**Query 4) Display the total number of customers based on gender who have placed individual orders of worth at least Rs.3000.**

SELECT

COUNT(t2.cus\_gender) AS NoOfCustomers, t2.cus\_gender

FROM

(SELECT

t1.cus\_id, t1.cus\_gender, t1.ord\_amount, t1.cus\_name

FROM

(SELECT

`order`.\*, customer.cus\_gender, customer.cus\_name

FROM

`order`

INNER JOIN customer ON `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;

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

SELECT

product.pro\_name as product\_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;

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

SELECT \* 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;

**Query 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 category.cat\_id, category.cat\_name, MIN(t3.min\_price) AS Min\_Price

FROM category

INNER JOIN

(SELECT product.cat\_id, product.pro\_name AS p, 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;

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

Select product.PRO\_ID , product.PRO\_Name FROM `order`

INNER JOIN supplier\_pricing

ON `order`.pricing\_ID=supplier\_pricing.Pricing\_ID

INNER JOIN product ON

product.PRO\_ID= supplier\_pricing.PRO\_ID

WHERE`order`.ord\_date > '2021-10-05';

**Query 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' ;

**Query 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. 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**.

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