Time-Series Sales Analysis

❖ **Used the Olist Brazilian E-Commerce Dataset (public dataset from Kaggle).**
It consists of 9 interrelated CSV files:

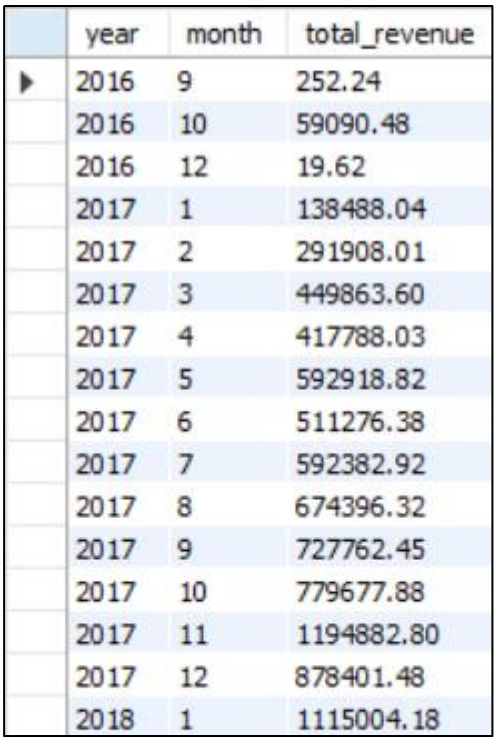
1. olist_customers_dataset.csv → Customer information (city, state, zip).
2. olist_orders_dataset.csv → Order details (timestamps, status).
3. olist_products_dataset.csv → Product metadata (category, size, weight).
4. olist_sellers_dataset.csv → Seller information (location, ID).
5. olist_order_items_dataset.csv → Items included in each order (price, shipping).
6. olist_order_payments_dataset.csv → Payment methods and payment value.
7. olist_order_reviews_dataset.csv → Customer reviews (rating, comment, timestamp).
8. olist_geolocation_dataset.csv → Geolocation data (city, state, lat/long).
9. product_category_name_translation.csv → Portuguese–English mapping of categories.

❖ **Datasets Used for This Analysis**
For time-series revenue and order volume analysis, we specifically use:

- orders table → Provides the order_purchase_timestamp (date/time of order).
- order_payments table → Provides the payment_value (revenue amount).

1. Monthly Revenue

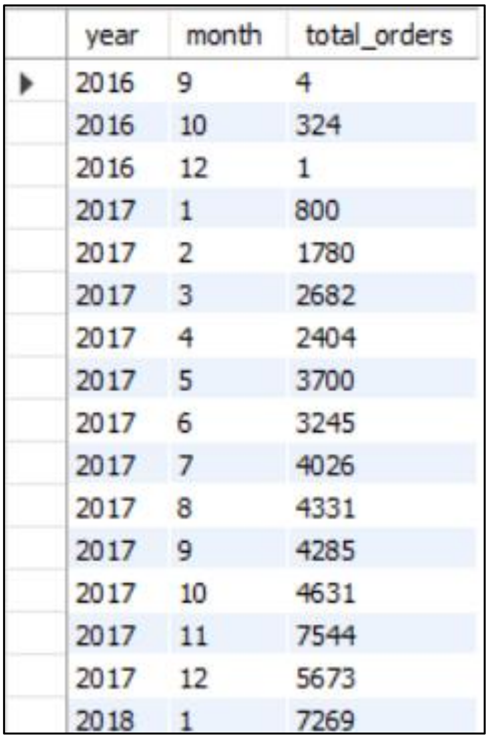
```
SELECT
   EXTRACT(YEAR FROM o.order_purchase_timestamp) AS year,
   EXTRACT(MONTH FROM o.order_purchase_timestamp) AS month,
   SUM(p.payment_value) AS total_revenue
FROM orders o
JOIN order_payments p ON o.order_id = p.order_id
GROUP BY year, month
ORDER BY year, month;
```



year	month	total_revenue
2016	9	252.24
2016	10	59090.48
2016	12	19.62
2017	1	138488.04
2017	2	291908.01
2017	3	449863.60
2017	4	417788.03
2017	5	592918.82
2017	6	511276.38
2017	7	592382.92
2017	8	674396.32
2017	9	727762.45
2017	10	779677.88
2017	11	1194882.80
2017	12	878401.48
2018	1	1115004.18

