```
# Data type of all columns in the "customers" table.

SELECT
    COLUMN_NAME,
    DATA_TYPE
FROM `scaler-dsml-sql-444401.Target_SQL_Buisness_Case.INFORMATION_SCHEMA.COLUMNS`
WHERE TABLE_NAME = 'customers';

# Get the time range between which the orders were placed.
SELECT
date_diff(recent_order,first_order,day) AS time_range
FROM
(
SELECT
MIN(order_purchase_timestamp) AS first_order,
MAX(order_purchase_timestamp) AS recent_order
from `Target_SQL_Buisness_Case.orders`
)

# Count the Cities & States of customers who ordered during the given period.

SELECT
COUNT(DISTINCT c.customer_city) as city_count,
COUNT(DISTINCT c.customer_state) AS state_count
FROM `Target_SQL_Buisness_Case.customers` c
INNER JOIN `Target_SQL_Buisness_Case.orders` o
ON c.customer_id = o.customer_id
WHERE EXTRACT(YEAR FROM o.order_purchase_timestamp) BETWEEN 2016 AND 2018;

# Is there a growing trend in the no. of orders placed over the past years?

select
order_year,
no_of_orders,
LAG(no_of_orders) OVER(ORDER BY order_year) AS prev_year_orders,
CASE
WHEN
LAG(no_of_orders) OVER(ORDER BY order_year) IS NULL
THEN NULL
ELSE
ROUND((no_of_orders - LAG(no_of_orders) OVER(ORDER BY
order_year))/LAG(no_of_orders) OVER(ORDER BY order_year)*100,2)
END AS year_by_year_growth_percentage
from
(
SELECT
EXTRACT(YEAR FROM order_purchase_timestamp ) AS order_year,
COUNT(order_id) AS no_of_orders
from `Target_SQL_Buisness_Case.orders`
GROUP BY order_year
)
ORDER BY order_year;


# Can we see some kind of monthly seasonality in terms of the no. of orders being
placed?

SELECT
```

```
EXTRACT(MONTH FROM order_purchase_timestamp) AS order_month,
COUNT(order_id) AS total_orders
FROM
`Target_SQL_Buisness_Case.orders`
GROUP BY
order_month
ORDER BY
order_month;

# During what time of the day, do the Brazilian customers mostly place their
orders? (Dawn, Morning, Afternoon or Night)
# 0-6 hrs : Dawn
# 7-12 hrs : Mornings
# 13-18 hrs : Afternoon
# 19-23 hrs : Night

SELECT
Order_Time,
tbl2.no_of_orders
FROM
(
SELECT
Order_Time,
COUNT(*) AS no_of_orders,
DENSE_RANK()OVER(ORDER BY COUNT(*) desc) AS rank
FROM
(
SELECT
CASE
WHEN EXTRACT(HOUR FROM order_purchase_timestamp) >=0 AND EXTRACT(HOUR FROM
order_purchase_timestamp)<=6 THEN 'Dawn'
WHEN EXTRACT(HOUR FROM order_purchase_timestamp) >= 7 AND EXTRACT(HOUR FROM
order_purchase_timestamp) <=12 THEN 'Mornings'
WHEN EXTRACT(HOUR FROM order_purchase_timestamp) >= 13 AND EXTRACT(HOUR FROM
order_purchase_timestamp) <= 18 THEN 'Afternoon'
ELSE 'NIGHT'
END AS Order_Time
FROM `Target_SQL_Buisness_Case.orders`
)tbl1
GROUP BY Order_Time
)tbl2
WHERE rank =1;

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

SELECT
c.customer_state,
EXTRACT(month from o.order_purchase_timestamp) AS month,
COUNT(o.order_id) AS no_of_orders
FROM `Target_SQL_Buisness_Case.orders` o
JOIN `Target_SQL_Buisness_Case.customers` c
ON o.customer_id = c.customer_id
GROUP BY c.customer_state,month
ORDER BY c.customer_state,month;


# How are the customers distributed across all the states?
```

```sql
SELECT
customer_state,
COUNT(customer_unique_id) AS total_customers
FROM
`Target_SQL_Buisness_Case.customers`
GROUP BY
customer_state
ORDER BY
total_customers DESC;

# Get the % increase in the cost of orders from year 2017 to 2018 (include months
between Jan to Aug only).
# You can use the "payment_value" column in the payments table to get the cost of
orders.

