



Target

# TARGET STORE ANALYSIS PROJECT



# PRESENTED BY

## USAMA ZAFAR

Business Intelligence Developer



# OBJECTIVE

- Evaluate sales performance and trends at Target stores.
- Analyze customer demographics and behaviors.
- Assess competitive positioning in the retail market.
- Identify opportunities for operational improvement.
- Provide actionable insights for strategic decision-making.



# CREATE TABLES TO IMPORT DATA FROM CSV FILES

```
CREATE TABLE geolocation  
(  
    geolocation_zip_code_prefix INT,  
    geolocation_lat DOUBLE,  
    geolocation_lng DOUBLE,  
    geolocation_city TEXT,  
    geolocation_state TEXT  
);
```

```
CREATE TABLE customers (  
    customer_id TEXT,  
    customer_unique_id TEXT,  
    customer_zip_code_prefix INT,  
    customer_city TEXT,  
    customer_state TEXT  
);
```

```
CREATE TABLE order_items (  
    order_id TEXT,  
    order_item_id INT,  
    product_id TEXT,  
    seller_id TEXT,  
    shipping_limit_date datetime,  
    price DOUBLE,  
    freight_value DOUBLE  
);
```

```
CREATE TABLE payments (  
    order_id TEXT,  
    payment_sequential INT,  
    payment_type TEXT,  
    payment_installments INT,  
    payment_value DOUBLE  
);
```



TARGET



**TARGET**

```
CREATE TABLE sellers (
    seller_id TEXT,
    seller_zip_code_prefix TEXT NULL,
    seller_city TEXT NULL,
    seller_state TEXT NULL
)j

CREATE TABLE products (
    product_id TEXT,
    product_category TEXT NULL,
    product_name_length int NULL,
    product_description_length int NULL,
    product_photos_qty int NULL,
    product_weight_g int NULL,
    product_length_cm int NULL,
    product_height_cm int NULL,
    product_width_cm int NULL
)j

#drop table if exists orders;
CREATE TABLE orders (
    order_id TEXT,
    customer_id TEXT,
    order_status TEXT,
    order_purchase_timestamp datetime null,
    order_approved_at datetime null,
    order_delivered_carrier_date datetime null,
    order_delivered_customer_date datetime null,
    order_estimated_delivery_date datetime null
)j
```

# IMPORT DATA FROM CSV FILES

```
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/customers.csv'  
INTO TABLE customers  
FIELDS TERMINATED BY ','  
LINES TERMINATED BY '\n'  
IGNORE 1 LINES;
```

```
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/geolocation.csv'  
INTO TABLE geolocation  
FIELDS TERMINATED BY ','  
LINES TERMINATED BY '\n'  
IGNORE 1 LINES;
```

```
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/order_items.csv'  
INTO TABLE order_items  
FIELDS TERMINATED BY ','  
LINES TERMINATED BY '\n'  
IGNORE 1 LINES;
```

# IMPORT DATA FROM CSV FILES

```
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/orders.csv'
INTO TABLE orders
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
IGNORE 1 LINES;

LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/payments.csv'
INTO TABLE payments
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
IGNORE 1 LINES;

LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/products.csv'
INTO TABLE products
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
IGNORE 1 LINES;

LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/sellers.csv'
INTO TABLE sellers
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
IGNORE 1 LINES;
```

# TABLES

The image shows a screenshot of a database management system interface. At the top, there's a header bar with a magnifying glass icon and the word "Tables". Below the header, there's a list of seven tables, each preceded by a blue downward-pointing arrow icon and a small grid icon representing a table structure. The tables listed are: "customers", "geolocation", "order\_items", "orders", "payments", "products", and "sellers".

Table
customers
geolocation
order_items
orders
payments
products
sellers



# RETRIEVE ALL CUSTOMER INFORMATION ABOUT CUSTOMERS.

SELECT

\*

FROM

customers;



customer_city	customer_state
franca	SP
sao bernardo do campo	SP
sao paulo	SP
mogi das cruzes	SP
campinas	SP
...	...