2. Monthly Order Volume

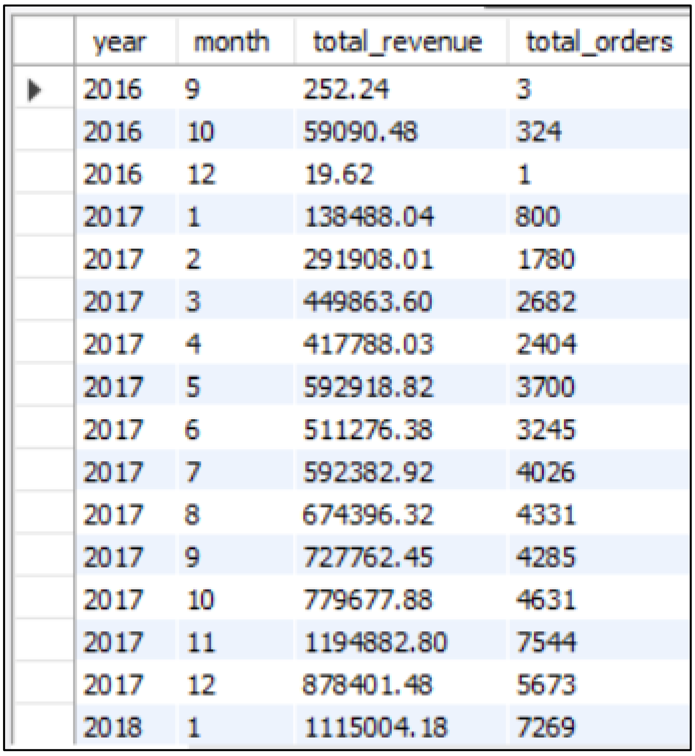
```
SELECT
   EXTRACT(YEAR FROM order_purchase_timestamp) AS year,
   EXTRACT(MONTH FROM order_purchase_timestamp) AS month,
   COUNT(DISTINCT order_id) AS total_orders
FROM orders
GROUP BY year, month
ORDER BY year, month;
```



year	month	total_orders
2016	9	4
2016	10	324
2016	12	1
2017	1	800
2017	2	1780
2017	3	2682
2017	4	2404
2017	5	3700
2017	6	3245
2017	7	4026
2017	8	4331
2017	9	4285
2017	10	4631
2017	11	7544
2017	12	5673
2018	1	7269

3. Monthly Revenue & Order Volume Together

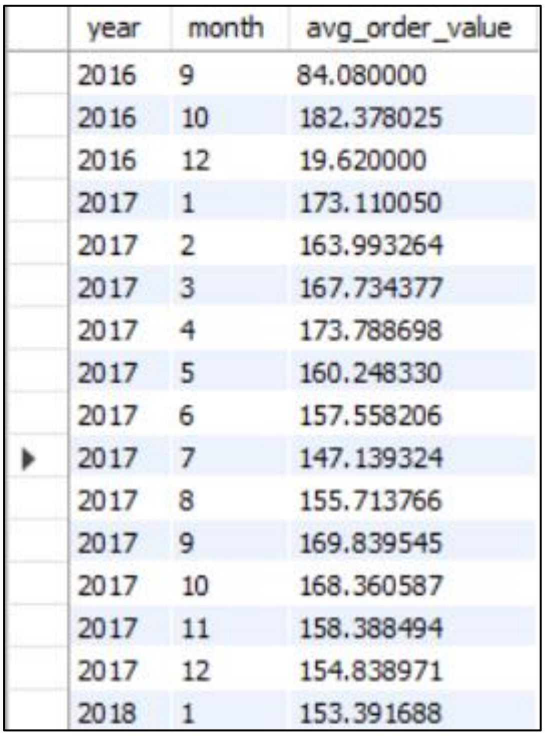
```
SELECT
   EXTRACT(YEAR FROM o.order_purchase_timestamp) AS year,
   EXTRACT(MONTH FROM o.order_purchase_timestamp) AS month,
   SUM(p.payment_value) AS total_revenue,
   COUNT(DISTINCT o.order_id) AS total_orders
FROM orders o
JOIN order_payments p ON o.order_id = p.order_id
GROUP BY year, month
ORDER BY year, month;
```



