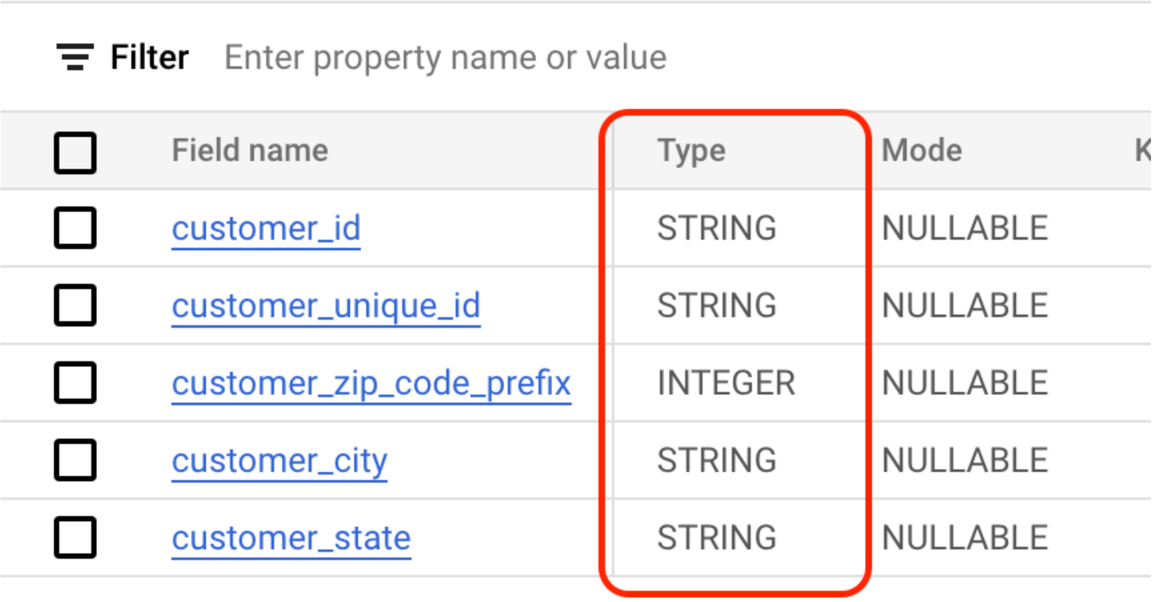
TARGET CASE STUDY

1. Datatype of all columns in the “customers” table.



□	Field name	Type	Mode	K
□	customer_id	STRING	NULLABLE	
□	customer_unique_id	STRING	NULLABLE	
□	customer_zip_code_prefix	INTEGER	NULLABLE	
□	customer_city	STRING	NULLABLE	
□	customer_state	STRING	NULLABLE	

2. Get the time range between which the orders were placed.

```sql
select order_status, Min(order_purchase_timestamp) as first_purchase,Max(order_purchase_timestamp) as latest_purchase
from `target.orders`
group by order_status
order by Min(order_purchase_timestamp) asc, Max(order_purchase_timestamp) desc
limit 2;
```

3. Count the number of Cities and States in our dataset

```sql
select c.customer_id, g.geolocation_city, g.geolocation_state
from `target.customers` as c inner join `target.geolocation` as g on c.customer_city = g.geolocation_city
where c.customer_id is not NULL or not trim(customer_id) = ""
order by c.customer_id asc
limit 10;
```

4. Is there a growing trend in the number of orders placed over the past years?

```sql
SELECT
DATE_TRUNC(order_purchase_timestamp, MONTH) AS month,
COUNT(DISTINCT order_id) AS num_orders
FROM
`target.orders`
GROUP BY month
```

5. What time do Brazilian customers tend to buy (Dawn, Morning, Afternoon or Night)?

```sql
SELECT
CASE
WHEN EXTRACT(HOUR FROM order_purchase_timestamp) < 6 THEN 'Dawn'
WHEN EXTRACT(HOUR FROM order_purchase_timestamp) < 12 THEN 'Morning'
WHEN EXTRACT(HOUR FROM order_purchase_timestamp) < 18 THEN 'Afternoon'
ELSE 'Night'
END AS time_of_day,
COUNT(order_id) AS num_orders,
FROM `target.orders`
GROUP BY time_of_day
ORDER BY time_of_day;
```

6. **Get the month on month no. of orders placed in each state.**

```sql
SELECT
CAST(DATE_TRUNC(o.order_purchase_timestamp, MONTH) AS TIMESTAMP) AS month,
COUNT(DISTINCT o.order_id) AS num_orders, g.geolocation_state, FROM `target.orders` o LEFT JOIN `target.customers` c ON o.customer_id = c.customer_id
LEFT JOIN `target.geolocation` g ON c.customer_zip_code_prefix = g.geolocation_zip_code_prefix
GROUP BY month, g.geolocation_state
ORDER BY month ASC, num_orders DESC;
```

