Thadomal Shahani Engineering College

Bandra (W.), Mumbai - 400 050.

CERTIFICATE

Certify that Mr./Miss PRATHAM ASNANI of IT Department, Semester III with Roll No. 08 has completed a course of the necessary experiments in the subject SQL LAB under my supervision in the **Thadomal Shahani Engineering College** Laboratory in the year 2023 - 2024

Reshma 31/10/2023

Teacher In- Charge

Head of the Department

Date 30/10/2023 Oct 31, 2023

Principal

CONTENTS

SQL LAB SE IT
NAME :PRATHAM ASNANI
ROLL NO.08
BATCH : S11

Assignment 1

HOSPITAL MANAGEMENT SYSTEM

Doctors: Doctor entity will store their name, specialization and shifts and are identified by unique Doctor_id values. "Experience and education" has attributes of Name of institute and years.

Nurses: Nurse entity will store their name, specialization and shifts and are identified by unique Nurse_id values. "Experience and education" has attributes of Name of institute and years.

Patients: Patient Entity stores Name, Date of birth, Age, Sex, Entry date, Exit date, Fees (Fee status, Amount, insurance etc) and medical history (Symptoms, previous ailments, duration etc.) They are assigned to different doctors based on their symptoms/disease and are uniquely identified by Patient_id values.

Non- medical staff (NMS): NMS entity stores data which includes receptionists, cleaning staff, ward boys, assistants, residency students, ambulance drivers, etc. They have unique NMS_id values.

Rooms: Every room has a unique room_id. A patient is assigned to the room and is also assigned doctors, nurses etc. It will store the entry and exit dates. This will also include special rooms such as OT, ICU, Blood bank, Pathology etc.

MAJOR CHARACTERISTICS:

A patient entering the hospital is assigned a unique patient_id and is then referred to a specific doctor after an initial diagnosis. If the diagnosis is serious, the patient is assigned a doctor, nurses and a room.

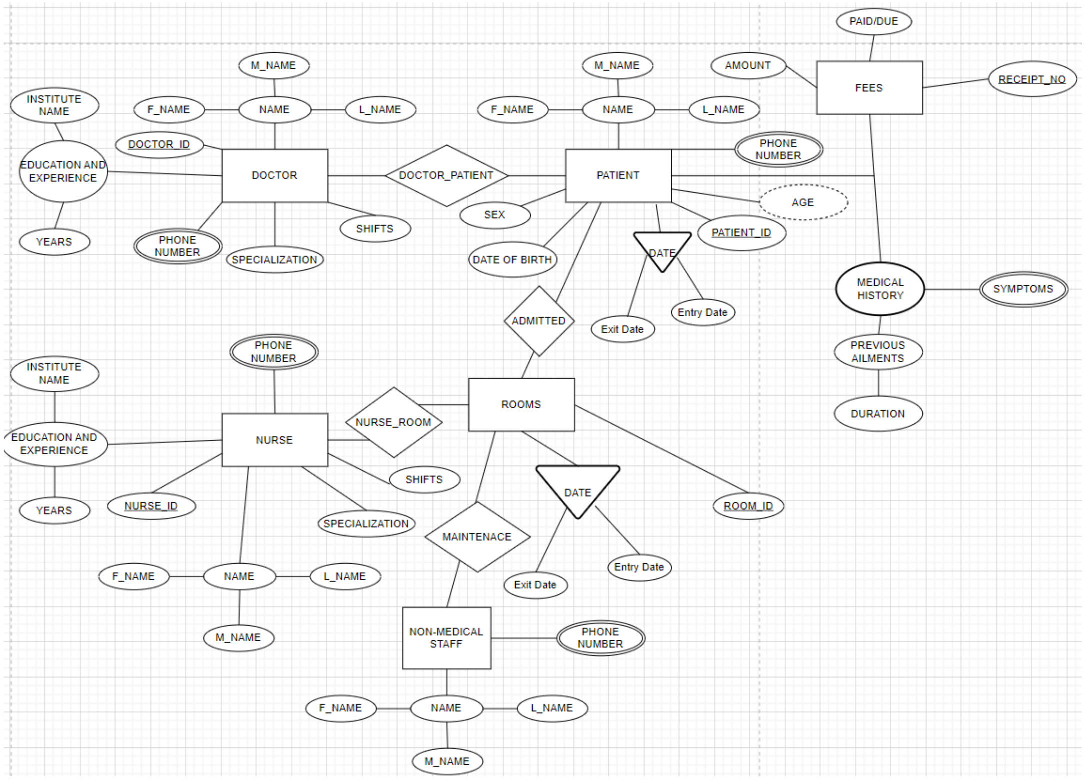
SQL LAB SE IT
NAME :PRATHAM ASNANI
ROLL NO.08
BATCH : S11

Assignment 2

Hospital Management System

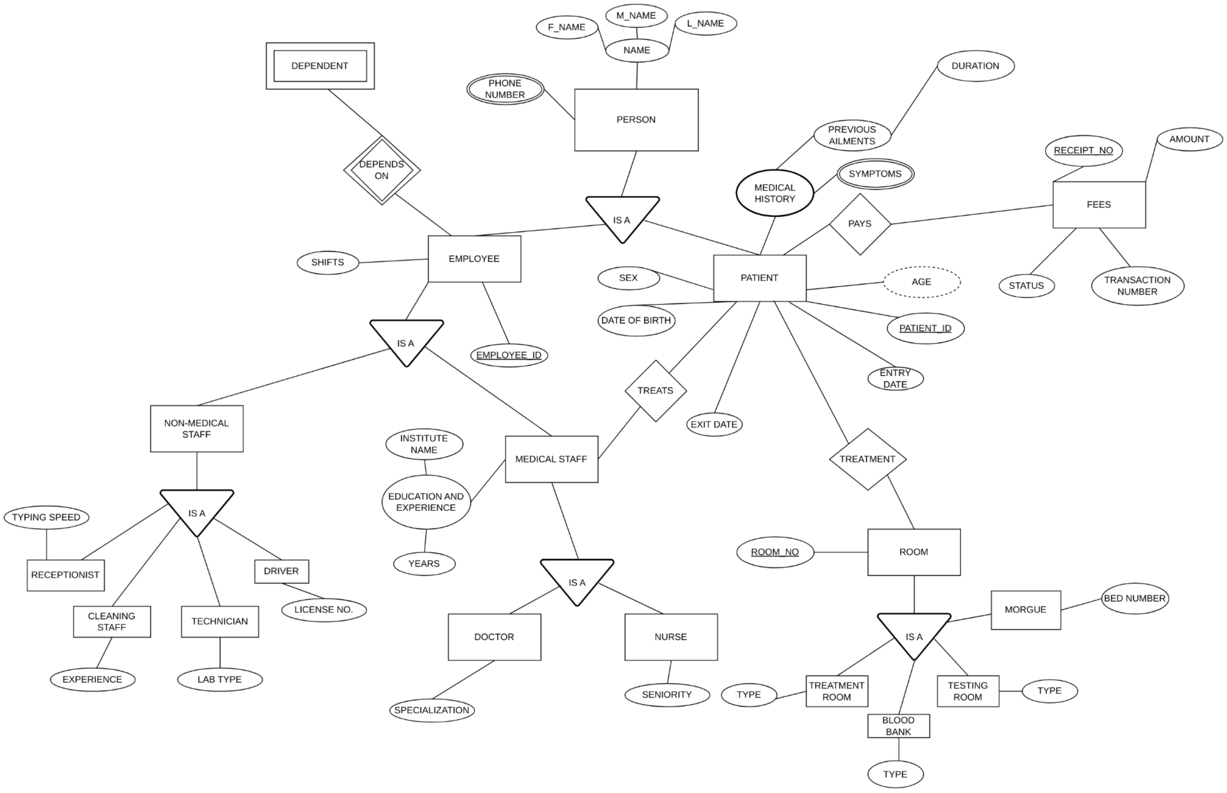
The idea is to design and implement an efficient database management system for a hospital that caters to the needs of managing various aspects of the hospital's operations. The system should be capable of handling the complexity and volume of data usually encountered in a hospital and provide smooth day-to-day operations. The system will have components like

Doctor assignment, Patient management, nurse assignment, room assignment etc. The patient will be diagnosed by a general physician and based on seriousness of illness, they will be assigned doctors, nurses and rooms. Doctors and nurses will have a general profile which will contain personal information like name, ID, qualification and experience, shifts etc. Fields like joining date and leaving date will help in calculating duration of their service. The hospital staff has to be available for the patient if they need them at any point of the day so it is important to keep track of employee shifts and working hours. Position of employment, salary and qualification that need to be maintained. Patients will also have a general profile which will contain information like their name, date of birth, age, previous medical history, symptoms etc. This will allow doctors and nurses to provide personalized treatment and improve the patient's health condition. Rooms will be assigned to the patients according to the seriousness of the illness and length for which the patient has to be accommodated. They can also be assigned to special rooms like ICUs, testing etc. It will manage and show types of rooms and their availability thus avoiding confusion. Separate sections will be there for Non-Medical staff. This will include services like blood bank, ambulances, pathology, receptionist etc. Receptionists will handle the fee payment procedure in collaboration with the patient/ family of patient. Billing Management System will handle accurate billing and invoice for patients including the entry and exit dates. It will show the additional charges for the additional services used like tests, food, air conditioning, ambulance service etc. Over time as the patients use the various services provided by the hospital, the equipment will start demanding maintenance, hence the cleaning and maintenance staff will be made aware about the cleaning schedules, maintenance requests and requirements. Recording and maintaining all these details and information in efficient format is very crucial for running any successful hotel. All these requirements are thus taken care of by the Hospital Management System.



SQL LAB SE IT
NAME :PRATHAM ASNANI
ROLL NO.08
BATCH : S11

Assignment 3



SQL LAB SE IT
NAME :PRATHAM ASNANI
ROLL NO.08
BATCH : S11

Assignment 4

Convert EER to relation

FEES
Receipt_no Status Transaction number Amount

ROOM
Room_id Room_Specialization Type Bed_number

EMPLOYEE
Employee_id Shifts Employee_type F_name M_name L_name Sex
Date of Birth

NON MEDICAL STAFF
Employee_id Role

RECEPTIONIST
Employee_id Typing Speed

CLEANING STAFF
Employee_id Role

TECHNICIAN
Employee_id Role

DRIVER
Employee_id Role

MEDICAL STAFF
Employee_id Institute
name Years of Experience Medical Staff Role Specialization Seniority

