SALES_TREND_ANALYSIS USING AGGREGATION

CREATE DATABASE

```
create database online_sales;

use online_sales;
```

IMPORT THE DATASET

Online-Store-Orders.csv

DATASET :

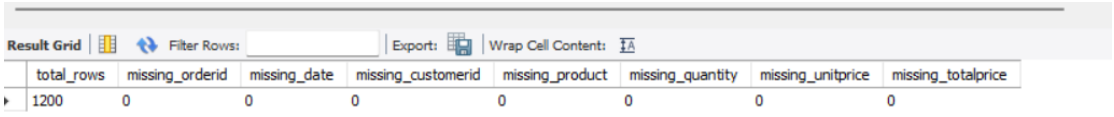
INSPECT THE DATA

```
Select * from sales_data limit 10;

Describe sales_data;

ALTER TABLE sales_data
MODIFY COLUMN Date DATE;

Describe sales_data;

SELECT * FROM sales_data LIMIT 10;
```

CLEAN THE DATA

 # check for nulls

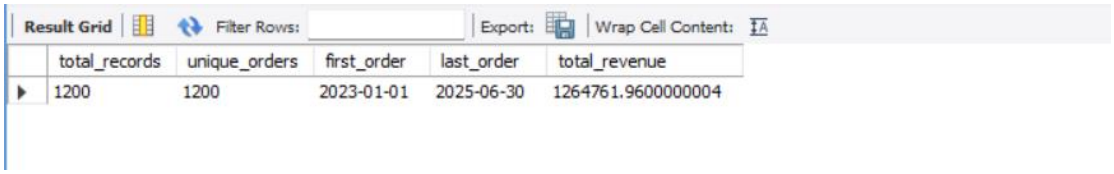
```
SELECT
    COUNT(*) AS total_rows,
    SUM(CASE WHEN OrderID IS NULL THEN 1 ELSE 0 END) AS
missing_orderid,
    SUM(CASE WHEN Date IS NULL THEN 1 ELSE 0 END) AS
missing_date,
    SUM(CASE WHEN CustomerID IS NULL THEN 1 ELSE 0 END)
AS missing_customerid,
    SUM(CASE WHEN Product IS NULL THEN 1 ELSE 0 END) AS
missing_product,
    SUM(CASE WHEN Quantity IS NULL THEN 1 ELSE 0 END) AS
missing_quantity,
    SUM(CASE WHEN UnitPrice IS NULL THEN 1 ELSE 0 END) AS
missing_unitprice,
    SUM(CASE WHEN TotalPrice IS NULL THEN 1 ELSE 0 END) AS
missing_totalprice
FROM sales_data;
```



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

total_rows	missing_orderid	missing_date	missing_customerid	missing_product	missing_quantity	missing_unitprice	missing_totalprice
1200	0	0	0	0	0	0	0

```
SELECT
    COUNT(*) AS total_records,
    COUNT(DISTINCT OrderID) AS unique_orders,
    MIN(Date) AS first_order,
    MAX(Date) AS last_order,
    SUM(TotalPrice) AS total_revenue
FROM sales_data;
```



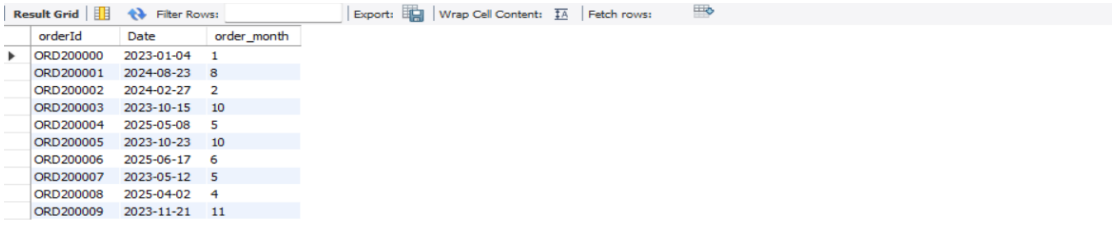
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

total_records	unique_orders	first_order	last_order	total_revenue
1200	1200	2023-01-01	2025-06-30	1264761.9600000004