customer_id	customer_unique_id	customer_zip_code_prefix
06b8999e2fba1a1fbc88172c00ba8bc7	861eff4711a542e4b93843c6dd7febb0	14409
18955e83d337fd6b2def6b18a428ac77	290c77bc529b7ac935b93aa66c333dc3	9790
4e7b3e00288586ebd08712fdd0374a03	060e732b5b29e8181a18229c7b0b2b5e	1151
b2b6027bc5c5109e529d4dc6358b12c3	259dac757896d24d7702b9adbbff3f3c	8775
4f2d8ab171c80ec8364f7c12e35b23ad	345ecd01c38d18a9036ed96c73b8d066	13056
0700c44-4-10b-0047379-07-07-171760	A-00744-010007-4-006F1A-CAD-011C-A	00000

# WHAT IS THE TOTAL REVENUE GENERATED?

SELECT

```
ROUND(SUM(payment_value), 2) AS total_Revenue_in_$
```

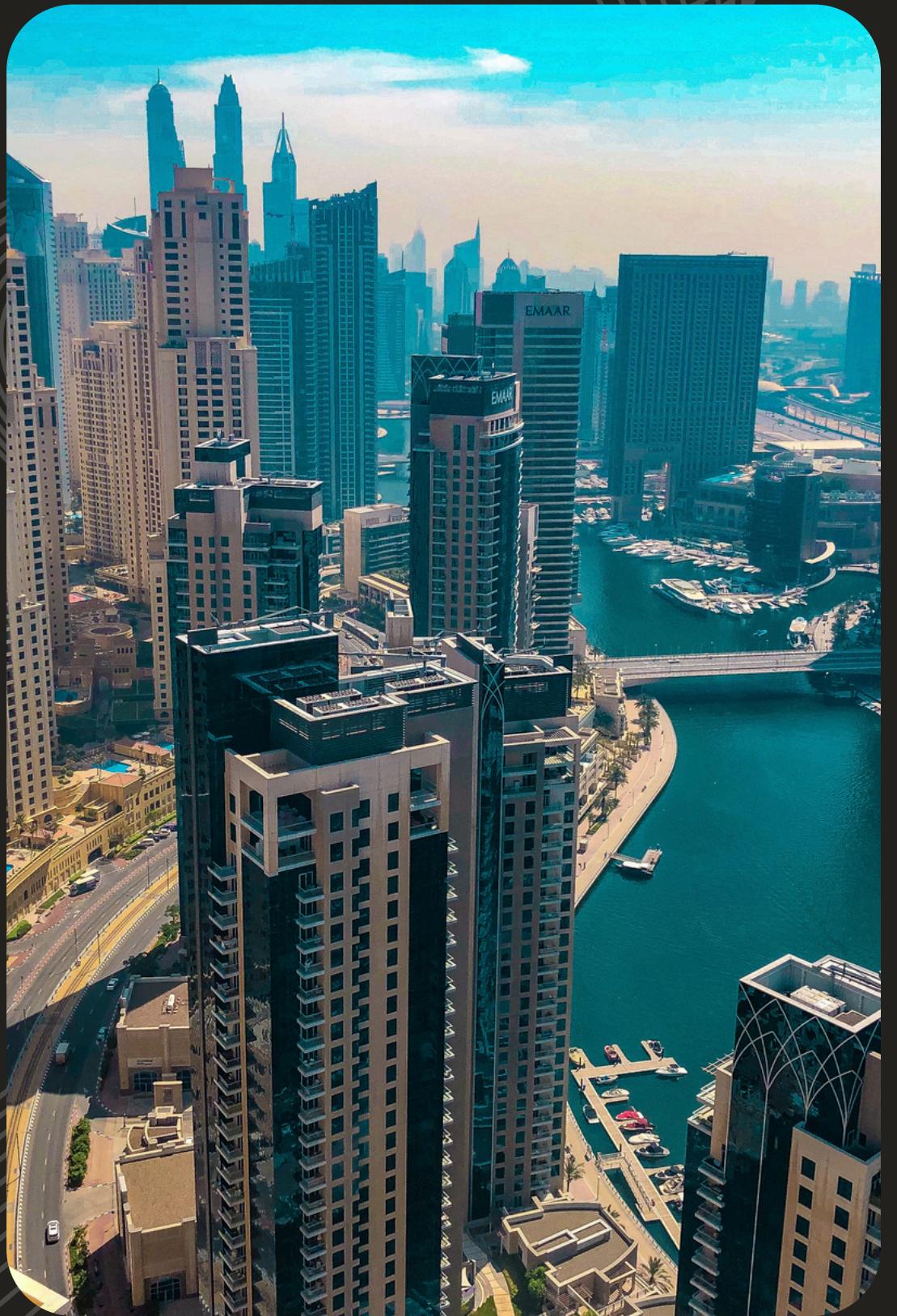
FROM

```
payments;
```

total\_Revenue\_in\_\$

16008872.12

# CALCULATE TOTAL ORDERS PLACED.



SELECT

COUNT(order\_id) AS Total\_orders

FROM

orders;

