Advanced E-commerce SQL Analysis

1. Total Revenue Generated

```
SELECT SUM(quantity * price) AS total_revenue
FROM amazon_sales;
```

2. Top 5 Revenue Generating Products

```
SELECT product_name, SUM(quantity * price) AS revenue
FROM amazon_sales
GROUP BY product_name
ORDER BY revenue DESC
LIMIT 5;
```

3. Monthly Sales Trend

```
SELECT DATE_FORMAT(order_date, '%Y-%m') AS month,
       SUM(quantity * price) AS monthly_revenue
FROM amazon_sales
GROUP BY month
ORDER BY month;
```

4. Rank Products by Sales (Window Function)

```
SELECT product_name,
       SUM(quantity) AS total_units,
       RANK() OVER (ORDER BY SUM(quantity) DESC) AS product_rank
FROM amazon_sales
GROUP BY product_name;
```

5. Find Repeat Customers

```
SELECT customer_id,
       COUNT(order_id) AS total_orders
FROM amazon_sales
GROUP BY customer_id
HAVING COUNT(order_id) > 1;
```

6. Category Contribution Percentage

```
SELECT category,
       SUM(quantity * price) AS revenue,
       ROUND(100 * SUM(quantity * price) /
       (SELECT SUM(quantity * price) FROM amazon_sales), 2) AS contribution_percent
FROM amazon_sales
GROUP BY category;
```

7. Running Total of Sales (Window Function)

```
SELECT order_date,
       SUM(quantity * price) AS daily_revenue,
       SUM(SUM(quantity * price)) OVER (ORDER BY order_date) AS running_total
FROM amazon_sales
GROUP BY order_date;
```

8. Previous Order Comparison (LAG)

```
SELECT order_date,
       SUM(quantity * price) AS daily_revenue,
       LAG(SUM(quantity * price)) OVER (ORDER BY order_date) AS previous_day_revenue
FROM amazon_sales
GROUP BY order_date;
```

9. Best Performing Region

```
SELECT customer_region,
       SUM(quantity * price) AS revenue
FROM amazon_sales
GROUP BY customer_region
ORDER BY revenue DESC;
```

10. Average Order Value

```
SELECT SUM(quantity * price) / COUNT(DISTINCT order_id) AS avg_order_value
FROM amazon_sales;
```