Part 2: SQL Test

Instructions: Please write SQL queries for each of the following questions. You may assume that all tables follow typical database conventions unless otherwise specified.

Tables: Assume you have the following tables in your database:

1.Sales

- sales_id (INT)
- customer_id (INT)
- product_id (INT)
- sale_date (DATE)
- quantity (INT)
- total_amount (DECIMAL)

2.Customers

- customer_id (INT)
- customer_name (VARCHAR)
- sales_region (VARCHAR)
- sign_up_date (DATE)

3.Products

- product_id (INT)
- product_name (VARCHAR)
- category (VARCHAR)
- price (DECIMAL)

Questions:

1.Write a query to return the customer_name, product_name, and total_amount for each sale in the last 30 days.

```
SELECT c.customer_name, p.product_name, s.total_amount

FROM Customers c

JOIN Sales s ON s.customer_id=c.customer_id

JOIN Products p ON p.product_id =s.product_id

WHERE s.sale_date >=DATEADD(DAY, -30,GETDATE());
```

2.Write a query to find the total revenue generated by each product category in the last year. The output should include the product category and the total revenue for that category.

```
SELECT p.category as Category ,SUM(s.total_amount) as Total_Revenue
FROM Sales s
JOIN Products p ON s.product_id=p.product_id
WHERE s.sale_date>=CURRENT_DATE-INTERVAL 1 YEAR
GROUP BY p.category
```

3.Write a query to return all customers who made purchases in 2023 and are located in the "West" region.

```
SELECT DISTINCT c.customer_id , c.customer_name
FROM Customer c
JOIN Sales s ON s.customer_id=c.customer_id
WHERE c.sales_region=”West”
AND
YEAR(s.sale_date)=2023
```

4.Write a query to display the total number of sales, total quantity sold, and total revenue for each customer. The result should include the customer_name, total sales, total quantity, and total revenue.

```
SELECT c.customer_name, COUNT(s.sale_id) As Total_Sale,
                SUM(s.quantity) AS Total_Quantity, SUM(s.total_amount) AS
Total_Revenue
FROM Sales s
JOIN Customers c ON s.customer_id=s.customer_id
GROUP BY c.customer_name
```

5.Write a query to find the top 3 customers (by total revenue) in the year 2023.

```
SELECT c.customer_name, SUM(s.total_amount) AS Total_revenue
FROM Customers c
JOIN Sales s ON s.customer_id = c.customer_id
WHERE YEAR(s.sale_date)=2023
GROUP BY c.customer_name ORDER BY Total_revenue
LIMIT 3
```

6.Write a query to rank products by their total sales quantity in 2023. The result should include the product_name, total quantity sold, and rank.

```
SELECT p.product_name, SUM(s.quantity), RANK() OVER(ORDER BY SUM(s.quantity)
DESC) AS Rank
FROM Sales s
JOIN  Products p ON s.product_id=s.product_id
WHERE YEAR(s.sale_date)=2023
GROUP BY p.product_name
```

7.Write a query that categorizes customers into "New" (if they signed up in the last 6 months) or "Existing" based on their sign_up_date. Include the customer_name, region, and category in the result.

```
SELECT  c.customer_name, c.sales_region,
        CASE
        WHEN c.signupdate >=CURRENT_DATE – INTERVAL 6 MONTH THEN
        ‘New’
        WHEN c.signupdate <CURRENT_DATE – INTERVAL 6 MONTH THEN
        ‘Existing’
END AS Category
FROM Customer c
JOIN Sales s ON s.customer_id=c.customer_id
```

8.Write a query to return the month and year along with the total sales for each month for the last 12 months.

```
SELECT DATE_FORMAT(s.sale_date,”%Y-%m’) AS sale_date,
SUM(s.total_amount) AS Total_Amount
FROM Sales s
WHERE s.sale_date >=CURRENT_DATE -INTERVAL 1 YEAR
GROUP BY sale_date
ORDER BY Sale_date
```

9.Write a query to return the product categories that generated more than $50,000 in revenue during the last 6 months.

```
SELECT p.category, SUM(s.total_amount) AS Revenue
FROM Products p
```

```
JOIN Sales s ON s.product_id=p.product_id
WHERE s.sale_date >=CURRENT_DATE -INTERVAL 6 MONTH
HAVING Revenue >=50000
GROUP BY p.category
```

10.Write a query to check for any sales where the total_amount doesn't match the expected value (i.e., quantity * price).

```
SELECT s.sale_id, c.customer_name,p.product_name, s.quantity , (s.quanitity * p.price)
AS Expected_value

FROM Customers c

JOIN Sales s ON c.customer_id=s.customer_id

JOIN Products p ON p.product_id = s.product_id

WHERE s.total_amount != Expected_value
```