Task-6:Sales Trend Analysis Using Aggregations

- Query:

```
SELECT
        YEAR(`Date`) AS year,
        MONTH(`Date`) AS month,
        SUM(`Total Revenue`) AS total_revenue
FROM `online_sales_data`
GROUP BY year, month
ORDER BY year, month;
```

- What It Does:

-Extracts year and month from the Date field.

-Calculates total revenue per month using SUM(Total Revenue).

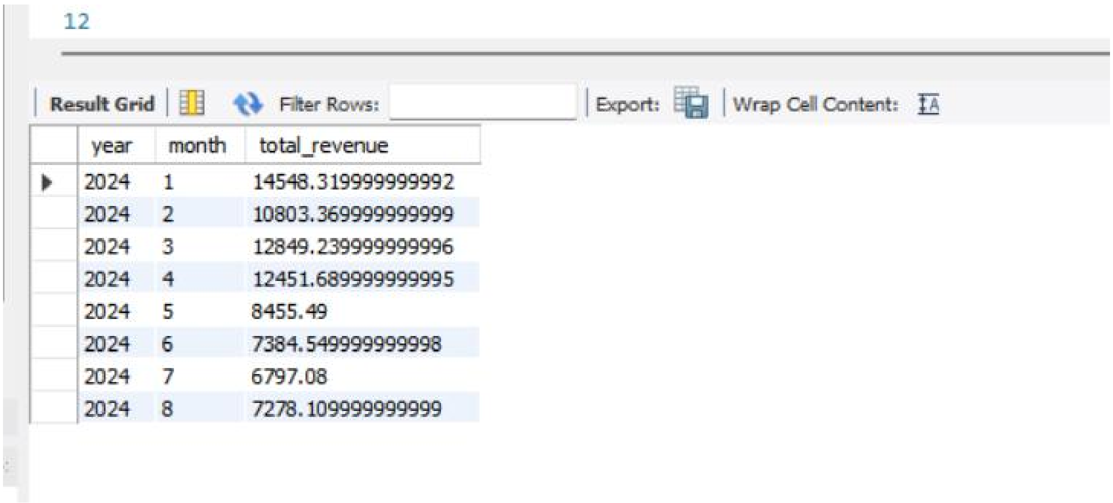
-Groups results by year and month to aggregate monthly data.

-Sorts the output in chronological order.

- Insight Gained:

Track how revenue trends over time, month by month.

- Output:



year	month	total_revenue
2024	1	14548.319999999992
2024	2	10803.369999999999
2024	3	12849.239999999996
2024	4	12451.689999999995
2024	5	8455.49
2024	6	7384.549999999998
2024	7	6797.08
2024	8	7278.109999999999

- Query:

```
SELECT
	`Product Category`,
	SUM(`Total Revenue`) AS revenue
FROM `online_sales_data`
GROUP BY `Product Category`
ORDER BY revenue DESC
LIMIT 5;
```

- What It Does:

-Aggregates total revenue per product category.

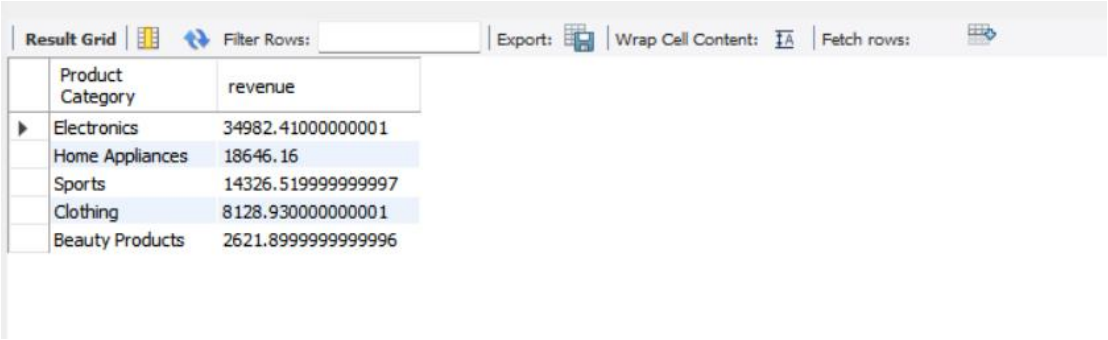
-Orders the result by revenue in descending order.

-Limits output to the top 5 categories.

- Insight Gained:

Identifies which product categories generate the most revenue.

