-- (Lab 4) Databse created as (order-directory)

CREATE DATABASE IF NOT EXISTS `order-directory`;

USE `ORDER-DIRECTORY`;

SHOW DATABASES;

SHOW TABLES;

-- Table name as (supplier) creation and attribuetes insertion.

CREATE TABLE IF NOT EXISTS 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 VALUES(1,"Rajesh Retails","Delhi",'1234567890');

INSERT INTO SUPPLIER VALUES(2,"Apario 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');

SELECT \* FROM SUPPLIER;

* Table name as (customer) creation and attribuetes insertion.

CREATE TABLE IF NOT EXISTS customer (CUST\_ID INT PRIMARY KEY, CUST\_NAME VARCHAR (20) NOT NULL,

CUS\_PHONE VARCHAR(10) NOT NULL, CUS\_CITY VARCHAR(30) NOT NULL, CUS\_GENDER CHAR);

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', "KOLKOTA",'F');

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

SELECT \* FROM CUSTOMER;

-- Table name as (category) creation and attribuetes insertion.

CREATE TABLE IF NOT EXISTS category (CAT\_ID INT PRIMARY KEY, CAT\_NAME VARCHAR(20) NOT NULL);

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");

SELECT \* FROM CATEGORY;

-- Table name as (product) creation and attribuetes insertion.

CREATE TABLE IF NOT EXISTS product (PRO\_ID INT PRIMARY KEY, PRO\_NAME VARCHAR(20) NOT NULL DEFAULT "Dummy", PRO\_DESC VARCHAR(60),

FOREIGN KEY (CAT\_ID) REFERENCES CATEFORY (CAT\_ID));

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);

SELECT \* FROM PRODUCT;

-- Table name as (supplier\_pricing) creation and attribuetes insertion.

CREATE TABLE IF NOT EXISTS supplier\_pricing (PRICING\_ID INT PRIMARY KEY, PRO\_ID INT NOT NULL, SUPP\_ID INT NOT NULL, SUPP\_PRICE INT DEFAULT 0,

FOREIGN KEY (PRO\_ID) REFERENCES RODDUCT (PRO\_ID), FOREIGN KEY (SUPP\_ID) REFERENCES SUPPLIER (SUPP\_ID));

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);

SELECT \* FROM SUPPLIER\_PRICING;

-- Table name as (order) creation and attribuetes insertion.

CREATE TABLE IF NOT EXISTS `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));

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);

SELECT \* FROM `ORDER`;

-- Table name as (rating) creation and attribuetes insertion.

CREATE TABLE IF NOT EXISTS rating(RAT\_ID INT PRIMARY KEY, ORD\_ID INT, RAT\_RATSTARS INT NOT NULL,

FOREIGN KEY (ORD\_ID)REFERENCES `order`(ORD\_ID));

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);

SELECT \* FROM RATING;

-- Query started from below.....

SELECT COUNT(t2.CUS\_GENDER) AS NoOfCustomers, t2.CUS\_GENDER FROM

(SELECT t1.CUS\_ID, t1.CUS\_GENDER, t1.ORD\_AMOUNT, t1.CUST\_NAME FROM

(SELECT `ORDER`.\*, CUSTOMER.CUS\_GENDER, CUSTOMER.CUS\_NAME FROM `ORDER` INNER JOIN CUSTOMER WHERE `ORDER`.CUS\_ID=CUSTOMER.CUS\_ID having

`ORDER`.ORD\_AMOUNT >=3000) AS t1 GROUP BY t1.CUS\_ID) AS t2 GROUP BY T2.CUS\_GENDER;

SELECT product.PRO\_NAME,`ORDER`.\* FROM `ORDER`,SUPPLIER\_PRICING.PRODUCT

WHERE `ORDER`.CUS\_ID = 2 AND `ORDER`.PRICING\_ID = SUPPLIER\_PRICING.PRICING\_ID AND SUPPLIER\_PRICING.PRO\_ID = PRODUCT.PRO\_ID;

SELECT SUPPLIER.\* FROM SUPPLIER WHERE SUPPLIER.SUPP\_ID in

(SELECT SUPP\_ID FROM SUPPLIER\_PRICING GROUP BY SUPP\_ID HAVING COUNT(SUPP\_ID) >1) GROUP BY SUPPLIER.SUPP\_ID;

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 PRODUCT.PRO\_ID, PRODUCT.PRO\_NAME FROM `ORDER` INNER JOIN SUPPLIER\_PRICING ON supplier\_pricing.PRICING\_ID=`ORDER`.PRICING\_ID INNER JOIN PRODUCT

ON PRODUCT.PRO\_ID=SUPPLIER\_PRICING.PRO\_ID WHERE `ORDER`.ORD\_DATE > "2021-10-05";

SELECT CUST\_NAME, CUS\_GENDER FROM CUSTOMER WHERE CUST\_NAME LIKE 'A%' OR CUST\_NAME LIKE '%A' ;

DELIMITER //

CREATE PROCEDURE GetAllProducts()

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.order\_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 = `ordeer`.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 GetAllProducts();