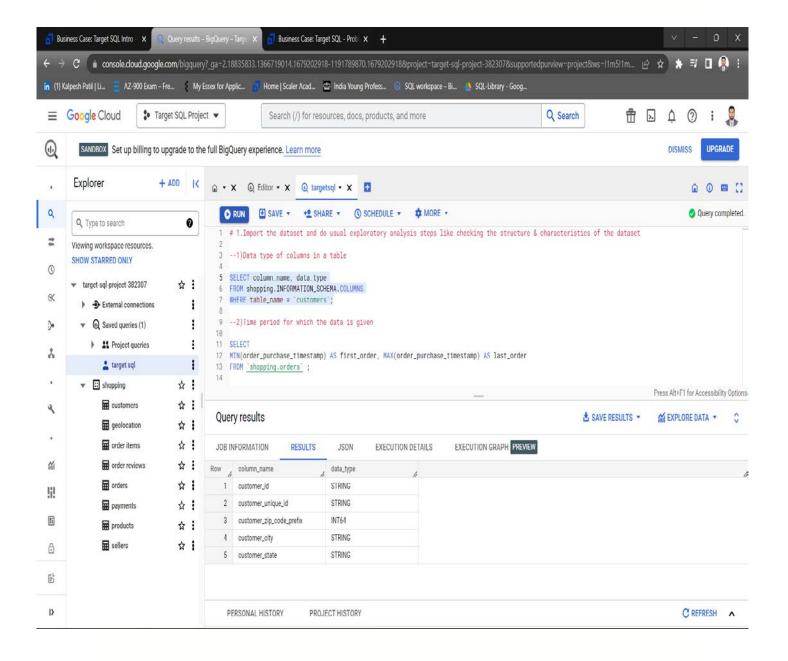
# **PROJECT:**

# Business Case: Target SQL

# 1.Import the dataset and do usual exploratory analysis steps like checking the st ructure & characteristics of the dataset

--1)Data type of columns in a table

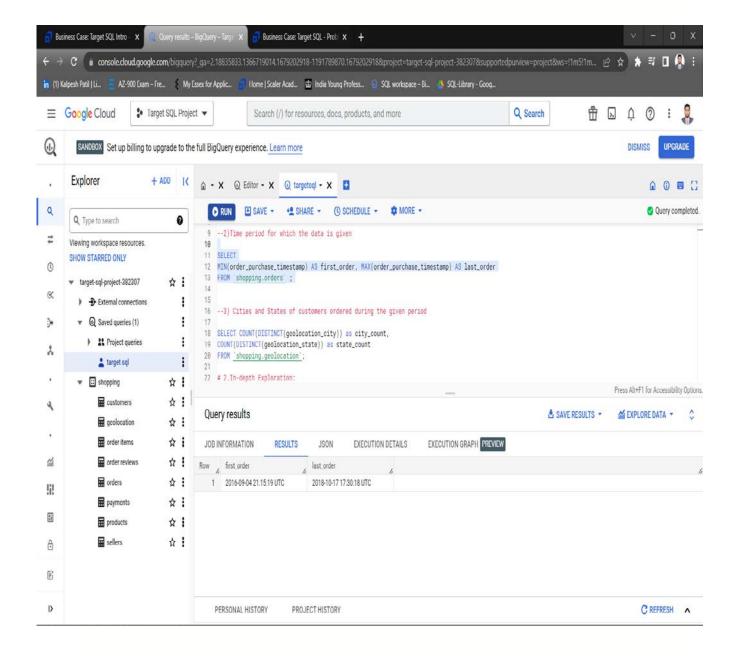
SELECT column\_name, data\_type
FROM shopping.INFORMATION\_SCHEMA.COLUMNS
WHERE table\_name = 'customers';



