1. ‘Data type of columns in a table’

SELECT

COLUMN\_NAME,

DATA\_TYPE

FROM

target-sql-359616.target\_data.INFORMATION\_SCHEMA.COLUMNS

WHERE

TABLE\_SCHEMA = 'target\_data'

AND TABLE\_NAME = 'sellers'

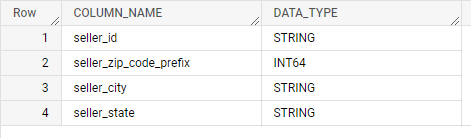
AND COLUMN\_NAME IN ('seller\_id',

'seller\_zip\_code\_prefix',

'seller\_city',

'seller\_state')

Output :



SELECT

COLUMN\_NAME,

DATA\_TYPE

FROM

target-sql-359616.target\_data.INFORMATION\_SCHEMA.COLUMNS

WHERE

TABLE\_SCHEMA = 'target\_data'

AND TABLE\_NAME = 'payments'

AND COLUMN\_NAME IN ('order\_id',

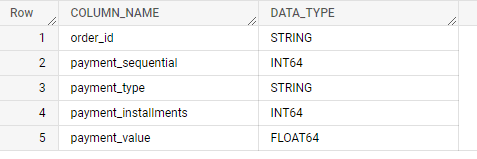
'payment\_sequential',

'payment\_type',

'payment\_installments',

'payment\_value')

Output :



1. ‘Time period for which the data is given’

Query:

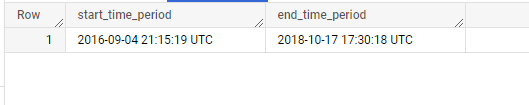
SELECT

min(order\_purchase\_timestamp) as start\_time\_period,max(order\_purchase\_timestamp) as end\_time\_period

FROM

Target-sql-359616.target\_data.orders

Output:



**OR**

SELECT

extract(year from min(order\_purchase\_timestamp)) as start\_time\_period, extract(year from max(order\_purchase\_timestamp)) as end\_time\_period

FROM

target-sql-359616.target\_data.orders

Output:



1. Cities and States covered in the dataset

Query :

WITH

base AS (

SELECT

DISTINCT customer\_city,

customer\_state

FROM

target-sql-359616.target\_data.customers),

base1 AS (

SELECT

DISTINCT seller\_city,

seller\_state

FROM

target-sql-359616.target\_data.sellers),

base2 AS (

SELECT

geolocation\_city,

geolocation\_state

FROM

target-sql-359616.target\_data.geolocation

GROUP BY

geolocation\_city, geolocation\_state)

SELECT

customer\_state as states,

customer\_city as cities

FROM(

SELECT \*

FROM base

UNION DISTINCT

SELECT \*

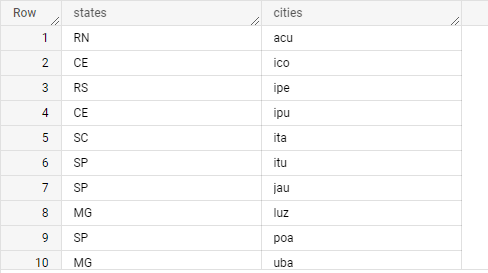
FROM base1

UNION DISTINCT

SELECT \*

FROM base2)

Output:



1. Is there a growing trend on e-commerce in Brazil?

Query:

SELECT

CONCAT(YEAR\_OF\_PURCHASE,'/',MONTH\_OF\_PURCHASE) AS purchase\_year\_month,

YEAR\_OF\_PURCHASE,

MONTH\_OF\_PURCHASE,

WEEK\_OF\_PURCHASE,

ORDER\_COUNT

FROM(

SELECT

EXTRACT(YEAR

FROM

order\_purchase\_timestamp) AS YEAR\_OF\_PURCHASE,

EXTRACT(MONTH FROM order\_purchase\_timestamp) As MONTH\_OF\_PURCHASE,

EXTRACT(WEEK FROM order\_purchase\_timestamp) AS WEEK\_OF\_PURCHASE,

COUNT(order\_id) AS ORDER\_COUNT

FROM

target-sql-359616.target\_data.orders

GROUP BY

EXTRACT(YEAR

FROM

order\_purchase\_timestamp),

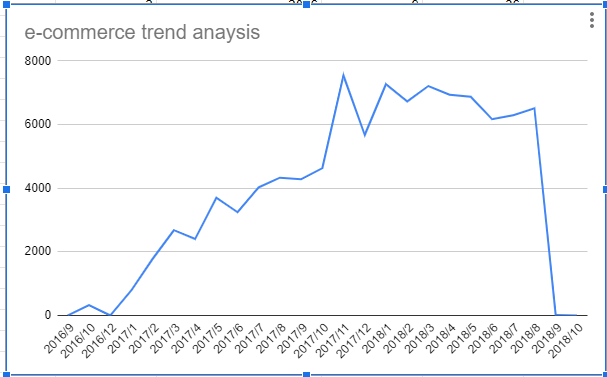
EXTRACT(MONTH FROM order\_purchase\_timestamp),

EXTRACT(WEEK FROM order\_purchase\_timestamp)

)

ORDER BY YEAR\_OF\_PURCHASE, MONTH\_OF\_PURCHASE, WEEK\_OF\_PURCHASE

Output:





Can we see some seasonality with peaks at specific months?

SELECT

EXTRACT(MONTH FROM order\_purchase\_timestamp) As MONTH\_OF\_PURCHASE,

COUNT(order\_id) AS ORDER\_COUNT

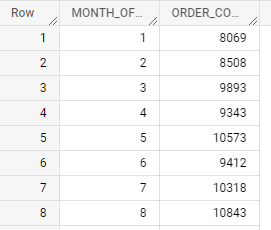
FROM

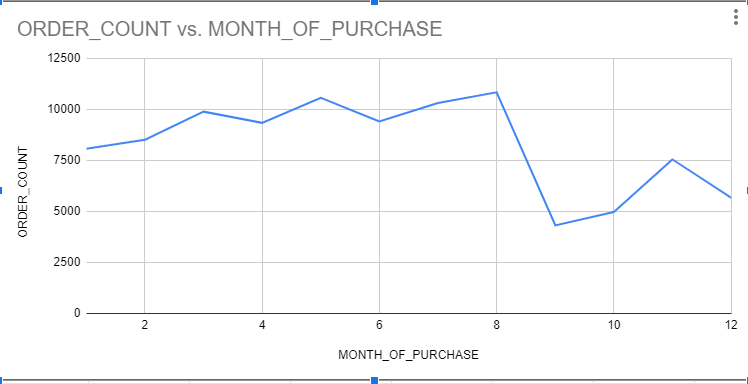
target-sql-359616.target\_data.orders

GROUP BY MONTH\_OF\_PURCHASE

ORDER BY MONTH\_OF\_PURCHASE

OUTPUT :





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

SELECT

CASE

WHEN EXTRACT(hour FROM timestamp(order\_purchase\_timestamp)) BETWEEN 0

AND 6

THEN 'dawn'

WHEN EXTRACT(hour FROM timestamp(order\_purchase\_timestamp)) BETWEEN 7

AND 12

THEN 'morning'

WHEN

EXTRACT(hour FROM timestamp(order\_purchase\_timestamp))

BETWEEN 13

AND 18

THEN 'afternoon'

WHEN

EXTRACT(hour FROM timestamp(order\_purchase\_timestamp))

BETWEEN 19

AND 23

THEN 'night'

