Business Case Study: Target SQL

Narendra Singh

Target is a globally renowned brand and a prominent retailer in the United States. Target makes itself a preferred shopping destination by offering outstanding value, inspiration, innovation and an exceptional guest experience that no other retailer can deliver.

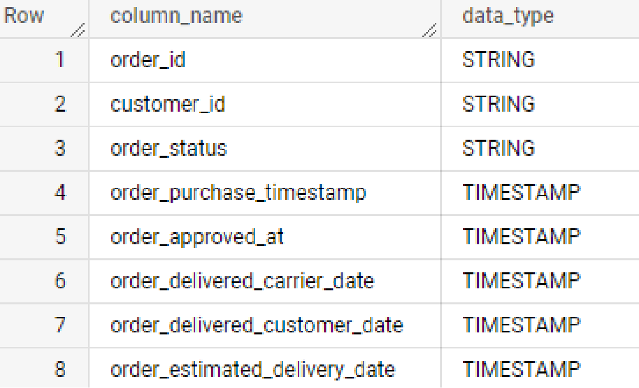
This particular business case focuses on the operations of Target in Brazil and provides insightful information about 100,000 orders placed between 2016 and 2018. The dataset offers a comprehensive view of various dimensions including the order status, price, payment and freight performance, customer location, product attributes, and customer reviews.

By analyzing this extensive dataset, it becomes possible to gain valuable insights into Target's operations in Brazil. The information can shed light on various aspects of the business, such as order processing, pricing strategies, payment and shipping efficiency, customer demographics, product characteristics, and customer satisfaction levels.

Q.1 Import the dataset and do the usual exploratory analysis steps like checking the structure & characteristics of the dataset.

1. The data type of columns in a table?

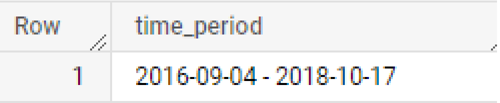
```sql
SELECT column_name, data_type
FROM target_sql.INFORMATION_SCHEMA.COLUMNS
WHERE TABLE_SCHEMA = 'target_sql' AND TABLE_NAME = 'orders';
```



Row	column_name	data_type
1	order_id	STRING
2	customer_id	STRING
3	order_status	STRING
4	order_purchase_timestamp	TIMESTAMP
5	order_approved_at	TIMESTAMP
6	order_delivered_carrier_date	TIMESTAMP
7	order_delivered_customer_date	TIMESTAMP
8	order_estimated_delivery_date	TIMESTAMP

2. Time period for which the data is given?

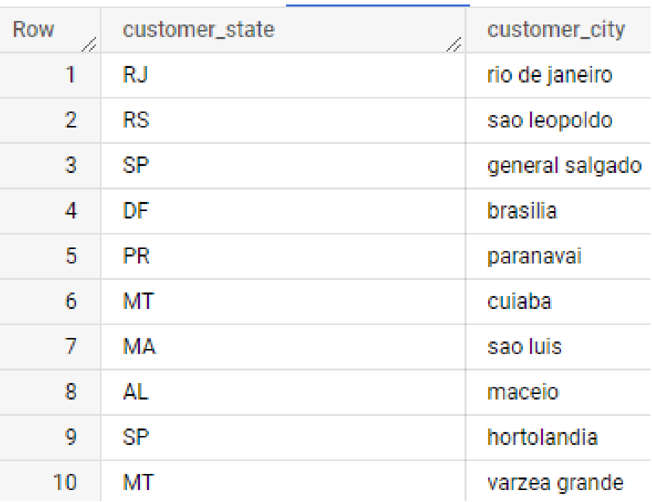
```
SELECT CONCAT(min(EXTRACT(date from order_purchase_timestamp))," - ",
max(EXTRACT(date from order_purchase_timestamp))) as time_period FROM
`target_sql.orders`
```



Row	time_period
1	2016-09-04 - 2018-10-17

3. Cities and States covered in the dataset

```
SELECT DISTINCT c.customer_city, c.customer_state
FROM `target_sql.orders` as o
JOIN `target_sql.customers` as c
ON o.customer_id = c.customer_id
```