QUERY

 # Show the month extracted from each order date to understand how month extraction works.

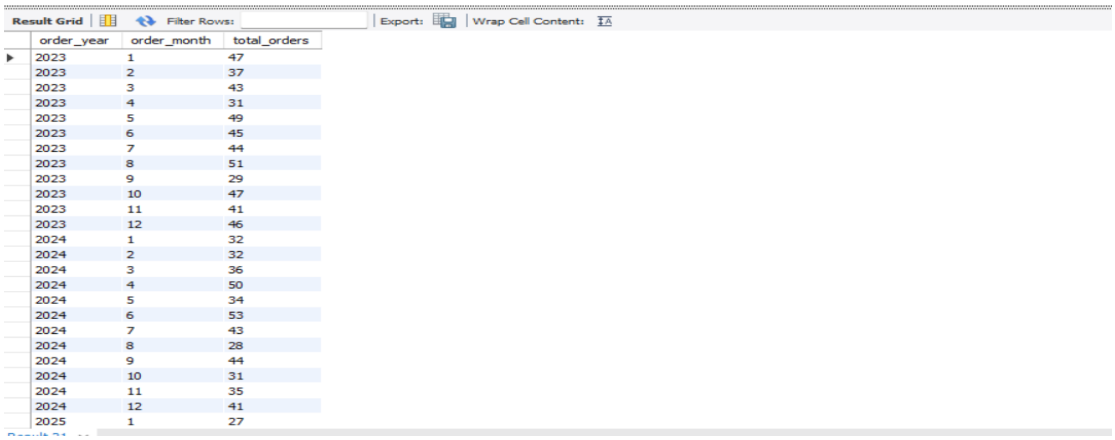
```
SELECT
    orderId,
    Date,
    EXTRACT(MONTH FROM Date) AS order_month
FROM sales_data
LIMIT 10;
```



orderId	Date	order_month
ORD200000	2023-01-04	1
ORD200001	2024-08-23	8
ORD200002	2024-02-27	2
ORD200003	2023-10-15	10
ORD200004	2025-05-08	5
ORD200005	2023-10-23	10
ORD200006	2025-06-17	6
ORD200007	2023-05-12	5
ORD200008	2025-04-02	4
ORD200009	2023-11-21	11

#Group sales data by year and month.

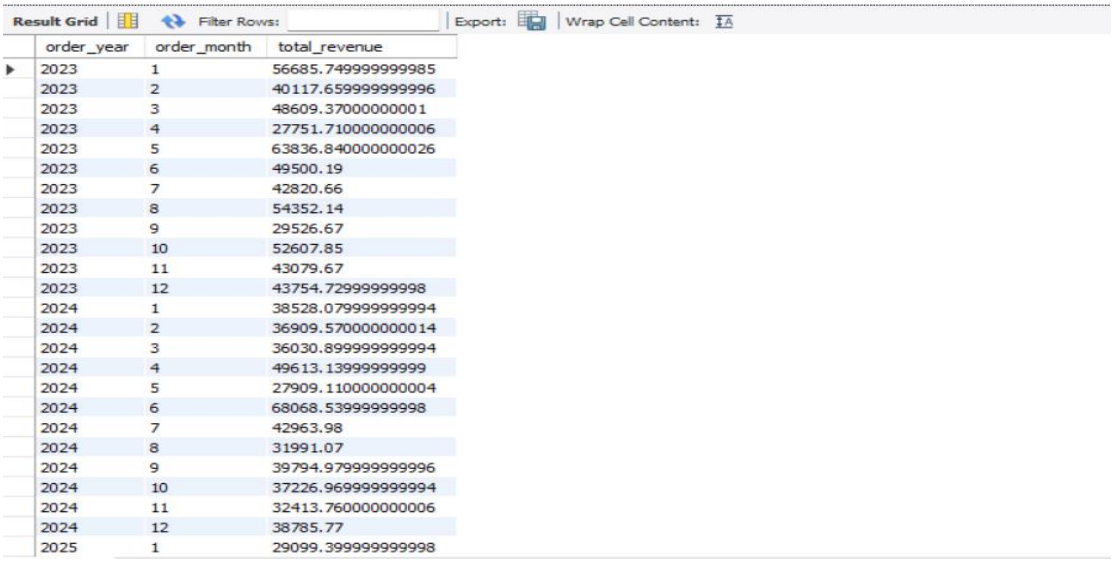
```
SELECT
    EXTRACT(YEAR FROM Date) AS order_year,
    EXTRACT(MONTH FROM Date) AS order_month,
    COUNT(*) AS total_orders
FROM sales_data
GROUP BY order_year, order_month
ORDER BY order_year, order_month;
```



order_year	order_month	total_orders
2023	1	47
2023	2	37
2023	3	43
2023	4	31
2023	5	49
2023	6	45
2023	7	44
2023	8	51
2023	9	29
2023	10	47
2023	11	41
2023	12	46
2024	1	32
2024	2	32
2024	3	36
2024	4	50
2024	5	34
2024	6	53
2024	7	43
2024	8	28
2024	9	44
2024	10	31
2024	11	35
2024	12	41
2025	1	27