# --2)Time period for which the data is given

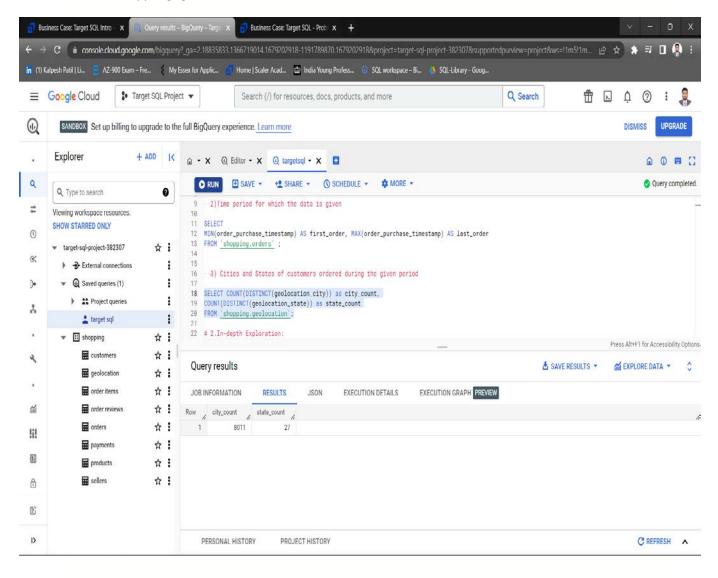
# **SELECT**

FROM `shopping.orders`;



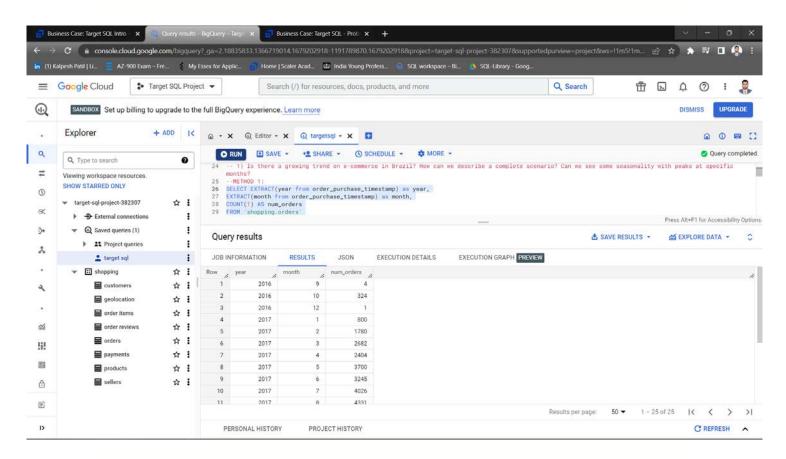
# --3) Cities and States of customers ordered during the given period

SELECT COUNT(DISTINCT(geolocation\_city)) as city\_count,
COUNT(DISTINCT(geolocation\_state)) as state\_count
FROM `shopping.geolocation`;

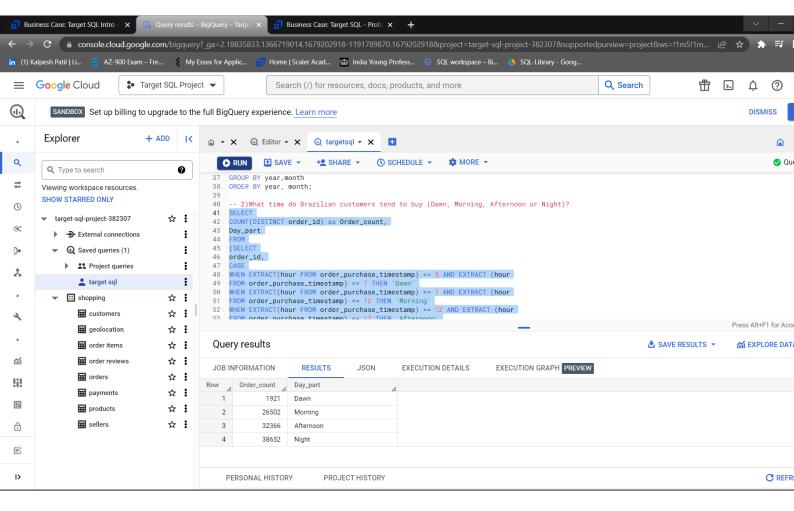


# # 2.In-depth Exploration:

```
-- 1) Is there a growing trend on e-
commerce in Brazil? How can we describe a complete scenario? Can we see some season
ality with peaks at specific months?
--METHOD 1:
SELECT EXTRACT(year from order_purchase_timestamp) as year,
EXTRACT(month from order_purchase_timestamp) as month,
COUNT(1) AS num_orders
FROM `shopping.orders`
GROUP BY year, month
ORDER BY year, month;
--METHOD 2:
SELECT DATE(order_purchase_timestamp) as year,
DATE(order_purchase_timestamp) as month,
COUNT(1) AS num_orders
FROM `shopping.orders`
GROUP BY year, month
ORDER BY year, month;
```