Row	customer_state	customer_city
1	RJ	rio de janeiro
2	RS	sao leopoldo
3	SP	general salgado
4	DF	brasilia
5	PR	paranavai
6	MT	cuiaba
7	MA	sao luis
8	AL	maceio
9	SP	hortolandia
10	MT	varzea grande

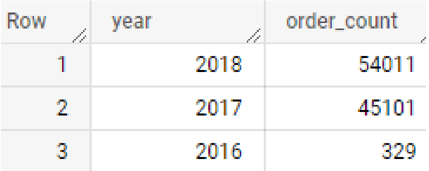
Q.2 In-depth Exploration:

1. Is there a growing trend on e-commerce in Brazil? How can we describe a complete scenario? Can we see some seasonality with peaks at specific months?

   A1. Year wise analysis

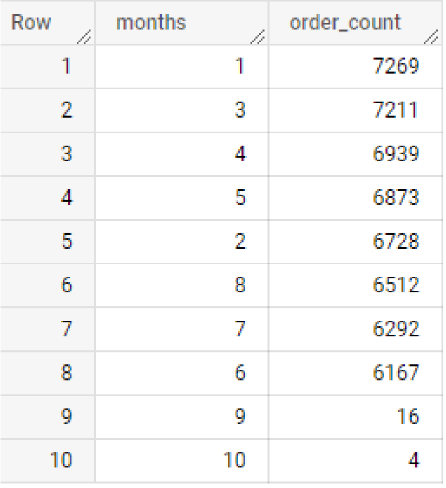
```
SELECT count(order_id) as order_count,
EXTRACT(YEAR from order_purchase_timestamp) as year
FROM `target_sql.orders`
GROUP BY year
```

```
ORDER BY year DESC
LIMIT 10
```



Row	year	order_count
1	2018	54011
2	2017	45101
3	2016	329

```
A2. Month wise analysis in year 2018

SELECT EXTRACT(month from order_purchase_timestamp) as months
count(order_id) as order_count
FROM `target_sql.orders`
GROUP BY months
ORDER BY order_count DESC
LIMIT 10
```



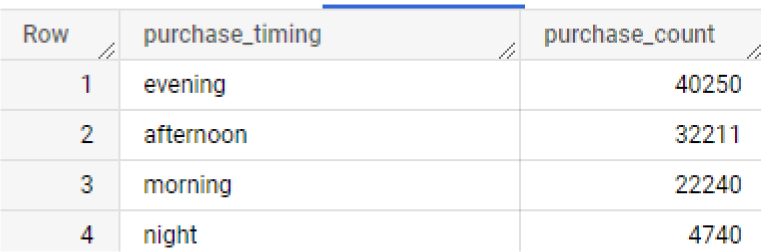
Row	months	order_count
1	1	7269
2	3	7211
3	4	6939
4	5	6873
5	2	6728
6	8	6512
7	7	6292
8	6	6167
9	9	16
10	10	4

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

Assuming:

Morning Time:	6.00 AM - 11.59AM (6.00-11.59)
Afternoon Time:	12.00 PM - 4.59 PM (12.00-16.59)
Evening Time:	5.00PM - 11.59PM (17.00 - 23.59)
Night Time:	12.00 AM - 5.59 AM (00.00 - 5.59)

```
SELECT
CASE
WHEN x.hours BETWEEN 6 AND 11 THEN "morning”
WHEN x.hours BETWEEN 12 AND 16 THEN "afternoon”
WHEN x.hours BETWEEN 17 AND 23 THEN "evening"
WHEN x.hours BETWEEN 0 AND 5 THEN "night"
END as purchase_timing,
count(x.order_id) as purchase_count
FROM (SELECT  EXTRACT(HOUR from order_purchase_timestamp) as hours,
EXTRACT(MINUTE from order_purchase_timestamp) as minutes,
order_id
FROM `target_sql.orders`) AS x
GROUP BY purchase_timing
ORDER BY purchase_count DESC
```