PATIENT
Patient_id Entry Date Exit Date Previous Ailments Duration

DEPENDENT
Employee_id Relation Dependent name

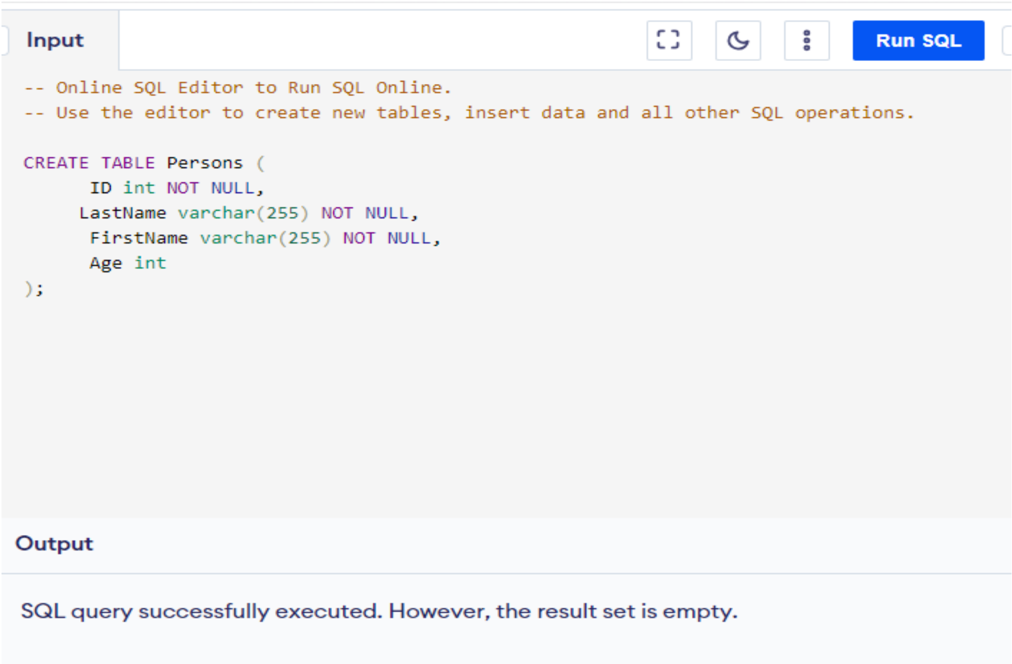
PAYS
Patient_id Receipt_no

TREATS
Employee_id Patient_id

TREATMENT
Patient_id Room_no

SQL LAB SE IT
NAME :PRATHAM ASNANI
ROLL NO.08
BATCH : S11

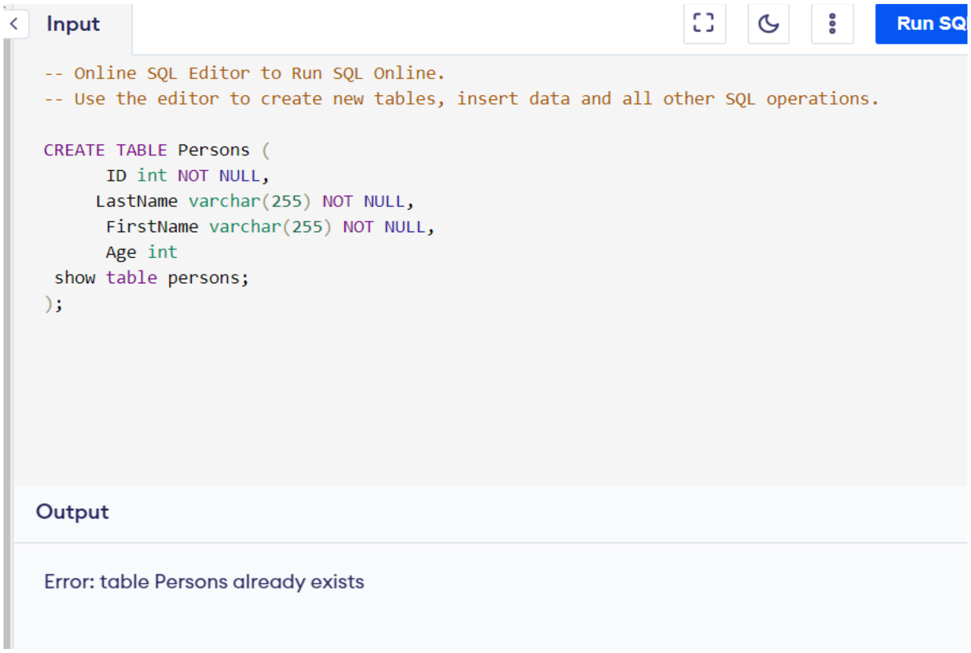
Assignment 5



```
-- Online SQL Editor to Run SQL Online.
-- Use the editor to create new tables, insert data and all other SQL operations.

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

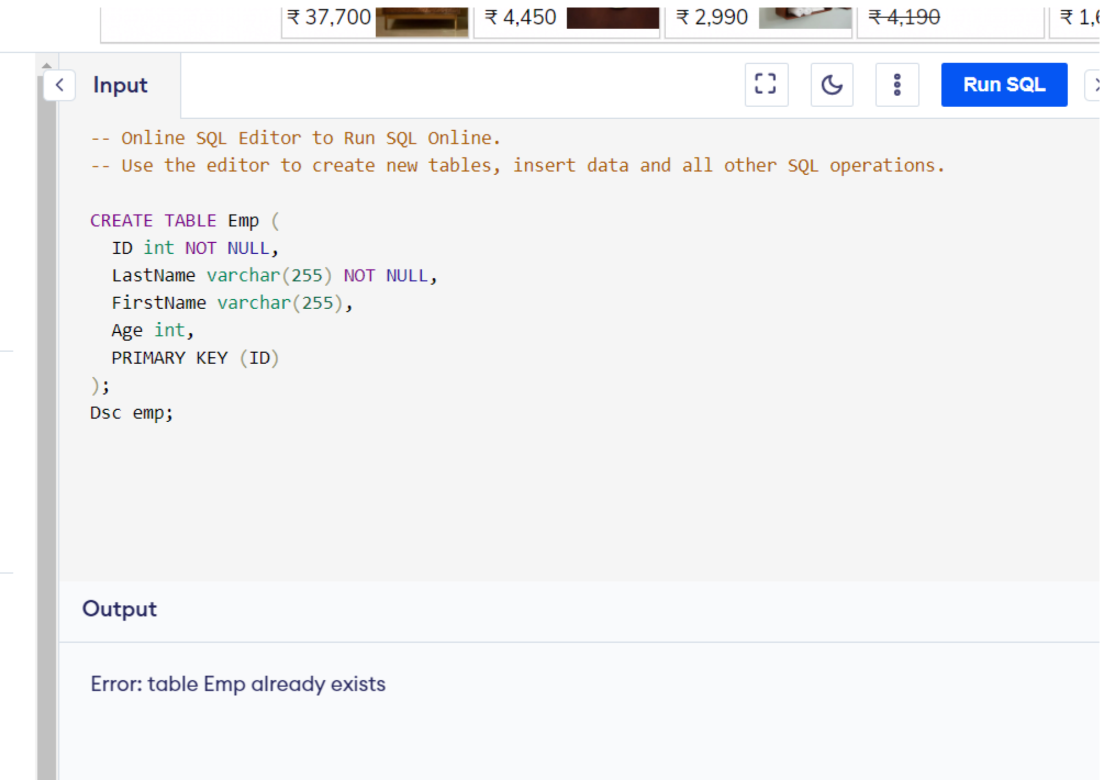
```
SQL query successfully executed. However, the result set is empty.
```



```
-- Online SQL Editor to Run SQL Online.
-- Use the editor to create new tables, insert data and all other SQL operations.

CREATE TABLE Persons (
      ID int NOT NULL,
     LastName varchar(255) NOT NULL,
      FirstName varchar(255) NOT NULL,
      Age int
 show table persons;
);
```

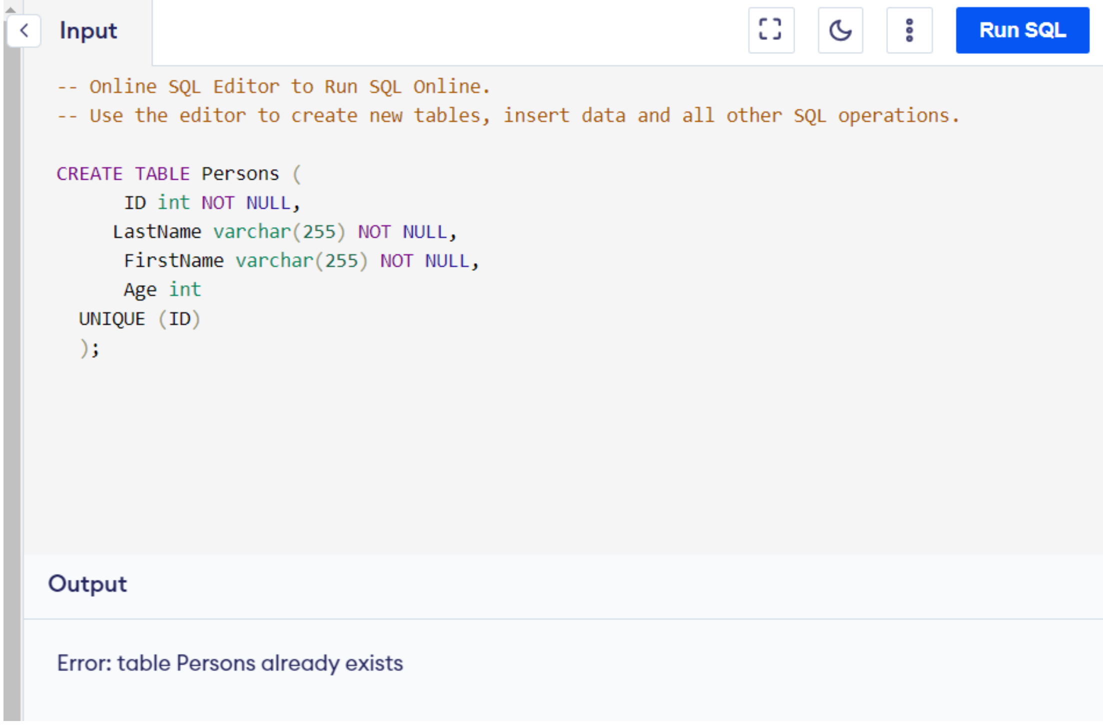
```
Error: table Persons already exists
```



```
-- Online SQL Editor to Run SQL Online.
-- Use the editor to create new tables, insert data and all other SQL operations.

CREATE TABLE Emp (
  ID int NOT NULL,
  LastName varchar(255) NOT NULL,
  FirstName varchar(255),
  Age int,
  PRIMARY KEY (ID)
);
Dsc emp;
```

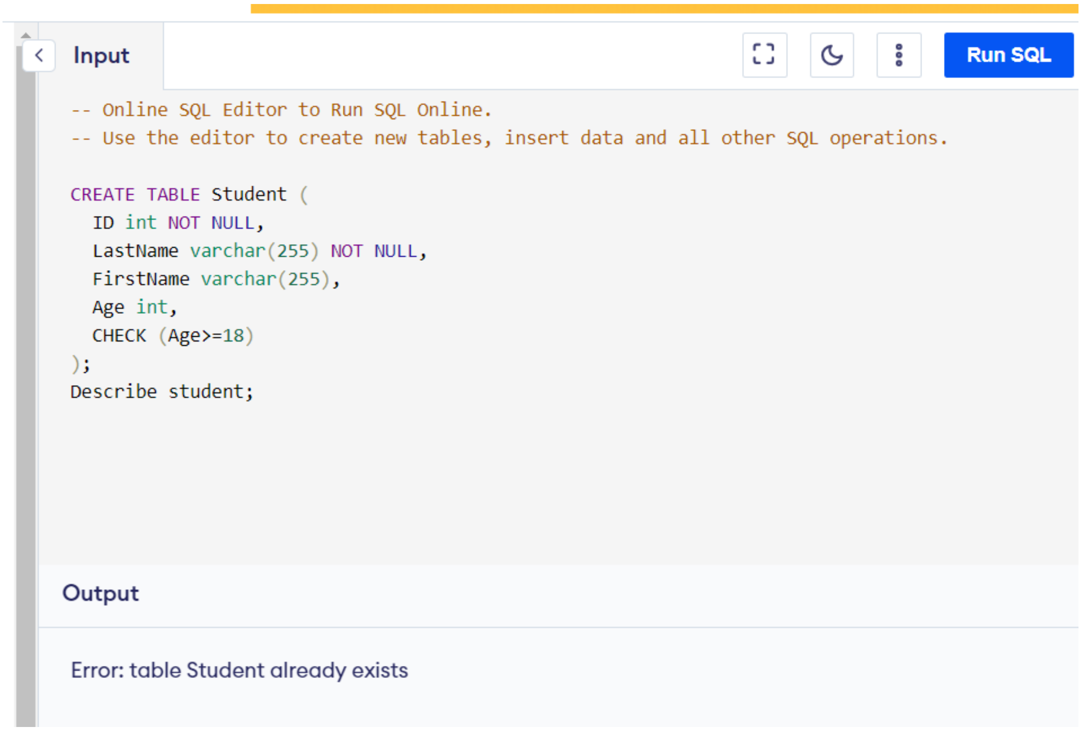
```
Error: table Emp already exists
```



```
-- Online SQL Editor to Run SQL Online.
-- Use the editor to create new tables, insert data and all other SQL operations.

