Create database ECommerceDB;

use ECommerceDB;

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

);

create table customer (

CUS\_ID INT primary key,

CUS\_NAME VARCHAR(20) NOT NULL,

CUS\_PHONE VARCHAR(10) NOT NULL,

CUS\_CITY VARCHAR(30) NOT NULL,

CUS\_GENDER CHAR

);

create table category (

CAT\_ID INT primary key,

CAT\_NAME VARCHAR(20) NOT NULL

);

create table product (

PRO\_ID INT primary key,

PRO\_NAME VARCHAR(20) NOT NULL DEFAULT "Dummy", PRO\_DESC VARCHAR(60),

CAT\_ID INT NOT NULL,

foreign key (CAT\_ID) references category (CAT\_ID)

);

create table supplier\_pricing (

PRICING\_ID INT primary key,

PRO\_ID INT NOT NULL,

SUPP\_ID INT NOT NULL,

foreign key (PRO\_ID) references product (PRO\_ID),

foreign key (SUPP\_ID) references supplier (SUPP\_ID),

SUPP\_PRICE INT DEFAULT 0

);

create table orders (

ORD\_ID INT primary key,

ORD\_AMOUNT INT NOT NULL,

ORD\_DATE DATE NOT NULL,

CUS\_ID INT NOT NULL,

PRICING\_ID INT NOT NULL,

foreign key (CUS\_ID) references customer (CUS\_ID),

foreign key (PRICING\_ID) references supplier\_pricing (PRICING\_ID)

);

create table rating (

RAT\_ID INT primary key,

ORD\_ID INT NOT NULL,

foreign key (ORD\_ID) references orders (ORD\_ID),

RAT\_RATSTARS INT NOT NULL

);

-- Insert data into Supplier Table

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');

-- Insert data into Customer Table

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');

-- Insert data into Category Table

INSERT INTO category (CAT\_ID, CAT\_NAME)

VALUES

(1, 'BOOKS'),

(2, 'GAMES'),

(3, 'GROCERIES'),

(4, 'ELECTRONICS'),

(5, 'CLOTHES');

-- Insert data into Product Table

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 15-inch 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);

-- Insert data into Supplier\_pricing Table

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);

-- Insert data into Order Table

INSERT INTO orders (ORD\_ID, ORD\_AMOUNT, ORD\_DATE, CUS\_ID, PRICING\_ID)

VALUES

(101, 1500, "2021-10-06", 2, 1),

(102, 1000, "2021-10-12", 3, 5),

(103, 30000, "2021-09-16", 5, 2),

(104, 1500, "2021-10-05", 1, 1),

(105, 3000, "2021-08-16", 4, 3),

(106, 1450, "2021-08-18", 1, 9),

(107, 789, "2021-09-01", 3, 7),

(108, 780, "2021-09-07", 5, 6),

(109, 3000, "2021-09-10", 5, 3),

(110, 2500, "2021-09-10", 2, 4),

(111, 1000, "2021-09-15", 4, 5),

(112, 789, "2021-09-16", 4, 7),

(113, 31000, "2021-09-16", 1, 8),

(114, 1000, "2021-09-16", 3, 5),

(115, 3000, "2021-09-16", 5, 3),

(116, 99, "2021-09-17", 2, 14);

-- Insert data into Rating Table

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);

select \* from supplier;

select \* from customer;

select \* from category;

select \* from product;

select \* from supplier\_pricing;

select \* from orders;

select \* from rating;

-- Display the total number of customers based on gender who have placed individual orders of worth at least Rs.3000

SELECT c.CUS\_GENDER, COUNT(c.CUS\_GENDER) as count

FROM customer c

Inner JOIN

(select c.CUS\_ID as customerId

from customer c

Inner Join orders o ON o.CUS\_ID = c.CUS\_ID

WHERE O.ORD\_AMOUNT >= 3000

GROUP BY c.CUS\_ID)

as v ON v.customerId = c.CUS\_ID

GROUP BY c.CUS\_GENDER;