- Output:



Product Category	revenue
Electronics	34982.41000000001
Home Appliances	18646.16
Sports	14326.519999999997
Clothing	8128.930000000001
Beauty Products	2621.8999999999996

- Query:

```
SELECT
        `Product Name`,
        SUM(`Units Sold`) AS total_units
FROM `online_sales_data`
GROUP BY `Product Name`
ORDER BY total_units DESC
LIMIT 10;
```

- What It Does:

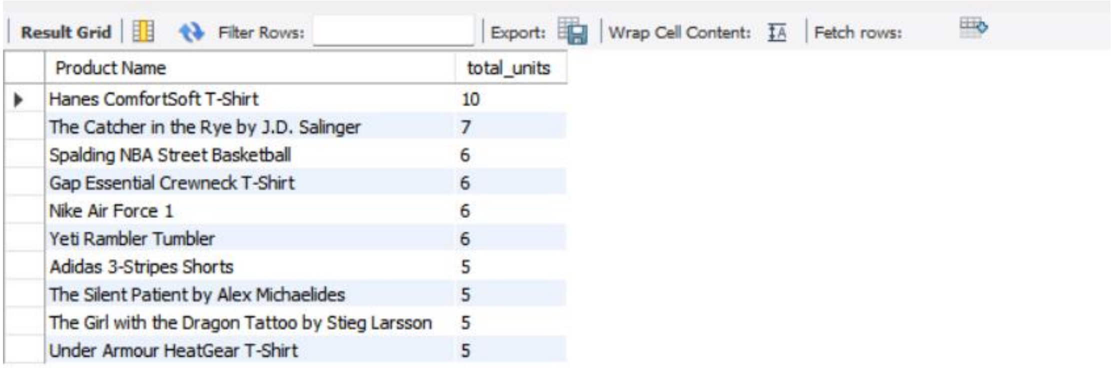
-Sums the Units Sold per product.

-Displays top 10 selling products by quantity sold.

- Insight Gained:

Understand which products are most popular with customers.

- Output:



Product Name	total_units
Hanes ComfortSoft T-Shirt	10
The Catcher in the Rye by J.D. Salinger	7
Spalding NBA Street Basketball	6
Gap Essential Crewneck T-Shirt	6
Nike Air Force 1	6
Yeti Rambler Tumbler	6
Adidas 3-Stripes Shorts	5
The Silent Patient by Alex Michaelides	5
The Girl with the Dragon Tattoo by Stieg Larsson	5
Under Armour HeatGear T-Shirt	5

- Query:

```
SELECT
	`Region`,
	SUM(`Total Revenue`) AS revenue
FROM `online_sales_data`
GROUP BY `Region`
ORDER BY revenue DESC;
```

- What It Does:

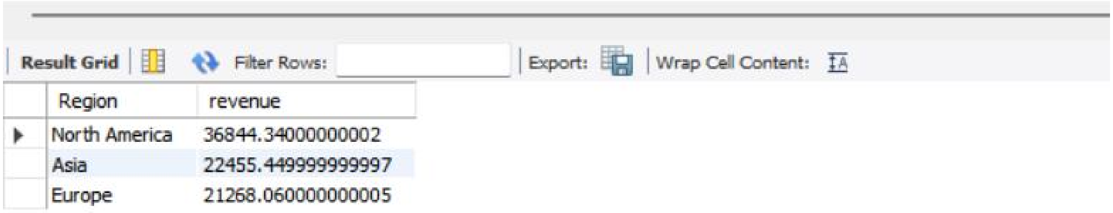
-Calculates total revenue per region.

-Sorts regions based on revenue in descending order.

- Insight Gained:

Shows which regions contribute the most to overall sales.

- Output:



Region	revenue
North America	36844.34000000002
Asia	22455.449999999997
Europe	21268.060000000005

- Query:

```
SELECT
	`Payment Method`,
	COUNT(*) AS total_transactions,
	SUM(`Total Revenue`) AS total_revenue
FROM `online_sales_data`
GROUP BY `Payment Method`
ORDER BY total_revenue DESC;
```

```
SELECT
	`Payment Method`,
	COUNT(*) AS total_transactions,
	SUM(`Total Revenue`) AS total_revenue
FROM `online_sales_data`
GROUP BY `Payment Method`
ORDER BY total_revenue DESC;
```

- What It Does:

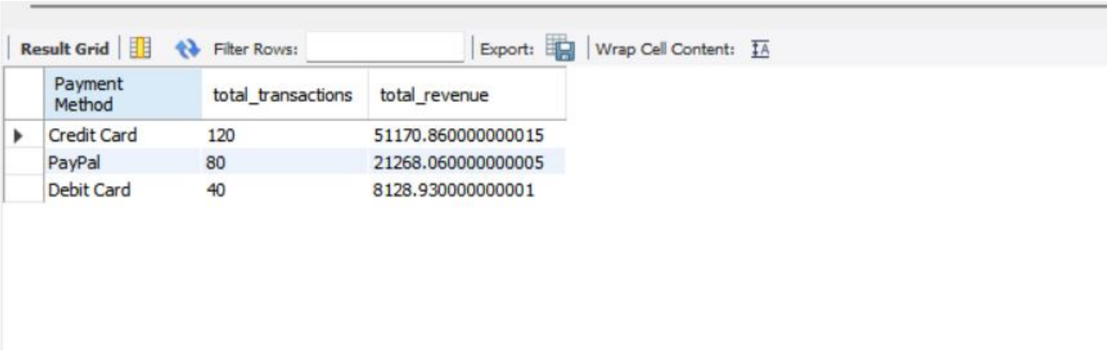
-Counts how many transactions were made per payment method.

-Calculates the revenue generated by each method.

- Insight Gained:

Reveals preferred payment options and their financial impact.

- Output:



Payment Method	total_transactions	total_revenue
Credit Card	120	51170.860000000015
PayPal	80	21268.060000000005
Debit Card	40	8128.930000000001

- Query:

```
SELECT
	`Date`,
	SUM(`Total Revenue`) AS daily_sales
FROM `online_sales_data`
GROUP BY `Date`
ORDER BY `Date`;
```

- What It Does:

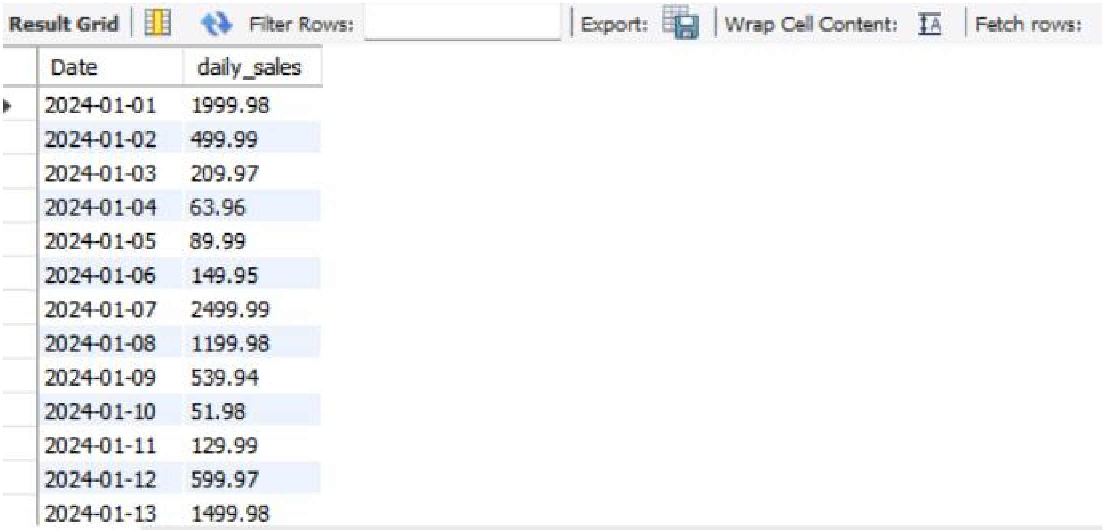
-Aggregates revenue per day.

-Displays chronological sales trends.

- Insight Gained:

Useful for plotting a time-series chart of sales.