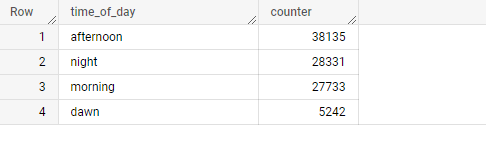
END AS time\_of\_day,

COUNT(DISTINCT order\_id) AS counter

FROM `target-sql-359616.target\_data.orders`

GROUP BY 1

ORDER BY 2 DESC;



1. Get month on month orders by region, states

SELECT g.geolocation\_state,EXTRACT(MONTH FROM o.order\_purchase\_timestamp) AS month, COUNT(o.order\_id) AS ORDER\_COUNT

FROM `target-sql-359616.target\_data.orders` o

JOIN `target-sql-359616.target\_data.customers` c

ON o.customer\_id = c.customer\_id

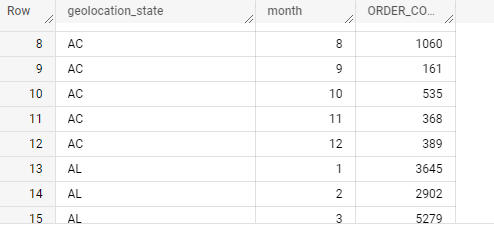
JOIN `target-sql-359616.target\_data.geolocation` g

ON c.customer\_zip\_code\_prefix = g.geolocation\_zip\_code\_prefix

GROUP BY g.geolocation\_state,month

ORDER BY g.geolocation\_state,month

OUTPUT:



1. How are customers distributed in Brazil

SELECT g.geolocation\_state,COUNT(DISTINCT c.customer\_id) AS Customer\_Count

FROM `target-sql-359616.target\_data.customers` c

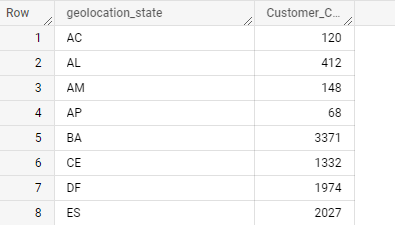
JOIN `target-sql-359616.target\_data.geolocation` g

ON c.customer\_zip\_code\_prefix = g.geolocation\_zip\_code\_prefix

GROUP BY g.geolocation\_state

ORDER BY g.geolocation\_state

OUTPUT:



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

WITH base AS(

SELECT EXTRACT(YEAR FROM o.order\_purchase\_timestamp) AS year, EXTRACT(MONTH FROM o.order\_purchase\_timestamp) AS month,sum(oi.price) AS SUM\_PRICE

FROM `target-sql-359616.target\_data.order\_items` oi

JOIN `target-sql-359616.target\_data.orders` o

ON oi.order\_id = o.order\_id

JOIN `target-sql-359616.target\_data.customers` c

ON o.customer\_id = c.customer\_id

GROUP BY year,month

ORDER BY year,month

),

base1 AS (

SELECT

SUM(CASE WHEN year = 2017 AND month BETWEEN 1 AND 8 THEN SUM\_PRICE else 0 END) AS price\_2017\_yr,

SUM(CASE WHEN year = 2018 AND month BETWEEN 1 AND 8 THEN SUM\_PRICE else 0 END) AS price\_2018\_yr,

FROM base

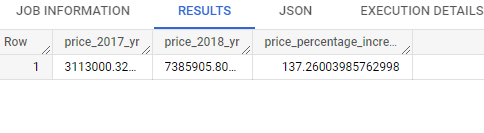
)

#SELECT \* FROM base

SELECT price\_2017\_yr,price\_2018\_yr,(price\_2018\_yr - price\_2017\_yr) / (price\_2017\_yr) \* 100 AS price\_percentage\_increase

FROM base1

OUTPUT:



1. Mean & Sum of price and freight value by customer state

SELECT c.customer\_state, sum(oi.price) AS SUM\_PRICE, sum(oi.price)/COUNT(DISTINCT oi.order\_id) As AVG\_PRICE,

sum(oi.freight\_value) AS SUM\_FV, sum(oi.freight\_value)/COUNT(DISTINCT oi.order\_id) AS AVG\_FV,