year	month	total_revenue	total_orders
2016	9	252.24	3
2016	10	59090.48	324
2016	12	19.62	1
2017	1	138488.04	800
2017	2	291908.01	1780
2017	3	449863.60	2682
2017	4	417788.03	2404
2017	5	592918.82	3700
2017	6	511276.38	3245
2017	7	592382.92	4026
2017	8	674396.32	4331
2017	9	727762.45	4285
2017	10	779677.88	4631
2017	11	1194882.80	7544
2017	12	878401.48	5673
2018	1	1115004.18	7269

4. Monthly Average Order Value (AOV)

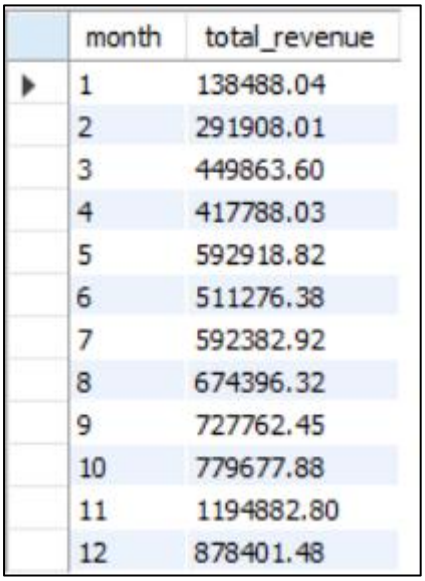
```
SELECT
    EXTRACT(YEAR FROM o.order_purchase_timestamp) AS year,
    EXTRACT(MONTH FROM o.order_purchase_timestamp) AS month,
    SUM(p.payment_value) / COUNT(DISTINCT o.order_id) AS avg_order_value
FROM orders o
JOIN order_payments p ON o.order_id = p.order_id
GROUP BY year, month
ORDER BY year, month;
```



year	month	avg_order_value
2016	9	84.080000
2016	10	182.378025
2016	12	19.620000
2017	1	173.110050
2017	2	163.993264
2017	3	167.734377
2017	4	173.788698
2017	5	160.248330
2017	6	157.558206
2017	7	147.139324
2017	8	155.713766
2017	9	169.839545
2017	10	168.360587
2017	11	158.388494
2017	12	154.838971
2018	1	153.391688

5. Revenue Trend for 2017 Only

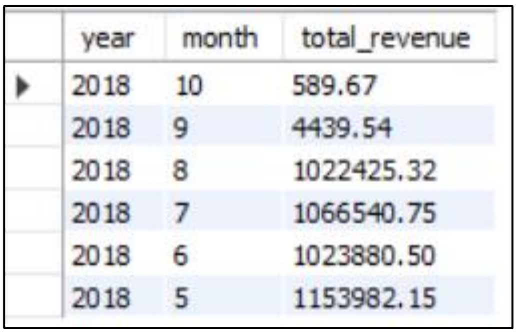
```
SELECT
   EXTRACT(MONTH FROM o.order_purchase_timestamp) AS month,
   SUM(p.payment_value) AS total_revenue
FROM orders o
JOIN order_payments p ON o.order_id = p.order_id
WHERE EXTRACT(YEAR FROM o.order_purchase_timestamp) = 2017
GROUP BY month
ORDER BY month;
```



month	total_revenue
1	138488.04
2	291908.01
3	449863.60
4	417788.03
5	592918.82
6	511276.38
7	592382.92
8	674396.32
9	727762.45
10	779677.88
11	1194882.80
12	878401.48