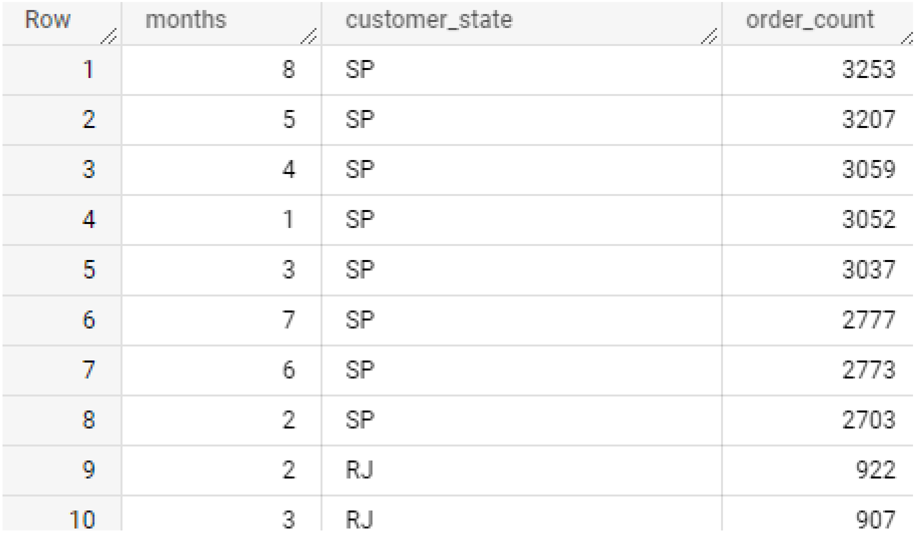


Row	purchase_timing	purchase_count
1	evening	40250
2	afternoon	32211
3	morning	22240
4	night	4740

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

1. Get month on month orders by region, states

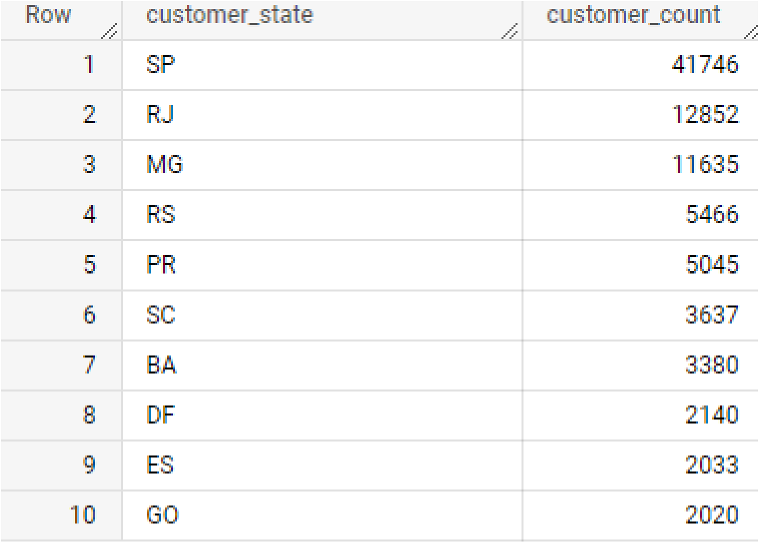
```
 WITH cte AS (SELECT EXTRACT(year from o.order_purchase_timestamp) as
years, EXTRACT(month from o.order_purchase_timestamp) as months,
o.order_id,  c.customer_id, c.customer_city, c.customer_state
FROM  `target_sql.orders` AS o
JOIN `target_sql.customers` AS c
ON o.customer_id = c.customer_id)
SELECT cte.months,cte.customer_state,count(cte.order_id) as
order_count                        FROM cte
WHERE cte.years = 2018 GROUP BY cte.months, cte.customer_state
ORDER BY order_count DESC
LIMIT 15
```



Row	months	customer_state	order_count
1	8	SP	3253
2	5	SP	3207
3	4	SP	3059
4	1	SP	3052
5	3	SP	3037
6	7	SP	2777
7	6	SP	2773
8	2	SP	2703
9	2	RJ	922
10	3	RJ	907