 # Find total monthly revenue.

```
SELECT
   EXTRACT(YEAR FROM Date) AS order_year,
   EXTRACT(MONTH FROM date) AS order_month,
   SUM(TotalPrice) AS total_revenue
FROM sales_data
GROUP BY order_year, order_month
ORDER BY order_year, order_month;
```

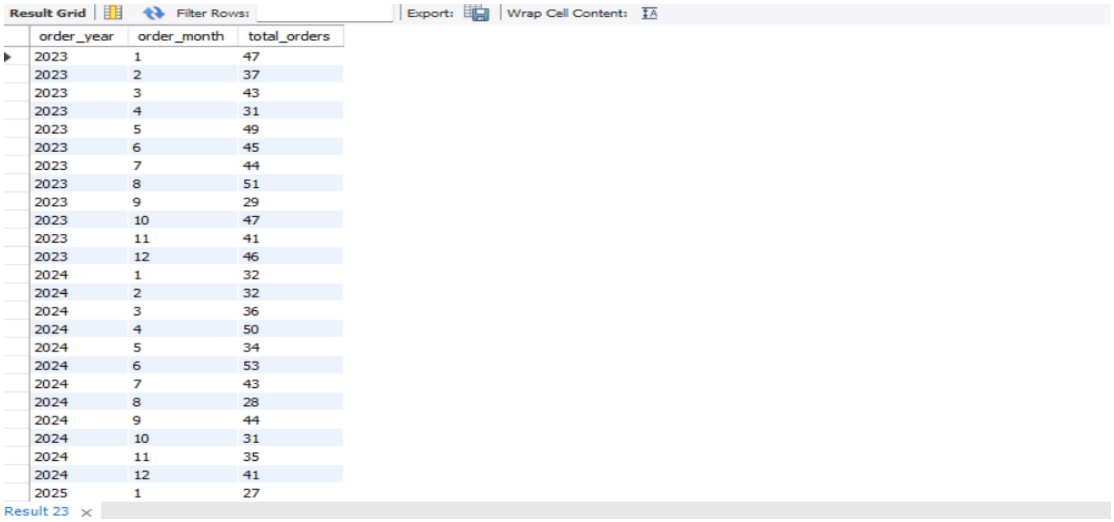


Result Grid | Filter Rows: | Export: | Wrap Cell Content:

order_year	order_month	total_revenue
2023	1	56685.749999999985
2023	2	40117.659999999996
2023	3	48609.37000000001
2023	4	27751.710000000006
2023	5	63836.840000000026
2023	6	49500.19
2023	7	42820.66
2023	8	54352.14
2023	9	29526.67
2023	10	52607.85
2023	11	43079.67
2023	12	43754.72999999998
2024	1	38528.079999999994
2024	2	36909.570000000014
2024	3	36030.899999999994
2024	4	49613.13999999999
2024	5	27909.110000000004
2024	6	68068.53999999998
2024	7	42963.98
2024	8	31991.07
2024	9	39794.979999999996
2024	10	37226.969999999994
2024	11	32413.760000000006
2024	12	38785.77
2025	1	29099.399999999998

#Find total number of unique orders each month.

```
SELECT
   EXTRACT(YEAR FROM DATE) AS order_year,
   EXTRACT(MONTH FROM DATE) AS order_month,
   COUNT(DISTINCT orderId) AS total_orders
FROM Sales_DATA
GROUP BY order_year, order_month
ORDER BY order_year, order_month;
```



