**LAB 4-DBMS SOLUTIONS**

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

select count(ID) as CustomerCount, Gender from

(select distinct customer.CUS\_ID as ID, customer.CUS\_GENDER as Gender from customer inner join `order` on customer.CUS\_ID=`order`.CUS\_ID where ORD\_AMOUNT>=3000) as Subquery group by Gender;

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

select Ord.ORD\_ID as Order\_ID,Ord.ORD\_AMOUNT as Order\_Amount, Ord.ORD\_Date, Product.PRO\_NAME as Product\_Name,ORd.CUS\_ID as Customer\_ID from `Order` Ord

inner join supplier\_pricing as Sp on Ord.pricing\_ID=Sp.pricing\_ID

inner join Product on Product.Pro\_ID=Sp.Pro\_id

where Ord.CUS\_ID=2;

**3. 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;

**4. Find the least expensive product from each category and print the table with category id, name, product name and price of the product**

select cat.CAT\_ID, cat.CAT\_NAME ,min(sp.SUPP\_PRICE) as Product\_Price from category cat

inner join product prod on cat.CAT\_ID=prod.CAT\_ID

inner join supplier\_pricing sp on sp.PRO\_ID=prod.PRO\_ID

group by cat.CAT\_ID;

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

select prod.PRO\_ID, prod.PRO\_NAME, ord.ORD\_DATE from product prod

inner join supplier\_pricing sp on sp.PRO\_ID=prod.PRO\_ID

inner join `order-directory`.`order` ord on ord.PRICING\_ID =sp.PRICING\_ID

where ord.ORD\_DATE >'2021-10-05';

**6. 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';

**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**

CREATE DEFINER=`root`@`localhost` PROCEDURE `RatingService`()

BEGIN

select supplier.SUPP\_ID, supplier.SUPP\_NAME, rating.RAT\_RATSTARS,

case

when rating.RAT\_RATSTARS=5 then 'Excellent Service'

when rating.RAT\_RATSTARS>4 then 'Good Service'

when rating.RAT\_RATSTARS>2 then 'Average Service'

else 'Poor Service'

end

as Type\_Of\_Service

from supplier

inner join supplier\_pricing on supplier\_pricing.SUPP\_ID=supplier.SUPP\_ID

inner join `order` on supplier\_pricing.PRICING\_ID=`order`.PRICING\_ID

inner join rating on rating.ORD\_ID=`order`.ORD\_ID;

END