7. **Distribution of customers across the states in Brazil**

```sql
select customer_state, count(customer_id) as customer_id
from `target.customers`
group by customer_state
order by customer_id desc;
```

8. Get % increase in cost of orders from 2017 to 2018 (include months between Jan to Aug only).

```
SELECT
(SUM(CASE
WHEN extract(year from o.order_purchase_timestamp) = 2018 AND extract(month
from o.order_purchase_timestamp) <= 8 THEN p.payment_value
WHEN extract(year from o.order_purchase_timestamp) = 2017 THEN -
p.payment_value
ELSE 0
END) / ABS(SUM(CASE WHEN extract(year from o.order_purchase_timestamp) =
2017 AND extract(month from o.order_purchase_timestamp) <= 8 THEN
p.payment_value ELSE 0 END))) * 100 AS percentage_increase
FROM `target.payments` p
INNER JOIN `target.orders` o ON p.order_id = o.order_id
WHERE extract(year from o.order_purchase_timestamp) IN (2017, 2018) AND
extract(month from o.order_purchase_timestamp) BETWEEN 1 AND 8
group by extract(year from o.order_purchase_timestamp) IN (2017, 2018) AND
extract(month from o.order_purchase_timestamp) BETWEEN 1 AND 8;
```

9. Calculate the total and average value of order price and freight for each state

```
select c.customer_state, sum(oi.price) sum_price, sum(oi.freight_value)
sum_freight_value, avg(oi.price) mean_price, avg(oi.freight_value)
mean_freight_value
       from `target.order_items` oi left join
       `target.orders` o on oi.order_id = o.order_id left join
       `target.customers` c on o.customer_id = c.customer_id
       group by c.customer_state
       order by c.customer_state;
```

10. Find the no. of days taken to deliver each order from the order's purchase date as delivery time.

```
select order_purchase_timestamp, order_delivered_customer_date,
order_estimated_delivery_date,
datetime_diff(order_purchase_timestamp, order_delivered_customer_date, day)
as diff_purchase_delivering,
datetime_diff(order_purchase_timestamp, order_estimated_delivery_date, day)
as diff_purchase_estimated_delivery,
datetime_diff(order_delivered_customer_date, order_estimated_delivery_date,
day) as diff_delivering_estimated_del
from `target.orders`
order by order_purchase_timestamp desc;
```

TARGET CASE STUDY

11. Top 5 states with highest/lowest average freight value - sort in desc/asc limit 5

```
select c.customer_state, avg(oi.freight_value) as avg_freight_value,
from `target.customers` c inner join
`target.orders` o on c.customer_id = o.customer_id inner join
`target.order_items` oi on o.order_id = oi.order_id
group by c.customer_state
order by avg_freight_value desc or asc
limit 5;
```

desc for highest and asc for lowest

14. Top 5 states with highest/lowest average time to delivery

```
select x.customer_state, round(avg(x.time_to_delivery),2) as
avg_time_to_delivery,
from (select c.customer_state,
datetime_diff(o.order_delivered_customer_date, o.order_purchase_timestamp,
day) as time_to_delivery,
from `target.customers` c inner join
`target.orders` o on c.customer_id = o.customer_id inner join
`target.order_items` oi on o.order_id = oi.order_id
group by c.customer_state, o.order_delivered_customer_date,
o.order_purchase_timestamp, order_estimated_delivery_date) x
group by x.customer_state
order by avg_time_to_delivery desc or asc;
```

desc for highest and asc for lowest

15. Top 5 states where delivery is really fast/ not so fast compared to estimated date

```
select c.customer_state, datetime_diff(o.order_delivered_customer_date, o.order_purchase_timestamp, day) as time_to_delivery,
datetime_diff(order_estimated_delivery_date, order_delivered_customer_date, day) as diff_estimated_delivery,
from `target.customers` c inner join
`target.orders` o on c.customer_id = o.customer_id inner join
`target.order_items` oi on o.order_id = oi.order_id
group by c.customer_state, o.order_delivered_customer_date, o.order_purchase_timestamp, order_estimated_delivery_date
order by diff_estimated_delivery desc
limit 5;
```

16. Month over Month count of orders for different payment types

```
SELECT
CAST(DATE_TRUNC(o.order_purchase_timestamp, MONTH) AS TIMESTAMP) AS month,
COUNT(DISTINCT p.order_id) AS count_of_orders, p.payment_type,
FROM `target.orders` o RIGHT JOIN `target.payments` p ON o.order_id = p.order_id
GROUP BY month, p.payment_type
ORDER BY month ASC, p.payment_type DESC;
```

17. Count of orders based on the no. of payment instalments.

```
SELECT payment_type,
count (distinct payment_installments) as no_payment_installments,
count(distinct order_id) as count_order_id,
from `target.payments`
group by payment_type
ORDER BY count_order_id DESC;
```