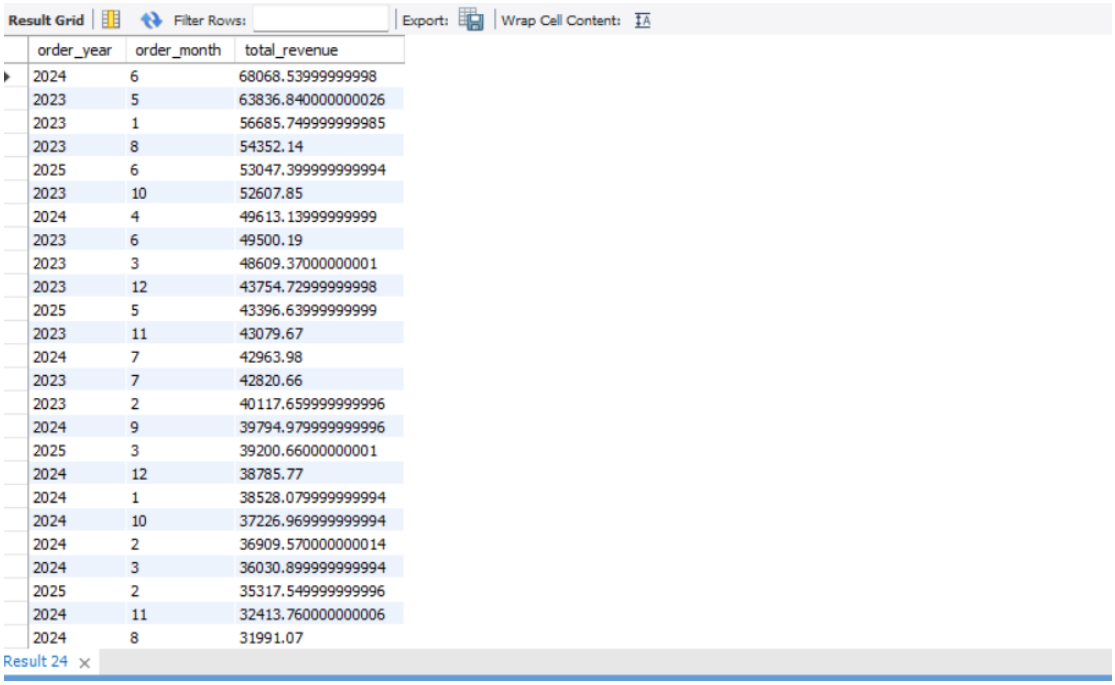
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

order_year	order_month	total_orders
2023	1	47
2023	2	37
2023	3	43
2023	4	31
2023	5	49
2023	6	45
2023	7	44
2023	8	51
2023	9	29
2023	10	47
2023	11	41
2023	12	46
2024	1	32
2024	2	32
2024	3	36
2024	4	50
2024	5	34
2024	6	53
2024	7	43
2024	8	28
2024	9	44
2024	10	31
2024	11	35
2024	12	41
2025	1	27

Result 23

#Sort the results from highest to lowest monthly revenue.

```
SELECT
    EXTRACT(YEAR FROM Date) AS order_year,
    EXTRACT(MONTH FROM Date) AS order_month,
    SUM(totalPrice) AS total_revenue
FROM sales_Data
GROUP BY order_year, order_month
ORDER BY total_revenue DESC;
```



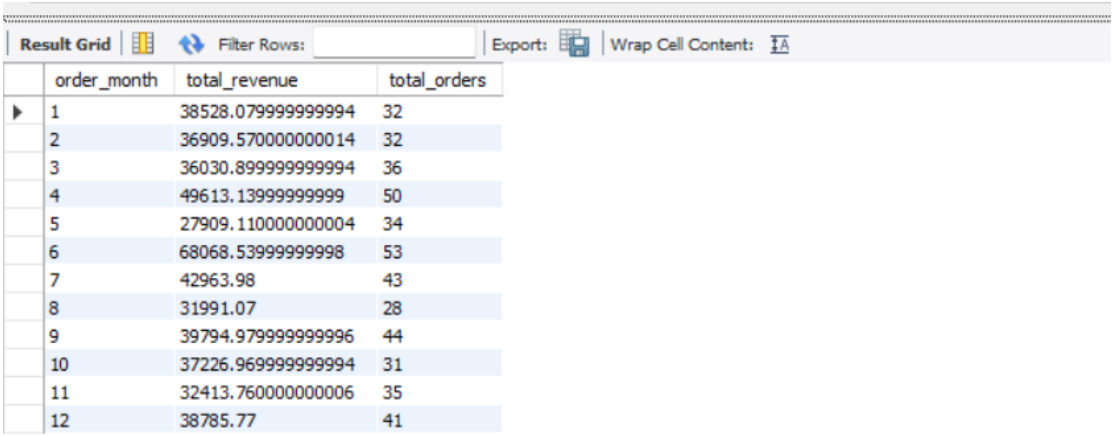
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

