create database superstore;

use superstore;

create table Category(

cat\_id int primary key,

cat\_name varchar(20) not null);

INSERT INTO Category (cat\_id, cat\_name) VALUES

(1, 'BOOKS'),

(2, 'GAMES'),

(3, 'GROCERIES'),

(4, 'ELECTRONICS'),

(5, 'CLOTHES');

CREATE TABLE Product (

pro\_id INT PRIMARY KEY,

pro\_name VARCHAR(20) NOT NULL,

pro\_desc VARCHAR(60),

cat\_id INT,

FOREIGN KEY (cat\_id) REFERENCES Category(cat\_id));

INSERT INTO Product (pro\_id, pro\_name, pro\_desc, cat\_id) VALUES

(1, 'GTA V', 'Windows 7 and above with i5 processor and 8GB RAM', 2),

(2, 'TSHIRT', 'SIZE-L with Black, Blue and White variations', 5),

(3, 'ROG LAPTOP', 'Windows 10 with 15inch screen, i7 processor, 1TB SSD', 4),

(4, 'OATS', 'Highly Nutritious from Nestle', 3),

(5, 'HARRY POTTER', 'Best Collection of all time by J.K Rowling', 1),

(6, 'MILK', '1L Toned MIlk', 3),

(7, 'Boat Earphones', '1.5Meter long Dolby Atmos', 4),

(8, 'Jeans', 'Stretchable Denim Jeans with various sizes and color', 5),

(9, 'Project IGI', 'compatible with windows 7 and above', 2),

(10, 'Hoodie', 'Black GUCCI for 13 yrs and above', 5),

(11, 'Rich Dad Poor Dad', 'Written by RObert Kiyosaki', 1),

(12, 'Train Your Brain', 'By Shireen Stephen', 1);

create table Supplier (

supp\_id int primary key,

supp\_name varchar(50) not null,

supp\_city varchar(50) not null,

supp\_phone varchar(50) not null);

INSERT INTO Supplier (supp\_id, supp\_name, supp\_city, supp\_phone) VALUES

(1, 'Rajesh Retails', 'Delhi', '1234567890'),

(2, 'Appario Ltd.', 'Mumbai', '2589631470'),

(3, 'Knome products', 'Bangalore', '9785462315'),

(4, 'Bansal Retails', 'Kochi', '8975463285'),

(5, 'Mittal Ltd.', 'Lucknow', '7898456532');

create table Supplier\_Pricing(

pricing\_id int primary key,

pro\_id int,

supp\_id int,

supp\_price INT DEFAULT 0,

FOREIGN KEY (pro\_id) REFERENCES product(pro\_id),

FOREIGN KEY (supp\_id) REFERENCES supplier(Supp\_id));

INSERT INTO Supplier\_Pricing (pricing\_id, pro\_id, supp\_id, supp\_price) VALUES

(1, 1, 2, 1500),

(2, 3, 5, 30000),

(3, 5, 1, 3000),

(4, 2, 3, 2500),

(5, 4, 1, 1000),

(6, 12, 2, 780),

(7, 12, 4, 789),

(8, 3, 1, 31000),

(9, 1, 5, 1450),

(10, 4, 2, 999),

(11, 7, 3, 549),

(12, 7, 4, 529),

(13, 6, 2, 105),

(14, 6, 1, 99),

(15, 2, 5, 2999),

(16, 5, 2, 2999);

create table customer(

cus\_id int primary key,

cus\_name varchar(20) not null,

cus\_city varchar(30) not null,

cus\_phone varchar(10) not null,

cus\_gender char(30));

INSERT INTO Customer (cus\_id, cus\_name, cus\_phone, cus\_city, cus\_gender) VALUES

(1, 'AAKASH', '9999999999', 'DELHI', 'M'),

(2, 'AMAN', '9785463215', 'NOIDA', 'M'),

(3, 'NEHA', '9999999999', 'MUMBAI', 'F'),

(4, 'MEGHA', '9994562399', 'KOLKATA', 'F'),

(5, 'PULKIT', '7895999999', 'LUCKNOW', 'M');

create table Orders (

ord\_id int primary key,

ord\_date date not null,

cus\_id int,

pricing\_id int,

foreign key (cus\_id) references customer(cus\_id),

foreign key (pricing\_id) references Supplier\_Pricing(pricing\_id));