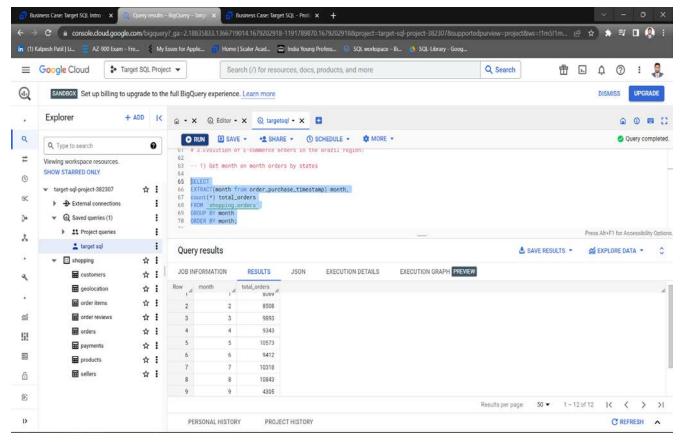
```
- 2)What time do Brazilian customers tend to buy (Dawn, Morning, Afternoon or Nigh?
COUNT(DISTINCT order_id) as Order_count,
Day_part
FROM
(SELECT
order_id,
CASE
WHEN EXTRACT(hour FROM order_purchase_timestamp) >= 5 AND EXTRACT (hour
FROM order_purchase_timestamp) <= 7 THEN 'Dawn'</pre>
WHEN EXTRACT(hour FROM order_purchase_timestamp) >= 7 AND EXTRACT (hour
FROM order_purchase_timestamp) <= 12 THEN 'Morning'
WHEN EXTRACT(hour FROM order_purchase_timestamp) >= 12 AND EXTRACT (hour
FROM order_purchase_timestamp) <= 17 THEN 'Afternoon'
ELSE 'Night'
END as Day_Part
FROM `shopping.orders`) Y
GROUP BY Day_Part
ORDER BY COUNT(DISTINCT order_id);
```



#### # 3.Evolution of E-commerce orders in the Brazil region:

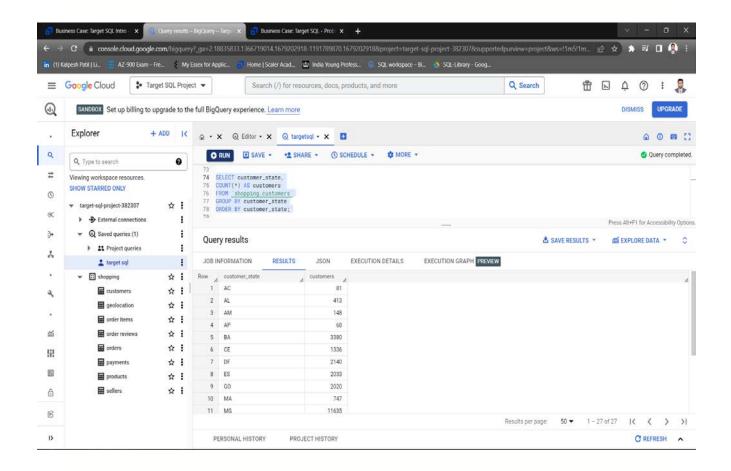
# -- 1) Get month on month orders by states

SELECT
EXTRACT(month from order\_purchase\_timestamp) month,
count(\*) total\_orders
FROM `shopping.orders`
GROUP BY month
ORDER BY month;

