Graded Lab 4 - DBMS

DROP DATABASE IF exists `order-directory`;

create database `order-directory`;

use `order-directory`;

CREATE TABLE IF NOT EXISTS supplier(

SUPP\_ID int primary key,

SUPP\_NAME varchar(50) NOT NULL,

SUPP\_CITY varchar(50),

SUPP\_PHONE varchar(10) NOT NULL

);

CREATE TABLE IF NOT EXISTS customer(

CUS\_ID INT NOT NULL,

CUS\_NAME VARCHAR(20) NOT NULL,

CUS\_PHONE VARCHAR(10) NOT NULL,

CUS\_CITY varchar(30) NOT NULL,

CUS\_GENDER CHAR,

PRIMARY KEY (CUS\_ID));

CREATE TABLE IF NOT EXISTS category (

CAT\_ID INT NOT NULL,

CAT\_NAME VARCHAR(20) NOT NULL,

PRIMARY KEY (CAT\_ID)

);

CREATE TABLE IF NOT EXISTS product (

PRO\_ID INT NOT NULL,

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

PRO\_DESC VARCHAR(60),

CAT\_ID INT NOT NULL,

PRIMARY KEY (PRO\_ID),

FOREIGN KEY (CAT\_ID) REFERENCES category (CAT\_ID)

);

CREATE TABLE IF NOT EXISTS supplier\_pricing (

PRICING\_ID INT NOT NULL,

PRO\_ID INT NOT NULL,

SUPP\_ID INT NOT NULL,

SUPP\_PRICE INT DEFAULT 0,

PRIMARY KEY (PRICING\_ID),

FOREIGN KEY (PRO\_ID) REFERENCES product (PRO\_ID),

FOREIGN KEY (SUPP\_ID) REFERENCES supplier(SUPP\_ID)

);

CREATE TABLE IF NOT EXISTS `order` (

ORD\_ID INT NOT NULL,

ORD\_AMOUNT INT NOT NULL,

ORD\_DATE DATE,

CUS\_ID INT NOT NULL,

PRICING\_ID INT NOT NULL,

PRIMARY KEY (ORD\_ID),

FOREIGN KEY (CUS\_ID) REFERENCES customer(CUS\_ID),

FOREIGN KEY (PRICING\_ID) REFERENCES supplier\_pricing(PRICING\_ID)

);

CREATE TABLE IF NOT EXISTS rating (

RAT\_ID INT NOT NULL,

ORD\_ID INT NOT NULL,

RAT\_RATSTARS INT NOT NULL,

PRIMARY KEY (RAT\_ID),

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);

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

select count(\*), CUS\_GENDER from `customer` where CUS\_ID IN (

select DISTINCT CUS\_ID from `order` where ORD\_AMOUNT >= 3000

)

GROUP BY CUS\_GENDER;

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

select A.\*, p.PRO\_NAME from product p inner join

(select o.ORD\_ID, o.ORD\_AMOUNT,o.ORD\_DATE, o.PRICING\_ID, o.CUS\_ID, sp.PRO\_ID from `order` o inner join supplier\_pricing sp

on o.PRICING\_ID=sp.PRICING\_ID where CUS\_ID=2) as A

on p.PRO\_ID=A.PRO\_ID ;

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

select s.\* from supplier\_pricing sp

inner join supplier s ON s.SUPP\_id = sp.SUPP\_ID

group by SUPP\_ID HAVING COUNT(\*) > 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 category.cat\_id,category.cat\_name, min(t3.min\_price) as Min\_Price from category inner join

(select product.cat\_id, product.pro\_name, t2.\* from product inner join

(select pro\_id, min(supp\_price) as Min\_Price from supplier\_pricing group by pro\_id)

as t2 where t2.pro\_id = product.pro\_id)

as t3 where t3.cat\_id = category.cat\_id group by t3.cat\_id;

#select c.CAT\_ID, c.CAT\_NAME, p.PRO\_NAME, supplier\_pricing.PRO\_ID, supplier\_pricing.minPrice from category c

#inner join product p ON p.CAT\_ID = c.CAT\_ID

#inner join (

# select MIN(sp.SUPP\_PRICE) as minPrice, sp.PRO\_ID from supplier\_pricing sp group by sp.PRO\_ID

#) supplier\_pricing ON supplier\_pricing.PRO\_ID = p.PRO\_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, o.ORD\_DATE from `order` o

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

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

where o.ORD\_DATE > "2021-10-05";

#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" OR CUS\_NAME LIKE "A%";

#Create a stored procedure to display supplier id, name, rating 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”.

CREATE DEFINER=`root`@`localhost` PROCEDURE `rating\_proc`()

BEGIN

select report.SUPP\_ID, report.SUPP\_NAME, report.r\_rating, (

CASE WHEN report.r\_rating = 5 THEN "Excellent Service"

WHEN report.r\_rating >= 4 THEN "Good Service"

WHEN report.r\_rating >= 2 THEN "Average Service"

ELSE "Poor Service"

END

) as Type\_of\_Service

from

(select s.SUPP\_ID as SUPP\_ID, s.SUPP\_NAME as SUPP\_NAME, (SUM(r.RAT\_RATSTARS) / COUNT(r.RAT\_RATSTARS)) as r\_rating from supplier s

inner join supplier\_pricing sp ON sp.SUPP\_ID = s.SUPP\_ID

inner join `order` o ON o.PRICING\_ID = sp.PRICING\_ID

inner join rating r ON r.ORD\_ID = o.ORD\_ID

GROUP BY s.SUPP\_ID) as report;

END