SQL CODING CHALLENGE

Ecommerce – SQL

NAME:JEFFRY JOSHUVA AMALAN

**QUESTIONS AND ANSWERS:**

**1.Update refrigerator product price to 800.**

update products set price = 800.00 where name = 'Refrigerator'

**2. Remove all cart items for a specific customer.**

delete from cart where customer\_id=7

**3.Retrieve Products Priced Below $100**

select \* from products where price<100

**4.Find Products with Stock Quantity Greater Than 5**

select \* from products where stockQuantity>5

**5.Retrieve Orders with Total Amount Between $500 and $1000.**

select \* from orders where total\_price between 500 and 1000

**6.Find Products which name end with letter ‘r’.**

select \* from products where name like '%r'

**7.Retrieve Cart Items for Customer 5**

select \* from cart where customer\_id = 5

**8.Find Customers Who Placed Orders in 2023.**

select distinct c.customer\_id, c.firstname+' '+ c.lastname, c.email from customers c inner join orders o on c.customer\_id = o.customer\_id where o.order\_date like '2023%'

**9.Determine the Minimum Stock Quantity for Each Product Category.**

select category,min(stockQuantity) as 'Minimum stock' from products group by category

**10.Calculate the Total Amount Spent by Each Customer.**

select c.customer\_id,c.firstname+' '+c.lastname as 'Customer Name',sum(o.total\_price) as 'Total price' from customers c inner join orders o on c.customer\_id = o.customer\_id group by c.customer\_id,c.firstname,c.lastname

**11. Find the Average Order Amount for Each Customer.**

select c.customer\_id,c.firstname+' '+c.lastname as 'Customer Name',avg(o.total\_price) as 'Average Amount' from customers c inner join orders o on c.customer\_id = o.customer\_id group by c.customer\_id,c.firstname,c.lastname

**12.Count the Number of Orders Placed by Each Customer.**

select c.customer\_id,c.firstname+' '+c.lastname as 'Customer Name',count(o.order\_id) as 'Orders' from customers c inner join orders o on c.customer\_id = o.customer\_id group by c.customer\_id,c.firstname,c.lastname

**13.Find the Maximum Order Amount for Each Customer.**

select c.customer\_id,c.firstname+' '+c.lastname as 'Customer Name',Max(o.total\_price) as 'Maximum Order Amount' from customers c inner join orders o on c.customer\_id = o.customer\_id group by c.customer\_id,c.firstname,c.lastname

**14.Get Customers Who Placed Orders Totaling Over $1000**

select c.customer\_id,c.firstname+' '+c.lastname as 'Customer Name',sum(o.total\_price) as 'Total Spent' from customers c inner join orders o on c.customer\_id = o.customer\_id group by c.customer\_id,c.firstname,c.lastname having sum(o.total\_price)>1000

**15.Subquery to Find Products Not in the Cart**

select \* from products where product\_id not in (select product\_id from cart)

**16.Subquery to Find Customers Who Havent Placed Orders**

select \* from customers where customer\_id not in (select customer\_id from orders)

**17. Subquery to Calculate the Percentage of Total Revenue for a Product.**

select p.product\_id,p.name,sum(o.itemAmount) as 'TotalAmount',(sum(o.itemAmount)\*100.00)/(select sum(itemAmount) from order\_items) as 'Percentage' from order\_items o inner join products p on o.product\_id = p.product\_id group by p.product\_id,p.name

**18.Subquery to Find Products with Low Stock.**

**--here I have taken the average products stock quantity and compared with each quantity**

**The average quantity involes all the products--**

select \* from products where stockQuantity<(select avg(stockQuantity) from products)

**19.Subquery to Find Customers Who Placed High-Value Orders.**

**--here i have assigned high value orders as one who purchase more than $1000--**

select \* from customers where customer\_id in (select customer\_id from orders group by customer\_id having sum(total\_price)>1000)