

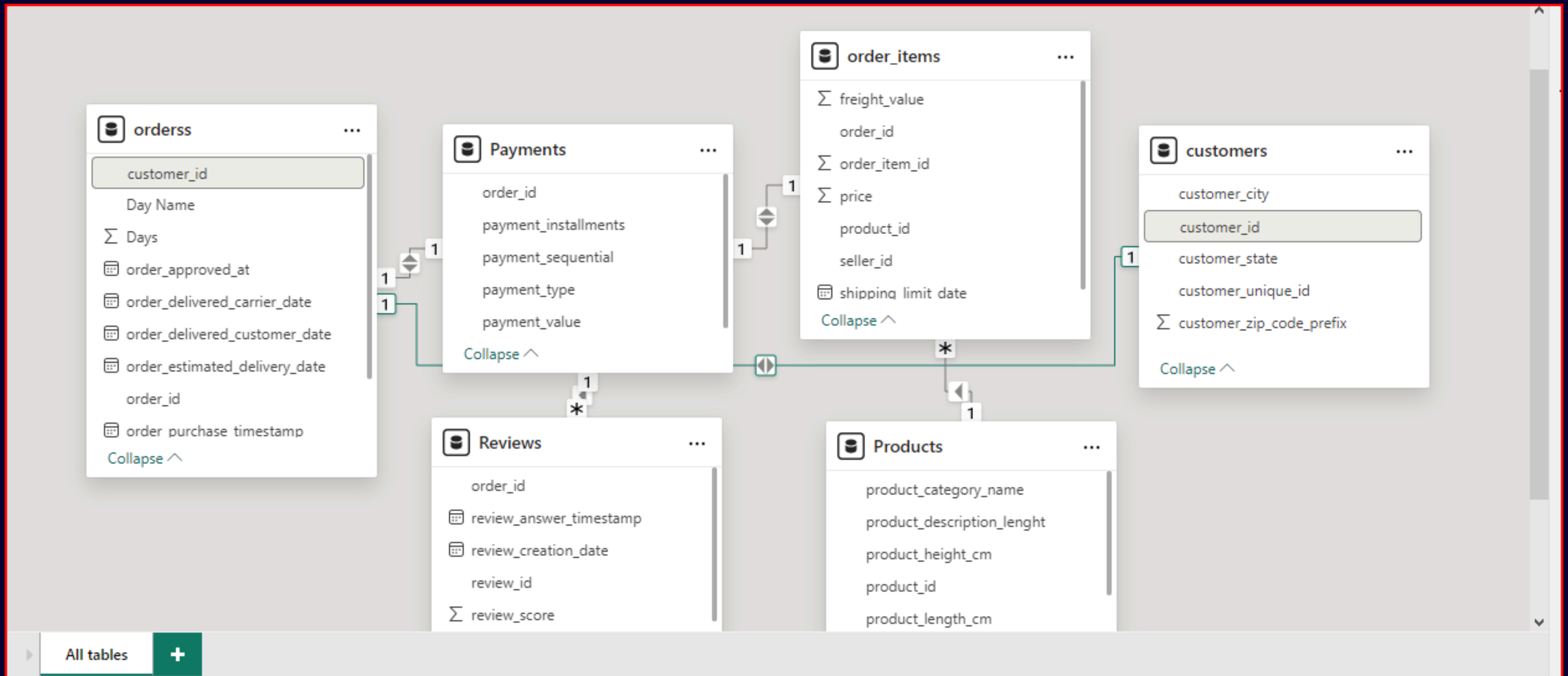
Olist Store Analysis

SQL Project

ABOUT

My name is Neha Soni. In this project, I analyzed the Olist dataset to extract insights on customer behavior and business performance. Key analyses included comparing weekday vs. weekend orders, payment statistics, and identifying orders with top review scores paid via credit card. I also calculated the average delivery time for pet shop products and analyzed average prices and payment values for customers in Sao Paulo. Lastly, I examined the relationship between shipping times and customer review scores to understand the impact of delivery speed on customer satisfaction.