Total\_orders

96463

# CALCULATE THE AVERAGE ORDER VALUE ACROSS ALL ORDERS.

```
SELECT  
    ROUND(AVG(payment_value), 2) AS average_order_value  
FROM  
    payments p  
    JOIN  
    orders o ON p.order_id = o.order_id;
```

average_order_value
153.07

# LIST THE NUMBER OF ORDERS PLACED FROM TOP 5 STATES.

```
SELECT  
    c.customer_state, COUNT(o.order_id) AS no_of_orders  
FROM  
    customers c  
        JOIN  
    orders o ON c.customer_id = o.customer_id  
GROUP BY customer_state  
ORDER BY no_of_orders DESC  
limit 5;
```

customer_state	no_of_orders
SP	40490
RJ	12351
MG	11352
RS	5343
PR	4923

# IDENTIFY THE SELLERS LOCATED IN A SAO PAULO CITY.

```
SELECT  
    seller_id, seller_city  
FROM  
    sellers  
WHERE  
    seller_city = 'sao paulo';
```

seller_id	seller_city
c0f3eea2e14555b6faeea3dd58c1b1c3	sao paulo
768a86e36ad6aae3d03ee3c6433d61df	sao paulo
8bd0f31cf0a614c658f6763bd02dea69	sao paulo
05a48cc8859962767935ab9087417fbb	sao paulo
f9ec7093df3a7b346b7bcf7864069ca3	sao paulo
f7496d659ca9fdaf323c0aae84176632	sao paulo
430315b7bb4b6e4b3c978f9dfa9b0558	sao paulo
e9e446d01bd10a97a8fffcfc4a3a20cb2	sao paulo
d9a84e1403de8da0c3aa531d6d108ba6	sao paulo

# COUNT THE TOTAL NUMBER OF UNIQUE PRODUCTS LISTED.

```
SELECT DISTINCT  
    COUNT(product_id) AS T_no_of_unique_products  
FROM  
    products;
```

T_no_of_unique_products
32340

# DETERMINE TOP 5 PRODUCTS BY REVENUE.

SELECT

```
p.product_id,  
ROUND(SUM(py.payment_value), 2) AS t_revenue_by_product
```

FROM

```
payments py  
JOIN  
order_items oi ON py.order_id = oi.order_id  
JOIN  
products p ON p.product_id = oi.product_id
```

GROUP BY product\_id

ORDER BY t\_revenue\_by\_product DESC LIMIT 5;

product_id	t_revenue_by_product
5769ef0a239114ac3a854af00df129e4	109312.64
bb50f2e236e5eea0100680137654686c	81887.42
422879e10f46682990de24d770e7f83d	79512.22
d1c427060a0f73f6b889a5c7c61f2ac4	70557.9
6cdd53843498f92890544667809f1595	64825.67

# DETERMINE THE AVERAGE DELIVERY TIME FOR ORDERS

```
SELECT  
    ROUND(AVG(TIMESTAMPDIFF(DAY,  
        order_purchase_timestamp,  
        order_delivered_customer_date)),  
    2) AS avg_delivery_time_Days  
FROM  
    orders;
```

avg_delivery_time_Days
12.09

# ANALYZE THE TOP 5 CUSTOMERS BASED ORDERS REPORT

SELECT

```
c.customer_unique_id,  
COUNT(o.order_id) AS no_of_repeat_orders  
FROM  
orders o  
JOIN  
customers c ON c.customer_id = o.customer_id  
GROUP BY c.customer_unique_id  
ORDER BY no_of_repeat_orders DESC  
limit 5;
```

customer_unique_id	no_of_repeat_orders
8d50f5eadf50201ccdc6fb9e2ac8455	15
3e43e6105506432c953e165fb2acf44c	9
6469f99c1f9dfaef7733b25662e7f1782	7
ca77025e7201e3b30c44b472ff346268	7
1b6c7548a2a1f9037c1fd3ddfed95f33	7

