TASK 6: Sales Trend Analysis Using Aggregations

1. Monthly Revenue and Order Volume

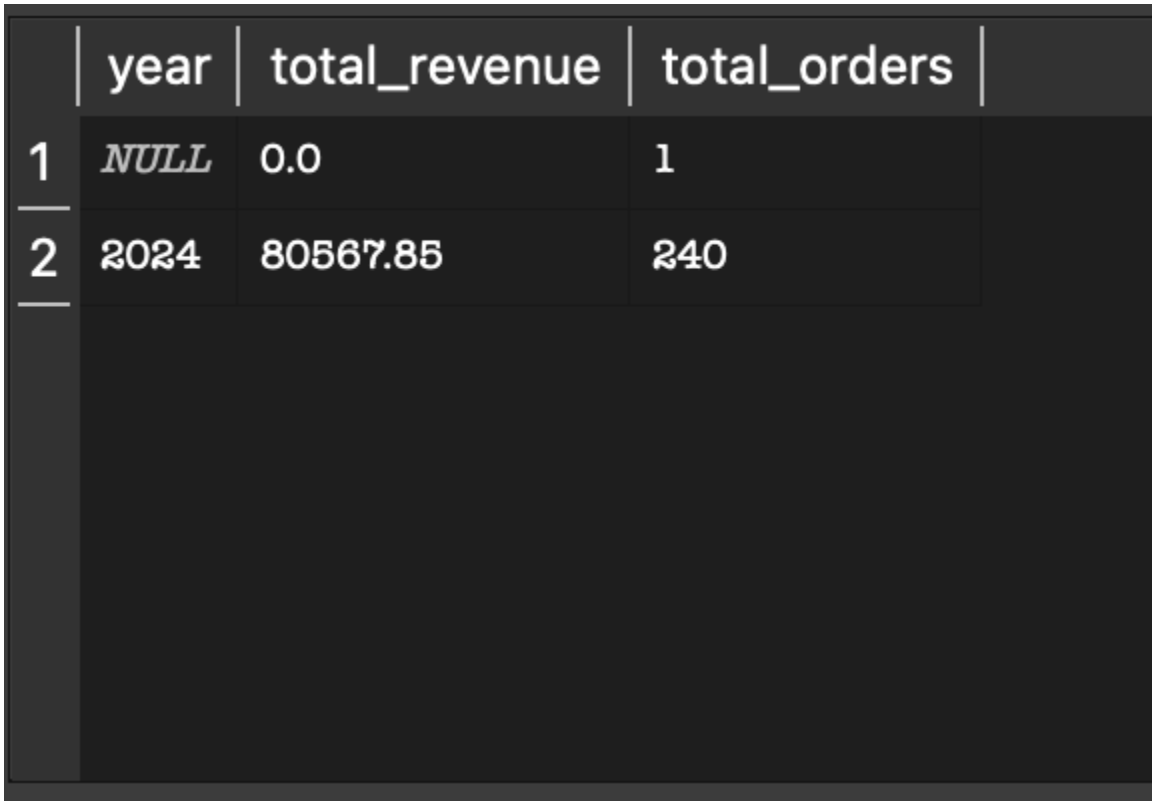
```
SELECT
  STRFTIME('%Y-%m', field2) AS month,
  SUM(field7) AS total_revenue,
  COUNT(DISTINCT field1) AS order_volume
FROM "Online Sales Data"
GROUP BY month
ORDER BY month;
```

	month	total_revenue	order_volume
1	*NULL*	0.0	1
2	2024-01	14548.32	31
3	2024-02	10803.37	29
4	2024-03	12849.24	31
5	2024-04	12451.69	30
6	2024-05	8455.49	31
7	2024-06	7384.55	30

2. Yearly Revenue Summary

```
SELECT
  STRFTIME('%Y', field2) AS year,
```

```
  SUM(field7) AS total_revenue,
  COUNT(DISTINCT field1) AS total_orders
FROM "Online Sales Data"
GROUP BY year
ORDER BY year;
```



	year	total_revenue	total_orders
1	*NULL*	0.0	1
2	2024	80567.85	240

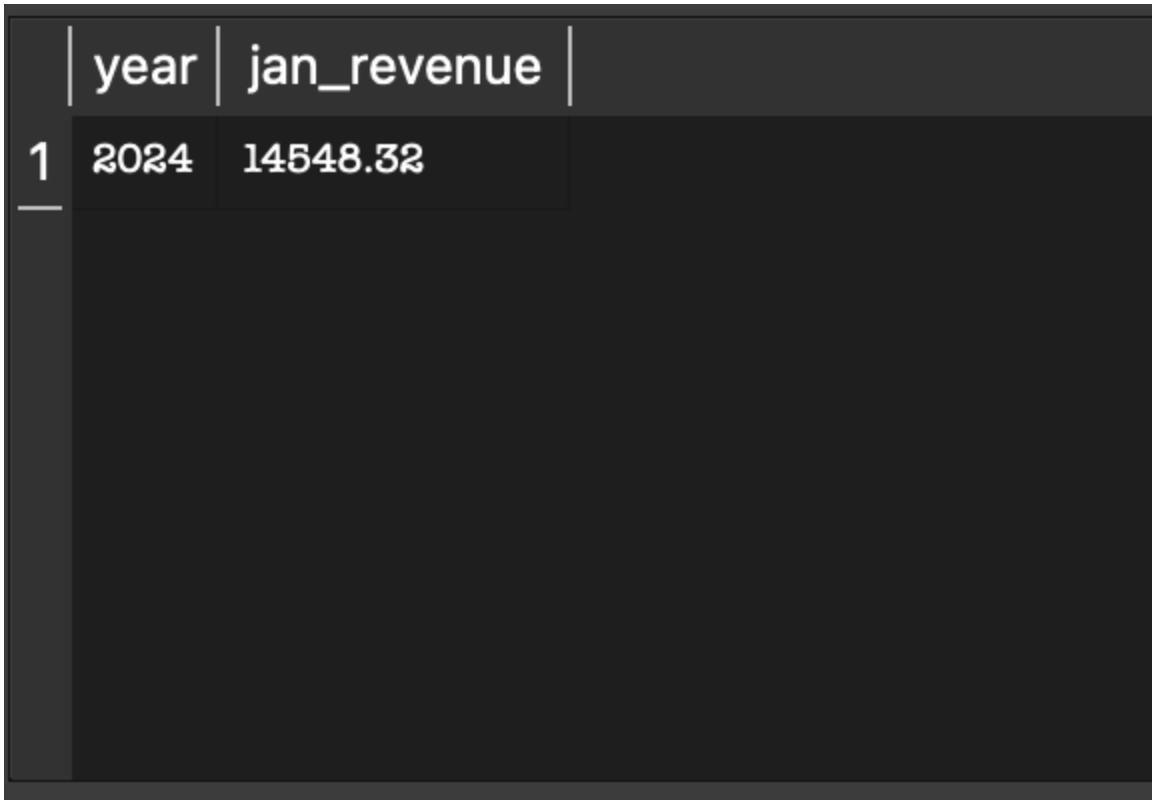
3. Revenue and Orders by Product Category (All Time)

```
SELECT
  field3 AS category,
  SUM(field7) AS revenue,
  COUNT(DISTINCT field1) AS orders
FROM "Online Sales Data"
GROUP BY field3
ORDER BY revenue DESC;
```