6. Last 6 Months Revenue

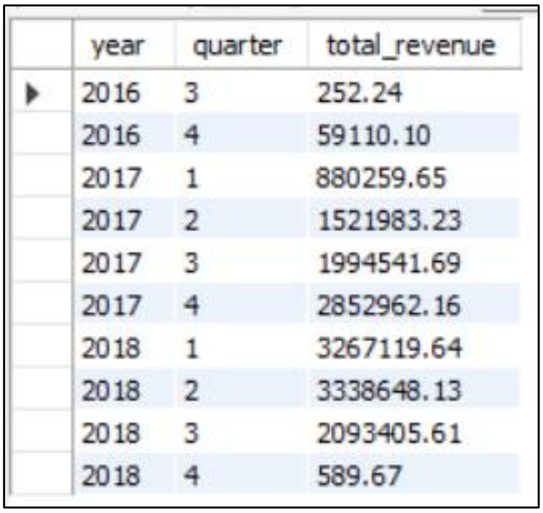
```
SELECT
   EXTRACT(YEAR FROM o.order_purchase_timestamp) AS year,
   EXTRACT(MONTH FROM o.order_purchase_timestamp) AS month,
   SUM(p.payment_value) AS total_revenue
FROM orders o
JOIN order_payments p ON o.order_id = p.order_id
GROUP BY year, month
ORDER BY year DESC, month DESC
LIMIT 6;
```



year	month	total_revenue
2018	10	589.67
2018	9	4439.54
2018	8	1022425.32
2018	7	1066540.75
2018	6	1023880.50
2018	5	1153982.15

7. Quarterly Revenue

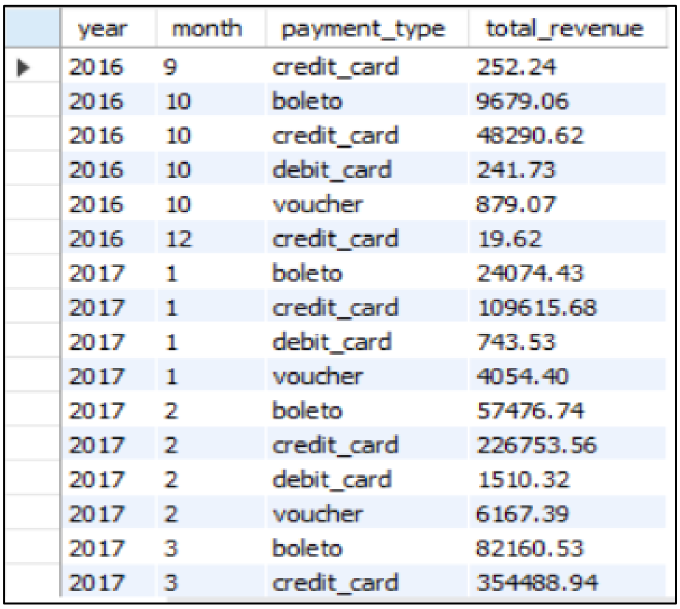
```
SELECT
   EXTRACT(YEAR FROM o.order_purchase_timestamp) AS year,
   QUARTER(o.order_purchase_timestamp) AS quarter,
   SUM(p.payment_value) AS total_revenue
FROM orders o
JOIN order_payments p ON o.order_id = p.order_id
GROUP BY year, quarter
ORDER BY year, quarter;
```



	year	quarter	total_revenue
▸	2016	3	252.24
	2016	4	59110.10
	2017	1	880259.65
	2017	2	1521983.23
	2017	3	1994541.69
	2017	4	2852962.16
	2018	1	3267119.64
	2018	2	3338648.13
	2018	3	2093405.61
	2018	4	589.67

