create database ecommerce;

use ecommerce;

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,

foreign key (CAT\_ID) references Category(CAT\_ID));

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));

Create table `order`(

ORD\_ID int primary key,

ORD\_AMOUNT int not null,

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));

create table rating(

RAT\_ID int primary key,

ORD\_ID int,

RAT\_RATSTARS INT not null,

foreign key(ORD\_ID) references order (ORD\_ID));

INSERT INTO SUPPLIER VALUES(1,"Rajesh Retails","Delhi",'1234567890');

INSERT INTO SUPPLIER VALUES(2,"Appario Ltd.","Mumbai",'2589631470');

INSERT INTO SUPPLIER VALUES(3,"Knome products","Banglore",'9785462315');

INSERT INTO SUPPLIER VALUES(4,"Bansal Retails","Kochi",'8975463285');

INSERT INTO SUPPLIER VALUES(5,"Mittal Ltd.","Lucknow",'7898456532');

INSERT INTO CUSTOMER VALUES(1,"AAKASH",'9999999999',"DELHI",'M');

INSERT INTO CUSTOMER VALUES(2,"AMAN",'9785463215',"NOIDA",'M');

INSERT INTO CUSTOMER VALUES(3,"NEHA",'9999999999',"MUMBAI",'F');

INSERT INTO CUSTOMER VALUES(4,"MEGHA",'9994562399',"KOLKATA",'F');

INSERT INTO CUSTOMER VALUES(5,"PULKIT",'7895999999',"LUCKNOW",'M');

INSERT INTO CATEGORY VALUES( 1,"BOOKS");

INSERT INTO CATEGORY VALUES(2,"GAMES");

INSERT INTO CATEGORY VALUES(3,"GROCERIES");

INSERT INTO CATEGORY VALUES (4,"ELECTRONICS");

INSERT INTO CATEGORY VALUES(5,"CLOTHES");

INSERT INTO PRODUCT VALUES(1,"GTA V","Windows 7 and above with i5 processor and 8GB RAM",2);

INSERT INTO PRODUCT VALUES(2,"TSHIRT","SIZE-L with Black, Blue and White variations",5);

INSERT INTO PRODUCT VALUES(3,"ROG LAPTOP","Windows 10 with 15inch screen, i7 processor, 1TB SSD",4);

INSERT INTO PRODUCT VALUES(4,"OATS","Highly Nutritious from Nestle",3);

INSERT INTO PRODUCT VALUES(5,"HARRY POTTER","Best Collection of all time by J.K Rowling",1);

INSERT INTO PRODUCT VALUES(6,"MILK","1L Toned MIlk",3);

INSERT INTO PRODUCT VALUES(7,"Boat EarPhones","1.5Meter long Dolby Atmos",4);

INSERT INTO PRODUCT VALUES(8,"Jeans","Stretchable Denim Jeans with various sizes and color",5);

INSERT INTO PRODUCT VALUES(9,"Project IGI","compatible with windows 7 and above",2);

INSERT INTO PRODUCT VALUES(10,"Hoodie","Black GUCCI for 13 yrs and above",5);

INSERT INTO PRODUCT VALUES(11,"Rich Dad Poor Dad","Written by RObert Kiyosaki",1);

INSERT INTO PRODUCT VALUES(12,"Train Your Brain","By Shireen Stephen",1);

INSERT INTO SUPPLIER\_PRICING VALUES(1,1,2,1500);

INSERT INTO SUPPLIER\_PRICING VALUES(2,3,5,30000);

INSERT INTO SUPPLIER\_PRICING VALUES(3,5,1,3000);

INSERT INTO SUPPLIER\_PRICING VALUES(4,2,3,2500);

INSERT INTO SUPPLIER\_PRICING VALUES(5,4,1,1000);

INSERT INTO SUPPLIER\_PRICING VALUES(6,12,2,780);

INSERT INTO SUPPLIER\_PRICING VALUES(7,12,4,789);

INSERT INTO SUPPLIER\_PRICING VALUES(8,3,1,31000);

INSERT INTO SUPPLIER\_PRICING VALUES(9,1,5,1450);

INSERT INTO SUPPLIER\_PRICING VALUES(10,4,2,999);

INSERT INTO SUPPLIER\_PRICING VALUES(11,7,3,549);

INSERT INTO SUPPLIER\_PRICING VALUES(12,7,4,529);

INSERT INTO SUPPLIER\_PRICING VALUES(13,6,2,105);

INSERT INTO SUPPLIER\_PRICING VALUES(14,6,1,99);

INSERT INTO SUPPLIER\_PRICING VALUES(15,2,5,2999);

INSERT INTO SUPPLIER\_PRICING VALUES(16,5,2,2999);

INSERT INTO `ORDER` VALUES (101,1500,"2021-10-06",2,1);

INSERT INTO `ORDER` VALUES(102,1000,"2021-10-12",3,5);

INSERT INTO `ORDER` VALUES(103,30000,"2021-09-16",5,2);

INSERT INTO `ORDER` VALUES(104,1500,"2021-10-05",1,1);

INSERT INTO `ORDER` VALUES(105,3000,"2021-08-16",4,3);