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

```
Error: table Persons already exists
```



```sql
-- Online SQL Editor to Run SQL Online.
-- Use the editor to create new tables, insert data and all other SQL operations.

CREATE TABLE Student (
  ID int NOT NULL,
  LastName varchar(255) NOT NULL,
  FirstName varchar(255),
  Age int,
  CHECK (Age>=18)
);
Describe student;
```

```
Error: table Student already exists
```

PRATHAM ASNANI
SQL LAB
SE IT
S11 08

5A.create tables with given constraint

```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| asnani             |
| car_dealearship    |
| databse            |
| ems                |
| information_schema |
| mysql              |
| payroll            |
| performance_schema |
| pratham            |
| sys                |
+--------------------+
10 rows in set (0.00 sec)

mysql> use PRATHAM;
Database changed
```

1)Client master

```
mysql> Create table Client_master
    -> (
    -> Client_no varchar(6),
    -> Name varchar(20) NOT NULL,
    -> Address1 varchar (30),
    -> Address2 varchar(30),
    -> City varchar(15),
    -> State varchar(15),
    -> Pincode int(6),
    -> Bal_due float(10,2),
    -> primary key(Client_no),
    -> CHECK (Client_no like 'C%')
    -> );
Query OK, 0 rows affected, 2 warnings (0.04 sec)
```

```
mysql> desc Client_master;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| Client_no | varchar(6)  | NO   | PRI | NULL    |       |
| Name      | varchar(20) | NO   |     | NULL    |       |
| Address1  | varchar(30) | YES  |     | NULL    |       |
| Address2  | varchar(30) | YES  |     | NULL    |       |
| City      | varchar(15) | YES  |     | NULL    |       |
| State     | varchar(15) | YES  |     | NULL    |       |
| Pincode   | int         | YES  |     | NULL    |       |
| Bal_due   | float(10,2) | YES  |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
8 rows in set (0.02 sec)
```

2)Product master

```
mysql> create table product_master
    -> (
    -> Product_no varchar(6),
    -> Description varchar(25) NOT NULL,
    -> Profit_percent float(5,2) NOT NULL,
    -> Unit_measure varchar(10) NOT NULL,
    -> Qty_on_hand int(8),
    -> Reorder_lvl int(8) NOT NULL,
    -> Sell_price float(8,2) NOT NULL CHECK(Sell_price<>0),
    -> Cost_price float(8,2) NOT NULL CHECK(Cost_price<>0),
    -> primary key(Product_no),
    -> CHECK (Product_no like 'P%')
    -> );
Query OK, 0 rows affected, 5 warnings (0.04 sec)
mysql> desc Product_master;
+----------------+-------------+------+-----+---------+-------+
| Field          | Type        | Null | Key | Default | Extra |
+----------------+-------------+------+-----+---------+-------+
| Product_no     | varchar(6)  | NO   | PRI | NULL    |       |
| Description    | varchar(25) | NO   |     | NULL    |       |
| Profit_percent | float(5,2)  | NO   |     | NULL    |       |
| Unit_measure   | varchar(10) | NO   |     | NULL    |       |
| Qty_on_hand    | int         | YES  |     | NULL    |       |
| Reorder_lvl    | int         | NO   |     | NULL    |       |
| Sell_price     | float(8,2)  | NO   |     | NULL    |       |
| Cost_price     | float(8,2)  | NO   |     | NULL    |       |
+----------------+-------------+------+-----+---------+-------+
8 rows in set (0.01 sec)
```

3)Salesman master

```
mysql> create table salesman_master
    -> (
    -> Salesman_no varchar(6),
    -> Salesman_name varchar(20)NOT NULL,
    -> Address1 varchar(30)NOT NULL,
    -> Address2 varchar(30),
    -> City varchar(20),
    -> Pincode varchar(6),
    -> State varchar(20),
    -> Sal_amt float(8,2)NOT NULL CHECK(Sal_amt<>0),
    -> Tgt_to_get float(6,2)NOT NULL CHECK(Tgt_to_get<>0),
    -> Ytd_sales float(6,2)NOT NULL,Remarks varchar(60),
    -> primary key(Salesman_no),CHECK (Salesman_no like 'S%'));
Query OK, 0 rows affected, 3 warnings (0.04 sec)

mysql> desc Salesman_master;
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| Salesman_no   | varchar(6)  | NO   | PRI | NULL    |       |
| Salesman_name | varchar(20) | NO   |     | NULL    |       |
| Address1      | varchar(30) | NO   |     | NULL    |       |
| Address2      | varchar(30) | YES  |     | NULL    |       |
| City          | varchar(20) | YES  |     | NULL    |       |
| Pincode       | varchar(6)  | YES  |     | NULL    |       |
| State         | varchar(20) | YES  |     | NULL    |       |
| Sal_amt       | float(8,2)  | NO   |     | NULL    |       |
| Tgt_to_get    | float(6,2)  | NO   |     | NULL    |       |
| Ytd_sales     | float(6,2)  | NO   |     | NULL    |       |
| Remarks       | varchar(60) | YES  |     | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
11 rows in set (0.01 sec)
```

4)sales order

```
mysql> create table sales_order
    -> (
    -> S_order_no varchar(6),
    -> S_order_date date,
    -> Client_no varchar(6),
    -> Dely_addr varchar(25),
    -> Salesman_no varchar(6),
    -> Dely_type char(1) CHECK(Dely_type in ('P','F','D')) ,
    -> Billed_yn char(1),
    -> Dely_date date,CHECK (Dely_date>=S_order_date),
    -> Order_status varchar(10)CHECK(order_status in('IP','F','B','C')),
    -> primary key(S_order_no),
    -> foreign key(Client_no)references Client_master(Client_no),
    -> foreign key(Salesman_no )references salesman_master(Salesman_no )
    -> );
Query OK, 0 rows affected (0.06 sec)
mysql> desc sales_order ;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| S_order_no   | varchar(6)  | NO   | PRI | NULL    |       |
| S_order_date | date        | YES  |     | NULL    |       |
| Client_no    | varchar(6)  | YES  | MUL | NULL    |       |
| Dely_addr    | varchar(25) | YES  |     | NULL    |       |
| Salesman_no  | varchar(6)  | YES  | MUL | NULL    |       |
| Dely_type    | char(1)     | YES  |     | NULL    |       |
| Billed_yn    | char(1)     | YES  |     | NULL    |       |
| Dely_date    | date        | YES  |     | NULL    |       |
| Order_status | varchar(10) | YES  |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
9 rows in set (0.01 sec)
```

5)sales order details

```
mysql> create table sales_order_details
    -> (
    -> S_order_no varchar(6),
    -> Product_no varchar(6),
    -> Qty_ordered int(8),
    -> Qty_disp int(8),
    -> Product_rate float(10,2),
    -> primary key(S_order_no,Product_no),
    -> foreign key(S_order_no)references sales_order(S_order_no),
    -> foreign key(Product_no)references product_master(Product_no)
    -> );
Query OK, 0 rows affected, 3 warnings (0.06 sec)
mysql> desc sales_order_details;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| S_order_no   | varchar(6)  | NO   | PRI | NULL    |       |
| Product_no   | varchar(6)  | NO   | PRI | NULL    |       |
| Qty_ordered  | int         | YES  |     | NULL    |       |
| Qty_disp     | int         | YES  |     | NULL    |       |
| Product_rate | float(10,2) | YES  |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
5 rows in set (0.01 sec)
```

6)challan header

```
mysql> create table Challan_Header
    -> (
    -> Challan_no varchar(6),
    -> S_order_no varchar(6),
    -> Challan_date date NOT NULL,
    -> Billed_yn char(1) DEFAULT ('N') check(Billed_yn in ('Y','N')),
    -> foreign key (S_order_no) references sales_order(S_order_no),
    -> primary key(Challan_no)
    -> );
Query OK, 0 rows affected (0.04 sec)
mysql> desc Challan_Header;
+--------------+------------+------+-----+-------------+-------------------+
| Field        | Type       | Null | Key | Default     | Extra             |
+--------------+------------+------+-----+-------------+-------------------+
| Challan_no   | varchar(6) | NO   | PRI | NULL        |                   |
| S_order_no   | varchar(6) | YES  | MUL | NULL        |                   |
| Challan_date | date       | NO   |     | NULL        |                   |
| Billed_yn    | char(1)    | YES  |     | _cp850\'N\' | DEFAULT_GENERATED |
+--------------+------------+------+-----+-------------+-------------------+
4 rows in set (0.01 sec)
```

7)challan details

```
mysql> create table Challan_details
    -> (
    -> Challan_no varchar(6) ,
    -> Product_no varchar(6) ,
    -> Qty_disp int(8),
    -> primary key(Qty_disp) ,
    -> foreign key(Challan_no) references Challan_Header(Challan_no),
    -> foreign key(Product_no) references product_master(Product_no)
    -> );
Query OK, 0 rows affected, 1 warning (0.03 sec)
mysql> desc Challan_details;
+------------+------------+------+-----+---------+-------+
| Field      | Type       | Null | Key | Default | Extra |
+------------+------------+------+-----+---------+-------+
| Challan_no | varchar(6) | YES  | MUL | NULL    |       |
| Product_no | varchar(6) | YES  | MUL | NULL    |       |
| Qty_disp   | int        | NO   | PRI | NULL    |       |
+------------+------------+------+-----+---------+-------+
3 rows in set (0.01 sec)
```

8)show tables

```
mysql> show tables;
+---------------------+
| Tables_in_pratham   |
+---------------------+
| challan_details     |
| challan_header      |
| client_master       |
| product_master      |
| sales_order         |
| sales_order_details |
| salesman_master     |
+---------------------+
7 rows in set (0.01 sec)
```

PRATHAM ASNANI
SQL LAB
SE IT
S11 08

6.Insert Records to all the tables.