-- Display all the orders along with product name ordered by a customer having Customer\_Id=2

SELECT p.PRO\_NAME, o.\*

FROM orders o

Inner join customer c ON c.CUS\_ID = o.CUS\_ID

Inner join supplier\_pricing sp ON sp.PRICING\_ID = o.PRICING\_ID

Inner join product p ON p.PRO\_ID = sp.PRO\_ID

WHERE c.CUS\_ID = 2;

-- Display the Supplier details who can supply more than one product

-- Find the least expensive product from each category and print the table with category id, name, product name and price of the product

-- Display the Id and Name of the Product ordered after “2021-10-05”

-- Display customer name and gender whose names start or end with character 'A'

-- 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 supplie

-- Display the total number of customers based on gender who have placed individual orders of worth at least Rs.3000.

select c.CUS\_GENDER , count(c.CUS\_GENDER) as count from customer c inner join

(

select c.CUS\_ID as customerId

from customer c inner join orders o on o.CUS\_ID=c.CUS\_ID where o.ORD\_AMOUNT>=3000 group by c.CUS\_ID

) as v on v.customerId= c.CUS\_ID group by c.CUS\_GENDER;

-- Display all the orders along with product name ordered by a customer having Customer\_Id=2

select p.PRO\_NAME, o.\*

from orders o inner join customer c on c.CUS\_ID= o.CUS\_ID inner join supplier\_pricing sp on sp.pricing\_id= o.pricing\_id

inner join product p on p.pro\_id =sp.pro\_id

where c.CUS\_ID=2;

-- Display the Supplier details who can supply more than one product.

select s.\*, v.totalProducts from supplier s inner join (

select sp.supp\_id ,count(sp.pro\_id) as totalProducts from supplier\_pricing sp group by sp.SUPP\_ID)

as v on v.supp\_id= s.supp\_id where v.totalProducts >1;

-- Find the least expensive product from each category and print the table with category id, name, product name and price of the product

select c.cat\_id, c.cat\_name, min(vv.min\_price) as min\_price from category c inner join (

select p.\*, v.min\_price from product p inner join(

select sp.pro\_id, min(sp.supp\_price) as min\_price from supplier\_pricing sp group by sp.PRO\_ID)

as v on p.pro\_id = v.PRO\_ID )

as vv on vv.cat\_id =c.cat\_id group by c.cat\_id;

-- Display the Id and Name of the Product ordered after “2021-10-05”.

select p.pro\_id, p.pro\_name from orders o inner join supplier\_pricing sp on sp.PRICING\_ID= o.PRICING\_ID

inner join product p on p.PRO\_ID=sp.PRO\_ID where o.ORD\_DATE> '2021-10-05' ;

-- Display customer name and gender whose names start or end with character 'A'

select \* from customer c

where c.CUS\_NAME like 'A%'

or c.CUS\_NAME like '%A';

-- Create a stored procedure to display supplier id, name, Rating(Average rating of all the products sold by every supplier) 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 report.supp\_id, report.supp\_name,

case

when report.average=5 then 'Exellent Service'

when report.average>4 then 'Good Service'

when report.average>2 then 'Average Service'

else 'Poor Service'

end as type\_of\_service from

( select s.\*, final.average from supplier s inner join ( select vv.supp\_id, avg (vv.rat\_ratstars) as average from (

select sp.pricing\_id, sp.pro\_id, sp.supp\_id, sp.supp\_price, v.ord\_id, v.rat\_ratstars from supplier\_pricing sp inner join(

select o.ord\_id, o.pricing\_id, r.rat\_ratstars from orders o inner join rating r on r.ord\_id= o.ord\_id)

as v on sp.pricing\_id= v.pricing\_id)

as vv group by vv.supp\_id)

as final on final.supp\_id= s.supp\_id)

as report;