	category	revenue	orders
1	Electronics	34982.41	40
2	Home Appliances	18646.16	40
3	Sports	14326.52	40
4	Clothing	8128.93	40
5	Beauty Products	2621.9	40
6	Books	1861.93	40
7	Product Category	0.0	1

4. Revenue Trend for January Only (All Years)

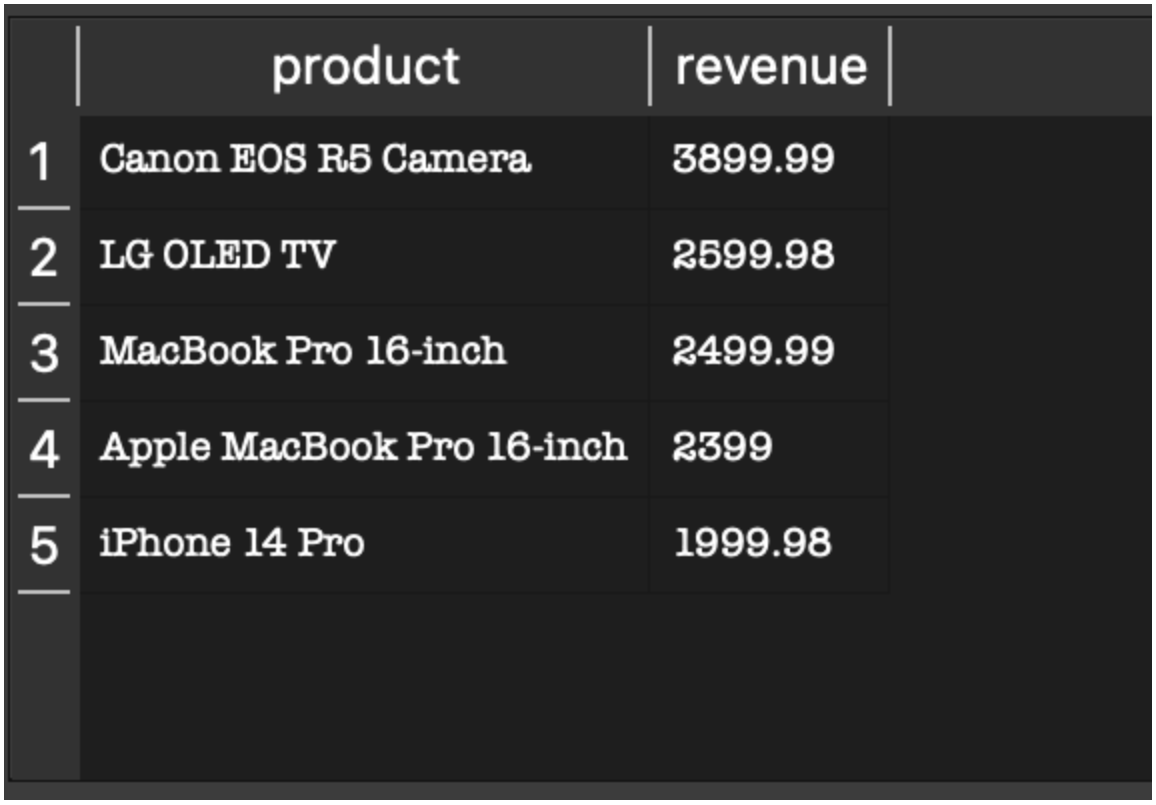
```
SELECT
  STRFTIME('%Y', field2) AS year,
  SUM(field7) AS jan_revenue
FROM "Online Sales Data"
WHERE STRFTIME('%m', field2) = '01'
GROUP BY year
ORDER BY year;
```



	year	jan_revenue
1	2024	14548.32

5. Top 5 Selling Products by Revenue

```
SELECT
  field4 AS product,
  SUM(field7) AS revenue
FROM "Online Sales Data"
GROUP BY field4
ORDER BY revenue DESC
LIMIT 5;
```



	product	revenue
1	Canon EOS R5 Camera	3899.99
2	LG OLED TV	2599.98
3	MacBook Pro 16-inch	2499.99
4	Apple MacBook Pro 16-inch	2399
5	iPhone 14 Pro	1999.98