1)INSERT INTO CLIENT MASTER

```
mysql> INSERT INTO
client_master(Client_no,Name,City,State,Pincode,Bal_due)values('C00001','Ivan
    '> Bayross','Bombay','Maharashtra',400054,15000);
Query OK, 1 row affected (0.02 sec)

mysql> INSERT INTO
client_master(Client_no,Name,City,State,Pincode,Bal_due)values('C00002','Vandana
    '> Saitwal','Madras','Tamil Nadu',780001,0);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO
client_master(Client_no,Name,City,State,Pincode,Bal_due)values('C00003','Pramada
    '> Jaguste','Bombay','Maharashtra',400057,5000);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO
client_master(Client_no,Name,City,State,Pincode,Bal_due)values('C00004','Basu
    '> Navindgi','Bombay','Maharashtra',400056,0);
Query OK, 1 row affected (0.00 sec)
```

```
mysql> INSERT INTO
client_master(Client_no,Name,City,State,Pincode,Bal_due)values('C00005','Ravi
    '> Sreedharan','Delhi','',100001,2000);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO
    ->
client_master(Client_no,Name,City,State,Pincode,Bal_due)values('C00006','Rukmini','Bombay','Ma
harashtr
    '> a',400050,15000);
Query OK, 1 row affected (0.00 sec)

mysql> SELECT*FROM client_master;
+-----------+-----------------+----------+----------+--------+--------------+---------+----------+
| Client_no | Name            | Address1 | Address2 | City   | State        | Pincode | Bal_due  |
+-----------+-----------------+----------+----------+--------+--------------+---------+----------+
| C00001    | Ivan
Bayross     | NULL     | NULL     | Bombay | Maharashtra  |  400054 | 15000.00 |
| C00002    | Vandana
Saitwal | NULL     | NULL     | Madras | Tamil Nadu   |  780001 |     0.00 |
| C00003    | Pramada
Jaguste | NULL     | NULL     | Bombay | Maharashtra  |  400057 |  5000.00 |
| C00004    | Basu
Navindgi    | NULL     | NULL     | Bombay | Maharashtra  |  400056 |     0.00 |
| C00005    | Ravi
Sreedharan | NULL     | NULL     | Delhi  |              |  100001 |  2000.00 |
| C00006    | Rukmini         | NULL     | NULL     | Bombay | Maharashtra
a |  400050 | 15000.00 |
+-----------+-----------------+----------+----------+--------+--------------+---------+----------+
6 rows in set (0.00 sec)
```

2)INSERT INTO PRODUCT MASTER

```
mysql> insert into product_master values('P00001','1.44 Floppies',5,'Piece',100,20,525,500);
Query OK, 1 row affected (0.01 sec)

mysql> insert into product_master values('P03453','Monitors',6,'Piece',10,3,12000,11280);
Query OK, 1 row affected (0.00 sec)

mysql> insert into product_master values('P06734','Mouse',5,'Piece',20,5,1050,1000);
Query OK, 1 row affected (0.00 sec)

mysql> insert into product_master values('P07865','1.22 Floppies',5,'Piece',100,20,525,500);
```

```
Query OK, 1 row affected (0.00 sec)

mysql> insert into product_master values('P07868','Keyboards',2,'Piece',10,3,3150,3050);
Query OK, 1 row affected (0.00 sec)

mysql> insert into product_master values('P07885','CD Drive',2.5,'Piece',10,3,5250,5100);
Query OK, 1 row affected (0.00 sec)

mysql> insert into product_master values('P07965','HDD',4,'Piece',10,3,8400,8000);
Query OK, 1 row affected (0.00 sec)

mysql> insert into product_master values('P07975','1.44 Drive',5,'Piece',10,3,1050,1000);
Query OK, 1 row affected (0.00 sec)

mysql> insert into product_master values('P08865','1.22 Drive',5,'Piece',2,3,1050,1000);
Query OK, 1 row affected (0.00 sec)

mysql> SELECT*FROM product_master;
+------------+---------------+----------------+--------------+-------------+-------------+------------+------------+
| Product_no | Description   | Profit_percent | Unit_measure | Qty_on_hand | Reorder_lvl |
Sell_price | Cost_price |
+------------+---------------+----------------+--------------+-------------+-------------+------------+------------+
| P00001     | 1.44 Floppies |           5.00 | Piece        |         100 |          20 |     525.00 |     500.00 |
| P03453     | Monitors      |           6.00 | Piece        |          10 |           3 |   12000.00 |   11280.00 |
| P06734     | Mouse         |           5.00 | Piece        |          20 |           5 |    1050.00 |    1000.00 |
| P07865     | 1.22 Floppies |           5.00 | Piece        |         100 |          20 |     525.00 |     500.00 |
| P07868     | Keyboards     |           2.00 | Piece        |          10 |           3 |    3150.00 |    3050.00 |
| P07885     | CD Drive      |           2.50 | Piece        |          10 |           3 |    5250.00 |    5100.00 |
| P07965     | HDD           |           4.00 | Piece        |          10 |           3 |    8400.00 |    8000.00 |
| P07975     | 1.44 Drive    |           5.00 | Piece        |          10 |           3 |    1050.00 |    1000.00 |
| P08865     | 1.22 Drive    |           5.00 | Piece        |           2 |           3 |    1050.00 |    1000.00 |
+------------+---------------+----------------+--------------+-------------+-------------+------------+------------+
9 rows in set (0.00 sec)
```

3)INSERT INTO SALESMAN MASTER

```
mysql> insert into salesman_master values('S00001', 'Kiran', 'A/14', 'Worli', 'Bombay', '400002',
'MAH', 3000,
    -> 100, 50,'Good');
Query OK, 1 row affected (0.02 sec)

mysql> insert into salesman_master values('S00002', 'Manish', '65', 'Nariman', 'Bombay', '400001',
'MAH', 3000,
   -> 200, 100,'Good');
Query OK, 1 row affected (0.00 sec)

mysql> insert into salesman_master  values('S00003', 'Ravi', 'P-7', 'Bandra', 'Bombay', '400032',
'MAH', 3000, 200,
   -> 100,'Good');
Query OK, 1 row affected (0.00 sec)

mysql> insert into salesman_master values('S00004', 'Ashish', 'A/5', 'Juhu', 'Bombay', '400044',
'MAH', 3000,
   -> 200, 150,'Good');
Query OK, 1 row affected (0.01 sec)

mysql> SELECT*FROM salesman_master;
+-------------+---------------+----------+----------+--------+---------+-------+---------+------------+-----------+---------+
| Salesman_no | Salesman_name | Address1 | Address2 | City   | Pincode | State | Sal_amt |
Tgt_to_get | Ytd_sales | Remarks |
+-------------+---------------+----------+----------+--------+---------+-------+---------+------------+-----------+---------+
| S00001      | Kiran         | A/14     | Worli    | Bombay | 400002  | MAH   | 3000.00 |     100.00 |
50.00 | Good    |
| S00002      | Manish        | 65       | Nariman  | Bombay | 400001  | MAH   | 3000.00 |     200.00 |
100.00 | Good    |
| S00003      | Ravi          | P-7      | Bandra   | Bombay | 400032  | MAH   | 3000.00 |     200.00 |
100.00 | Good    |
| S00004      | Ashish        | A/5      | Juhu     | Bombay | 400044  | MAH   | 3000.00 |     200.00 |
150.00 | Good    |
+-------------+---------------+----------+----------+--------+---------+-------+---------+------------+-----------+---------+
4 rows in set (0.00 sec)
```

4)INSERT INTO SALES ORDER

```
mysql> insert into
sales_order(S_order_no,S_order_date,Client_no,Salesman_no,Dely_type,Billed_yn,Dely_date,Orde
r_status)
   -> values('O19002','1996-01-25','C00002','S00002','P','N','1996-01-27','C');
Query OK, 1 row affected (0.02 sec)

mysql> insert into
   ->
sales_order(S_order_no,S_order_date,Client_no,Salesman_no,Dely_type,Billed_yn,Dely_date,Orde
r_status) values('O46865','1996-02-18','C00003','S00003','F','Y','1996-02-20','F');
Query OK, 1 row affected (0.01 sec)
```

```
mysql> insert into
    ->
sales_order(S_order_no,S_order_date,Client_no,Salesman_no,Dely_type,Billed_yn,Dely_date,Orde
r_status) values('O19003','1996-04-03','C00001','S00001','F','Y','1996-04-07','F');
Query OK, 1 row affected (0.00 sec)

mysql> insert into
    ->
sales_order(S_order_no,S_order_date,Client_no,Salesman_no,Dely_type,Billed_yn,Dely_date,Orde
r_status) values('O46866','1996-05-20','C00004','S00002','P','N','1996-05-22','C');
Query OK, 1 row affected (0.00 sec)

mysql> insert into
    ->
sales_order(S_order_no,S_order_date,Client_no,Salesman_no,Dely_type,Billed_yn,Dely_date,Orde
r_status) values('O10008','1996-05-24','C00005','S00004','F','N','1996-05-26','IP');
Query OK, 1 row affected (0.00 sec)

mysql> SELECT*FROM sales_order;
+------------+--------------+-----------+-----------+-------------+-----------+-----------+------------+--------------+
| S_order_no | S_order_date | Client_no | Dely_addr | Salesman_no | Dely_type | Billed_yn |
Dely_date  | Order_status |
+------------+--------------+-----------+-----------+-------------+-----------+-----------+------------+--------------+
| O10008     | 1996-05-24   | C00005    | NULL      | S00004      | F         | N         | 1996-05-26 | IP
|
| O19002     | 1996-01-25   | C00002    | NULL      | S00002      | P         | N         | 1996-01-27 | C
|
| O19003     | 1996-04-03   | C00001    | NULL      | S00001      | F         | Y         | 1996-04-07 | F
|
| O46865     | 1996-02-18   | C00003    | NULL      | S00003      | F         | Y         | 1996-02-20 | F
|
| O46866     | 1996-05-20   | C00004    | NULL      | S00002      | P         | N         | 1996-05-22 | C
|
+------------+--------------+-----------+-----------+-------------+-----------+-----------+------------+--------------+
5 rows in set (0.00 sec)
```

