E-COMMERCE TRENDS & INSIGHTS WITH SQL

- *A CASE STUDY ON THE OLIST BRAZILIAN DATASET*

By Ahmad Shahzad · July 2025

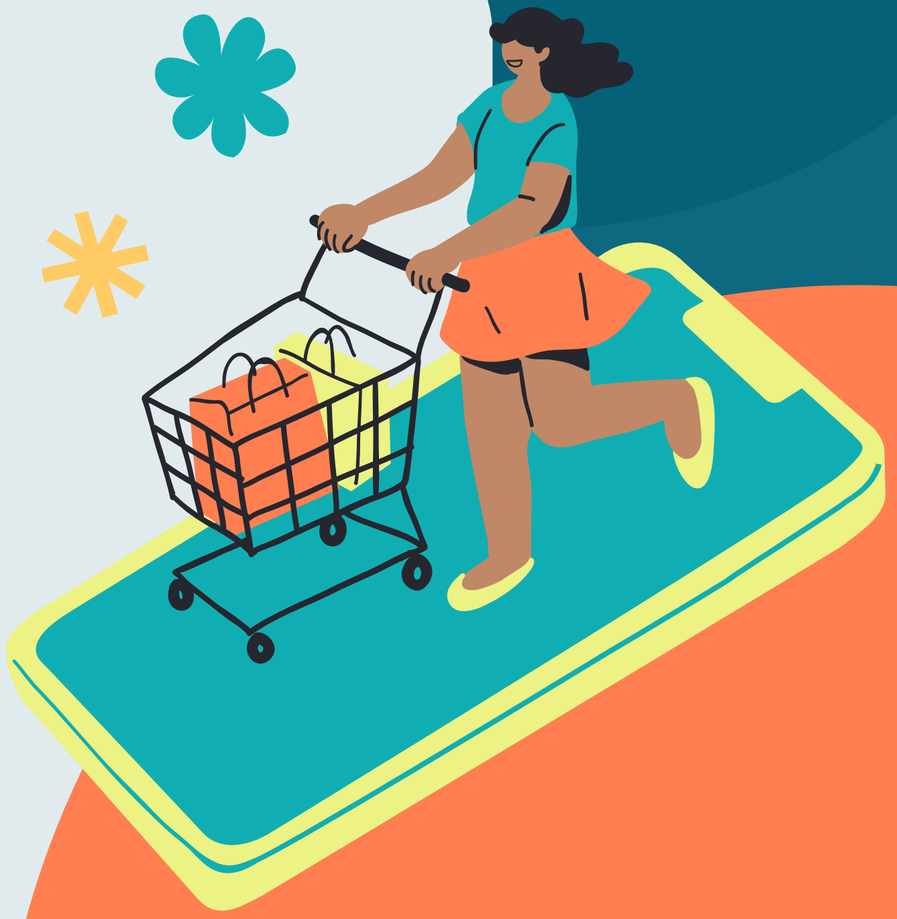
TOOLS & TECH STACK

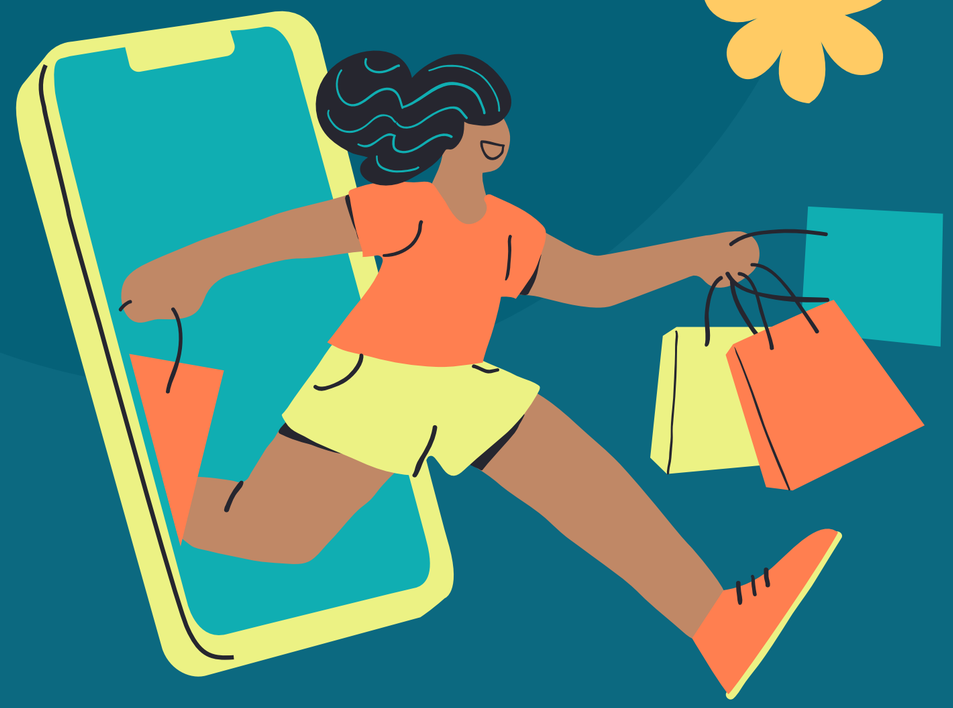
1. MYSQL
2. KAGGLE
3. CANVA
4. EXCEL

- SQL FOR DATA EXPLORATION
- OVER 100K+ ROWS ANALYZED

E-COMMERCE GROWTH OVERVIEW

The e-commerce industry has seen exponential growth over the last decade, with the global market expanding rapidly. As more consumers shift to online shopping, businesses are adapting to meet this rising demand. Understanding this growth is essential for identifying opportunities and challenges in the market





PROJECT OVERVIEW

- **Dataset:** Olist Brazilian E-commerce
- **Objective**: Extract insights and trends that help understand customer behavior and optimize sales/delivery
- **Tables used:** orders, order_items, products, customers, reviews, payments, etc.