Database schema overview



Weekday Vs Weekend (order_purchase_timestamp) Payment Statistics

```
With T3 as (select *,
Case when dayname(order_purchase_timestamp) = "sunday" or
dayname(order_purchase_timestamp) = "saturday" then "weekend"
else "weekday"
end as week_statics from orders)
select t3. week_statics, concat(round((count(p.order_id)/99441)*100), ' ', '%') as contribution
from t3 inner join payments p
on t3.order_id=p.order_id
group by t3.week_statics;
```

week_statics	contribution
weekend	23 %
weekday	77 %

Number of Orders with review score 5 and payment type as credit card.

```
SELECT reviews.review_score, payments.payment_type,  
       COUNT(orders.order_id) AS Numberoforders  
FROM  
  orders  
   INNER JOIN  
  payments ON orders.order_id = payments.order_id  
   INNER JOIN  
  reviews ON payments.order_id = reviews.order_id  
WHERE  
  reviews.review_score = 5  
   AND payments.payment_type = 'credit_card';
```

review_score	payment_type	Numberoforders
5	credit_card	43523

Average number of days taken for order_delivered_customer_date for pet_shop

```
SELECT p.product_category_name,  
       floor(AVG(DATEDIFF(o.order_delivered_customer_date,  
                           o.order_purchase_timestamp)))AS avg_shipping_days  
FROM  
  orders o  
    JOIN  
  orders_item it ON o.order_id = it.order_id  
    JOIN  
  products p ON it.product_id = p.product_id  
WHERE  
  product_category_name = 'petshop'  
group by p.product_category_name;
```

product_category_name	avg_shipping_days
petshop	11

Average price and payment values from customers of sao paulo city

```
WITH S1 AS (  
  SELECT  
    UPPER(C.CUSTOMER_CITY) AS CITY,  
    CONCAT(FLOOR(AVG(P.PAYMENT_VALUE)), ' ', '$') AS AVG_PAYMENT,  
    AVG(P.PAYMENT_VALUE) AS PAYMENT FROM ORDERS O  
  INNER JOIN PAYMENTS P ON O.ORDER_ID = P.ORDER_ID  
  INNER JOIN CUSTOMERS C ON O.CUSTOMER_ID = C.CUSTOMER_ID  
  WHERE UPPER(C.CUSTOMER_CITY) = 'SAO PAULO'  
  GROUP BY C.CUSTOMER_CITY ),  
S2 AS (  
  SELECT  
    UPPER(C.CUSTOMER_CITY) AS CITY,  
    CONCAT(FLOOR(AVG(I.PRICE)), ' ', '$') AS AVG_PRICE,  
    AVG(I.PRICE) AS PRICE FROM ORDERS O  
  INNER JOIN ORDERS_ITEM I ON O.ORDER_ID = I.ORDER_ID  
  INNER JOIN CUSTOMERS C ON O.CUSTOMER_ID = C.CUSTOMER_ID  
  WHERE UPPER(C.CUSTOMER_CITY) = 'SAO PAULO'  
  GROUP BY C.CUSTOMER_CITY)  
SELECT S1.CITY, S2.AVG_PRICE, S1.AVG_PAYMENT FROM S1 INNER JOIN S2 ON S1.CITY = S2.CITY;
```

	CITY	AVG_PRICE	AVG_PAYMENT
	SAO PAULO	108 \$	139 \$

Relationship between shipping days (order_delivered_customer_date - order_purchase_timestamp) Vs review scores.

```
SELECT
    r.review_score,
    Floor(AVG(DATEDIFF(o.order_delivered_customer_date,
                      o.order_purchase_timestamp))) AS shipping_days
FROM
    orders o
    JOIN
    reviews r ON o.order_id = r.order_id
GROUP BY r.review_score
ORDER BY r.review_score;
```

review_score	shipping_days
1	21
2	17
3	14
4	12
5	10

THANKYOU