5)INSERT INTO SALES ORDER DETAILS

```
mysql> insert into sales_order_details values ('O19001', 'P00001', 4,4,525);
Query OK, 1 row affected (0.02 sec)

mysql> insert into sales_order_details values ('O19001', 'P07965', 2,1,8400);
```

```
Query OK, 1 row affected (0.00 sec)

mysql> insert into sales_order_details values ('O19001', 'P07885', 2,1,5250);
Query OK, 1 row affected (0.00 sec)

mysql> insert into sales_order_details values ('O19002', 'P00001', 10,0,525);
Query OK, 1 row affected (0.00 sec)

mysql> insert into sales_order_details values ('O46865', 'P07868', 3,3,3150);
Query OK, 1 row affected (0.00 sec)

mysql> insert into sales_order_details values ('O46865', 'P07885', 3,1,5250);
Query OK, 1 row affected (0.00 sec)

mysql> insert into sales_order_details values ('O46865', 'P00001', 10,10,525);
Query OK, 1 row affected (0.00 sec)

mysql> insert into sales_order_details values ('O46865', 'P03453', 4,4,1050);
Query OK, 1 row affected (0.00 sec)

mysql> insert into sales_order_details values ('O19003', 'P03453', 2,2,1050);
Query OK, 1 row affected (0.00 sec)

mysql> insert into sales_order_details values ('O19003', 'P06734', 1,1,12000);
Query OK, 1 row affected (0.00 sec)

mysql> insert into sales_order_details values ('O46866', 'P07965', 1,0,8400);
Query OK, 1 row affected (0.00 sec)

mysql> insert into sales_order_details values ('O46866', 'P07975', 1,0,1050);
Query OK, 1 row affected (0.00 sec)

mysql> insert into sales_order_details values ('O10008', 'P00001', 10,5,525);
Query OK, 1 row affected (0.00 sec)

mysql> insert into sales_order_details values ('O10008', 'P07975', 5,3,1050);
Query OK, 1 row affected (0.01 sec)


mysql> SELECT*FROM sales_order_details;
+------------+------------+-------------+----------+--------------+
| S_order_no | Product_no | Qty_ordered | Qty_disp | Product_rate |
+------------+------------+-------------+----------+--------------+
| O10008     | P00001     |          10 |        5 |       525.00 |
| O10008     | P07975     |           5 |        3 |      1050.00 |
| O19001     | P00001     |           4 |        4 |       525.00 |
| O19001     | P07885     |           2 |        1 |      5250.00 |
| O19001     | P07965     |           2 |        1 |      8400.00 |
| O19002     | P00001     |          10 |        0 |       525.00 |
| O19003     | P03453     |           2 |        2 |      1050.00 |
| O19003     | P06734     |           1 |        1 |     12000.00 |
| O46865     | P00001     |          10 |       10 |       525.00 |
| O46865     | P03453     |           4 |        4 |      1050.00 |
| O46865     | P07868     |           3 |        3 |      3150.00 |
| O46865     | P07885     |           3 |        1 |      5250.00 |
| O46866     | P07965     |           1 |        0 |      8400.00 |
| O46866     | P07975     |           1 |        0 |      1050.00 |
+------------+------------+-------------+----------+--------------+
14 rows in set (0.00 sec)
```

6)INSERT INTO CHALLAN HEADER

```
insert into Challan_Header values('CH9001', 'O19001', '1995-12-12', 'Y');
insert into Challan_Header values('CH6865', 'O46865', '1995-11-12', 'Y');
insert into Challan_Header  values('CH3965', 'O10008', '1995-11-12', 'Y');
mysql> SELECT*FROM Challan_Header;
+------------+------------+--------------+-----------+
| Challan_no | S_order_no | Challan_date | Billed_yn |
+------------+------------+--------------+-----------+
| CH3965     | O10008     | 1995-11-12   | Y         |
| CH6865     | O46865     | 1995-11-12   | Y         |
| CH9001     | O19001     | 1995-12-12   | Y         |
+------------+------------+--------------+-----------+
3 rows in set (0.00 sec)
```

7)INSERT INTO CHALLAN VALUES

```
insert into Challan_details values('CH9001', 'P00001', 4);
insert into Challan_details values('CH9001', 'P07965', 1);
insert into Challan_details values('CH6865', 'P07868', 3);
insert into Challan_details values('CH6865', 'P00001', 10);
insert into Challan_details  values('CH3965', 'P00001', 5);
insert into Challan_details  values('CH3965', 'P07975', 2);

mysql> select*from Challan_details;
+------------+------------+----------+
| Challan_no | Product_no | Qty_disp |
+------------+------------+----------+
| CH9001     | P07965     |        1 |
| CH3965     | P07975     |        2 |
| CH6865     | P07868     |        3 |
| CH9001     | P00001     |        4 |
| CH3965     | P00001     |        5 |
| CH6865     | P00001     |       10 |
+------------+------------+----------+
6 rows in set (0.00 sec)
```

PRATHAM ASNANI
SQL LAB
SE IT
S11 08

7.Sql queries

1)Find out the names of all clients.

```
mysql> select name from Client_master;
+-----------------+
| name            |
+-----------------+
| Ivan
Bayross     |
| Vandana
Saitwal |
| Pramada
Jaguste |
| Basu
Navindgi    |
| Ravi
Sreedharan |
| Rukmini         |
+-----------------+
6 rows in set (0.00 sec)
```

2)Print the entire client_master table

```
mysql> select * from Client_master;
+-----------+-----------------+----------+----------+--------+-------------+---------+----------+
| Client_no | Name            | Address1 | Address2 | City   | State       | Pincode | Bal_due  |
+-----------+-----------------+----------+----------+--------+-------------+---------+----------+
| C00001    | Ivan
Bayross    | NULL     | NULL     | Bombay | Maharashtra  |  400054 | 15000.00 |
| C00002    | Vandana
Saitwal | NULL     | NULL     | Madras | Tamil Nadu   |  780001 |     0.00 |
| C00003    | Pramada
Jaguste | NULL     | NULL     | Bombay | Maharashtra  |  400057 |  5000.00 |
| C00004    | Basu
Navindgi   | NULL     | NULL     | Bombay | Maharashtra  |  400056 |     0.00 |
| C00005    | Ravi
Sreedharan | NULL     | NULL     | Delhi  |             |  100001 |  2000.00 |
| C00006    | Rukmini         | NULL     | NULL     | Bombay | Maharashtr
a |  400050 | 15000.00 |
+-----------+-----------------+----------+----------+--------+-------------+---------+----------+
6 rows in set (0.00 sec)
```

3)Retrieve the list of names and the cities of all the clients

```
mysql> select name,city from Client_master;
+-----------------+--------+
| name            | city   |
+-----------------+--------+
| Ivan
Bayross    | Bombay |
| Vandana
Saitwal | Madras |
| Pramada
Jaguste | Bombay |
| Basu
Navindgi   | Bombay |
| Ravi
Sreedharan | Delhi  |
| Rukmini         | Bombay |
+-----------------+--------+
6 rows in set (0.00 sec)
```

4)List the various products available from the product_master

```
mysql> select Description from product_master;
+---------------+
| Description   |
+---------------+
| 1.44 Floppies |
| Monitors      |
| Mouse         |
| 1.22 Floppies |
| Keyboards      |
| CD Drive       |
| HDD            |
| 1.44 Drive     |
| 1.22 Drive     |
+---------------+
9 rows in set (0.00 sec)
```

5)Find the name of all clients having 'a' as the second letter in their names

```
mysql> select name from Client_master where name like '_a%' ;
+-----------------+
| name            |
+-----------------+
| Vandana
Saitwal |
| Basu
Navindgi   |
| Ravi
Sreedharan |
+-----------------+
3 rows in set (0.00 sec)
```

6)Find out the clients who stay in city whose second letter is 'a'

```
mysql> select name,city from Client_master where city like '_a%' ;
+-----------------+--------+
| name            | city   |
+-----------------+--------+
| Vandana
Saitwal | Madras |
+-----------------+--------+
1 row in set (0.00 sec)
```

7)Find the list of all clients who stay in bombay or city delhi or city madras

```
mysql> select name,city from Client_master where city="Bombay" OR city="Delhi" OR
    -> city="Madras";
+-----------------+--------+
| name            | city   |
+-----------------+--------+
| Ivan
Bayross    | Bombay |
| Vandana
Saitwal | Madras |
| Pramada
Jaguste | Bombay |
| Basu
Navindgi   | Bombay |
| Ravi
Sreedharan | Delhi  |
| Rukmini         | Bombay |
+-----------------+--------+
6 rows in set (0.01 sec)
```

8)List all the clients who are located in 'Bombay'

```
mysql> select name,city from Client_master where city="Bombay" ;
+-----------------+--------+
| name            | city   |
+-----------------+--------+
| Ivan
Bayross    | Bombay |
| Pramada
Jaguste | Bombay |
| Basu
Navindgi   | Bombay |
| Rukmini         | Bombay |
+-----------------+--------+
4 rows in set (0.00 sec)
```

9)Print the list of clients whose bal_due are greater than value 10000

```
mysql> select name, Bal_due from Client_master where Bal_due>10000;
+--------------+----------+
| name         | Bal_due  |
+--------------+----------+
| Ivan
Bayross | 15000.00 |
| Rukmini      | 15000.00 |
+--------------+----------+
2 rows in set (0.01 sec)
```

10)Print information for client number C00001 & C00002

```
mysql> select * from sales_order where S_order_date BETWEEN '1996-01-01' AND
'1996-03-31';
+------------+--------------+-----------+-----------+-------------+-----------+-----------+------------+--------------+
| S_order_no | S_order_date | Client_no | Dely_addr | Salesman_no | Dely_type | Billed_yn |
Dely_date  | Order_status |
+------------+--------------+-----------+-----------+-------------+-----------+-----------+------------+--------------+
| O19001     | 1996-01-12   | C00001    | NULL      | S00001      | F         | N         | 1996-01-20 | IP
|
| O19002     | 1996-01-25   | C00002    | NULL      | S00002      | P        | N        | 1996-01-27 | C
|
| O46865     | 1996-02-18   | C00003    | NULL      | S00003      | F        | Y        | 1996-02-20 | F
|
+------------+--------------+-----------+-----------+-------------+-----------+-----------+------------+--------------+
3 rows in set (0.00 sec)
```

11)Display order information for client_no 'c00001' and 'c00002'

```
mysql> select * from Client_master where Client_no='C00001' OR
    -> Client_no='C00002';
+-----------+-----------------+----------+----------+--------+-------------+---------+----------+
| Client_no | Name            | Address1 | Address2 | City   | State       | Pincode | Bal_due  |
+-----------+-----------------+----------+----------+--------+-------------+---------+----------+
| C00001    | Ivan
Bayross    | NULL     | NULL     | Bombay | Maharashtra |  400054 | 15000.00 |
| C00002    | Vandana
Saitwal | NULL     | NULL     | Madras | Tamil Nadu  |  780001 |     0.00 |
+-----------+-----------------+----------+----------+--------+-------------+---------+----------+
2 rows in set (0.01 sec)
```

12)Find the products with description as '1.44 drive' and '1.22 drive'

```
mysql> select*from Product_master
    -> where Sell_price BETWEEN 2000 AND 5000;
+------------+-------------+----------------+--------------+-------------+-------------+------------+------------+
| Product_no | Description | Profit_percent | Unit_measure | Qty_on_hand | Reorder_lvl |
Sell_price | Cost_price |
+------------+-------------+----------------+--------------+-------------+-------------+------------+------------+
| P07868     | Keyboards   |           2.00 | Piece        |          10 |           3 |    3150.00 |    3050.00 |
+------------+-------------+----------------+--------------+-------------+-------------+------------+------------+
1 row in set (0.01 sec)
```

13)Find the product whose selling price is greater than 2000 and less than or equal to 5000

```
mysql> select Product_no, Description from product_master
    -> where Description='1.44
    '> Drive' or Description='1.22 Drive';
+------------+-------------+
| Product_no | Description |
+------------+-------------+
| P08865     | 1.22 Drive  |
+------------+-------------+
1 row in set (0.00 sec)
```

14)Find the product whose selling price is more than 1500 and also find the new selling price as original price *15

```
mysql> select Product_no,Description,Sell_price, Sell_price*15 as original_price
    -> from product_master
    -> where Sell_price >1500;
+------------+-------------+------------+----------------+
| Product_no | Description | Sell_price | original_price |
+------------+-------------+------------+----------------+
| P03453     | Monitors    |   12000.00 |      180000.00 |
| P07868     | Keyboards   |    3150.00 |       47250.00 |
| P07885     | CD Drive    |    5250.00 |       78750.00 |
| P07965     | HDD         |    8400.00 |      126000.00 |
+------------+-------------+------------+----------------+
4 rows in set (0.01 sec)
```

