SQL CODING CHALLENGE

Ecommerce – SQL

NAME:JEFFRY JOSHUVA AMALAN

**Creating a database:**

create database Ecommerce

**using the database:**

use Ecommerce

**creating the table customers,products,cart,orders,order\_items:**

create table customers(customer\_id int not null primary key,firstname varchar(50),lastname varchar(50),email varchar(50),password varchar(50))

create table products(product\_id int not null primary key,name varchar(50),price decimal(10,2),description varchar(255),stockQuantity int)

create table cart(cart\_id int not null primary key,customer\_id int,product\_id int,quantity int,foreign key(customer\_id) references customers(customer\_id),foreign key(product\_id) references products(product\_id))

create table orders(order\_id int not null primary key,customer\_id int,order\_date varchar(50),total\_price decimal(10,2),shipping\_address varchar(255),foreign key(customer\_id) references customers(customer\_id))

create table order\_items(order\_item\_id int not null primary key,order\_id int, product\_id int,quantity int,foreign key(order\_id) references orders(order\_id),foreign key(product\_id) references products(product\_id))

**inserting the values for the given table :**

**products table:**

insert into products (product\_id, name, price, description, stockQuantity) values

(1, 'Laptop', 800.00, 'High-performance laptop', 10),

(2, 'Smartphone', 600.00, 'Latest smartphone', 15),

(3, 'Tablet', 300.00, 'Portable tablet', 20),

(4, 'Headphones', 150.00, 'Noise-canceling', 30),

(5, 'TV', 900.00, '4K Smart TV', 5),

(6, 'Coffee Maker', 50.00, 'Automatic coffee maker', 25),

(7, 'Refrigerator', 700.00, 'Energy-efficient', 10),

(8, 'Microwave Oven', 80.00, 'Countertop microwave', 15),

(9, 'Blender', 70.00, 'High-speed blender', 20),

(10, 'Vacuum Cleaner', 120.00, 'Bagless vacuum cleaner', 10)

**Customer table:**

insert into customers (customer\_id, firstname, lastname, email, password) values

(1, 'John', 'Doe', 'johndoe@example.com', 'john123'),

(2, 'Jane', 'Smith', 'janesmith@example.com', 'smith@123'),

(3, 'Robert', 'Johnson', 'robert@example.com', 'robert#41'),

(4, 'Sarah', 'Brown', 'sarah@example.com', '1234@sarah'),

(5, 'David', 'Lee', 'david@example.com', 'davidlee#%t'),

(6, 'Laura', 'Hall', 'laura@example.com', 'laura123#hall'),

(7, 'Michael', 'Davis', 'michael@example.com', 'Michael@2025'),

(8, 'Emma', 'Wilson', 'emma@example.com', 'emmawilson#2378'),

(9, 'William', 'Taylor', 'william@example.com', 'william123@$'),

(10, 'Olivia', 'Adams', 'olivia@example.com', '123456#@adam')

**Orders table:**

insert into orders (order\_id, customer\_id, order\_date, total\_price, shipping\_address) values

(1, 1, '2025-01-05', 1200.00, '12/B cross street, Nazareth'),

(2, 2, '2025-02-10', 900.00, '45-A East Road,chennai'),

(3, 3, '2025-03-15', 300.00, '7/A North street,bangalore'),

(4, 4, '2025-04-20', 150.00, '1st cross street,kovilpatti'),

(5, 5, '2025-05-25', 1800.00, '12C 2nd street,Erode'),

(6, 6, '2025-06-01', 400.00, '5-c,South street, Nazareth'),

(7, 7, '2025-06-05', 700.00, '159 church street, bangalore'),

(8, 8, '2025-06-10', 160.00, '8 north street,chennai'),

(9, 9, '2025-06-15', 140.00, '3rd street,Nagarkovil'),

(10, 10, '2025-06-16', 1400.00, '33/C 2nd cross street, erode')

**Order items table:**

insert into order\_items (order\_item\_id, order\_id, product\_id, quantity, itemAmount) values

(1, 1, 1, 2, 1600.00),

(2, 1, 3, 1, 300.00),

(3, 2, 2, 3, 1800.00),

(4, 3, 5, 2, 1800.00),

(5, 4, 4, 4, 600.00),

(6, 4, 6, 1, 50.00),

(7, 5, 1, 1, 800.00),

(8, 5, 2, 2, 1200.00),

(9, 6, 10, 2, 240.00),

(10, 6, 9, 3, 210.00)

**Cart table:**

insert into cart (cart\_id, customer\_id, product\_id, quantity) values

(1, 1, 1, 2),

(2, 1, 3, 1),

(3, 2, 2, 3),

(4, 3, 4, 4),

(5, 3, 5, 2),

(6, 4, 6, 1),

(7, 5, 1, 1),

(8, 6, 10, 2),

(9, 6, 9, 3),

(10, 7, 7, 2)

**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)