# HOW MANY UNIQUE CUSTOMERS HAVE MADE PURCHASES IN EACH TOP 5 STATE?

```
CREATE INDEX idx_customer_zip_code_prefix ON customers(customer_zip_code_prefix);
CREATE INDEX idx_geolocation_zip_code_prefix ON geolocation(geolocation_zip_code_prefix);
SELECT
    geolocation_state,
    COUNT(DISTINCT c.customer_id) AS num_unique_customers
FROM
    customers c
        JOIN
    geolocation g ON g.geolocation_zip_code_prefix = c.customer_zip_code_prefix
GROUP BY geolocation_state
ORDER BY num_unique_customers DESC LIMIT 5 ;
```

geolocation_state	num_unique_customers
SP	41731
RJ	12839
MG	11624
RS	5473
PR	5034

# WHICH PRODUCT CATEGORIES HAVE THE HIGHEST AVERAGE ORDER VALUES (INCLUDING BOTH PRODUCT PRICE AND FREIGHT COST)?

SELECT

```
p.product_category,  
round(AVG(oi.price + oi.freight_value),2) AS avg_order_value
```

FROM

```
products p  
JOIN  
order_items oi ON p.product_id = oi.product_id  
JOIN  
orders o ON oi.order_id = o.order_id
```

WHERE

```
o.order_status = 'delivered'  
GROUP BY p.product_category  
ORDER BY avg_order_value DESC;
```

product_category	avg_order_value
PCs	1147.49
HOUSE PASTA'S OVEN AND CAFE	674.6
ELECTRICES 2	511.73
Agro Industria e Comercio	369.92
musical instruments	310.58

# CALCULATE RFM METRICS

```
WITH customer_rfm AS (
    SELECT
        customer_id,
        DATEDIFF(NOW(), MAX(order_purchase_timestamp)) AS recency,
        COUNT(DISTINCT o.order_id) AS frequency,
        SUM(p.payment_value) AS monetary
    FROM orders o
    JOIN payments p ON o.order_id = p.order_id
    WHERE o.order_status = 'delivered' -- Consider only delivered orders
    GROUP BY customer_id
)
SELECT
    customer_id,
    recency,
    frequency,
    monetary,
    CASE
        WHEN recency >= 0 AND recency <= 30 THEN 'Active'
        WHEN recency > 30 AND recency <= 90 THEN 'Inactive'
        ELSE 'Lost'
    END AS customer_segment
FROM customer_rfm
ORDER BY monetary DESC, recency DESC, frequency DESC;
```

customer_id	recency	frequency	monetary	customer_segment
e6f959bf384d1d53b6d68826699bba12	2836	1	154.57	Lost
b8cf418e97ae795672d326288dfab7a7	2836	1	133.46	Lost
dc607dc98d6a11d5d04d9f2a70aa6c34	2836	1	92.27	Lost
6f989332712d3222b6571b1cf5b835ce	2836	1	53.73	Lost
355077684019f7f60a031656bd7262b8	2836	1	45.46	Lost
7812fcebf5e8065d31e1bb5f0017dae	2836	1	40.95	Lost

# CREATE FUNCTION TO CALCULATE AVERAGE PROCESSING TIME

```
drop function if exists avg_processing_time;  
DELIMITER $$  
CREATE FUNCTION avg_processing_time()  
RETURNS DECIMAL(10, 2)  
DETERMINISTIC  
BEGIN  
    DECLARE avg_time DECIMAL(10, 2);  
  
    SELECT AVG(TIMESTAMPDIFF(DAY, order_approved_at, order_delivered_customer_date))  
    INTO avg_time  
    FROM orders  
    WHERE order_status = 'delivered'  
        AND order_approved_at IS NOT NULL  
        AND order_delivered_customer_date IS NOT NULL;  
  
    RETURN avg_time;  
END$$  
  
DELIMITER ;  
  
select avg_processing_time();
```

avg_processing_time()
11.64

# KPIS

- Total Sales 16008872.12\$
- Total Orders Placed 96463
- Total number of unique products 32340
- Top 5 Customers
- Top 5 states by sale
- Average Processing Time 11.64 Days



# RECOMMENDATIONS

- Implement demand forecasting for optimized inventory management.
- Segment customers for targeted marketing and promotions.
- Streamline order processing to reduce average time to 11.64 days.
- Increase product variety with 32,340 unique items.
- Focus marketing on top 5 states by sales to maximize revenue.



Target  
TARGET

# Thank You

## Contact Us



+923026059857



usamabhutta035@gmail.com



[www.reallygreatsite.com](http://www.reallygreatsite.com)

# Usama Zafar

*Gerente General*