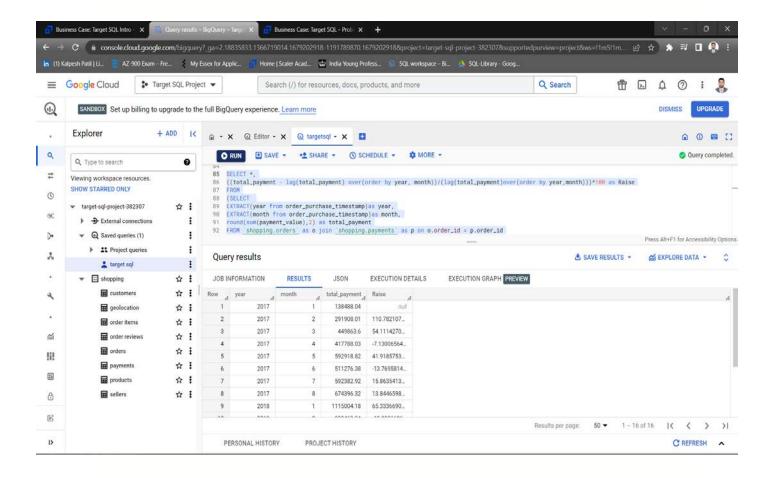


# -- 2) Distribution of customers across the states in Brazil

SELECT customer\_state, COUNT(\*) AS customers FROM `shopping.customers` GROUP BY customer\_state ORDER BY customer\_state;



```
# 4. Impact on Economy: Analyze the money movement by e-
commerce by looking at order prices, freight and others.
1) Get % increase in cost of orders from 2017 to 2018 (include months between Jan t
o Aug only) - You can use "payment_value" column in payments table
SELECT *.
((total_payment - lag(total_payment) over(order by year, month))/(lag(total_payment
)over(order by year, month)))*100 as Raise
FROM
(SELECT
EXTRACT(year from order_purchase_timestamp)as year,
EXTRACT(month from order_purchase_timestamp)as month,
round(sum(payment_value),2) as total_payment
FROM `shopping.orders` as o join `shopping.payments` as p on o.order_id = p.order_i
GROUP BY year, month
ORDER BY year, month )t1
WHERE year in (2017, 2018) and month not in (9,10,11,12)
ORDER BY year, month;
```



# --2) Mean & Sum of price and freight value by customer state

```
SELECT s.seller_state,

AVG(price) AS mean_price,

SUM(price) AS total_price,

AVG(freight_value) AS mean_freight,

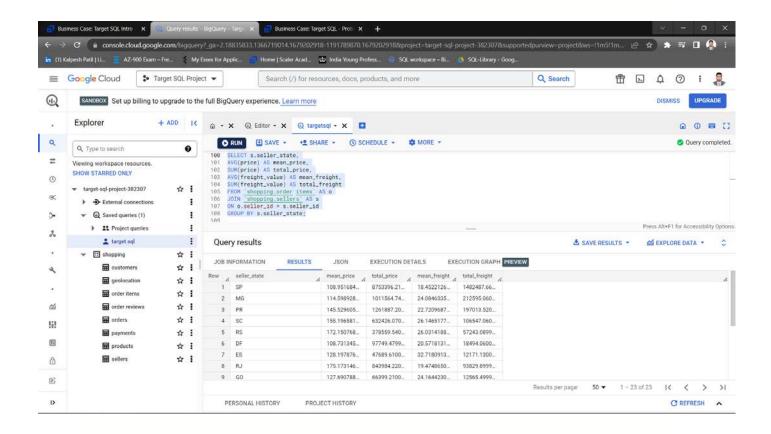
SUM(freight_value) AS total_freight

FROM `shopping.order items` AS o

JOIN `shopping.sellers` AS s

ON o.seller_id = s.seller_id

GROUP BY s.seller_state;
```



# #5. Analysis on sales, freight and delivery time

# --1) Calculate days between purchasing, delivering and estimated delivery

#### **SELECT**

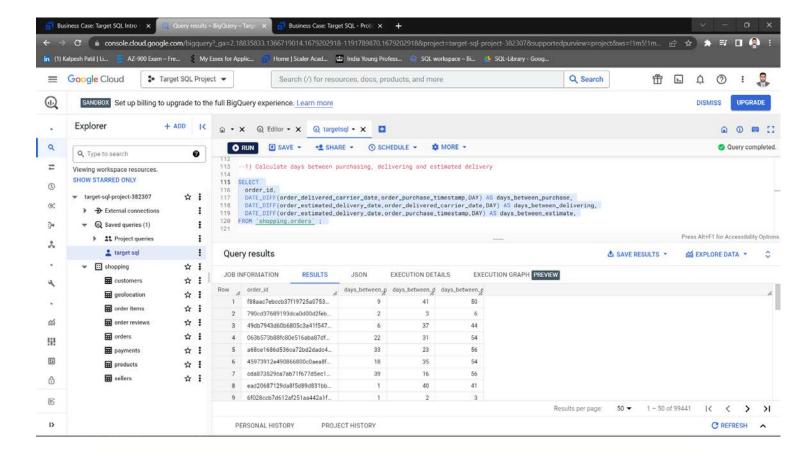
order\_id,

DATE\_DIFF(order\_delivered\_carrier\_date,order\_purchase\_timestamp,DAY) AS days\_betw een\_purchase,