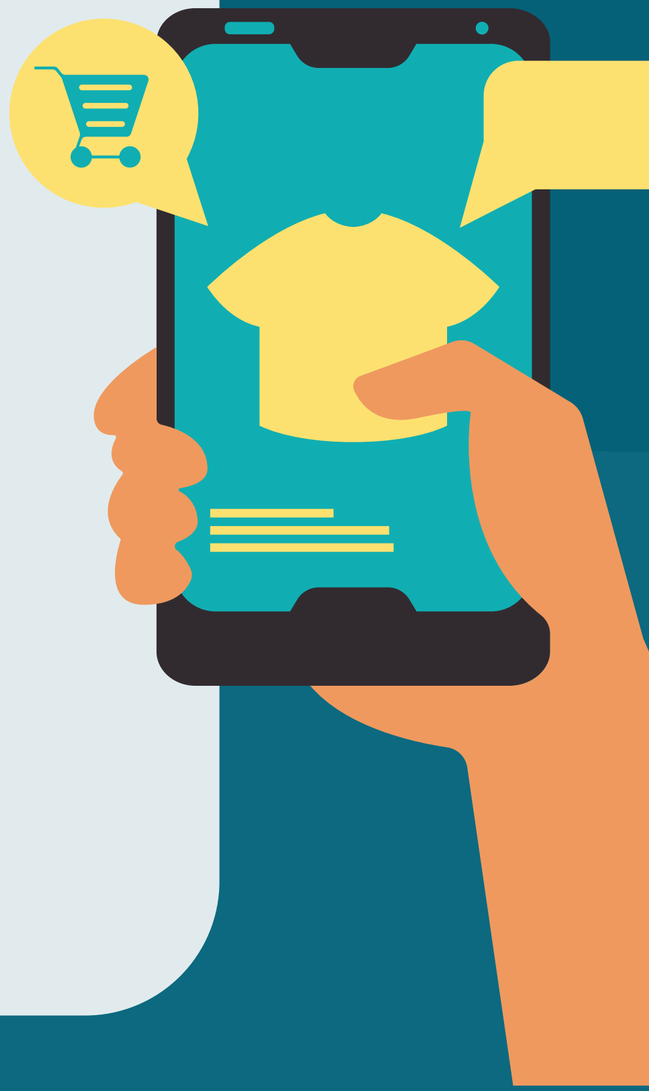
KEY BUSINESS QUESTIONS

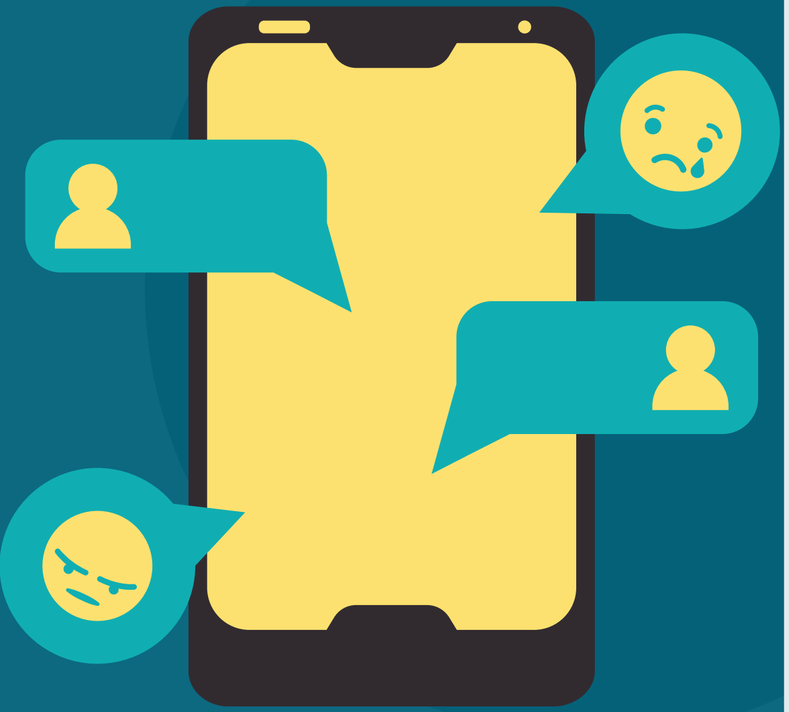
- COUNT THE NUMBER OF CUSTOMERS PER STATE.
- FIND THE TOTAL NUMBER OF ORDERS PER MONTH
- LIST THE TOP 10 CITIES WITH THE MOST ORDERS.
- FIND TOP 5 PRODUCT CATEGORIES BY TOTAL REVENUE.
- WHAT IS THE AVERAGE DELIVERY TIME?
- IDENTIFY PRODUCTS WITH HIGH REVENUE BUT POOR REVIEWS.
- WHICH MONTHS HAD THE HIGHEST SALES SPIKES? WHY?

1. COUNT THE NUMBER OF CUSTOMERS PER STATE.

```sql
SELECT
    customer_state, COUNT(*) AS customer_count
FROM
    customers
GROUP BY customer_state;
```

customer_state	customer_count
SP	41746
MG	11635
ES	2033