15)Rename the new in the above query as new_price

```
mysql> select Product_no,Description,Sell_price, Sell_price*15 as new_price from
    -> product_master
    -> where Sell_price >1500;
+------------+-------------+------------+-----------+
| Product_no | Description | Sell_price | new_price |
+------------+-------------+------------+-----------+
| P03453     | Monitors    |   12000.00 | 180000.00 |
| P07868     | Keyboards   |    3150.00 |  47250.00 |
| P07885     | CD Drive    |    5250.00 |  78750.00 |
| P07965     | HDD         |    8400.00 | 126000.00 |
+------------+-------------+------------+-----------+
4 rows in set (0.00 sec)
```

16)Find the product whose cost price is less than 1500

```
mysql> select Product_no,Description,Cost_price from product_master
    -> where
    -> Cost_price<1500;
+------------+---------------+------------+
| Product_no | Description   | Cost_price |
+------------+---------------+------------+
| P00001     | 1.44 Floppies |     500.00 |
| P06734     | Mouse         |    1000.00 |
| P07865     | 1.22 Floppies |     500.00 |
| P07975     | 1.44 Drive    |    1000.00 |
| P08865     | 1.22 Drive    |    1000.00 |
+------------+---------------+------------+
5 rows in set (0.00 sec)
```

17)List the product in sorted order of their description

```
mysql> select Description from product_master
```

```
    -> ORDER BY Description ASC;
+---------------+
| Description   |
+---------------+
| 1.22 Drive    |
| 1.22 Floppies |
| 1.44 Drive    |
| 1.44 Floppies |
| CD Drive      |
| HDD           |
| Keyboards     |
| Monitors      |
| Mouse         |
+---------------+
9 rows in set (0.00 sec)
```

18)Calculate the square root of price of each product

```
mysql> select Product_no,Description,Cost_price,sqrt(Cost_price) as
    -> square_root_of_cost_price from product_master;
+------------+---------------+------------+---------------------------+
| Product_no | Description   | Cost_price | square_root_of_cost_price |
+------------+---------------+------------+---------------------------+
| P00001     | 1.44 Floppies |     500.00 |        22.360679774997898 |
| P03453     | Monitors      |   11280.00 |        106.20734437881403 |
| P06734     | Mouse         |    1000.00 |        31.622776601683793 |
| P07865     | 1.22 Floppies |     500.00 |        22.360679774997898 |
| P07868     | Keyboards     |    3050.00 |        55.226805085936306 |
| P07885     | CD Drive      |    5100.00 |          71.4142842854285 |
| P07965     | HDD           |    8000.00 |         89.44271909999159 |
| P07975     | 1.44 Drive    |    1000.00 |        31.622776601683793 |
| P08865     | 1.22 Drive    |    1000.00 |        31.622776601683793 |
+------------+---------------+------------+---------------------------+
9 rows in set (0.00 sec)
```

19)Divide the cost of product '540 HDD' by difference between its price and 100

```
mysql> select Cost_price/(Cost_price-100) as Difference from product_master
    -> where
    -> Description = 'HDD' ;
+------------+
| Difference |
+------------+
|   1.012658 |
+------------+
1 row in set (0.01 sec)
```

20)List the names,city,state of clients not in the state of 'Maharashtra'

```
mysql> select name,city,state from Client_master where NOT state ='Maharashtra';
+-----------------+--------+--------------+
| name            | city   | state        |
+-----------------+--------+--------------+
| Vandana
Saitwal | Madras | Tamil Nadu   |
| Ravi
Sreedharan | Delhi  |              |
| Rukmini         | Bombay | Maharashtra
a |
+-----------------+--------+--------------+
3 rows in set (0.00 sec)
```

21)List the product_no,description,sell_price of products whose description begin with letter 'M'

```
mysql> select Product_no,Description,Sell_price from product_master
    -> where Description
    -> like 'M%';
+------------+-------------+------------+
| Product_no | Description | Sell_price |
+------------+-------------+------------+
| P03453     | Monitors    |   12000.00 |
| P06734     | Mouse       |    1050.00 |
+------------+-------------+------------+
2 rows in set (0.00 sec)
```

22)List of all orders that were canceled in month of March

```
mysql> select * from sales_order where Order_status='C' AND S_order_date BETWEEN
'1996-03-01' AND
    -> '1996-03-31';
Empty set (0.00 sec)
```

PRATHAM ASNANI
SQL LAB
SE IT
S11 08

8.Sql queries (complex queries)

1) Count the total no of orders

```
mysql> Select count(*) from sales_order;
+----------+
| count(*) |
+----------+
|        6 |
+----------+
1 row in set (0.02 sec)
```

2) Calculate the average price of all the products

```
mysql> select round(avg(Sell_price)) from product_master;
+------------------------+
| round(avg(Sell_price)) |
+------------------------+
|                   3667 |
+------------------------+
1 row in set (0.01 sec)
```

3) Calculate the minimum price of product

```
mysql> select min(Sell_price) from product_master;
+-----------------+
| min(Sell_price) |
+-----------------+
|          525.00 |
+-----------------+
1 row in set (0.00 sec)
```

4) Determine the maximum and minimum product price. Rename the title as max_price and min_price respectively

```
mysql> select min(Cost_price) as min_price ,max(Cost_price) as max_price from
    -> product_master;
+-----------+-----------+
| min_price | max_price |
+-----------+-----------+
|    500.00 |  11280.00 |
+-----------+-----------+
1 row in set (0.00 sec)
```

5) Count the number of product having price greater than or equal to 1500

```
mysql> select count(*) from product_master
    -> where Sell_price>=1500;
+----------+
| count(*) |
+----------+
|        4 |
+----------+
1 row in set (0.00 sec)
```

6) Find all products whose qty_on_hand is less than reorder level

```
mysql> select Description from product_master
    -> where Qty_on_hand<Reorder_lvl;
+-------------+
| Description |
+-------------+
| 1.22 Drive  |
+-------------+
1 row in set (0.00 sec)
```

7)Count no. of products whose qty_on_hand is less than reorder level

```
mysql> select count(*) from product_master
    -> where Qty_on_hand<Reorder_lvl;
+----------+
| count(*) |
+----------+
|        1 |
+----------+
1 row in set (0.00 sec)
```

8) Print the description and total quantity sold for each product

```
mysql> select s.Product_no,p.Description, sum(s.Qty_ordered) from
    -> sales_order_details s,product_master p where p.Product_no=s.Product_no group by
    -> s.Product_no,p.Description;
+------------+---------------+--------------------+
| Product_no | Description   | sum(s.Qty_ordered) |
+------------+---------------+--------------------+
| P00001     | 1.44 Floppies |                 34 |
| P03453     | Monitors      |                 6 |
| P06734     | Mouse         |                 1 |
| P07868     | Keyboards     |                 3 |
| P07885     | CD Drive      |                 5 |
| P07965     | HDD           |                 3 |
| P07975     | 1.44 Drive    |                 6 |
+------------+---------------+--------------------+
7 rows in set (0.01 sec)
```

9) Find the value of each product sold

```
mysql> select s.Product_no,p.Description,sum(s.Qty_disp*s.Product_rate) "Sales Per
    "> Product" from sales_order_details s,product_master p where
    -> p.Product_no=s.Product_no group by s.Product_no,p.Description;
+------------+---------------+--------------------+
| Product_no | Description   | Sales Per
Product |
+------------+---------------+--------------------+
| P00001     | 1.44 Floppies |            9975.00 |
| P03453     | Monitors      |            6300.00 |
| P06734     | Mouse         |           12000.00 |
| P07868     | Keyboards     |            9450.00 |
| P07885     | CD Drive      |           10500.00 |
| P07965     | HDD           |            8400.00 |
| P07975     | 1.44 Drive    |            3150.00 |
+------------+---------------+--------------------+
7 rows in set (0.00 sec)
```

10) find out the product which has been sold to 'ivan bayroos'.

```
mysql> select distinct s.Client_no,c.Name from sales_order_details d,sales_order
    -> s,product_master p,Client_master c
    -> where p.Product_no=d.Product_no and s.S_order_no=d.S_order_no and
    -> c.Client_no=s.Client_no and p.Description='CD Drive';
+-----------+-----------------+
| Client_no | Name            |
+-----------+-----------------+
| C00001    | IvanBayross     |
| C00003    | Pramada Jaguste |
+-----------+-----------------+
2 rows in set (0.00 sec)
```

11) Find the product no and description of moving products

```
mysql> select d.Product_no,d.S_order_no from sales_order_details d,sales_order
    -> s,product_master p where s.S_order_no=d.S_order_no and p.Product_no=d.Product_no
    -> and d.Qty_ordered<5 and p.Description='1.44 Floppies';
```

```
+------------+------------+
| Product_no | S_order_no |
+------------+------------+
| P00001     | O19001     |
+------------+------------+
1 row in set (0.00 sec)
```

12)Find the names of the clients who have purchased CD Drive

```
mysql>select d.Product_no,p.Description,sum(Qty_ordered)"Qty Ordered" from
sales_order_details d,sales_order s,product_master p,Client_master c where
s.S_order_no=d.S_order_no and p.Product_no=d.Product_no and c.Client_no=s.Client_no
and (c.name='Ivan Bayross' or c.name='Vandana Saitwal') group by
d.Product_no,p.Description;

+------------+---------------+-------------+
| Product_no | Description | Qty Ordered |
+------------+---------------+-------------+
| P00001 | 1.44 Floppies | 14 |
| P03453 | Monitors | 2 |
| P06734 | Mouse | 1 |
| P07885 | CD Drive | 2 |
| P07965 | HDD | 2 |
+------------+---------------+-------------+
5 row in set (0.00 sec)
```

13) List the product no and s order no of customers having quantity ordered less than 5 from the order detail product 1.44 Floppies

```
mysql>select s.Client_no,d.Product_no,p.Description ,sum(Qty_ordered)"Qty_ordered"
    -> from sales_order s, sales_order_details d,product_master p,Client_master c
    -> where s.s_order_no=d.s_order_no and d.Product_no=p.Product_no and
    -> s.Client_no=c.Client_no
    -> group by s.Client_no,d.Product_no,p.Description
    -> having s.Client_no='C00001' or s.Client_no='C00002';
+-----------+------------+---------------+-------------+
| Client_no | Product_no | Description   | Qty_ordered |
+-----------+------------+---------------+-------------+
| C00001    | P00001     | 1.44 Floppies |           4 |
| C00001    | P07885     | CD Drive      |           2 |
| C00001    | P07965     | HDD           |           2 |
| C00001    | P03453     | Monitors      |           2 |
| C00001    | P06734     | Mouse         |           1 |
| C00002    | P00001     | 1.44 Floppies |          10 |
+-----------+------------+---------------+-------------+
6 rows in set (0.00 sec)
```

14)Find the products and their quantities for the orders placed by Vandana Saitwal and Ivan Bayross

```
mysql> select Product_no,Description from product_master
    -> where Product_no not
    -> in(select Product_no from sales_order_details);
+------------+---------------+
| Product_no | Description   |
+------------+---------------+
| P07865     | 1.22 Floppies |
| P08865     | 1.22 Drive    |
+------------+---------------+
2 rows in set (0.00 s)
```