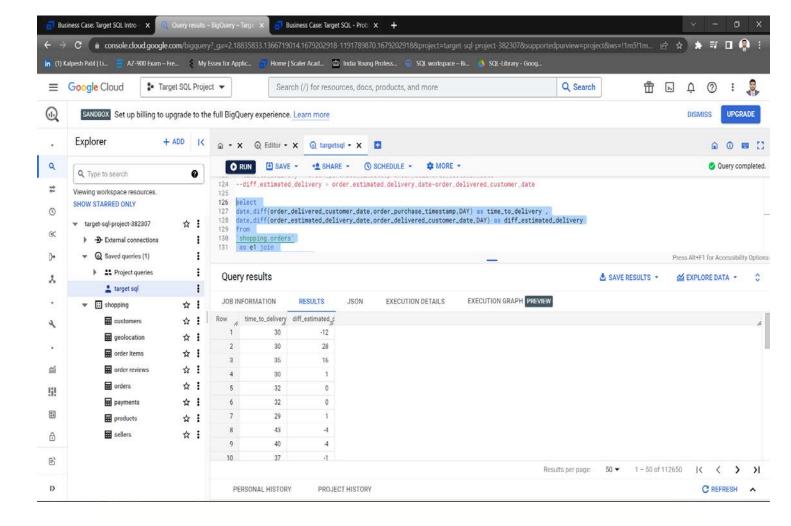
DATE\_DIFF(order\_estimated\_delivery\_date,order\_delivered\_carrier\_date,DAY) AS days
\_between\_delivering,

 $\label{eq:delivery_date} \begin{picture}(c) DATE\_DIFF(order\_estimated\_delivery\_date, order\_purchase\_timestamp, DAY) AS days\_between\_estimate, \end{picture}$ 

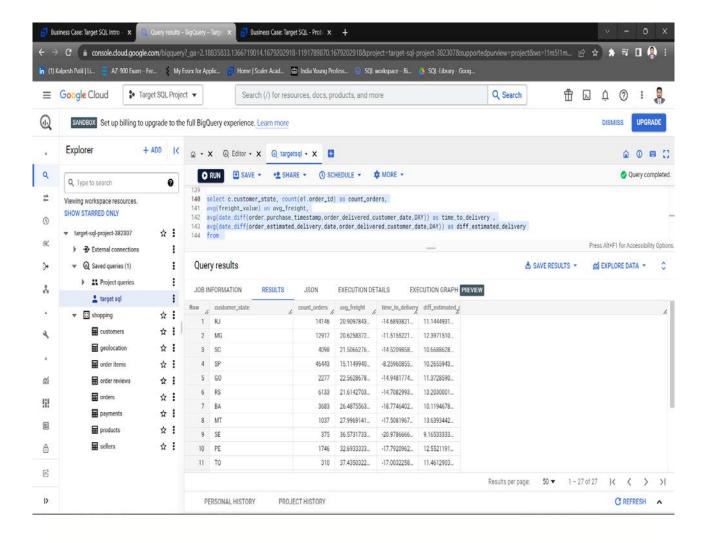
FROM `shopping.orders` ;



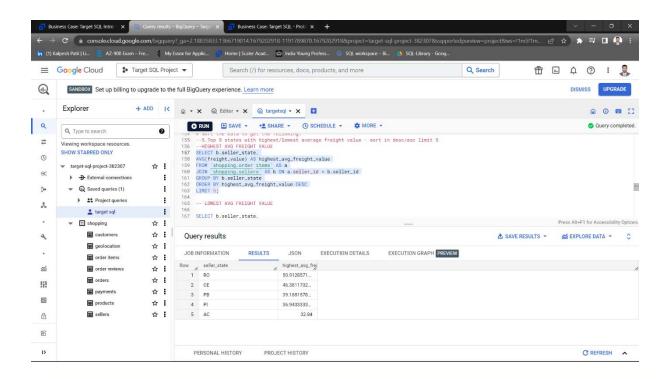
```
2) Find time_to_delivery & diff_estimated_delivery. Formula for the same given belo
--time_to_delivery = order_purchase_timestamp-order_delivered_customer_date
--diff_estimated_delivery = order_estimated_delivery_date-
order_delivered_customer_date
select
date_diff(order_delivered_customer_date,order_purchase_timestamp,DAY) as time_to_de
livery ,
date_diff(order_estimated_delivery_date,order_delivered_customer_date,DAY) as diff_
estimated_delivery
from
`shopping.orders`
as e1 join
`shopping.order items`
as e2 on e1.order_id=e2.order_id
join `shopping.customers`
as c on e1.customer_id = c.customer_id;
```



```
3) Group data by state, take mean of freight_value, time_to_delivery, diff_estimate
d_delivery
select c.customer_state, count(e1.order_id) as count_orders,
avg(freight_value) as avg_freight,
avg(date_diff(order_purchase_timestamp,order_delivered_customer_date,DAY)) as time_
to_delivery ,
avg(date_diff(order_estimated_delivery_date,order_delivered_customer_date,DAY)) as
diff_estimated_delivery
`shopping.orders`
as e1 join
`shopping.order items`
as e2 on e1.order_id=e2.order_id
join `shopping.customers`
as c on e1.customer_id = c.customer_id
where order_delivered_customer_date is not null
group by c.customer_state ;
```