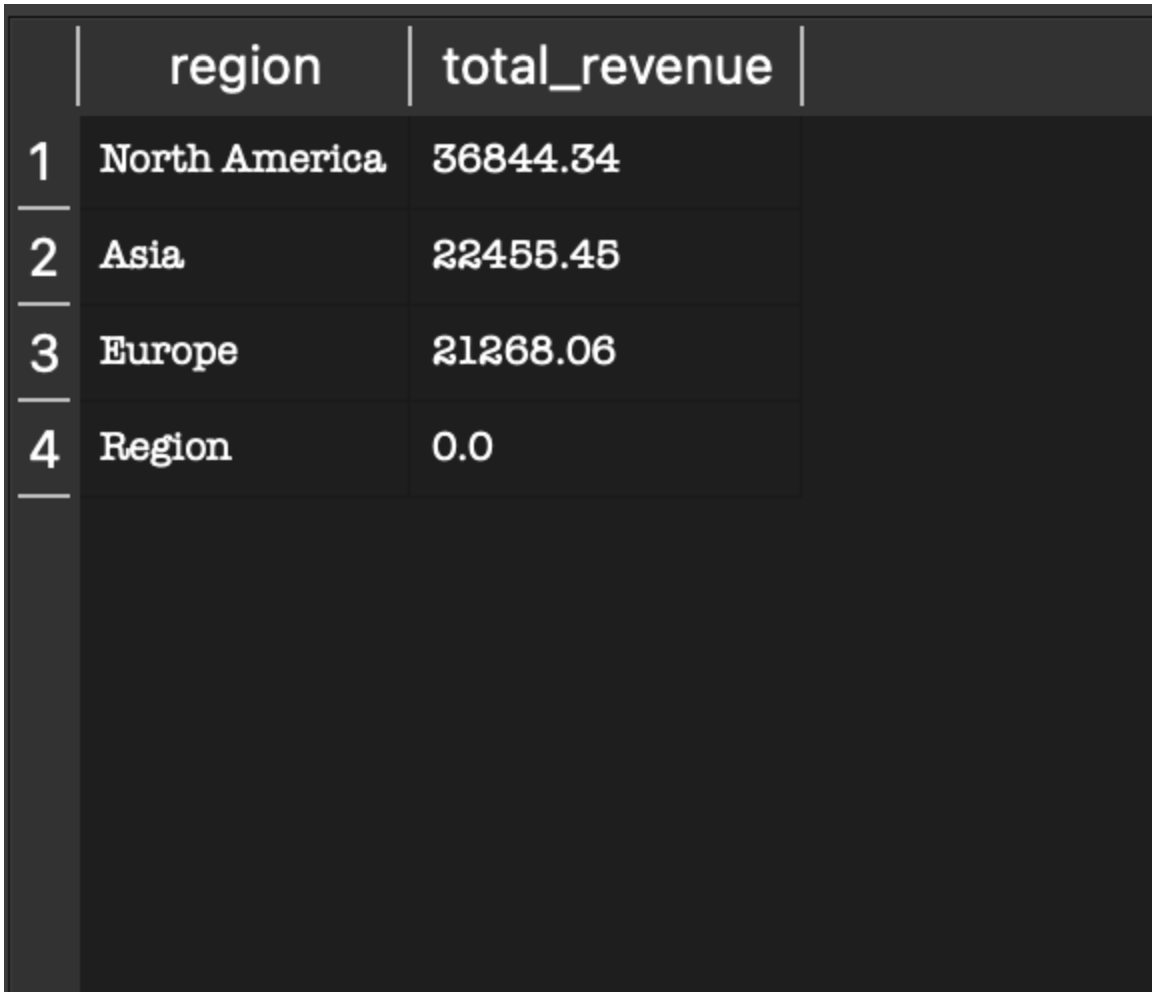
6. Monthly Average Revenue per Order

```
SELECT
  STRFTIME('%Y-%m', field2) AS month,
  SUM(field7) / COUNT(DISTINCT field1) AS avg_order_value
FROM "Online Sales Data"
GROUP BY month
ORDER BY month;
```

	month	avg_order_value
1	*NULL*	0.0
2	2024-01	469.30064516129
3	2024-02	372.53
4	2024-03	414.491612903226
5	2024-04	415.056333333333
6	2024-05	272.757741935484
7	2024-06	246.151666666667
8	2024-07	219.26064516129
9	2024-08	269.55962962963

7. Region-wise Revenue Distribution

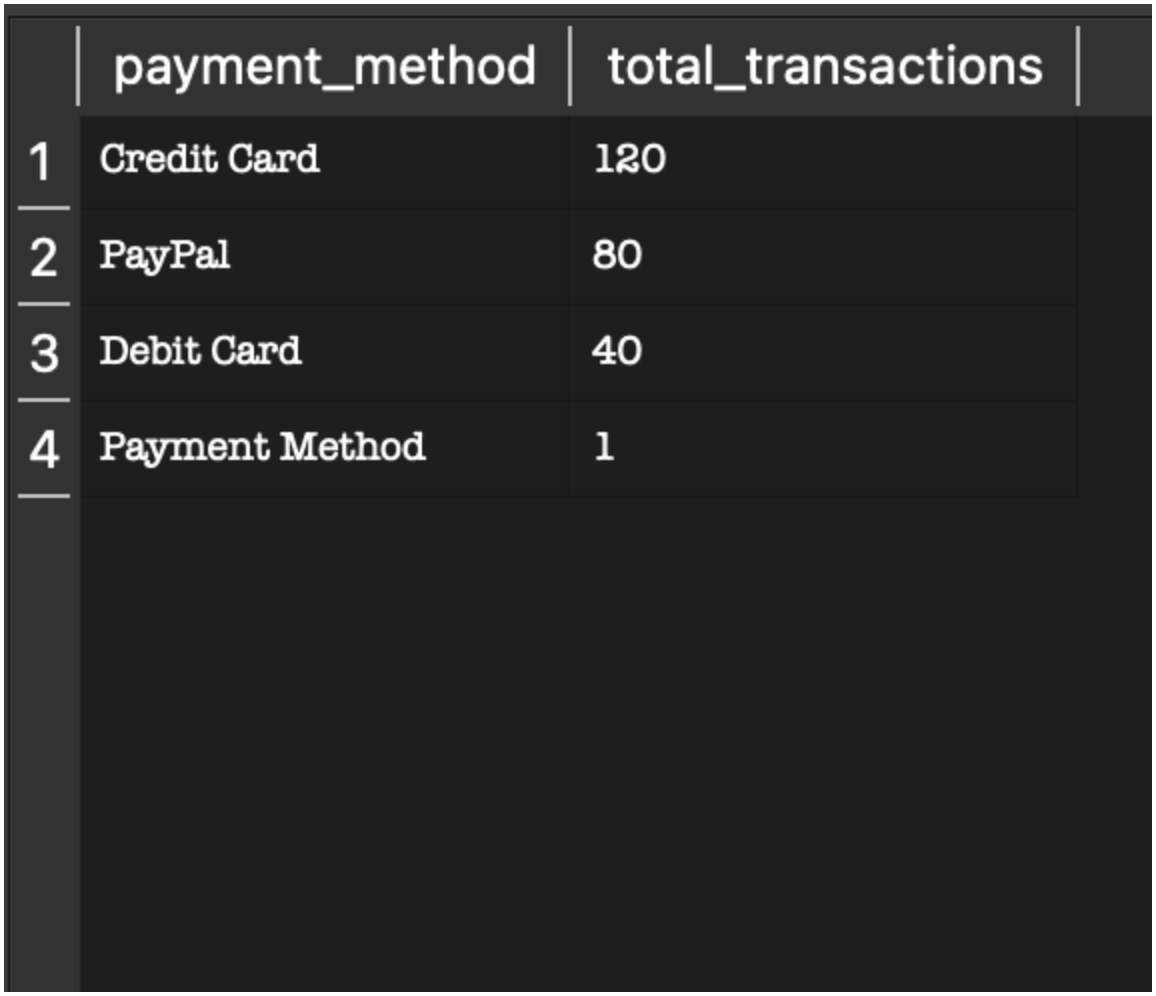
```
SELECT
  field8 AS region,
  SUM(field7) AS total_revenue
FROM "Online Sales Data"
GROUP BY field8
ORDER BY total_revenue DESC;
```