15)Find the products and their quantities for the orders placed by client no C00001 andC00002

```
mysql> select Name,Address1,Address2,City,Pincode from Client_master where
    -> Client_no in (select Client_no from sales_order where S_order_no='O19001');
+--------------+----------+----------+--------+---------+
| Name         | Address1 | Address2 | City   | Pincode |
+--------------+----------+----------+--------+---------+
| Ivan Bayross | NULL     | NULL     | Bombay |  400054 |
+--------------+----------+----------+--------+---------+
1 row in set (0.00 sec)
```

16)Find the product no and description of non moving products

```
mysql> select Product_no,Description from product_master where Product_no not IN(select
product_no from sales_order_details);'
+------------+---------------+
| Product_no | Description   |
+------------+---------------+
| P07865     | 1.22 Floppies |
| P08865     | 1.22 Drive    |
+------------+---------------+
2 rows in set (0.00 sec)
```

17)Find the customers name ,city and pincode for the client who has placed order no O19001

```
mysql> select Name,Address1,Address2,City,Pincode from Client_master where
    -> Client_no in (select Client_no from sales_order where S_order_no='O19001');
+--------------+----------+----------+--------+---------+
| Name         | Address1 | Address2 | City   | Pincode |
+--------------+----------+----------+--------+---------+
| Ivan
```

```
Bayross | NULL     | NULL     | Bombay |  400054 |
+--------------+----------+----------+--------+---------+
1 row in set (0.00 sec)
```

PRATHAM ASNANI
SQL LAB
SE IT
S11 08

9.create view using single table and multiple tables

1)CREATE A VIEW FORM V1

```
mysql> SELECT * FROM V1;
+-----------+----------------+----------+----------+--------+-------------+---------+----------+
| Client_no | Name           | Address1 | Address2 | City   | State       | Pincode | Bal_due  |
+-----------+----------------+----------+----------+--------+-------------+---------+----------+
| C00001    | Ivan
Bayross    | NULL     | NULL     | Bombay | Maharashtra  |  400054 | 15000.00 |
| C00002    | Vandana
Saitwal | NULL     | NULL     | Madras | Tamil Nadu   |  780001 |     0.00 |
| C00003    | Pramada
Jaguste | NULL     | NULL     | Bombay | Maharashtra  |  400057 |  5000.00 |
| C00004    | Basu
Navindgi   | NULL     | NULL     | Bombay | Maharashtra  |  400056 |     0.00 |
| C00005    | Ravi
Sreedharan | NULL     | NULL     | Delhi  |             |  100001 |  2000.00 |
| C00006    | Rukmini        | NULL     | NULL     | Bombay | Maharashtra
a |  400050 | 15000.00 |
+-----------+----------------+----------+----------+--------+-------------+---------+----------+
6 rows in set (0.00 sec)
```

2)CREATE A VIEW FORM V3

```
mysql> create view v3 as select client_master.client_no,product_master.product_no from
client_master,product_master;
Query OK, 0 rows affected (0.02 sec)
mysql> select * from v3;
+-----------+------------+
| client_no | product_no |
+-----------+------------+
| C00006   | P00001    |
| C00005   | P00001    |
| C00004   | P00001    |
| C00003   | P00001    |
| C00002   | P00001    |
| C00001   | P00001    |
| C00006   | P03453    |
| C00005   | P03453    |
| C00004   | P03453    |
| C00003   | P03453    |
| C00002   | P03453    |
| C00001   | P03453    |
| C00006   | P06734    |
| C00005   | P06734    |
| C00004   | P06734    |
| C00003   | P06734    |
| C00002   | P06734    |
| C00001   | P06734    |
| C00006   | P07865    |
| C00005   | P07865    |
| C00004   | P07865    |
| C00003   | P07865    |
| C00002   | P07865    |
| C00001   | P07865    |
| C00006   | P07868    |
| C00005   | P07868    |
| C00004   | P07868    |
| C00003   | P07868    |
| C00002   | P07868    |
| C00001   | P07868    |
| C00006   | P07885    |
| C00005   | P07885    |
| C00004   | P07885    |
| C00003   | P07885    |
| C00002   | P07885    |
| C00001   | P07885    |
| C00006   | P07965    |
| C00005   | P07965    |
| C00004   | P07965    |
| C00003   | P07965    |
| C00002   | P07965    |
| C00001   | P07965    |
| C00006    | P07975     |
| C00005    | P07975     |
| C00004    | P07975     |
| C00003    | P07975     |
| C00002    | P07975     |
| C00001    | P07975     |
| C00006    | P08865     |
| C00005    | P08865     |
| C00004    | P08865     |
| C00003    | P08865     |
| C00002    | P08865     |
| C00001    | P08865     |
+-----------+------------+
54 rows in set (0.00 sec)
```

PRATHAM ASNANI
SQL LAB
SE IT
S11 08

10.Create trigger

```
mysql> create table Student(
roll_no int(2),
name varchar(20),
city varchar(15),
state varchar(20)
);
delimiter $$
create trigger tg1
before insert on Student
for each row
begin
set new.name= upper(new.name);
set new.city= upper(new.city); e
nd;
```

```
$$
```

Creating Triggers Step 3. Insert values into student table :

```
mysql> insert into Student values (28, 'Shania', 'mumbai', 'Maharashtra');
-> $$
mysql> insert into Student values (36, 'Simran', 'ahmedabad', 'Gujrat');
-> $$ mysql> insert into Student values (12, 'Neha', 'gangtok', 'Sikkim');
> $$
select * from Student;
-> $$
+---------+--------+-----------+-------------+
| roll_no | name | city | state |
+---------+--------+-----------+-------------+
| 28 | SHANIA | MUMBAI | Maharashtra |
| 36 | SIMRAN | AHMEDABAD | Gujrat |
| 12 | NEHA | GANGTOK | Sikkim
| +---------+--------+-----------+-------------+

insert into person values ('leena','kirtikar',43);
-> $$ insert into person values ('aakash','kapadia',38);
-> $$ insert into person values ('parth','shroff',63);
-> $$
mysql> select * from person;
-> $$
+--------+----------+----+
| fname | lname | id |
+--------+----------+----+
| aakash | kapadia | 38 |
| leena | kirtikar | 43 |
| parth | shroff | 63 |
+--------+----------+----+
create table audit_log (
ofname char(10),
olname char(10),
nfname char(10),
nname char(10),
c when date );
$$
delimiter $$
create trigger t2
after update on person
for each row
begin
insert into audit_log
values(old.fname,old.lname,new.fname,new.lname,curdate())
```

```
);
end;
$$
-> update person set fname='anish' where id like 43;
-> $$
mysql> update person set fname='anvay' where id like 63;
-> $$
mysql> update person set fname='aman' where id like 38;
-> $$
select *from audit_log;
-> $$
+--------+----------+--------+----------+------------+
| ofname | olname | nfname | nname | cwhen |
+--------+----------+--------+----------+------------+
| leena | kirtikar | anish | kirtikar | 2020-11-26 |
| parth | shroff | anvay | shroff | 2020-11-26 |
| aakash | kapadia | aman | kapadia | 2020-11-26 |
+--------+----------+--------+----------+------------+
-> select *from person;
-> $$
 +-------+----------+----+
| fname | lname | id |
+-------+----------+----+
| aman | kapadia | 38 |
| anish | kirtikar | 43 |
| anvay | shroff | 63 |
+-------+----------+----+
```

PRATHAM ASNANI
SQL LAB
SE IT
S11 08

11.write PL/SQL code to create function and procedure

```
mysql> delimiter &&
mysql> create procedure my_pro(IN client_no int)
```

```
   -> begin
   -> select * from Client_master LIMIT client_no ;
   -> end;
   -> &&
Query OK, 0 rows affected (0.04 sec)

mysql> call my_pro(2);
   -> &&
+-----------+-----------------+----------+----------+--------+------------+---------+----------+
| Client_no | Name           | Address1 | Address2 | City   | State      | Pincode | Bal_due  |
+-----------+-----------------+----------+----------+--------+------------+---------+----------+
| C00001    | Ivan
Bayross    | NULL     | NULL     | Bombay | Maharashtra |  400054 | 15000.00 |
| C00002    | Vandana
Saitwal | NULL     | NULL     | Madras | Tamil Nadu  |  780001 |     0.00 |
+-----------+-----------------+----------+----------+--------+------------+---------+----------+
2 rows in set (0.01 sec)

Query OK, 0 rows affected (0.04 sec)


mysql> delimiter &&
mysql> create procedure my_pro1(OUT client_no int)
   -> begin
   -> select COUNT(*) INTO client_no FROM Client_master;
   -> end;
   -> &&
Query OK, 0 rows affected (0.01 sec)

mysql> call my_pro1(@a);
   -> &&
Query OK, 1 row affected (0.01 sec)

mysql> select @a;
   -> &&
+------+
| @a   |
+------+
|    6 |
+------+
1 row in set (0.01 sec)
```

```
mysql> delimiter $$
mysql> create function proprice(sell_price double)RETURNS varchar(20)
    -> DETERMINISTIC
    -> BEGIN
    -> DECLARE lvl varchar(20);
    -> IF sell_price<1000 THEN
    -> SET lvl='CHEAP';
    -> ELSEIF sell_price>3000 THEN
    -> SET lvl='EXPENSIVE';
    -> END IF;
    -> RETURN(lvl);
    -> END;
    -> $$
Query OK, 0 rows affected (0.02 sec)

mysql> select product_no,proprice(sell_price) from product_master;
    -> $$
+------------+----------------------+
| product_no | proprice(sell_price) |
+------------+----------------------+
| P00001     | CHEAP                |
| P03453     | EXPENSIVE            |
| P06734     | NULL                 |
| P07865     | CHEAP                |
| P07868     | EXPENSIVE            |
| P07885     | EXPENSIVE            |
| P07965     | EXPENSIVE            |
| P07975     | NULL                 |
| P08865     | NULL                 |
+------------+----------------------+
9 rows in set (0.01 sec)
```

PRATHAM ASNANI
SQL LAB
SE IT
S11 08

12.Write PL/SQL code to create Cursor

```
mysql> CREATE procedure CPr()
    -> BEGIN
    -> DECLARE done INT(10) DEFAULT 0;
    -> DECLARE i varchar(20);
    -> DECLARE n varchar(100);
    -> declare curs1 CURSOR FOR select roll_no,name from Student;
    -> declare CONTINUE HANDLER FOR NOT FOUND set done=1;
    -> OPEN curs1;
    -> read_loop:LOOP
    -> fetch curs1 INTO i,n;
    -> If done=1 THEN
    -> leave read_loop;
    -> end if;
    -> select i,n;
    -> end loop read_loop;
    -> close curs1;
    -> end;
    -> $$
Query OK, 0 rows affected, 1 warning (0.02 sec)

mysql> CALL CPr();
    -> $$
+------+--------+
| i    | n      |
+------+--------+
| 28   | SHANIA |
+------+--------+
1 row in set (0.01 sec)

+------+--------+
| i    | n      |
+------+--------+
| 36   | SIMRAN |
+------+--------+
1 row in set (0.02 sec)

+------+------+
| i    | n    |
+------+------+
| 12   | NEHA |
+------+------+
1 row in set (0.02 sec)

Query OK, 0 rows affected (0.03 sec)
```