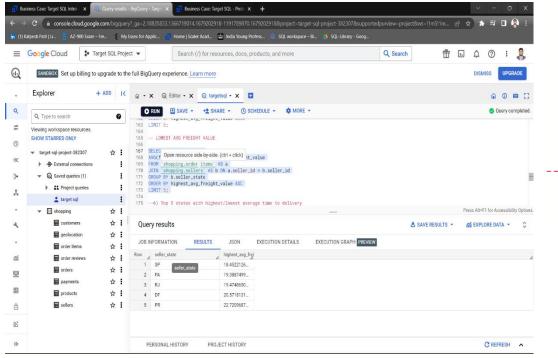


```
# Sort the data to get the following:
--
5.Top 5 states with highest/lowest average freight value - sort in desc/asc limit 5
--HIGHEST AVG FREIGHT VALUE
SELECT b.seller_state,
AVG(freight_value) AS highest_avg_freight_value
FROM `shopping.order items` AS a
JOIN `shopping.sellers` AS b ON a.seller_id = b.seller_id
GROUP BY b.seller_state
ORDER BY highest_avg_freight_value DESC
LIMIT 5;
```



# -- LOWEST AVG FREIGHT VALUE

```
SELECT b.seller_state,
AVG(freight_value) AS highest_avg_freight_value
FROM `shopping.order items` AS a
JOIN `shopping.sellers` AS b ON a.seller_id = b.seller_id
GROUP BY b.seller_state
ORDER BY highest_avg_freight_value ASC
LIMIT 5;
```