FROM `target-sql-359616.target\_data.order\_items` oi

JOIN `target-sql-359616.target\_data.orders` o

ON oi.order\_id = o.order\_id

JOIN `target-sql-359616.target\_data.customers` c

ON o.customer\_id = c.customer\_id

GROUP BY c.customer\_state

ORDER BY c.customer\_state

OUTPUT:



1. Calculate days between purchasing, delivering and estimated delivery

SELECT

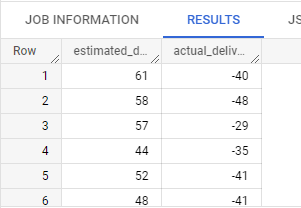
date\_diff(o.order\_estimated\_delivery\_date,o.order\_purchase\_timestamp,DAY) AS estimated\_delivery\_days,

date\_diff(o.order\_delivered\_customer\_date,o.order\_estimated\_delivery\_date,DAY) AS actual\_delivery\_days

FROM `target-sql-359616.target\_data.orders` o

WHERE order\_status = 'delivered'

OUTPUT:



1. Create columns:
   * time\_to\_delivery = order\_purchase\_timestamp-order\_delivered\_customer\_date
   * diff\_estimated\_delivery = order\_estimated\_delivery\_date-order\_delivered\_customer\_date

SELECT

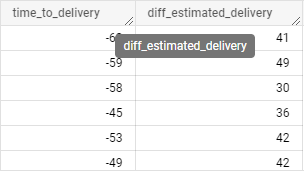
date\_diff(date(o.order\_purchase\_timestamp),date(o.order\_estimated\_delivery\_date), Day) AS time\_to\_delivery ,

date\_diff(date(o.order\_estimated\_delivery\_date),date(o.order\_delivered\_customer\_date), Day) AS diff\_estimated\_delivery ,

FROM `target-sql-359616.target\_data.orders` o

WHERE o.order\_status = 'delivered'

OUTPUT:



1. Group data by state, take mean of freight\_value, time\_to\_delivery, diff\_estimated\_delivery

SELECT

geolocation\_state,geolocation\_city,

SUM(freight\_value)/COUNT(DISTINCT order\_id) AS AVG\_FV,

SUM(time\_to\_delivery)/COUNT(DISTINCT order\_id) AS AVG\_time\_to\_delivery,

SUM(diff\_estimated\_delivery)/COUNT(DISTINCT order\_id) AS AVG\_diff\_estimated\_delivery

FROM(

SELECT o.order\_id,oi.freight\_value,g.geolocation\_state,g.geolocation\_city,

ABS(date\_diff(date(o.order\_purchase\_timestamp),date(o.order\_estimated\_delivery\_date), Day)) AS time\_to\_delivery ,

ABS(date\_diff(date(o.order\_estimated\_delivery\_date),date(o.order\_delivered\_customer\_date), Day)) AS diff\_estimated\_delivery ,

FROM `target-sql-359616.target\_data.orders` o

JOIN `target-sql-359616.target\_data.order\_items` oi

ON o.order\_id = oi.order\_id

JOIN `target-sql-359616.target\_data.customers` c

ON o.customer\_id = c.customer\_id

JOIN `target-sql-359616.target\_data.geolocation` g

ON c.customer\_zip\_code\_prefix = g.geolocation\_zip\_code\_prefix

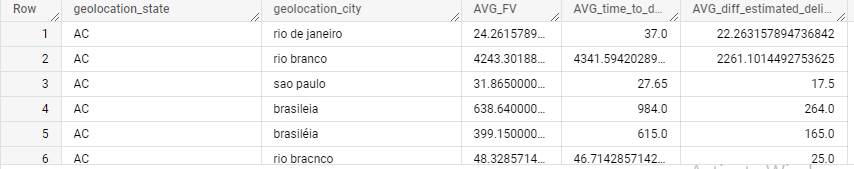
WHERE o.order\_status = 'delivered'