	region	total_revenue
1	North America	36844.34
2	Asia	22455.45
3	Europe	21268.06
4	Region	0.0

8. Payment Method Usage Count

```
SELECT
  field9 AS payment_method,
  COUNT(*) AS total_transactions
FROM "Online Sales Data"
GROUP BY field9
ORDER BY total_transactions DESC;
```



	payment_method	total_transactions
1	Credit Card	120
2	PayPal	80
3	Debit Card	40
4	Payment Method	1

9. Most Sold Product by Units

```
SELECT
  field4 AS product,
  SUM(field5) AS total_units_sold
FROM "Online Sales Data"
GROUP BY field4
ORDER BY total_units_sold DESC
LIMIT 1;
```

	product	total_units_sold
1	Hanes ComfortSoft T-Shirt	10

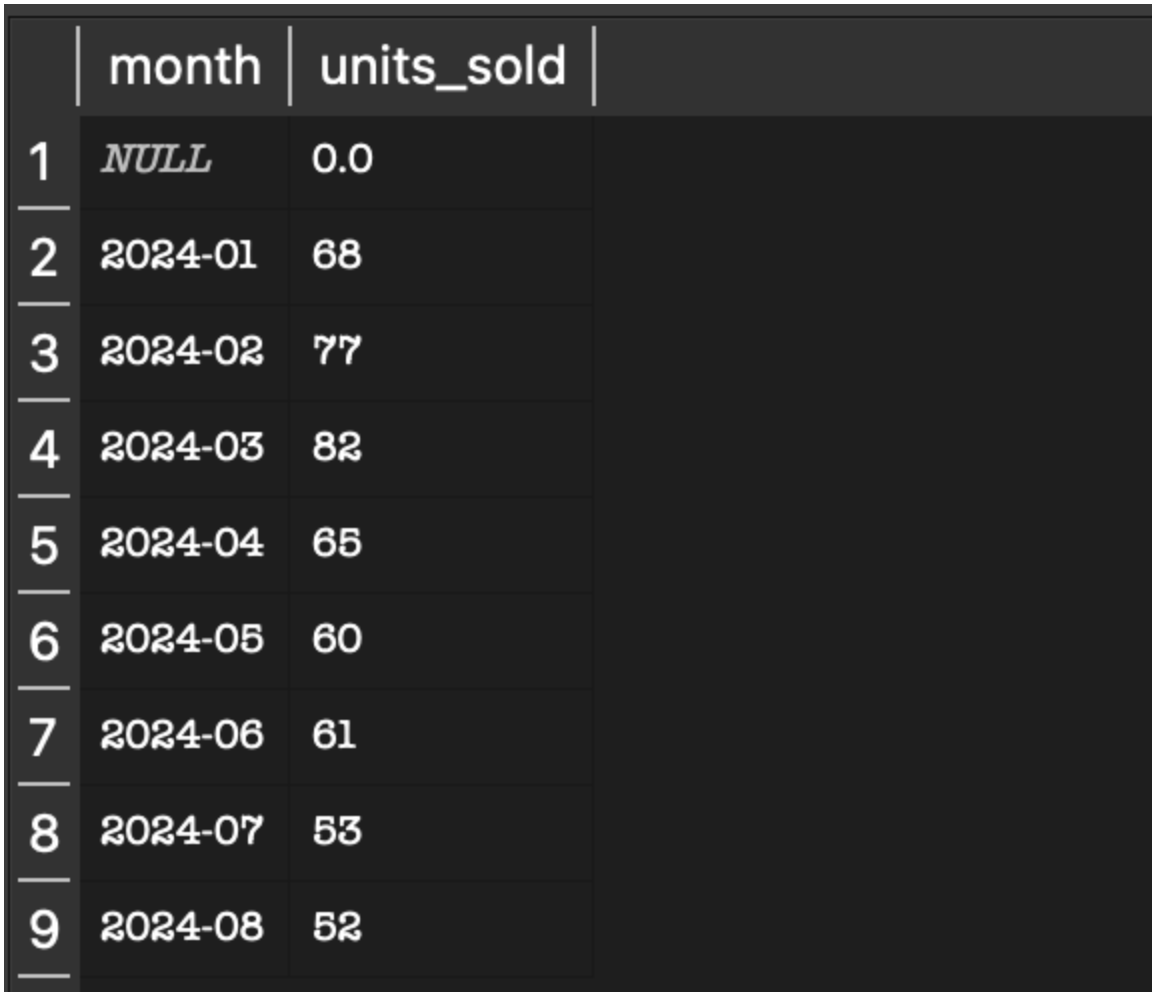
10. Daily Revenue (Top 7 Days)

```
SELECT
  field2 AS date,
  SUM(field7) AS revenue
FROM "Online Sales Data"
GROUP BY field2
ORDER BY revenue DESC
LIMIT 7;
```

	date	revenue
1	2024-04-12	3899.99
2	2024-03-26	2599.98
3	2024-01-07	2499.99
4	2024-08-04	2399
5	2024-01-01	1999.98
6	2024-02-05	1895
7	2024-03-13	1599.99