6) Top 5 states with highest/lowest average time to delivery

PERSONAL HISTORY

PROJECT HISTORY

C REFRESH A

D

```
--HIGHEST TIME TO DELIVERY
                SELECT b.customer_state,
                ROUND(AVG(DATE_DIFF(order_delivered_customer_date,order_purchase_timestamp,DAY)),2)
                   as highest_avg_delivery
                FROM `shopping.orders` AS a
                JOIN `shopping.customers` AS b
                ON b.customer_id = a.customer_id
                GROUP BY b.customer_state
                ORDER BY highest_avg_delivery desc
                LIMIT 5;
              ss Case: Target SQL Intro X
                                                                       ierył. ga = 2.18835833.1366719014.1679202918-1191789870.16792029188project = target = sql - project. 382307&supported purview = project 8ws = 11 m511 m.... 😢 🎓 🏇 🗾 🔳 🧛
in (1) Kalpesh Patil | Li... Z-900 Exam – Fre... My Essex for Applic...
                                                                                                     Home | Scaler Acad... 🖒 India Young Profess... 🧕 SQL workspace – Bi... 🍐 SQL-Library - Goog
                                                                                                                                                                                                                                                                              Search (/) for resources, docs, products, and more
                                                                                                                                                                                                                                         Q Search
0
               SANDBOX Set up billing to upgrade to the full BigQuery experience. Learn more
                                                                                                                                                                                                                                                                                                  DISMISS UPGRADE
                                                    + ADD K A - X Q Editor - X Q targetsql - X
              Explorer
                                                                                                                                                                                                                                                                                                             @ 0 m ()
                                                                                 THE RUN SAVE - SHARE - SCHEDULE - MORE -
 Q
                                                                                                                                                                                                                                                                                                            Query completed.
              Q Type to search
                                                                              174

-6) Top 5 states with highest/lowest average time to delivery
175

-RIGHEST TIME TO DELIVERY
177

SELECT b.customer_state,
178

ROUND(ANG/CHRILDFIF(order_delivered_customer_date,order_purchase_timestamp,DAY)),2) as highest_avg_delivery
179

FROM *ahopping_orders_AS a

BU JOUN *ahopping_orders_AS b

ON b.customer_id = a.customer_id

**ROUND **Ahopping_orders_AS b

ON b.customer_id = a.customer_id

**ROUND **Ahopping_orders_AS b

ON b.customer_id = a.customer_id

**ROUND **Ahopping_orders_AS b

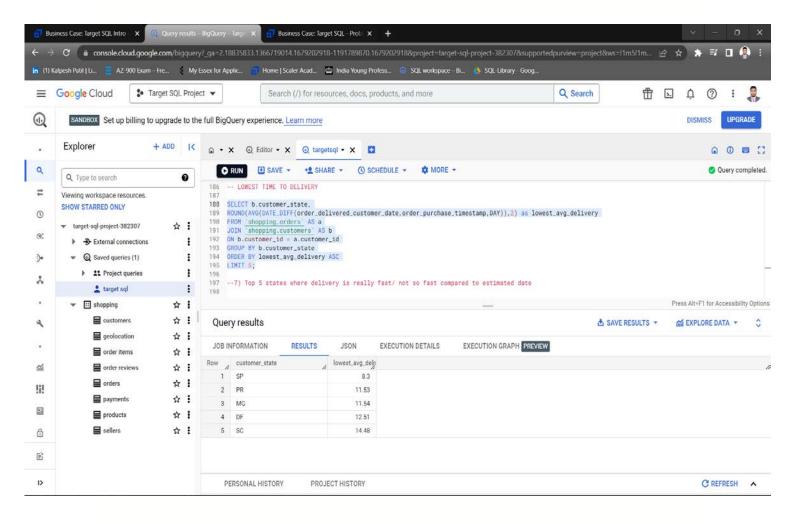
**ROUND
  #
             SHOW STARRED ONLY
  0

▼ target-sql-project-382307

                                                             ☆ :
  80
                ▶ ⊕ External connections
                                                                              GROUP BY b.customer_state
183 ORDER BY highest_avg_delivery desc
184 LIMIT 5;
                3+
                                                                   .
                    Project queries
                                                                    :
  ٨
                                                                              186 -- LOWEST TIME TO DELIVERY
                                                             .
                       1 target sql
                                                                                                                                                                                                                                                                                Press Alt+F1 for Accessibility Options.
                ▼ 🖽 shopping
                                                              ☆ :
                                                             ☆ : Query results
                                                                                                                                                                                                                                                          & SAVE RESULTS ▼ MEXPLORE DATA ▼ C
  ٩
                          geolocation
                                                             ☆ :
                                                                               JOB INFORMATION
                                                                                                                  RESULTS
                                                                                                                                        JSON
                                                                                                                                                         EXECUTION DETAILS
                                                                                                                                                                                              EXECUTION GRAPH PREVIEW
                           d order items
                                                              ☆ :
                                                                                                                           highest_avg_dell
                           order reviews
  sá.
                                                             ☆ :
                                                                                                                                              28.98
                           orders
                                                              ☆ :
  111
                                                                                  2 AP
                                                                                                                                               26.73
                           payments
                                                             ☆ :
  1
                           m products
                                                             ☆ :
                           sellers
                                                             ☆!
  0
  0
```

# -- LOWEST TIME TO DELIVERY

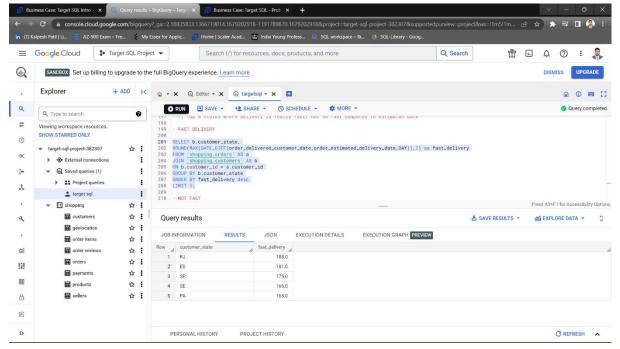
```
SELECT b.customer_state,
ROUND(AVG(DATE_DIFF(order_delivered_customer_date,order_purchase_timestamp,DAY)),2)
as lowest_avg_delivery
FROM `shopping.orders` AS a
JOIN `shopping.customers` AS b
ON b.customer_id = a.customer_id
GROUP BY b.customer_state
ORDER BY lowest_avg_delivery ASC
LIMIT 5;
```



