderdate)) as orderdate from sales_order;
+-----+
| orderdate   |
+-----+
| 12-June-2002 |
| 25-June-2002 |
| 03-April-2002 |
| 24-May-2002  |
| 20-May-2002  |
+-----+
5 rows in set (0.00 sec)
```

d. List the date, 15 days after today's date.

```
mysql> select now() + interval 15 day;
+-----+
| now() + interval 15 day |
+-----+
| 2023-01-29 13:03:51      |
+-----+
1 row in set (0.01 sec)
```