11. Monthly Units Sold

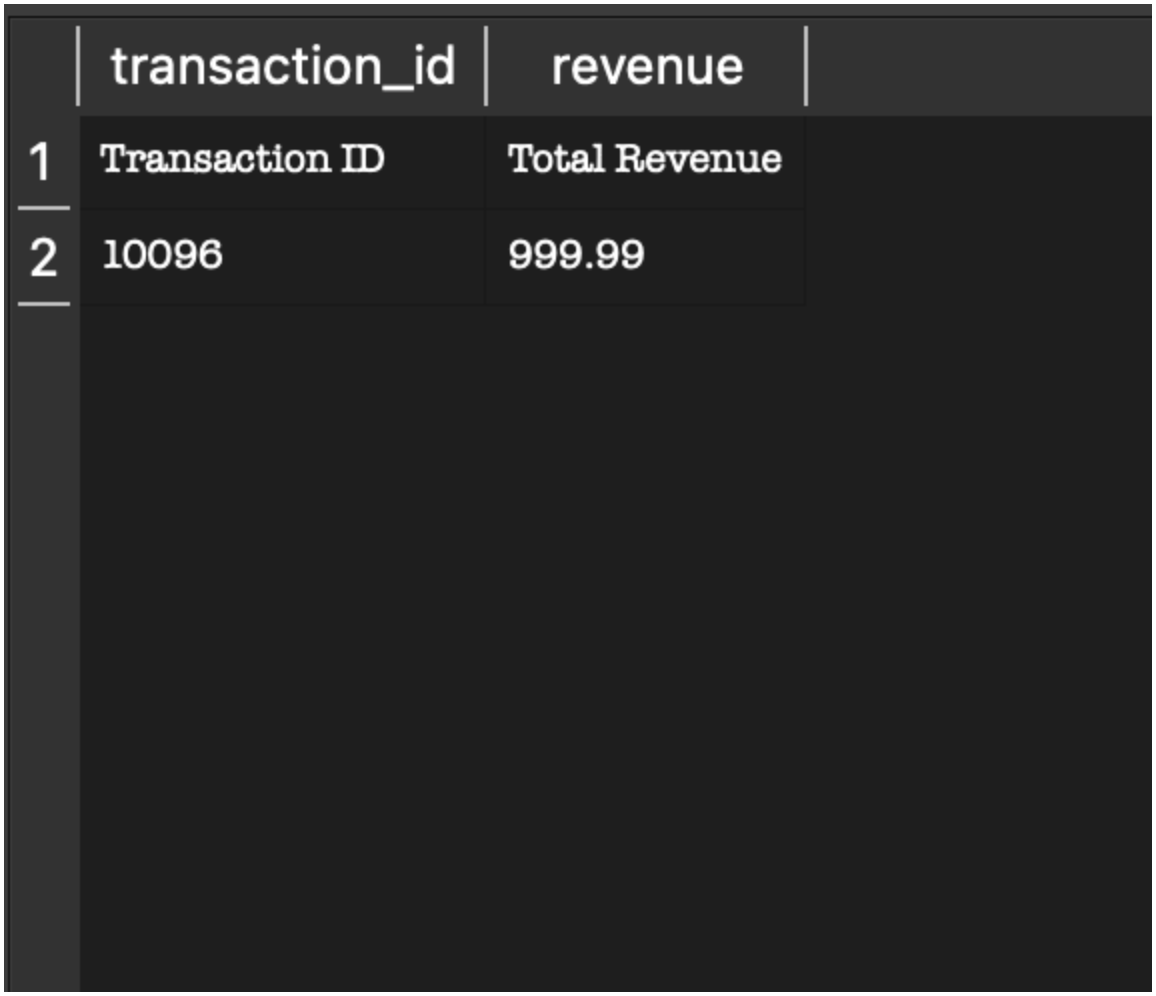
```
SELECT
  STRFTIME('%Y-%m', field2) AS month,
  SUM(field5) AS units_sold
FROM "Online Sales Data"
GROUP BY month
ORDER BY month;
```



	month	units_sold
1	*NULL*	0.0
2	2024-01	68
3	2024-02	77
4	2024-03	82
5	2024-04	65
6	2024-05	60
7	2024-06	61
8	2024-07	53
9	2024-08	52

12. Highest Revenue Transaction

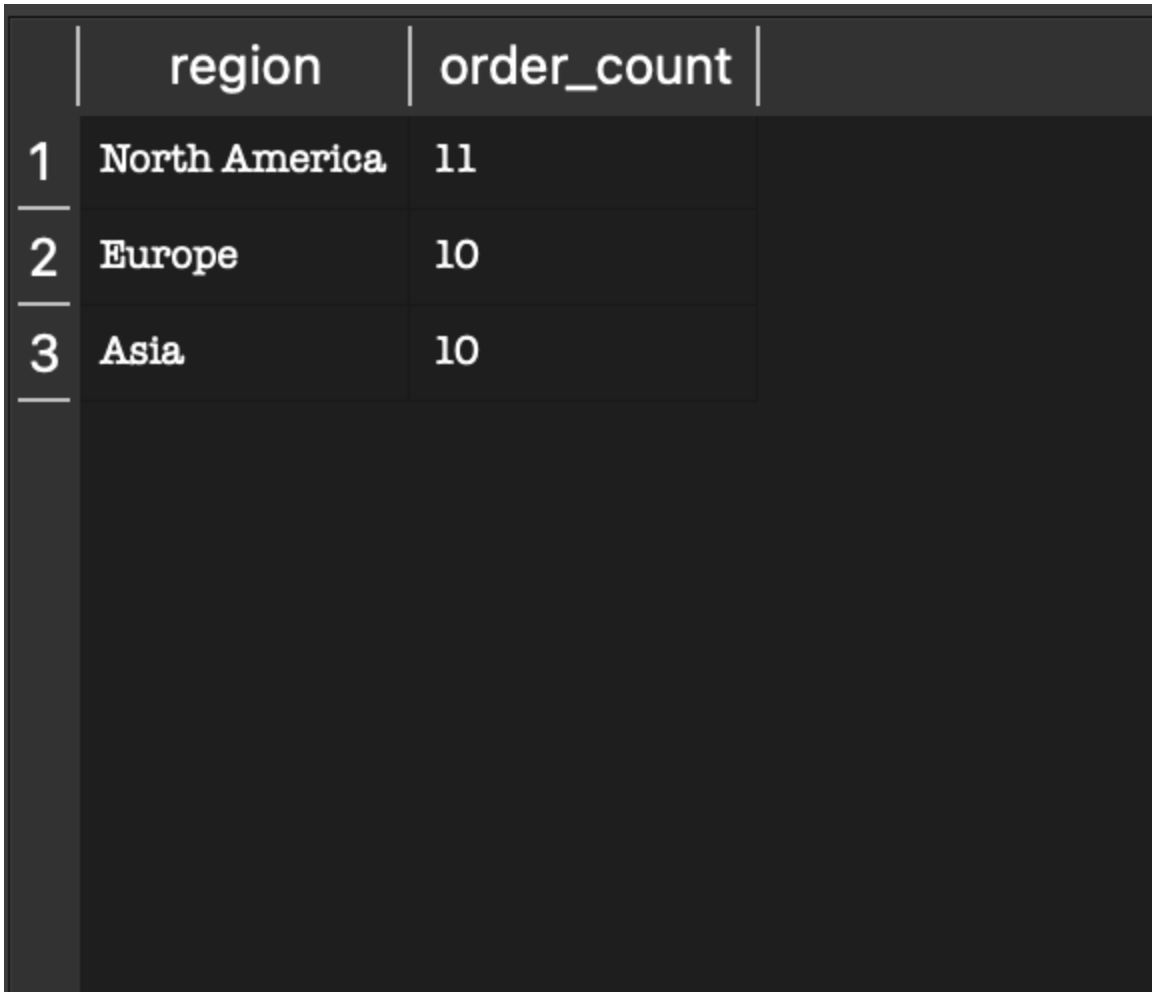
```
SELECT
  field1 AS transaction_id,
  field7 AS revenue
FROM "Online Sales Data"
ORDER BY revenue DESC
LIMIT 1;
```