RJ	12852
RS	5466





FIND THE TOTAL NUMBER OF ORDERS PER MONTH

```
SELECT DATE_FORMAT(order_purchase_timestamp, '%Y-%m') AS order_month, COUNT(*) AS total_orders
from orders
group by order_month
order by order_month;
```

order_month	total_orders
2016-09	4
2016-10	324
2016-12	1
2017-01	800
2017-02	1780

LIST THE TOP 10 CITIES WITH THE MOST ORDERS.

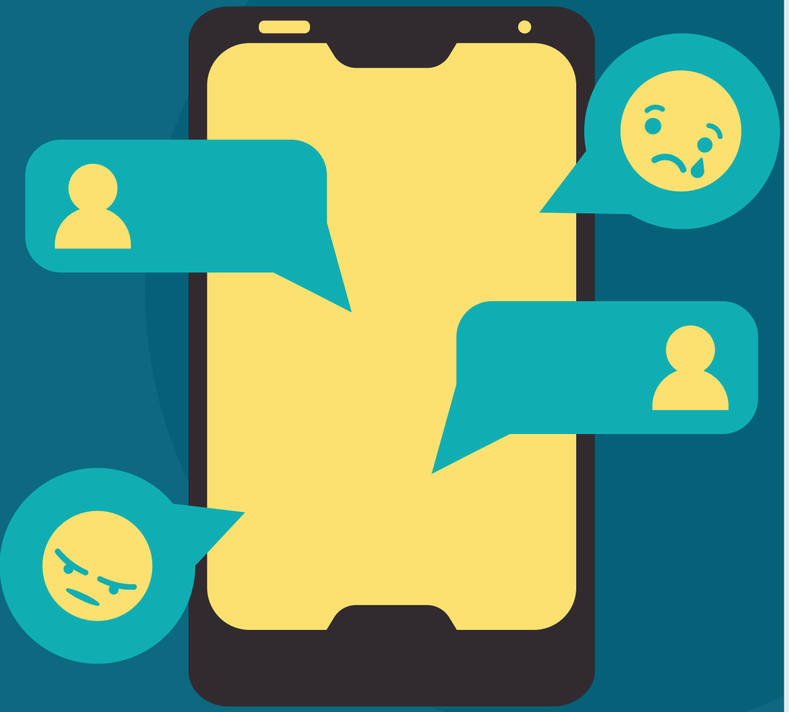
```sql
SELECT
    customers.customer_city,
    COUNT(orders.order_id) AS total_order
FROM
    customers
        JOIN
    orders ON customers.customer_id = orders.customer_id
GROUP BY customers.customer_city
ORDER BY total_order DESC
LIMIT 10;
```