) AS temp

GROUP BY geolocation\_state,geolocation\_city

ORDER BY geolocation\_state

OUTPUT :



1. Sort the data to get the following:
   1. Top 5 states with highest/lowest average freight value - sort in desc/asc limit 5
   2. Top 5 states with highest/lowest average time to delivery
   3. Top 5 states where delivery is really fast/ not so fast compared to estimated date

SELECT

geolocation\_state,

SUM(freight\_value)/COUNT(DISTINCT order\_id) AS AVG\_FV,

SUM(time\_to\_delivery)/COUNT(DISTINCT order\_id) AS AVG\_time\_to\_delivery,

SUM(diff\_estimated\_delivery)/COUNT(DISTINCT order\_id) AS AVG\_diff\_estimated\_delivery

FROM(

SELECT o.order\_id,oi.freight\_value,g.geolocation\_state,g.geolocation\_city,

ABS(date\_diff(date(o.order\_purchase\_timestamp),date(o.order\_estimated\_delivery\_date), Day)) AS time\_to\_delivery ,

ABS(date\_diff(date(o.order\_estimated\_delivery\_date),date(o.order\_delivered\_customer\_date), Day)) AS diff\_estimated\_delivery ,

FROM `target-sql-359616.target\_data.orders` o

JOIN `target-sql-359616.target\_data.order\_items` oi

ON o.order\_id = oi.order\_id

JOIN `target-sql-359616.target\_data.customers` c

ON o.customer\_id = c.customer\_id

JOIN `target-sql-359616.target\_data.geolocation` g

ON c.customer\_zip\_code\_prefix = g.geolocation\_zip\_code\_prefix

WHERE o.order\_status = 'delivered'

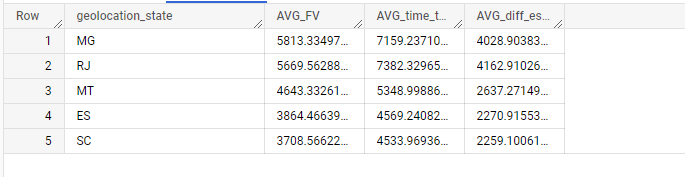
) AS temp

GROUP BY geolocation\_state

ORDER BY AVG\_FV DESC,geolocation\_state ASC

LIMIT 5

OUTPUT:



SELECT

geolocation\_state,

SUM(freight\_value)/COUNT(DISTINCT order\_id) AS AVG\_FV,

SUM(time\_to\_delivery)/COUNT(DISTINCT order\_id) AS AVG\_time\_to\_delivery,

SUM(diff\_estimated\_delivery)/COUNT(DISTINCT order\_id) AS AVG\_diff\_estimated\_delivery

FROM(

SELECT o.order\_id,oi.freight\_value,g.geolocation\_state,g.geolocation\_city,

ABS(date\_diff(date(o.order\_purchase\_timestamp),date(o.order\_estimated\_delivery\_date), Day)) AS time\_to\_delivery ,

ABS(date\_diff(date(o.order\_estimated\_delivery\_date),date(o.order\_delivered\_customer\_date), Day)) AS diff\_estimated\_delivery ,

FROM `target-sql-359616.target\_data.orders` o

JOIN `target-sql-359616.target\_data.order\_items` oi

ON o.order\_id = oi.order\_id

JOIN `target-sql-359616.target\_data.customers` c

ON o.customer\_id = c.customer\_id

JOIN `target-sql-359616.target\_data.geolocation` g

ON c.customer\_zip\_code\_prefix = g.geolocation\_zip\_code\_prefix

WHERE o.order\_status = 'delivered'