8. Monthly Revenue by Payment Type

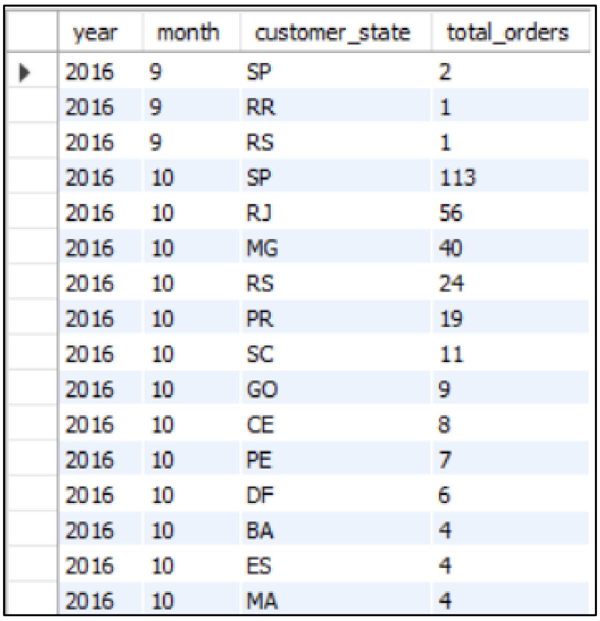
```
SELECT
   EXTRACT(YEAR FROM o.order_purchase_timestamp) AS year,
   EXTRACT(MONTH FROM o.order_purchase_timestamp) AS month,
   p.payment_type,
   SUM(p.payment_value) AS total_revenue
FROM orders o
JOIN order_payments p ON o.order_id = p.order_id
GROUP BY year, month, p.payment_type
ORDER BY year, month;
```



	year	month	payment_type	total_revenue
▸	2016	9	credit_card	252.24
	2016	10	boleto	9679.06
	2016	10	credit_card	48290.62
	2016	10	debit_card	241.73
	2016	10	voucher	879.07
	2016	12	credit_card	19.62
	2017	1	boleto	24074.43
	2017	1	credit_card	109615.68
	2017	1	debit_card	743.53
	2017	1	voucher	4054.40
	2017	2	boleto	57476.74
	2017	2	credit_card	226753.56
	2017	2	debit_card	1510.32
	2017	2	voucher	6167.39
	2017	3	boleto	82160.53
	2017	3	credit_card	354488.94

9. Monthly Orders by Customer State

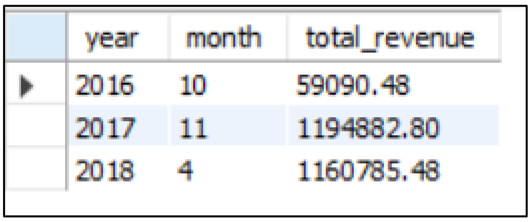
```
SELECT
    EXTRACT(YEAR FROM o.order_purchase_timestamp) AS year,
    EXTRACT(MONTH FROM o.order_purchase_timestamp) AS month,
    c.customer_state,
    COUNT(DISTINCT o.order_id) AS total_orders
FROM orders o
JOIN customers c ON o.customer_id = c.customer_id
GROUP BY year, month, c.customer_state
ORDER BY year, month, total_orders DESC;
```



	year	month	customer_state	total_orders
▶	2016	9	SP	2
	2016	9	RR	1
	2016	9	RS	1
	2016	10	SP	113
	2016	10	RJ	56
	2016	10	MG	40
	2016	10	RS	24
	2016	10	PR	19
	2016	10	SC	11
	2016	10	GO	9
	2016	10	CE	8
	2016	10	PE	7
	2016	10	DF	6
	2016	10	BA	4
	2016	10	ES	4
	2016	10	MA	4

10. Best Month by Revenue Each Year

```
SELECT year, month, total_revenue
FROM (
    SELECT
        EXTRACT(YEAR FROM o.order_purchase_timestamp) AS year,
        EXTRACT(MONTH FROM o.order_purchase_timestamp) AS month,
        SUM(p.payment_value) AS total_revenue,
        RANK() OVER (PARTITION BY EXTRACT(YEAR FROM o.order_purchase_timestamp)
                     ORDER BY SUM(p.payment_value) DESC) AS rn
    FROM orders o
    JOIN order_payments p ON o.order_id = p.order_id
    GROUP BY year, month
) ranked
WHERE rn = 1;
```



	year	month	total_revenue
▶	2016	10	59090.48
	2017	11	1194882.80
	2018	4	1160785.48