7) Top 5 states where delivery is really fast/ not so fast compared to estimated date

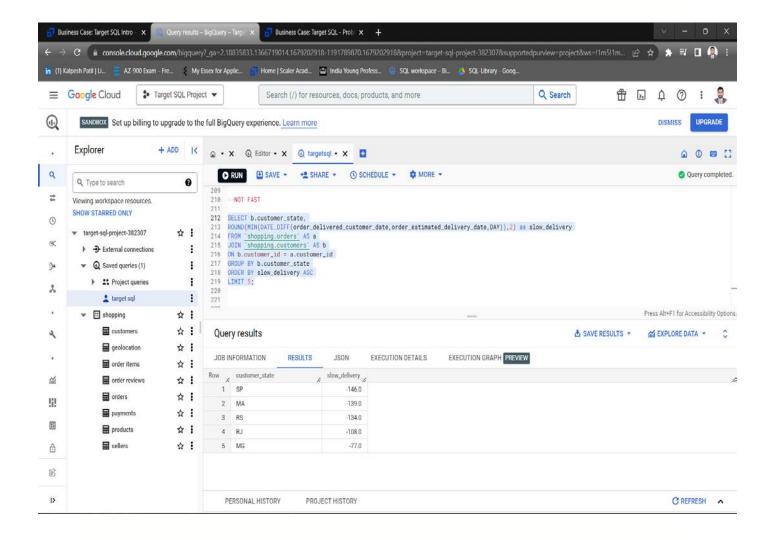
# --FAST DELIVERY

```
SELECT b.customer_state,
ROUND(MAX(DATE_DIFF(order_delivered_customer_date,order_estimated_delivery_date,DAY
)),2) as fast_delivery
FROM `shopping.orders` AS a
JOIN `shopping.customers` AS b
ON b.customer_id = a.customer_id
GROUP BY b.customer_state
ORDER BY fast_delivery desc
LIMIT 5;
```



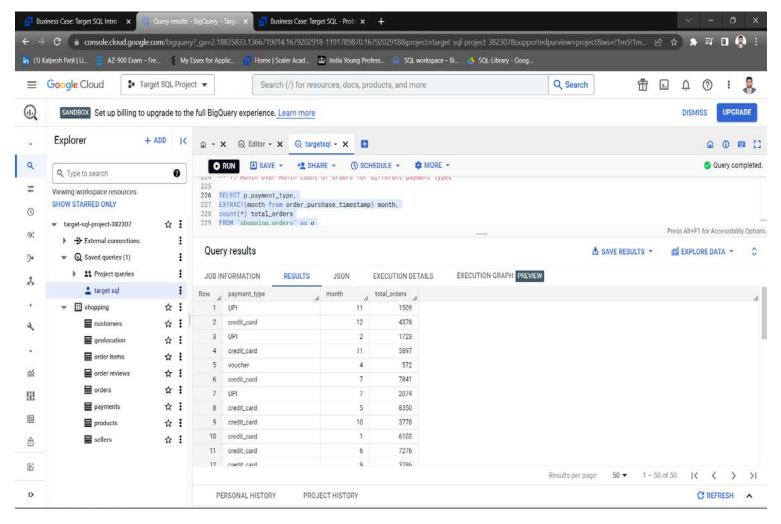
# --NOT FAST

```
SELECT b.customer_state,
ROUND(MIN(DATE_DIFF(order_delivered_customer_date, order_estimated_delivery_date, DAY
)),2) as slow_delivery
FROM `shopping.orders` AS a
JOIN `shopping.customers` AS b
ON b.customer_id = a.customer_id
GROUP BY b.customer_state
ORDER BY slow_delivery ASC
LIMIT 5;
```



```
#6. Payment type analysis:
-- 1) Month over Month count of orders for different payment types

SELECT p.payment_type,
EXTRACT(month from order_purchase_timestamp) month,
count(*) total_orders
FROM `shopping.orders` as o
JOIN `shopping.payments` as p
ON o.order_id=p.order_id
GROUP BY month,p.payment_type
ORDER BY month,p.payment_type;
```



# --2) Count of orders based on the no. of payment installments

```
SELECT p.payment_installments,
count(*) total_orders
FROM `shopping.orders` as o
JOIN `shopping.payments` as p
ON o.order_id=p.order_id
GROUP BY p.payment_installments
ORDER BY p.payment_installments;
```