	transaction_id	revenue
1	Transaction ID	Total Revenue
2	10096	999.99

13. Top 3 Regions by Order Count in Jan 2024

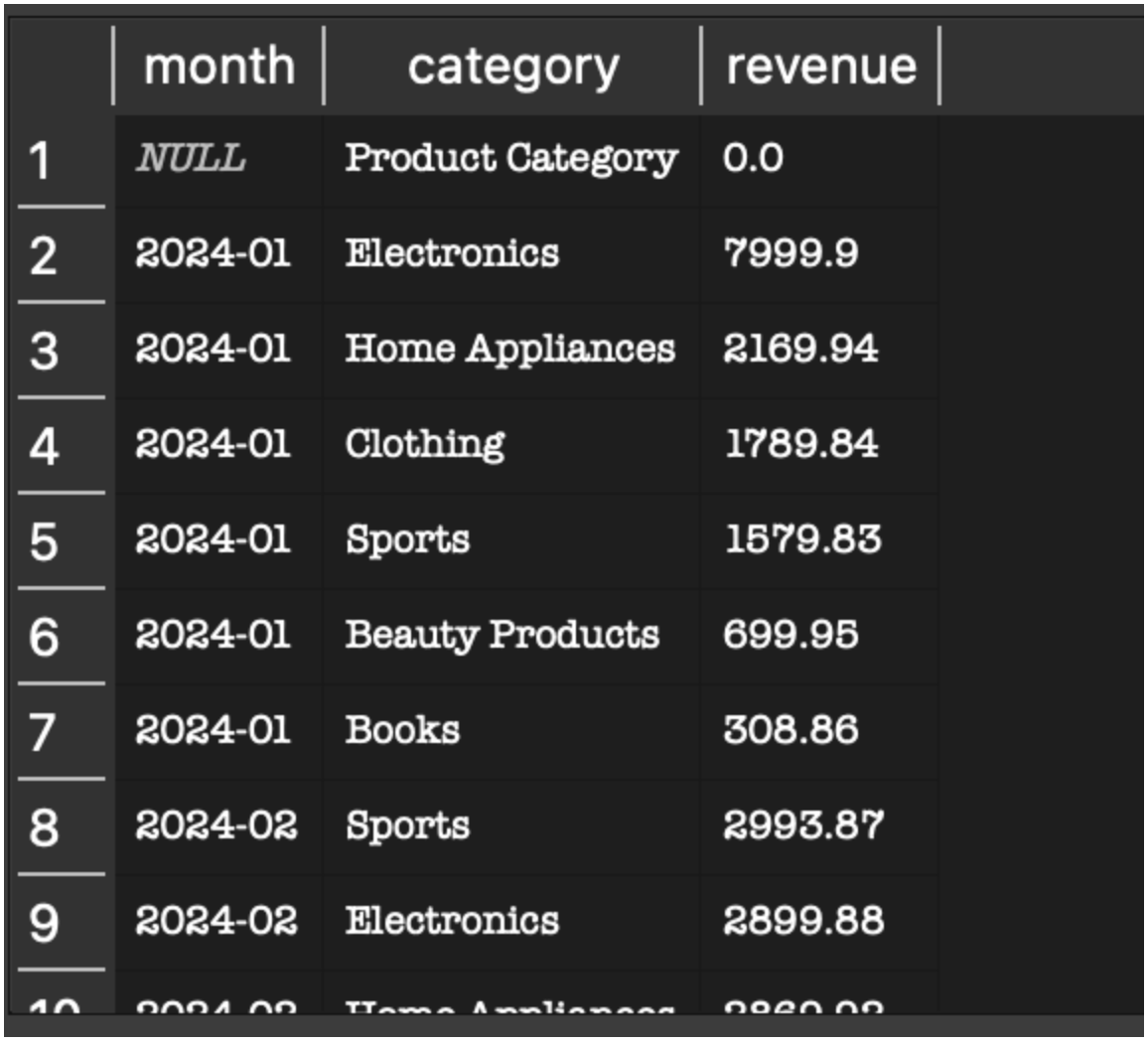
```
SELECT
  field8 AS region,
  COUNT(field1) AS order_count
FROM "Online Sales Data"
WHERE STRFTIME('%Y-%m', field2) = '2024-01'
GROUP BY region
ORDER BY order_count DESC
LIMIT 3;
```



	region	order_count
1	North America	11
2	Europe	10
3	Asia	10

14. Product Category Trend (Monthly Revenue)

```
SELECT
  STRFTIME('%Y-%m', field2) AS month,
  field3 AS category,
  SUM(field7) AS revenue
FROM "Online Sales Data"
GROUP BY month, category
ORDER BY month, revenue DESC;
```



	month	category	revenue
1	*NULL*	Product Category	0.0
2	2024-01	Electronics	7999.9
3	2024-01	Home Appliances	2169.94
4	2024-01	Clothing	1789.84
5	2024-01	Sports	1579.83
6	2024-01	Beauty Products	699.95
7	2024-01	Books	308.86
8	2024-02	Sports	2993.87
9	2024-02	Electronics	2899.88