order_year	order_month	total_revenue
2024	6	68068.53999999998
2023	5	63836.840000000026
2023	1	56685.749999999985
2023	8	54352.14
2025	6	53047.399999999994
2023	10	52607.85
2024	4	49613.13999999999
2023	6	49500.19
2023	3	48609.37000000001
2023	12	43754.72999999998
2025	5	43396.63999999999
2023	11	43079.67
2024	7	42963.98
2023	7	42820.66
2023	2	40117.659999999996
2024	9	39794.979999999996
2025	3	39200.66000000001
2024	12	38785.77
2024	1	38528.079999999994
2024	10	37226.969999999994
2024	2	36909.570000000014
2024	3	36030.899999999994
2025	2	35317.549999999996
2024	11	32413.760000000006
2024	8	31991.07

Result 24

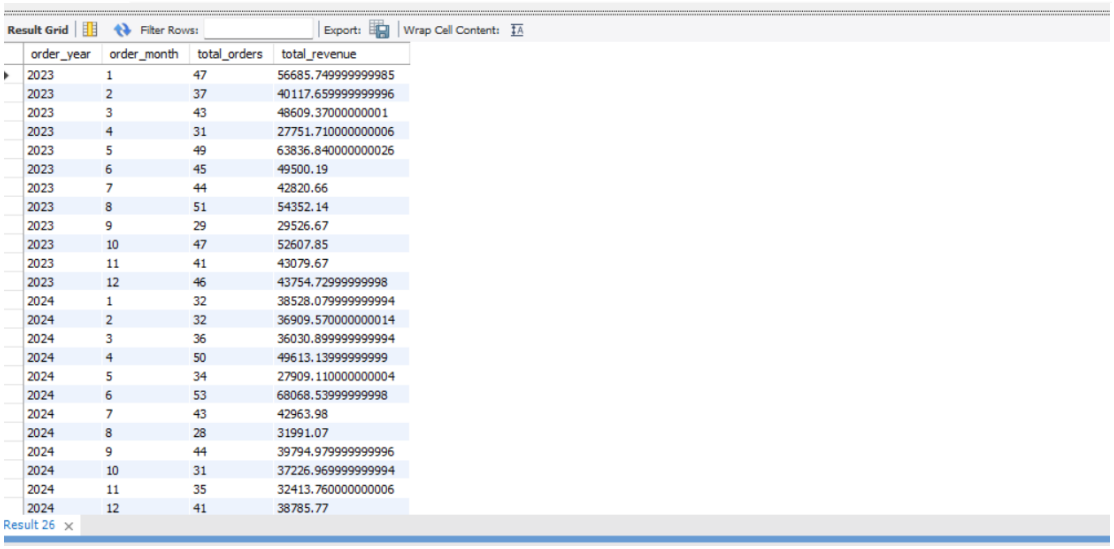
#Show sales data only for 2024.

```
SELECT
    EXTRACT(MONTH FROM date) AS order_month,
    SUM(totalPrice) AS total_revenue,
    COUNT(DISTINCT orderId) AS total_orders
FROM sales_Data
WHERE EXTRACT(YEAR FROM date) = 2024
GROUP BY order_month
ORDER BY order_month;
```



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

order_month	total_revenue	total_orders
1	38528.079999999994	32
2	36909.570000000014	32
3	36030.899999999994	36
4	49613.13999999999	50
5	27909.110000000004	34
6	68068.53999999998	53
7	42963.98	43
8	31991.07	28
9	39794.979999999996	44
10	37226.969999999994	31
11	32413.760000000006	35
12	38785.77	41



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

order_year	order_month	total_orders	total_revenue
2023	1	47	56685.749999999985
2023	2	37	40117.659999999996
2023	3	43	48609.37000000001
2023	4	31	27751.710000000006
2023	5	49	63836.840000000026
2023	6	45	49500.19
2023	7	44	42820.66
2023	8	51	54352.14
2023	9	29	29526.67
2023	10	47	52607.85
2023	11	41	43079.67
2023	12	46	43754.72999999998
2024	1	32	38528.079999999994
2024	2	32	36909.570000000014
2024	3	36	36030.899999999994
2024	4	50	49613.13999999999
2024	5	34	27909.110000000004
2024	6	53	68068.53999999998
2024	7	43	42963.98
2024	8	28	31991.07
2024	9	44	39794.979999999996
2024	10	31	37226.969999999994
2024	11	35	32413.760000000006
2024	12	41	38785.77

Result 26