SELECT *,
LAG(total_cost)OVER(ORDER BY year) AS prev_year_cost,
CASE
WHEN LAG(total_cost)OVER(ORDER BY year) IS NULL THEN NULL
ELSE ROUND((total_cost - LAG(total_cost)OVER(ORDER BY
year))/LAG(total_cost)OVER(ORDER BY year)*100,2)
END AS percentage_increase_in_cost
FROM
(
SELECT
EXTRACT(YEAR FROM o.order_purchase_timestamp) AS year,
SUM(p.payment_value) AS total_cost
FROM `Target_SQL_Buisness_Case.orders` o
JOIN `Target_SQL_Buisness_Case.payments` p
ON o.order_id = p.order_id
WHERE EXTRACT(YEAR FROM o.order_purchase_timestamp) IN (2017,2018) AND
EXTRACT(MONTH FROM o.order_purchase_timestamp) IN (1,2,3,4,5,6,7,8)
GROUP BY year
ORDER BY year
)
ORDER BY year;

# Calculate the Total & Average value of order price for each state.

SELECT c.customer_state,
SUM(p.payment_value) AS total_price,
ROUND(SUM(p.payment_value)/COUNT(DISTINCT o.order_id),2) AS avg_price
FROM `Target_SQL_Buisness_Case.customers` c
JOIN `Target_SQL_Buisness_Case.orders` o
ON c.customer_id = o.customer_id
JOIN `Target_SQL_Buisness_Case.payments` p
ON o.order_id = p.order_id
GROUP BY c.customer_state
order by c.customer_state;

# Calculate the Total & Average value of order freight for each state.

SELECT c.customer_state,
SUM(oi.freight_value) AS total_freight,
ROUND(SUM(oi.freight_value)/COUNT(DISTINCT o.order_id),2) AS avg_freight
FROM `Target_SQL_Buisness_Case.customers` c
JOIN `Target_SQL_Buisness_Case.orders` o
ON c.customer_id = o.customer_id
```

```sql
JOIN `Target_SQL_Buisness_Case.order_items` oi
ON o.order_id = oi.order_id
GROUP BY c.customer_state
order by c.customer_state;

#Find the no. of days taken to deliver each order from the order’s purchase date as
delivery time.
#Also, calculate the difference (in days) between the estimated & actual delivery
date of an order.
#Do this in a single query.
#You can calculate the delivery time and the difference between the estimated &
actual delivery date using the given formula:
#time_to_deliver = order_delivered_customer_date - order_purchase_timestamp
#diff_estimated_delivery = order_delivered_customer_date -
order_estimated_delivery_date

SELECT
order_id,
TIMESTAMP_DIFF(order_delivered_customer_date, order_purchase_timestamp, day) AS
delivery_time,
TIMESTAMP_DIFF(order_delivered_customer_date, order_estimated_delivery_date, day)
AS diff_estimated_delivery
FROM `Target_SQL_Buisness_Case.orders`
WHERE order_delivered_customer_date IS NOT NULL;


# Find out the top 5 states with the highest & lowest average freight value.

SELECT
c.customer_state,
ROUND(SUM(oi.freight_value)/COUNT(DISTINCT oi.order_id),2) AS avg_freight_value
FROM `Target_SQL_Buisness_Case.customers` c
JOIN `Target_SQL_Buisness_Case.orders` o
ON c.customer_id = o.customer_id
JOIN `Target_SQL_Buisness_Case.order_items` oi
ON o.order_id = oi.order_id
GROUP BY c.customer_state
ORDER BY avg_freight_value desc
LIMIT 5


UNION ALL

SELECT
c.customer_state,
ROUND(SUM(oi.freight_value)/COUNT(DISTINCT oi.order_id),2) AS avg_freight_value
FROM `Target_SQL_Buisness_Case.customers` c
JOIN `Target_SQL_Buisness_Case.orders` o
ON c.customer_id = o.customer_id
JOIN `Target_SQL_Buisness_Case.order_items` oi
ON o.order_id = oi.order_id
GROUP BY c.customer_state
ORDER BY avg_freight_value
LIMIT 5;

# Find out the top 5 states where the order delivery is really fast as compared to
the estimated date of delivery.
```

```sql
#You can use the difference between the averages of actual & estimated delivery date to figure out how fast the delivery was for each state.

SELECT
c.customer_state,
ROUND(AVG(TIMESTAMP_DIFF(o.order_delivered_customer_date,o.order_estimated_delivery_date,day)),2) AS avg_actual_delivery
FROM `Target_SQL_Buisness_Case.orders` o
JOIN `Target_SQL_Buisness_Case.customers` c
ON o.customer_id = c.customer_id
WHERE order_delivered_customer_date IS NOT NULL
GROUP BY c.customer_state
HAVING avg_actual_delivery < 0
ORDER BY avg_actual_delivery
limit 5;

# Find the month on month no. of orders placed using different payment types.

SELECT
EXTRACT(MONTH FROM o.order_purchase_timestamp) AS month,
EXTRACT(YEAR FROM o.order_purchase_timestamp) AS year,
p.payment_type,
COUNT(o.order_id) AS no_of_orders
FROM `Target_SQL_Buisness_Case.orders` o
JOIN `Target_SQL_Buisness_Case.payments` p
ON o.order_id = p.order_id
GROUP BY year,month, payment_type
order by year , month, no_of_orders desc

# Find the no. of orders placed on the basis of the payment installments that have been paid.

SELECT payment_installments,
COUNT(distinct order_id) AS total_orders
FROM `Target_SQL_Buisness_Case.payments`
GROUP BY payment_installments
order by payment_installments;
```