SQL Queries:

1. Select Products with price range 1000 - 2000 :

```
SELECT *
FROM Product
WHERE Product_Price >= 1000 and Product_Price <= 2000;
```

2. Show all delivery agents with a rating above 80:

```
SELECT *
FROM DeliveryAgent
WHERE DeliveryAgent_Rating > 80;
```

3. Show all products in a customer's cart:

```
SELECT Product.Product_ID, Product.Product_Name, Product.Product_Price,
Product.Product_Discount, Product.Product_Description,
Product.Product_Image, Product.Category_ID, Product.Retailer_ID
FROM Cart
INNER JOIN Product ON Cart.Product_ID = Product.Product_ID
WHERE Cart.User_ID = 1;
```

4. Order history of a particular customer / Show all orders for a specific customer:

```
SELECT *
FROM Orders
WHERE User_ID = 8;
```

5. Place a new order

```
INSERT INTO Orders (User_ID, DeliveryAgent_ID, Order_Amount, Order_Date,
Order_Day, Order_Month, Order_Year)
VALUES (1, 1, 999, '2023-02-16', DAY('2023-02-16'), MONTH('2023-02-16'),
YEAR('2023-02-16'));
```

6. Add a product to a customer's cart

```
INSERT INTO Cart (User_ID, Product_ID)
VALUES (1, 5);
```

7. Show all the products sold by a specific retailer

```
SELECT *
FROM Product
WHERE Retailer_ID = 1;
```

8. Show all products under a specific category:

```
SELECT *
FROM Product
WHERE Category_ID = 1;
```

9. Add a product to a specific category:

```
INSERT INTO Product (Product_Name, Product_Price, Product_Quantity,
Product_Discount, Product_Description, Product_Image, Category_ID,
Retailer_ID)
VALUES ('Samsung Galaxy S21', 10000, 50, 10, 'A new smartphone from
Samsung', 's21.jpg', 1, 1);
```

10. Display all the products ordered by a particular customer in an order

```
SELECT Product.Product_ID, Product.Product_Name, Product.Product_Price,
Product.Product_Discount, Product.Product_Description, Product.Product_Image
FROM Orders
INNER JOIN Cart ON Orders.User_ID = Cart.User_ID
INNER JOIN Product ON Cart.Product_ID = Product.Product_ID
WHERE Orders.User_ID = 1;
```

11. Show the total sales made by each retailer for the current month.

```
SELECT r.Retailer_fName, r.Retailer_ID, SUM(o.Order_Amount) AS total_sales
FROM Retailer r
INNER JOIN Product p ON p.Retailer_ID = r.Retailer_ID
INNER JOIN Orders o ON o.User_ID = p.Product_ID
WHERE MONTH(o.Order_Date) = MONTH(CURDATE())
GROUP BY r.Retailer_ID;
```

12. List the top 5 delivery agents with the highest average rating, including their names and average rating.

```
SELECT CONCAT(d.DeliveryAgent_fName, ' ', d.DeliveryAgent_mName, ' ',
d.DeliveryAgent_lName) AS delivery_agent_name,
AVG(d.DeliveryAgent_Rating) AS avg_rating
FROM DeliveryAgent d
JOIN Orders o ON d.DeliveryAgent_ID = o.DeliveryAgent_ID
GROUP BY d.DeliveryAgent_ID
ORDER BY avg_rating DESC
LIMIT 5;
```

13. Show the total amount of money spent by each customer in the last month.

```
SELECT c.User_fName, SUM(o.Order_Amount) AS total_spent
FROM Customer c
INNER JOIN Orders o ON o.User_ID = c.User_ID
WHERE o.Order_Date >= DATE_SUB(CURDATE(), INTERVAL 1 MONTH)
GROUP BY c.User_ID;
```

14. Find the top 2 customers who have spent the most on their orders in the past 6 months, along with the total amount spent by each customer.

```
SELECT c.User_fName, SUM(o.Order_Amount) AS total_spent
FROM Customer c
JOIN Orders o ON c.User_ID = o.User_ID
WHERE o.Order_Date BETWEEN DATE_SUB(NOW(), INTERVAL 6 MONTH)
AND NOW()
GROUP BY c.User_ID
ORDER BY Total_Spent DESC
LIMIT 2;
```

15. Update the product quantity by subtracting the count of products in the user's cart and then deleting all items from the cart for that user whenever a Cart is converted to an Order

```
UPDATE Product
SET Product_Quantity = Product_Quantity - (
  SELECT COUNT(*)
  FROM Cart
  WHERE User_ID = 8 AND Product_ID = Product.Product_ID
)
WHERE Product_ID IN (
  SELECT Product_ID
  FROM Cart
  WHERE User_ID = 8
);

DELETE FROM Cart
WHERE User_ID = 8;
```

Relational Schema:

Relational Model