Lab 4 DBMS

Create Tables  
  
-- Creating the supplier table  
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  
);  
  
-- Creating the customer table  
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  
);  
  
-- Creating the category table  
CREATE TABLE category (  
 CAT\_ID INT PRIMARY KEY,  
 CAT\_NAME VARCHAR(20) NOT NULL  
);  
  
-- Creating the product table  
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)  
);  
  
-- Creating the supplier\_pricing table  
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)  
);  
  
-- Creating the order table  
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)  
);  
  
-- Creating the rating table  
CREATE TABLE rating (  
 RAT\_ID INT PRIMARY KEY,  
 ORD\_ID INT,  
 RAT\_RATSTARS INT NOT NULL,  
 FOREIGN KEY (ORD\_ID) REFERENCES order(ORD\_ID)  
);

3. Insert Values into the table:  
-- Inserting data into the 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', 'Banglore', '9785462315'),  
(4, 'Bansal Retails', 'Kochi', '8975463285'),  
(5, 'Mittal Ltd.', 'Lucknow', '7898456532');  
  
-- Inserting data into the 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');  
  
-- Inserting data into the category table  
INSERT INTO category (CAT\_ID, CAT\_NAME) VALUES  
(1, 'BOOKS'),  
(2, 'GAMES'),  
(3, 'GROCERIES'),  
(4, 'ELECTRONICS'),  
(5, 'CLOTHES');  
  
-- Inserting data into the 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 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);  
  
-- Inserting data into the 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);  
  
-- Inserting data into the order table  
INSERT INTO order (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);  
  
-- Inserting data into the 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);

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(\*) AS Total\_Customers  
FROM customer  
JOIN order ON customer.CUS\_ID = order.CUS\_ID  
WHERE ORD\_AMOUNT >= 3000  
GROUP BY CUS\_GENDER;

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

SELECT o.ORD\_ID, o.ORD\_AMOUNT, o.ORD\_DATE, p.PRO\_NAME  
FROM order o  
JOIN supplier\_pricing sp ON o.PRICING\_ID = sp.PRICING\_ID  
JOIN product p ON sp.PRO\_ID = p.PRO\_ID  
WHERE o.CUS\_ID = 2;

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

SELECT DISTINCT s.SUPP\_ID, s.SUPP\_NAME  
FROM supplier s  
JOIN supplier\_pricing sp ON s.SUPP\_ID = sp.SUPP\_ID  
GROUP BY s.SUPP\_ID, s.SUPP\_NAME  
HAVING COUNT(DISTINCT sp.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 c.CAT\_ID, c.CAT\_NAME, p.PRO\_NAME, MIN(sp.SUPP\_PRICE) AS Least\_Expensive\_Price  
FROM category c  
JOIN product p ON c.CAT\_ID = p.CAT\_ID  
JOIN supplier\_pricing sp ON p.PRO\_ID = sp.PRO\_ID  
GROUP BY c.CAT\_ID, c.CAT\_NAME, p.PRO\_NAME;

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

SELECT p.PRO\_ID, p.PRO\_NAME  
FROM product p  
JOIN supplier\_pricing sp ON p.PRO\_ID = sp.PRO\_ID  
JOIN order o ON sp.PRICING\_ID = o.PRICING\_ID  
WHERE o.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%' OR 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.

DELIMITER //  
CREATE PROCEDURE SupplierServiceRating()  
BEGIN  
 SELECT sp.SUPP\_ID, s.SUPP\_NAME, AVG(r.RAT\_RATSTARS) AS Average\_Rating,  
 CASE  
 WHEN AVG(r.RAT\_RATSTARS) = 5 THEN 'Excellent Service'  
 WHEN AVG(r.RAT\_RATSTARS) > 4 THEN 'Good Service'  
 WHEN AVG(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 rating r ON sp.PRICING\_ID = r.ORD\_ID  
 GROUP BY sp.SUPP\_ID, s.SUPP\_NAME;  
END //  
DELIMITER ;

To execute the stored procedure, use the following command:

CALL SupplierServiceRating();