- Output:



Date	daily_sales
2024-01-01	1999.98
2024-01-02	499.99
2024-01-03	209.97
2024-01-04	63.96
2024-01-05	89.99
2024-01-06	149.95
2024-01-07	2499.99
2024-01-08	1199.98
2024-01-09	539.94
2024-01-10	51.98
2024-01-11	129.99
2024-01-12	599.97
2024-01-13	1499.98

- Query:

```
SELECT
        `Product Category`,
        AVG(`Total Revenue`) AS avg_order_value
FROM `online_sales_data`
GROUP BY `Product Category`
ORDER BY avg_order_value DESC;
```

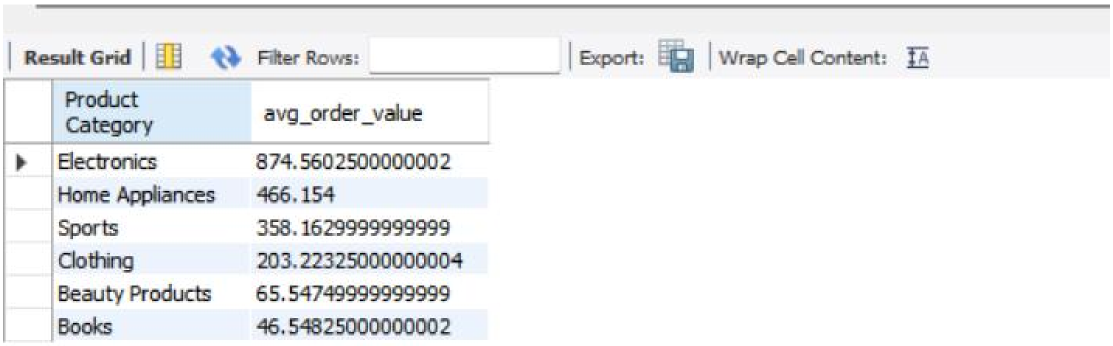
- What It Does:

-Calculates average revenue per order within each product category.

- Insight Gained:

Identifies high-value product categories — good for pricing strategy or promotions.

- Output:



Product Category	avg_order_value
Electronics	874.5602500000002
Home Appliances	466.154
Sports	358.1629999999999
Clothing	203.22325000000004
Beauty Products	65.54749999999999
Books	46.54825000000002

- Query:

```
SELECT
        YEAR(`Date`) AS year,
        MONTH(`Date`) AS month,
        SUM(`Total Revenue`) AS total_revenue,
        COUNT(DISTINCT `Transaction ID`) AS order_volume
FROM `online_sales_data`
GROUP BY year, month
ORDER BY year, month;
```

- What We Did:

-YEAR(Date) and MONTH(Date): Extract the year and month from each transaction date.

-SUM(Total Revenue): Calculates the total revenue earned in each month.

-COUNT(DISTINCT Transaction ID): Counts the number of unique transactions (orders) in each month.

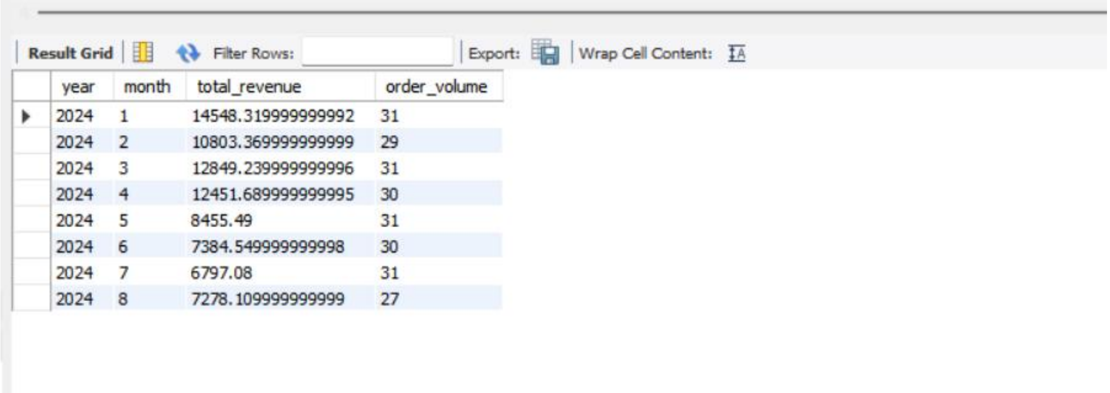
-GROUP BY year, month: Groups all the data by month and year, so we can perform monthly analysis.

-ORDER BY year, month: Sorts the results in chronological order.

- Purpose:

This gives a clear view of how much revenue and how many orders happened month by month, which helps identify sales trends over time.

- Output:



year	month	total_revenue	order_volume
2024	1	14548.319999999992	31
2024	2	10803.369999999999	29
2024	3	12849.239999999996	31
2024	4	12451.689999999995	30
2024	5	8455.49	31
2024	6	7384.549999999998	30
2024	7	6797.08	31
2024	8	7278.109999999999	27