15. Monthly Revenue from Electronics Only

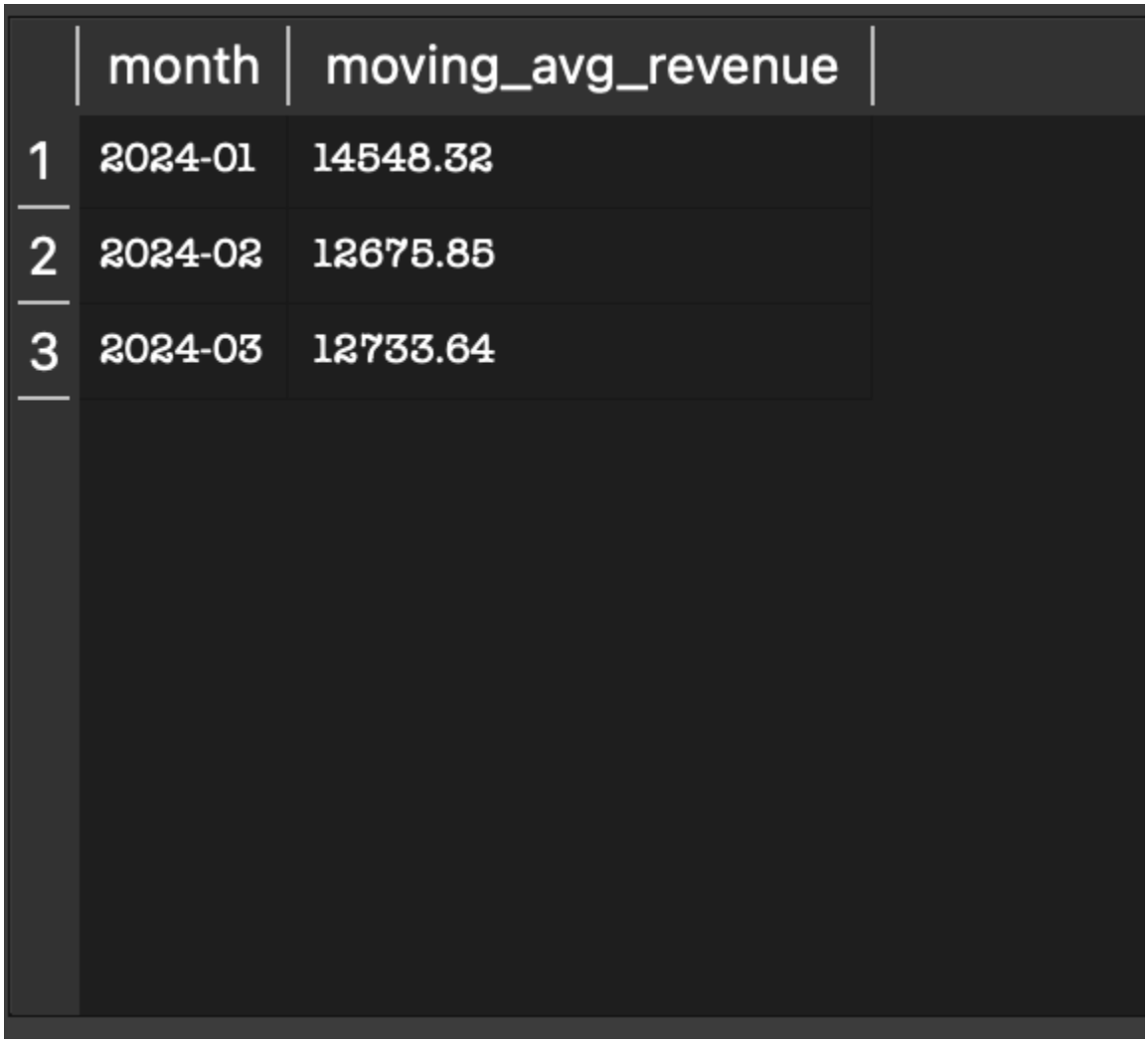
```
SELECT
  STRFTIME('%Y-%m', field2) AS month,
  SUM(field7) AS electronics_revenue
FROM "Online Sales Data"
WHERE field3 = 'Electronics'
GROUP BY month
ORDER BY month;
```

	month	electronics_revenue
1	2024-01	7999.9
2	2024-02	2899.88
3	2024-03	4499.9
4	2024-04	6709.91
5	2024-05	4198.96
6	2024-06	3599.93
7	2024-07	2006.97
8	2024-08	3066.96

16.3-Month Moving Average of Revenue (per Month)

```
WITH monthly_revenue AS (
  SELECT
    STRFTIME('%Y-%m', field2) AS month,
    SUM(field7) AS revenue
  FROM "Online Sales Data"
  GROUP BY month
),
moving_avg AS (
  SELECT
```

```
    m1.month,
    ROUND(AVG(m2.revenue), 2) AS moving_avg_revenue
  FROM monthly_revenue m1
  JOIN monthly_revenue m2
    ON m2.month <= m1.month
  GROUP BY m1.month
  HAVING COUNT(*) <= 3
)
SELECT * FROM moving_avg;
```



	month	moving_avg_revenue
1	2024-01	14548.32
2	2024-02	12675.85
3	2024-03	12733.64

17.Month-over-Month Growth %

```
WITH monthly AS (
  SELECT
    STRFTIME('%Y-%m', field2) AS month,
    SUM(field7) AS revenue
  FROM "Online Sales Data"
  GROUP BY month
),
growth AS (
  SELECT
    month,
    revenue,
    LAG(revenue) OVER (ORDER BY month) AS prev_month_revenue
  FROM monthly
)
SELECT
  month,
  revenue,
  prev_month_revenue,
  ROUND(((revenue - prev_month_revenue) * 100.0) /
prev_month_revenue, 2) AS growth_percent
FROM growth
WHERE prev_month_revenue IS NOT NULL;
```

	month	revenue	prev_month_revenue	growth_percent
1	2024-01	14548.32	0.0	*NULL*
2	2024-02	10803.37	14548.32	-25.74
3	2024-03	12849.24	10803.37	18.94
4	2024-04	12451.69	12849.24	-3.09
5	2024-05	8455.49	12451.69	-32.09
6	2024-06	7384.55	8455.49	-12.67
7	2024-07	6797.08	7384.55	-7.96
8	2024-08	7278.11	6797.08	7.08