INSERT INTO Orders (ord\_id, ord\_date, cus\_id, pricing\_id) VALUES

(101, '2021-10-06', 2, 1),

(102, '2021-10-12', 3, 5),

(103, '2021-09-16', 5, 2),

(104, '2021-10-05', 1, 1),

(105, '2021-08-16', 4, 3),

(106, '2021-08-18', 1, 9),

(107, '2021-09-01', 3, 7),

(108, '2021-09-07', 5, 6),

(109, '2021-09-10', 5, 3),

(110, '2021-09-10', 2, 4),

(111, '2021-09-15', 4, 5),

(112, '2021-09-16', 4, 7),

(113, '2021-09-16', 1, 8),

(114, '2021-09-16', 3, 5),

(115, '2021-09-16', 5, 3),

(116, '2021-09-17', 2, 14),

(117, '2023-05-02', 4, 3);

create table rating(

rat\_id int primary key,

rat\_ratstars int,

ord\_id int,

foreign key (ord\_id) references Orders(ord\_id));

INSERT INTO rating (rat\_ID, ord\_id,rat\_ratstars) VALUES

(1 ,101, 4),

(2 ,102 ,3),

(3, 103, 1),

(4 ,104 ,2),

(5, 105, 4),

(6, 106, 3),

(7, 107, 4),

(8, 108, 4),

(9, 109, 3),

(10, 110, 5),

(11, 111 ,3),

(12, 112, 4),

(13, 113, 2),

(14, 114, 1),

(15, 115, 1),

(16, 116, 0);

/\*4) Display the total number of customers based on gender who have placed individual orders of worth at least Rs. 3000:\*/

select cus\_gender,count(cus\_gender)

from

customer inner join

orders

using(cus\_id)

inner join supplier\_pricing

using(pricing\_id)

where

supp\_price>=3000

group by cus\_gender;

/\*5) Display all the orders along with product name ordered by a customer having Customer\_Id=2:\*/

select cus\_gender,count(cus\_gender)

from

customer inner join

orders

using(cus\_id)

inner join supplier\_pricing

using(pricing\_id)

where

supp\_price>=3000

group by cus\_gender;

/\*6) Display the Supplier details who can supply more than one product:\*/

select supp\_id,supp\_name,supp\_city,supp\_phone

from supplier s

inner join supplier\_pricing sp

using(supp\_id)

group by supp\_id

having count(pro\_id)>1;

/\*7) Find the least expensive product from each category and print the table with category id, name, product name, and price of the product:\*/

select cat\_id, cat\_name,pro\_name,supp\_price

from category

inner join product

inner join supplier\_pricing

using (pro\_id)

where supp\_price=(

select min(supp\_price)

from supplier\_pricing

)

/\*8) Display the Id and Name of the Product ordered after “2021-10-05”:\*/

select pro\_id,pro\_name

from product

inner join supplier\_pricing

using(pro\_id)

inner join orders

using(pricing\_id)

where ord\_date>'2021-10-05';

/\*9) Display customer name and gender whose names start or end with character 'A':\*/

select cus\_name,cus\_gender

from customer

where cus\_name like '%a%';

/\*10) Create a stored procedure to display supplier id, name, Rating(Average rating of all the products sold by every customer) and

Type\_of\_Service. For Type\_of\_Service, If rating =5, print “Excellent Service”,If rating >4 print “Good Service”, If rating >2 print “Average

Service” else print “Poor Service”. Note that there should be one rating per supplier\*/

SELECT

s.supp\_id,

s.supp\_name,

r.rat\_ratstars,

CASE

WHEN r.rat\_ratstars = 5 THEN 'Excellent Service'

WHEN r.rat\_ratstars > 4 THEN 'Good Service'

WHEN r.rat\_ratstars > 2 THEN 'Average Service'

ELSE 'Poor Service'

END AS Type\_of\_Service

FROM Supplier s

JOIN Supplier\_Pricing sp ON s.supp\_id = sp.supp\_id

JOIN Product p ON sp.pro\_id = p.pro\_id

JOIN Orders o ON sp.pricing\_id = o.pricing\_id

LEFT JOIN rating r ON o.ord\_id = r.ord\_id;