customer_city	total_order
sao paulo	15540
rio de janeiro	6882
belo horizonte	2773
brasilia	2131
curitiba	1521
campinas	1444
porto alegre	1379
salvador	1245
guarulhos	1189

FIND TOP 5 PRODUCT CATEGORIES BY TOTAL REVENUE.

```sql
SELECT
    products.product_category_name,
    SUM(order_items.price + order_items.freight_value) AS total_revenue
FROM
    products
        JOIN
    order_items ON products.product_id = order_items.product_id
GROUP BY products.product_category_name
ORDER BY total_revenue DESC
LIMIT 5;
```

product_category_name	total_revenue
beleza_saude	1441248.07
relogios_presentes	1305541.61
cama_mesa_banho	1241681.72
esporte_lazer	1156656.48
informatica_acessorios	1059272.40



WHAT IS THE AVERAGE DELIVERY TIME

```
SELECT
    ROUND(AVG(DATEDIFF(order_delivered_customer_date,
                       order_purchase_timestamp)),2) AS avg_delivery_time
FROM
    orders
WHERE
    order_delivered_customer_date IS NOT NULL;
```

	avg_delivery_time
▶	12.50

IDENTIFY PRODUCTS WITH HIGH REVENUE BUT POOR REVIEWS.

```sql
SELECT
    order_items.product_id,
    ROUND(SUM(order_items.price + order_items.freight_value),
            2) AS revenue,
    ROUND(AVG(order_reviews.review_score), 2) AS avg_score
FROM
    order_items
        JOIN
    order_reviews ON order_reviews.order_id = order_items.order_id
GROUP BY order_items.product_id
HAVING revenue > 100 AND avg_score < 3
ORDER BY revenue DESC;
```

product_id	revenue	avg_score
25c38557cf793876c5abdd5931f922db	39286.47	2.68
fd0065af7f09af4b82a0ca8f3eed1852	22367.88	1.18
1dec4c88c685d5a07bf01dcb0f8bf9f8	21585.32	2.74
fb01a5fc09b9b9563c2ee41a22f07d54	16786.31	2.87
5769ef0a239114ac3a854af00df129e4	13664.08	1.00
19936fa4f614ee0590d3b77ac83fd648	10633.12	1.33
b4436da747c3a53ab07ac0e71de17dcd	10052.68	1.83
3db0b74faf0d26a6b252528659d6b849	9618.88	1.00
b5e13c9a353102f79c6206ff5cb61a50	9368.14	2.89

WHICH MONTHS HAD THE HIGHEST SALES SPIKES?

```sql
SELECT order_month,
       total_orders,
       total_orders - LAG(total_orders) OVER (ORDER BY order_month) AS month_over_month_change
FROM (
  SELECT DATE_FORMAT(order_purchase_timestamp,'%Y-%m') AS order_month,
         COUNT(*) AS total_orders
  FROM   orders
  GROUP  BY order_month
) AS sub
ORDER BY month_over_month_change DESC
LIMIT 5;
```

order_month	total_orders	month_over_month_chang
2017-11	7544	2913
2018-01	7269	1596
2017-05	3700	1296
2017-02	1780	980
2017-03	2682	902

CHALLENGES & LEARNINGS

- IMPORTING LARGE CSVS (100K+ ROWS) WITH MYSQL IMPORT WIZARD AND LOCAL INFILE
- FIXING DATETIME FORMATS AND NULL VALUES
- CREATING FOREIGN KEYS, INDEXING FOR PERFORMANCE
- LEARNING HOW SQL REVEALS PATTERNS THROUGH DATA

THANK YOU

LET’S CONNECT! THANKS FOR READING! I'D LOVE TO HEAR YOUR THOUGHTS.