Data analyst

Interview questions

for entry-level roles

Part I

1. Calculate the Rolling Average of Sales Over the Past 30 Days

- **Problem Statement:** You need to analyze the rolling average of daily sales to understand recent trends. Specifically, calculate the average sales amount over the past 30 days for each day.

- **Solution:** Use window functions to compute the rolling average.

```sql
SELECT
    order_date,
    sales_amount,
    AVG(sales_amount) OVER (
        ORDER BY order_date
        ROWS BETWEEN 29 PRECEDING AND CURRENT ROW
    ) AS rolling_avg
FROM sales;
```

2. Identify the Top 5 Products by Revenue in the Last Quarter

Problem Statement: Your task is to find the top 5 products that generated the highest revenue in the last quarter. This will help prioritize inventory and marketing efforts.

Solution: Use aggregate functions with filtering and ordering.

```sql
SELECT
    product_id,
    SUM(revenue) AS total_revenue
FROM sales
WHERE order_date BETWEEN DATEADD(QUARTER, -1, GETDATE()) AND GETDATE()
GROUP BY product_id
ORDER BY total_revenue DESC
LIMIT 5;
```

3. Find Customers Who Have Not Made a Purchase in the Last 6 Months

Problem Statement: Determine which customers have not placed any orders in the past 6 months. This information can be useful for re-engagement campaigns.

Solution: Use LEFT JOIN to find customers with no recent purchases.

```sql
SELECT c.customer_id, c.customer_name
FROM customers c
LEFT JOIN sales s ON c.customer_id = s.customer_id
AND s.order_date > DATEADD(MONTH, -6, GETDATE())
WHERE s.customer_id IS NULL;
```

4. Calculate Year-Over-Year Growth Percentage in Sales for Each Product

Problem Statement: You need to measure how sales performance has changed compared to the previous year for each product. Calculate the year-over-year growth percentage.

Solution: Compare current year sales with previous year sales.

```sql
WITH sales_per_year AS (
    SELECT
        product_id,
        YEAR(order_date) AS year,
        SUM(sales_amount) AS total_sales
    FROM sales
    GROUP BY product_id, YEAR(order_date)
)
SELECT
    s1.product_id,
    s1.year AS current_year,
    s1.total_sales AS current_year_sales,
    s2.year AS previous_year,
    s2.total_sales AS previous_year_sales,
    ((s1.total_sales - s2.total_sales) / s2.total_sales) * 100 AS
yoy_growth
FROM sales_per_year s1
LEFT JOIN sales_per_year s2
    ON s1.product_id = s2.product_id AND s1.year = s2.year + 1
WHERE s1.year = YEAR(GETDATE());
```

5. Calculate the Average Time Taken for an Order to be Delivered After It's Placed

Problem Statement: Determine the average delivery time from when an order is placed to when it is delivered. This metric will help assess delivery performance.

Solution: Compute the average duration between order placement and delivery.

```sql
SELECT
    AVG(DATEDIFF(day, order_date, delivery_date)) AS
avg_delivery_time
FROM orders
WHERE delivery_date IS NOT NULL;
```

Found this helpful? Repost!