INSERT INTO `ORDER` VALUES(106,1450,"2021-08-18",1,9);

INSERT INTO `ORDER` VALUES(107,789,"2021-09-01",3,7);

INSERT INTO `ORDER` VALUES(108,780,"2021-09-07",5,6);

INSERT INTO `ORDER` VALUES(109,3000,"2021-09-10",5,3);

INSERT INTO `ORDER` VALUES(110,2500,"2021-09-10",2,4);

INSERT INTO `ORDER` VALUES(111,1000,"2021-09-15",4,5);

INSERT INTO `ORDER` VALUES(112,789,"2021-09-16",4,7);

INSERT INTO `ORDER` VALUES(113,31000,"2021-09-16",1,8);

INSERT INTO `ORDER` VALUES(114,1000,"2021-09-16",3,5);

INSERT INTO `ORDER` VALUES(115,3000,"2021-09-16",5,3);

INSERT INTO `ORDER` VALUES(116,99,"2021-09-17",2,14);

INSERT INTO RATING VALUES(1,101,4);

INSERT INTO RATING VALUES(2,102,3);

INSERT INTO RATING VALUES(3,103,1);

INSERT INTO RATING VALUES(4,104,2);

INSERT INTO RATING VALUES(5,105,4);

INSERT INTO RATING VALUES(6,106,3);

INSERT INTO RATING VALUES(7,107,4);

INSERT INTO RATING VALUES(8,108,4);

INSERT INTO RATING VALUES(9,109,3);

INSERT INTO RATING VALUES(10,110,5);

INSERT INTO RATING VALUES(11,111,3);

INSERT INTO RATING VALUES(12,112,4);

INSERT INTO RATING VALUES(13,113,2);

INSERT INTO RATING VALUES(14,114,1);

INSERT INTO RATING VALUES(15,115,1);

INSERT INTO RATING VALUES(16,116,0);

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

Select count(C.CUS\_ID) as count, C.CUS\_GENDER

FROM `ORDER` ord

INNER JOIN customer c ON C.CUS\_ID = ORD.CUS\_ID

WHERE ORD.ORD\_AMOUNT>=3000

GROUP BY C.CUS\_GENDER

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

Select ord.\*,cus.cus\_name, prd.pro\_name, prd.pro\_desc from `ORDER` ord

INNER JOIN SUPPLIER\_PRICING sp ON sp.PRICING\_ID = ord.PRICING\_ID

INNER JOIN PRODUCT prd ON prd.PRO\_ID = sp.PRO\_ID

INNER JOIN CUSTOMER cus ON cus.CUS\_ID = ord.CUS\_ID

WHERE ORD.CUS\_ID = 2

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

SELECT \* FROM supplier sp

INNER JOIN

(

SELECT SUPP\_ID, COUNT(PRO\_ID) as count

FROM SUPPLIER\_PRICING

GROUP BY SUPP\_ID

) as sup on sup.SUPP\_ID = sp.supp\_id

WHERE sup.count>2

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(t1.minPrice)

FROM Category c

INNER JOIN

(

SELECT PRD.CAT\_ID, PRD.PRO\_NAME, t2.\* FROM PRODUCT PRD

INNER JOIN(

SELECT sp.pro\_id, MIN(sp.SUPP\_PRICE) as minPrice FROM SUPPLIER\_PRICING sp

GROUP BY sp.PRO\_ID

) AS T2 ON PRD.PRO\_ID = T2.PRO\_ID

) AS T1 ON T1.CAT\_ID = C.CAT\_ID

GROUP BY T1.CAT\_ID

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

SELECT p.PRO\_ID, p.PRO\_NAME from `ORDER` ord

INNER JOIN SUPPLIER\_PRICING sp ON sp.PRICING\_ID = ord.PRICING\_ID

INNER JOIN PRODUCT p ON p.PRO\_ID = sp.PRO\_ID

WHERE ord.ORD\_DATE > "2021-10-05"

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

SELECT cus.CUS\_NAME, cus.CUS\_GENDER FROM CUSTOMER cus

WHERE cus.CUS\_NAME like 'A%' OR cus.CUS\_NAME like '%A';

1. 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.

DELIMITER &&

CREATE PROCEDURE details()

BEGIN

select report.supp\_id,report.supp\_name,report.Average,

CASE

WHEN report.Average =5 THEN "Excellent 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 final.supp\_id, supplier.supp\_name, final.Average from

(select test2.supp\_id, sum(test2.rat\_ratstars)/count(test2.rat\_ratstars) as Average from

(select supplier\_pricing.supp\_id, test.ORD\_ID, test.RAT\_RATSTARS from supplier\_pricing inner join

(select `order`.pricing\_id, rating.ORD\_ID, rating.RAT\_RATSTARS from `order` inner join rating on rating.`ord\_id` = `order`.ord\_id ) as test

on test.pricing\_id = supplier\_pricing.pricing\_id)

as test2 group by supplier\_pricing.supp\_id)

as final inner join supplier where final.supp\_id = supplier.supp\_id) as report;

END &&

DELIMITER ;

call details();