Example 2 :

```
mysql> alter table person add column email varchar(50);
    -> $$
Query OK, 0 rows affected (0.04 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> update person set email='aman@gmail.com' where id like 38;update person set
email='anish@gmail.com' where id like 43;  update person set email='anvay@gmail.com' where
id like 63;
    -> $$
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

Query OK, 1 row affected (0.02 sec)
Rows matched: 1  Changed: 1  Warnings: 0

Query OK, 1 row affected (0.02 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from person;
    -> $$
+--------+----------+----+-----------------+
| fname  | lname    | id | email           |
+--------+----------+----+-----------------+
| aakash | kapadia  | 38 | aman@gmail.com  |
| anish  | kirtikar | 43 | anish@gmail.com |
| parth  | shroff   | 63 | anvay@gmail.com |
+--------+----------+----+-----------------+
3 rows in set (0.00 sec)
mysql>  delimiter $$
mysql> CREATE procedure build_email_list(INOUT email_list varchar(400))
    -> BEGIN
    -> DECLARE v_finished INTEGER DEFAULT 0;
    -> declare v_email varchar(100) DEFAULT "";
    -> DECLARE email_cursor CURSOR FOR
    -> SELECT email from person;
    -> DECLARE CONTINUE HANDLER
    -> FOR NOT FOUND set v_finished=1;
    -> OPEN email_cursor;
    -> get_email:LOOP
    -> FETCH email_cursor INTO v_email;
    -> IF v_finished=1 THEN
    -> LEAVE get_email;
    -> END IF;
    -> SET email_list=CONCAT(v_email,";",email_list);
    -> END LOOP get_email;
    -> CLOSE email_cursor;
    -> END;
    -> $$
Query OK, 0 rows affected (0.01 sec)

mysql> SET @email_list="";
    -> $$
Query OK, 0 rows affected (0.01 sec)

mysql> CALL build_email_list(@email_list);
    -> $$
Query OK, 0 rows affected (0.01 sec)

mysql> select @email_list;
    -> $$
+-------------------------------------------------+
| @email_list                                     |
+-------------------------------------------------+
| anvay@gmail.com;anish@gmail.com;aman@gmail.com; |
+-------------------------------------------------+
1 row in set (0.00 sec)

mysql> SET @email_list="2";
    -> $$
Query OK, 0 rows affected (0.00 sec)

mysql> CALL build_email_list(@email_list);
    -> $$
Query OK, 0 rows affected (0.00 sec)

mysql> select @email_list;
    -> $$
+--------------------------------------------------+
| @email_list                                      |
+--------------------------------------------------+
| anvay@gmail.com;anish@gmail.com;aman@gmail.com;2 |
+--------------------------------------------------+
1 row in set (0.00 sec)
```

Pratham Asnani
08 S11

A schedule is serialization if it is equivalent to a serial schedule. Different forms of schedule equivalent give rise to the notion of

1. conflict serializ~~ation~~bility
2. View serializability

Conflicting Paris

1) l_i = read (2), l_j = read (2). l_i and l_j don't conflict
2) l_i = read (2), l_j = write (2). They conflict
3) l_i = write (2), l_j = read (2). They conflict
4) l_i = write (2), l_j = write (2). They conflict

Conflict seriabitity

If a schedule S can be transformes into a schedule S' by a series of swaps of non-conflicting instruction. We say that S and S' are conflict equivalent

We say that a schedule S is conflict serializable if it is conflict equivalent to a serial schedule.

Problem 1

T_1	T_2	T_3	T_4
$R_1(A)$			
	R(A)		
		R(A)	
			R(A)
W(B)			
	W(B)		
		W(B)	
			W(B)

Concurrenty Problem in DBMS

Concurrency problem in Transaction
→ Dirty Read problem
→ Unprepeatable read problem
→ Last Update problem
→ Phantom Read problem

Blind Write : Performing write before reating is Blind write

Serializable schedule are always:

=> Consistent
=> Recouerable
=> cascadels
=> strict

Check whether schedule is conflict serializable or not

Step 1: Find all the conflicting operations

step 2: Start Creating precedence graph by drawing one node for each transaction

Step 3: Draw an edge conflict pairs such that if Xi(V) and Yj(V) forms a conflict pair then draw an edge from Ti and j. This ensures that Ti gets execute before Tj.

Step 4: Check if ~~that~~ there is any cycle formed in the graph. If ther is not cycle form, then the schedule is conflict serializable otherwise not.

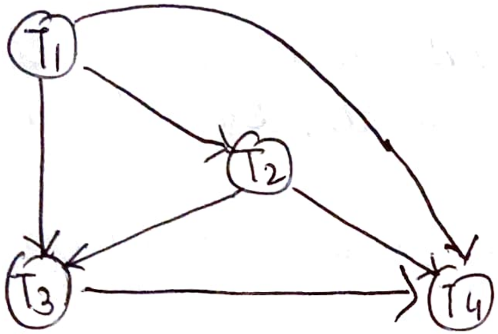
Check wheder given schedule is view serializable or Not

T_1	T_2	T_3	T_4
R(A)			
	R(A)		
		R(A)	
			R(A)
W(B)			
	W(B)		
		W(B)	~~w(~~
			W(B)

conflicting pairs

1	$W_1(B), W_2(B)$	$(T_1 \rightarrow T_2)$
2	$W_1(B), W_3(B)$	$(T_1 \rightarrow T_3)$
3	$W_1(B), W_4(B)$	$(T_1 \rightarrow T_4)$
4	$W_2(B), W_3(B)$	$(T_2 \rightarrow T_3)$
5	$W_2(B), W_4(B)$	$(T_2 \rightarrow T_4)$
6	$W_3(B), W_4(B)$	$(T_3 \rightarrow T_4)$

Draw precedence graph



There is No cycle. This is conflict serializable
Given schedule is also view serializable.

Name:Pratham Asnani
Roll no:08
Batch:s11

Written Assignment - 02

Relational algebra is a fundamental concept in Database Management Systems.It is procedural way to slove query languages and hence it is called as procedural query language. It allows users to describe queries on a relational database in a structured manner.It uses a series of operation to perform a query.There are a set of operation of relational algebra.

1) Selection operation
2) Project operation
3) Union operation
4) Set intersection
5) Set difference
6) Cartesian Product / Cross Product
7) Rename Operation
8) Join Operation

1. Select Operation (σ):
   Select Operation select tuples that satisfy a given condition.
   Example:-σ Department=”HR”(EMPLOYEE)

2. Projection Operation (∏):
   This operation shows the list of those attributes that we wish to appear in the result. Rest of the attributes are eliminated from the table.
   Example:- ∏CustomerName,Status(Customer)

3. Union Operation (U):
   Suppose there are two tuples R and S. The union operation contains all the tuples that are either in R or S or both in R & S.
   Example:-∏ CUSTOMER_NAME (BORROW) ∪ ∏ CUSTOMER_NAME (DEPOSITOR) .

4. Set Intersection (∩):
   Suppose there are two tuples R and S. The set intersection operation contains all tuples that are in both R & S.
   Example:-∏ CUSTOMER_NAME (LENDING) ∪ ∏ CUSTOMER_NAME (DEPOSITOR) .

5. Set Difference (-):
Suppose there are two tuples R and S. The set intersection operation contains all tuples that are in R but not in S.
Example:- ∏ EMPLOYEE_NAME (TATA_EMPLOYEE) -∏ EMP_NAME (DEPARTMENT)

6. Rename Operation (ρ):
The rename operation is used to rename the output relation. It is denoted by rho (ρ).
Example:-ρ NEWEMPLOYEE(EMPLOYEE)

7. Cross Product Operation (X):
This operation is used to show every tuple one by one from the left set(relation) will pair up with all the tuples in the right set(relation).
Example:-

EMPLOYEE

Sr_no.	**Fname**	**Lname**
1	Mohit	Gupta
2	Raj	Mishra

EMPLOYEE_DETAIL

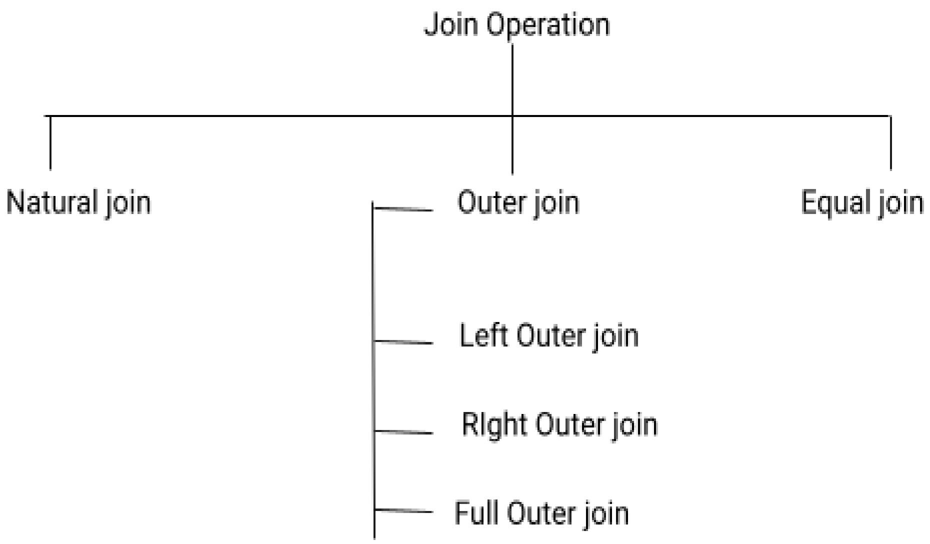
Emp_ID	**Department**
101	HR
102	Sales

On applying CROSS PRODUCT on EMPLOYEE and EMPLOYEE_DETAIL

Sr_no.	**Fname**	**Lname**	**Emp_ID**	**Department**
1	Mohit	Gupta	101	HR
1	Mohit	Gupta	101	HR
2	Raj	Mishra	102	Sales
2	Raj	Mishra	102	Sales

8.Join Operations:

A Join operation combines related tuples from different relations, if and only if a given join condition is satisfied.



Left Outer Join(R(⟕) S):-

All the tuples from the Left relation, R, are included in the resulting relation. If there are tuples in R without any matching tuple in the Right relation S, then the Sattributes of the resulting relation are made NULL.

Left RIght

A	B	C	D
1	Google	1	Active
2	Amazon	2	Inactive
3	Alibaba	4	Active

Company⟕ Status

A	B	C	D
1	Google	1	Active

2	Amazon	2	Inactive
3	Alibaba	_	_

Full Outer Join: (R⋈ S)

All the tuples from both participating relations are included in the resulting relation. If there are no matching tuples for both relations, their respective unmatched attributes are made NULL.

LEFT RIGHT

ENTRY_TIME	NAME	DEPT	EMP_ID	EMP_ID	SALARY	POSITION
9:30	RAJ	HR	E34	E34	40,000	TYPIST
9:40	RAHUL	IT	E56	E56	60,000	MANAGER
9:25	OM	SALED	E90	E88	90,000	MANAGER

FULL OUTER JOIN

ENTRY_TIME	NAME	DEPT	EMP_ID	EMP_ID	SALARY	POSITION
9:30	RAJ	HR	E34	E34	40,000	TYPIST
9:40	RAHUL	IT	E56	E56	60,000	MANAGER
9:25	OM	SALED	E90	_	_	_
_	_	_	_	E88	90,000	MANAGER