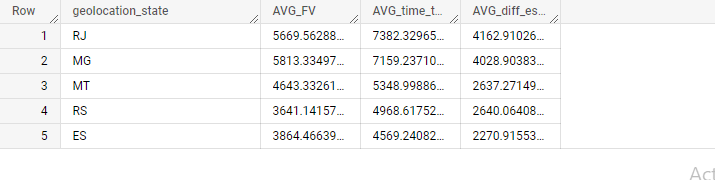
) AS temp

GROUP BY geolocation\_state

ORDER BY AVG\_time\_to\_delivery DESC,geolocation\_state ASC

LIMIT 5

OUTPUT:



SELECT

geolocation\_state,

SUM(freight\_value)/COUNT(DISTINCT order\_id) AS AVG\_FV,

SUM(time\_to\_delivery)/COUNT(DISTINCT order\_id) AS AVG\_time\_to\_delivery,

SUM(diff\_estimated\_delivery)/COUNT(DISTINCT order\_id) AS AVG\_diff\_estimated\_delivery

FROM(

SELECT o.order\_id,oi.freight\_value,g.geolocation\_state,g.geolocation\_city,

ABS(date\_diff(date(o.order\_purchase\_timestamp),date(o.order\_estimated\_delivery\_date), Day)) AS time\_to\_delivery ,

ABS(date\_diff(date(o.order\_estimated\_delivery\_date),date(o.order\_delivered\_customer\_date), Day)) AS diff\_estimated\_delivery ,

FROM `target-sql-359616.target\_data.orders` o

JOIN `target-sql-359616.target\_data.order\_items` oi

ON o.order\_id = oi.order\_id

JOIN `target-sql-359616.target\_data.customers` c

ON o.customer\_id = c.customer\_id

JOIN `target-sql-359616.target\_data.geolocation` g

ON c.customer\_zip\_code\_prefix = g.geolocation\_zip\_code\_prefix

WHERE o.order\_status = 'delivered'

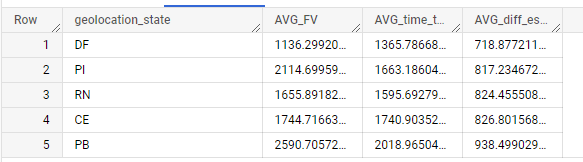
) AS temp

GROUP BY geolocation\_state

ORDER BY AVG\_diff\_estimated\_delivery ASC,geolocation\_state ASC

LIMIT 5

OUTPUT:



1. Payment type analysis:
   1. Month over Month count of orders for different payment types
   2. Distribution of payment installments and count of orders

SELECT

p.payment\_type,

#EXTRACT(YEAR FROM o.order\_purchase\_timestamp) AS year,

EXTRACT(MONTH FROM o.order\_purchase\_timestamp) AS month,

COUNT(o.order\_id) AS ORDER\_COUNT

FROM `target-sql-359616.target\_data.orders` o

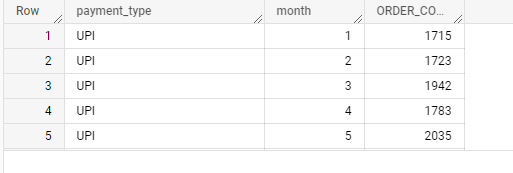
JOIN `target-sql-359616.target\_data.payments` p

ON o.order\_id = p.order\_id

GROUP BY p.payment\_type,month

ORDER BY p.payment\_type,month

OUTPUT:



SELECT

p.payment\_type,

#EXTRACT(YEAR FROM o.order\_purchase\_timestamp) AS year,

#EXTRACT(MONTH FROM o.order\_purchase\_timestamp) AS month,

COUNT(o.order\_id) AS ORDER\_COUNT

FROM `target-sql-359616.target\_data.orders` o

JOIN `target-sql-359616.target\_data.payments` p

ON o.order\_id = p.order\_id

GROUP BY p.payment\_type

ORDER BY p.payment\_type

OUTPUT:

