**Problem Statement**

An E-commerce website manages its data in the form of various tables.

1. You are required to create tables for **supplier,customer,category,product,supplier\_pricing,order,rating** to store the data for the E-commerce with the schema definition given below.

**Table:supplier**

**Columns:**

|  |  |
| --- | --- |
| **SUPP\_ID** | INT PK |
| SUPP\_NAME | varchar(50) NOT NULL |
| SUPP\_CITY | varchar(50)  NOT NULL |
| SUPP\_PHONE | varchar(50) NOT NULL |

**Table:customer**

**Columns:**

|  |  |
| --- | --- |
| **CUS\_ID** | INT PK |
| CUS\_NAME | VARCHAR(20) NOT NULL |
| CUS\_PHONE | VARCHAR(10) NOT NULL |
| CUS\_CITY | VARCHAR(30) NOT NULL |
| CUS\_GENDER | CHAR |

**Table:category**

**Columns:**

|  |  |
| --- | --- |
| **CAT\_ID** | INT PK |
| CAT\_NAME | VARCHAR(20) NOT NULL |

**Table:product**

**Columns:**

|  |  |
| --- | --- |
| **PRO\_ID** | INT PK |
| PRO\_NAME | VARCHAR(20) NOT NULL DEFAULT "Dummy" |
| PRO\_DESC | VARCHAR(60) |
| **CAT\_ID** | INT FK |

**Table:supplier\_pricing**

**Columns:**

|  |  |
| --- | --- |
| **PRICING\_ID** | INT PK |
| **PRO\_ID** | INT FK |
| **SUPP\_ID** | INT FK |
| SUPP\_PRICE | INT DEFAULT 0 |

**Table:order**

**Columns:**

|  |  |
| --- | --- |
| ORD\_ID | INT PK |
| ORD\_AMOUNT | INT  NOT NULL |
| ORD\_DATE | DATE  NOT NULL |
| **CUS\_ID** | INT FK |
| **PRICING\_ID** | INT FK |

**Table:rating**

**Columns:** **(Rating provided in this table is common for product and supplier)**

|  |  |
| --- | --- |
| **RAT\_ID** | INT PK |
| **ORD\_ID** | INT FK |
| RAT\_RATSTARS | INT  NOT NULL |

1. Insert the following data in the table created above

Supplier Table-

**SUPP\_ID SUPP\_NAME SUPP\_CITY SUPP\_PHONE**

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

Customer Table-

**CUS\_ID CUS\_NAME CUS\_PHONE CUS\_CITY CUS\_GENDER**

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

Category Table-

**CAT\_ID CAT\_NAME**

1 BOOKS

2 GAMES

3 GROCERIES

4 ELECTRONICS

5 CLOTHES

Product Table-

**PRO\_ID PRO\_NAME PRO\_DESC CAT\_ID**

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

Supplier\_pricing Table-

**PRICING\_ID PRO\_ID SUPP\_ID SUPP\_PRICE**

1 1 2 1500

2 3 5 30000

3 5 1 3000

4 2 3 2500

5 4 1 1000

Order Table-

**ORD\_ID ORD\_AMOUNT ORD\_DATE CUS\_ID PRICING\_ID**

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 2021-00-10 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

Rating table-

**RAT\_ID ORD\_ID RAT\_RATSTARS**

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

**Queries →**

Write queries for the following:

1. Display the total number of customers based on gender who have placed orders of worth at least Rs.3000.
2. Display all the orders along with product name ordered by a customer having Customer\_Id=2
3. Display the Supplier details who can supply more than one product.
4. Find the least expensive product from each category and print the table with category id, name, product name and price of the product
5. Display the Id and Name of the Product ordered after “2021-10-05”.
6. Display customer name and gender whose names start or end with character 'A'.
7. 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”.

**Queries**

3)

select cus\_gender,count(cus\_gender) from customer where cus\_id in

(select cus\_id from `order` group by cus\_id having sum(ord\_amount) >= 3000) group by cus\_gender;

4)

select p.PRO\_NAME, o.\* from product p,`order` o, supplier\_pricing sp

where p.PRO\_ID = sp.PRO\_ID

and o.PRICING\_ID = sp.PRICING\_ID

and o.CUS\_ID = 2;

5)

select s1.\*, b1.total\_orders from supplier s1 , (select sp.SUPP\_ID,count(\*) as total\_orders

from `order` o, supplier\_pricing sp, supplier s

where sp.SUPP\_ID = s.supp\_ID

and o.PRICING\_ID = sp.PRICING\_ID

group by sp.SUPP\_ID

having count(\*) >1) b1

where s1.supp\_id = b1.SUPP\_ID;

6)

select p1.\* from product as p1, (

select p.pro\_name,p.pro\_id,p.CAT\_ID, min(sp.SUPP\_PRICE) as Min\_price

from supplier\_pricing sp , product p

where sp.pro\_id = p.pro\_id

group by p.pro\_id;

7)

select sp.pro\_id, p.Pro\_Name

from `order` 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 ORD\_Date > '2021-10-05';

8)

select cus\_name,cus\_gender from customer where cus\_name like 'A%' or cus\_name like '%A';

9)

delimiter $$

ALTER procedure Rating\_procedure()

begin

select sp.SUPP\_ID, s.Supp\_Name, AVG(r.RAT\_RATSTARS) As 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 `order` o

INNER JOIN rating r ON r.ORD\_ID = o.ORD\_ID

INNER JOIN supplier\_pricing sp ON sp.Pricing\_ID = o.Pricing\_ID

INNER JOIN Supplier s ON s.Supp\_ID = sp.SUPP\_ID

GROUP BY sp.SUPP\_ID;

end $$

delimiter ;

CALL Rating\_procedure();