18.Running Total of Revenue Over Time

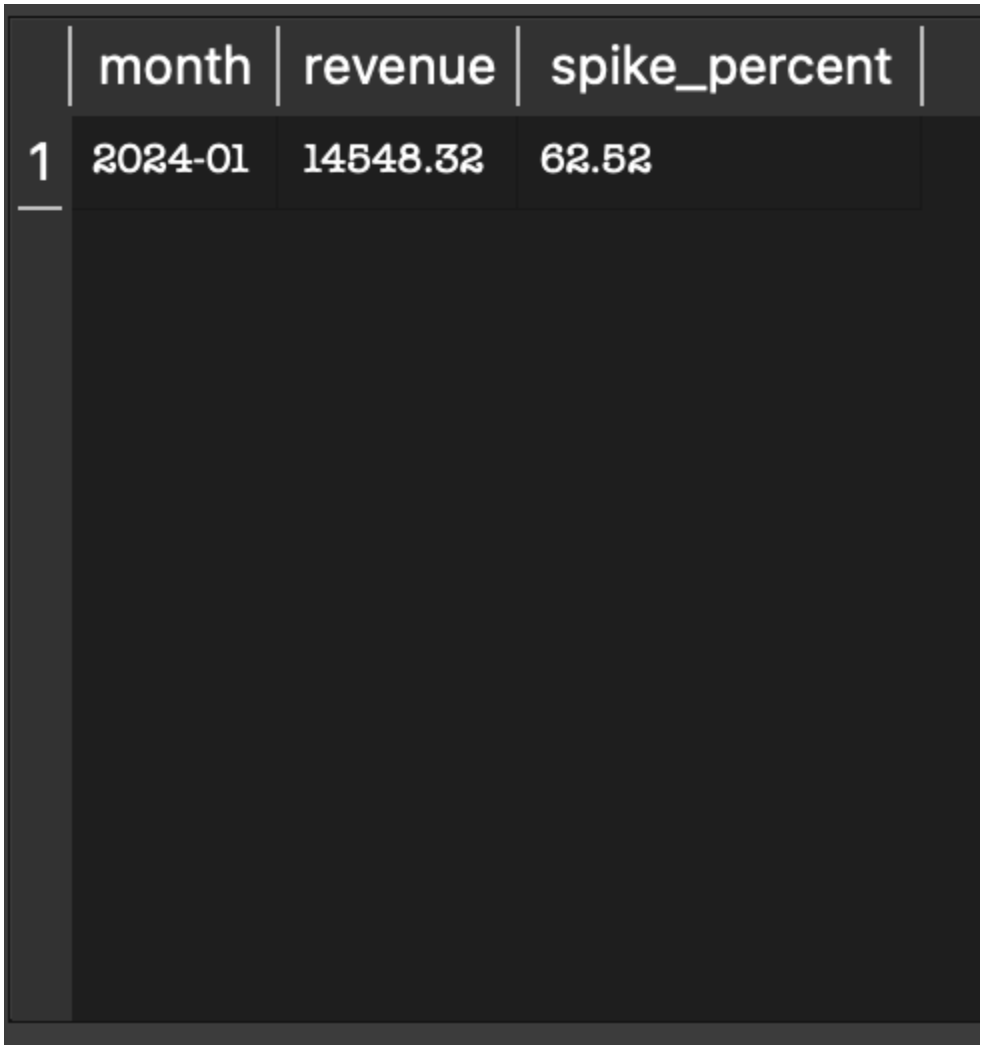
```
WITH monthly AS (
  SELECT
    STRFTIME('%Y-%m', field2) AS month,
    SUM(field7) AS revenue
  FROM "Online Sales Data"
  GROUP BY month
)
SELECT
  month,
  revenue,
  SUM(revenue) OVER (ORDER BY month ROWS BETWEEN
UNBOUNDED PRECEDING AND CURRENT ROW) AS running_total
FROM monthly;
```

	month	revenue	running_total
1	*NULL*	0.0	0.0
2	2024-01	14548.32	14548.32
3	2024-02	10803.37	25351.69
4	2024-03	12849.24	38200.93
5	2024-04	12451.69	50652.62
6	2024-05	8455.49	59108.11
7	2024-06	7384.55	66492.66
8	2024-07	6797.08	73289.74
9	2024-08	7278.11	80567.85

19.Revenue Spike Detection (Above Average by >50%)

```
WITH monthly AS (
  SELECT
    STRFTIME('%Y-%m', field2) AS month,
    SUM(field7) AS revenue
  FROM "Online Sales Data"
  GROUP BY month
),
avg_rev AS (
  SELECT AVG(revenue) AS avg_revenue FROM monthly
)
```

```
SELECT
  m.month,
  m.revenue,
  ROUND((m.revenue - a.avg_revenue) * 100.0 / a.avg_revenue, 2) AS
spike_percent
FROM monthly m, avg_rev a
WHERE m.revenue > 1.5 * a.avg_revenue;
```



	month	revenue	spike_percent
1	2024-01	14548.32	62.52

20.Top Products Whose Revenue Grew Month-over-Month

```
WITH monthly_product AS (
  SELECT
```

```
    STRFTIME('%Y-%m', field2) AS month,
    field4 AS product,
    SUM(field7) AS revenue
  FROM "Online Sales Data"
  GROUP BY month, product
),
with_lag AS (
  SELECT
    month,
    product,
    revenue,
    LAG(revenue) OVER (PARTITION BY product ORDER BY month)
AS prev_revenue
  FROM monthly_product
)
SELECT
  month,
  product,
  revenue,
  prev_revenue,
  ROUND(((revenue - prev_revenue) * 100.0) / prev_revenue, 2) AS
growth_percent
FROM with_lag
WHERE prev_revenue IS NOT NULL AND growth_percent > 0
ORDER BY growth_percent DESC
LIMIT 10;
```

	month	product	revenue	prev_revenue	growth_percent
1	2024-05	Dyson Supersonic Hair Dryer	799.98	399.99	100.0
2	2024-07	Keurig K-Elite Coffee Maker	339.98	189.99	78.95
3	2024-08	The Girl with the Dragon Tattoo by Stieg Larsson	32.97	19.98	65.02
4	2024-08	The Silent Patient by Alex Michaelides	80.97	53.98	50.0