2. How are customers distributed in Brazil

```
SELECT  customer_state, count(customer_id) as customer_count
FROM `target_sql.customers`
GROUP BY customer_state
ORDER BY customer_count DESC
LIMIT 10
```

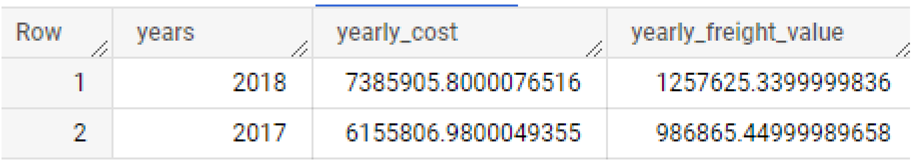


Row	customer_state	customer_count
1	SP	41746
2	RJ	12852
3	MG	11635
4	RS	5466
5	PR	5045
6	SC	3637
7	BA	3380
8	DF	2140
9	ES	2033
10	GO	2020

Q.4 Impact on the Economy: Analyze the money movemented 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 to Aug only)

```
WITH cte AS (SELECT  o.order_id, EXTRACT(date from
o.order_purchase_timestamp) as dates, EXTRACT(year from
o.order_purchase_timestamp) as years, ot.price,
ot.freight_value                 FROM `target_sql.orders` AS o
JOIN `target_sql.order_items` as ot
ON o.order_id = ot.order_id)
SELECT cte.years, SUM(cte.price) as yearly_cost,
sum(cte.freight_value) as yearly_freight_value
FROM cte
WHERE cte.dates BETWEEN "2017-01-01" AND "2018-08-30"
GROUP BY years
```

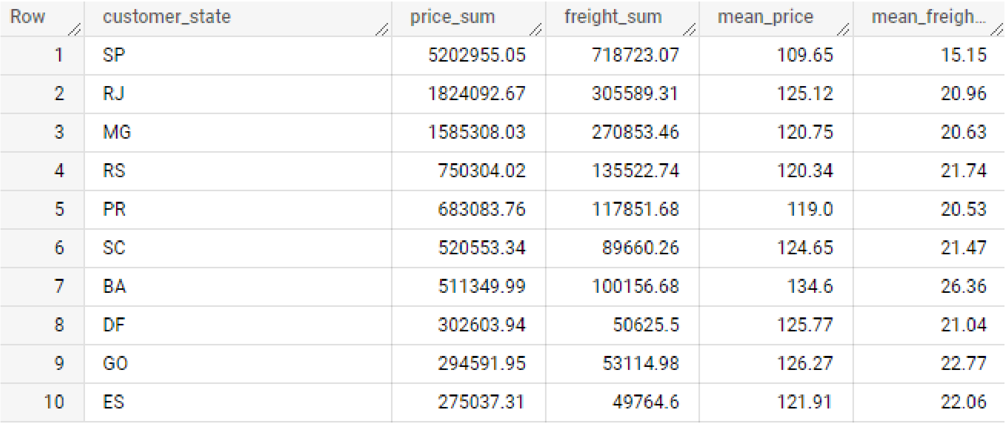


Row	years	yearly_cost	yearly_freight_value
1	2018	7385905.8000076516	1257625.3399999836
2	2017	6155806.9800049355	986865.44999989658

```
with base as (

select * from Ecommerce.orders a

inner join Ecommerce.payments b

on a.order_id = b.order_id

Where  extract(year from a.order_purchase_timestamp) between 2017 and 2018

And extract(month from a.order_purchase_timestamp) between 1 and 8

),

base_2 as ( select extract(year from order_purchase_timestamp) as year, sum(payment_value) as cost from bas group by 1 order by 1 asc

),

base_3 as ( select *, lead(cost, 1) over (order by year) as next_year_cost from base_2)

select *, (next_year_cost - cost)/ cost *100 as per_inc from base_3
```

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

```
WITH cte AS (SELECT  o.order_id, c.customer_id,
c.customer_state, ot.price, ot.freight_value FROM
`target_sql.customers` as c
JOIN `target_sql.orders` as o
ON c.customer_id = o.customer_id
JOIN `target_sql.order_items` as ot
ON o.order_id = ot.order_id)
SELECT  cte.customer_state ,ROUND(sum(cte.price),2) as
price_sum, ROUND(sum(cte.freight_value),2) as freight_sum
,ROUND(Avg(cte.price),2) as mean_price,
ROUND(Avg(cte.freight_value),2) as mean_freight_value
FROM cte
GROUP BY cte.customer_state
ORDER BY price_sum DESC
LIMIT 10
```

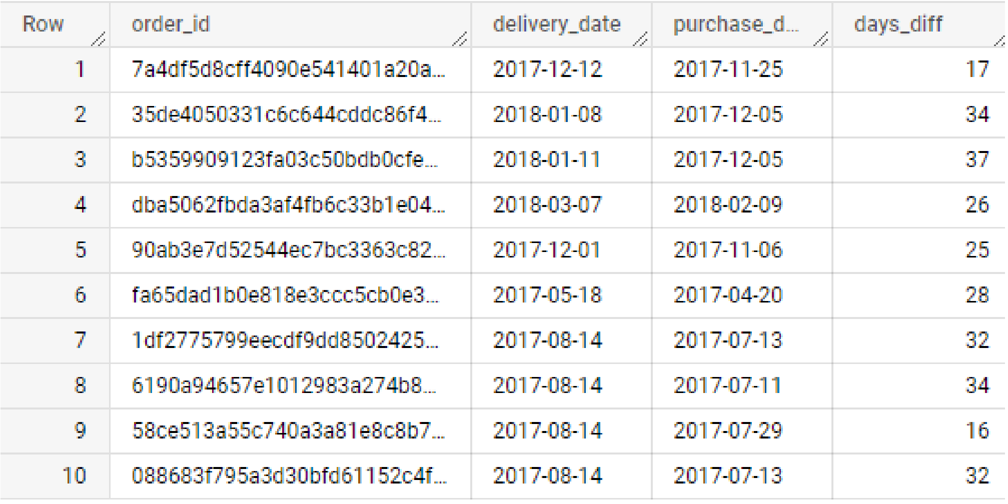


Row	customer_state	price_sum	freight_sum	mean_price	mean_freigh...
1	SP	5202955.05	718723.07	109.65	15.15
2	RJ	1824092.67	305589.31	125.12	20.96
3	MG	1585308.03	270853.46	120.75	20.63
4	RS	750304.02	135522.74	120.34	21.74
5	PR	683083.76	117851.68	119.0	20.53
6	SC	520553.34	89660.26	124.65	21.47
7	BA	511349.99	100156.68	134.6	26.36
8	DF	302603.94	50625.5	125.77	21.04
9	GO	294591.95	53114.98	126.27	22.77
10	ES	275037.31	49764.6	121.91	22.06

Q5. Analysis on sales, freight and delivery time

1. Calculate days between purchasing, delivering and estimated deliver

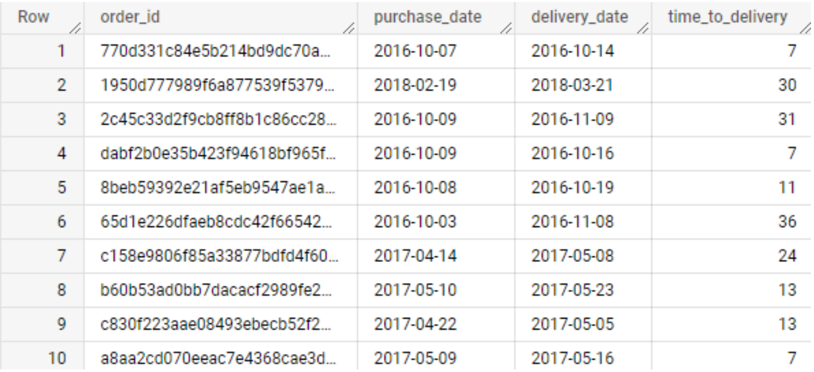
```
SELECT order_id,  EXTRACT(date from
order_estimated_delivery_date) as delivery_date,  EXTRACT(date
from order_purchase_timestamp) as purchase_date, DATE_DIFF(
EXTRACT(date from order_estimated_delivery_date),EXTRACT(date
from order_purchase_timestamp), DAY) AS days_diff
FROM `target_sql.orders`
LIMIT 10
```



Row	order_id	delivery_date	purchase_d...	days_diff
1	7a4df5d8cff4090e541401a20a...	2017-12-12	2017-11-25	17
2	35de4050331c6c644cddc86f4...	2018-01-08	2017-12-05	34
3	b5359909123fa03c50bdb0cfe...	2018-01-11	2017-12-05	37
4	dba5062fbda3af4fb6c33b1e04...	2018-03-07	2018-02-09	26
5	90ab3e7d52544ec7bc3363c82...	2017-12-01	2017-11-06	25
6	fa65dad1b0e818e3ccc5cb0e3...	2017-05-18	2017-04-20	28
7	1df2775799eecdf9dd8502425...	2017-08-14	2017-07-13	32
8	6190a94657e1012983a274b8...	2017-08-14	2017-07-11	34
9	58ce513a55c740a3a81e8c8b7...	2017-08-14	2017-07-29	16
10	088683f795a3d30bfd61152c4f...	2017-08-14	2017-07-13	32

2. Create columns:
    1. time_to_delivery = order_purchase_timestamp-order_delivered_customer_date

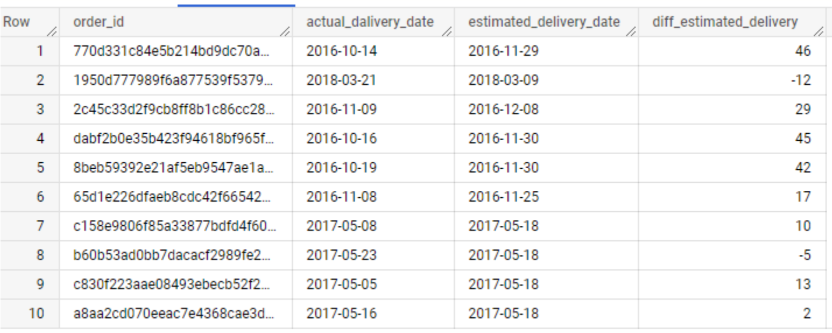
```
SELECT order_id,    EXTRACT(date from
order_purchase_timestamp) as
purchase_date,EXTRACT(date from
order_delivered_customer_date) as delivery_date,
DATE_DIFF( EXTRACT(date from
order_delivered_customer_date),EXTRACT(date from
order_purchase_timestamp), DAY) AS time_to_delivery
FROM `target_sql.orders`
WHERE order_delivered_customer_date IS NOT NULL
LIMIT 10
```



Row	order_id	purchase_date	delivery_date	time_to_delivery
1	770d331c84e5b214bd9dc70a...	2016-10-07	2016-10-14	7
2	1950d777989f6a877539f5379...	2018-02-19	2018-03-21	30
3	2c45c33d2f9cb8ff8b1c86cc28...	2016-10-09	2016-11-09	31
4	dabf2b0e35b423f94618bf965f...	2016-10-09	2016-10-16	7
5	8beb59392e21af5eb9547ae1a...	2016-10-08	2016-10-19	11
6	65d1e226dfaeb8cdc42f66542...	2016-10-03	2016-11-08	36
7	c158e9806f85a33877bdfd4f60...	2017-04-14	2017-05-08	24
8	b60b53ad0bb7dacacf2989fe2...	2017-05-10	2017-05-23	13
9	c830f223aae08493ebecb52f2...	2017-04-22	2017-05-05	13
10	a8aa2cd070eeac7e4368cae3d...	2017-05-09	2017-05-16	7

2. diff_estimated_delivery = order_estimated_delivery_date-order_delivered_customer_date

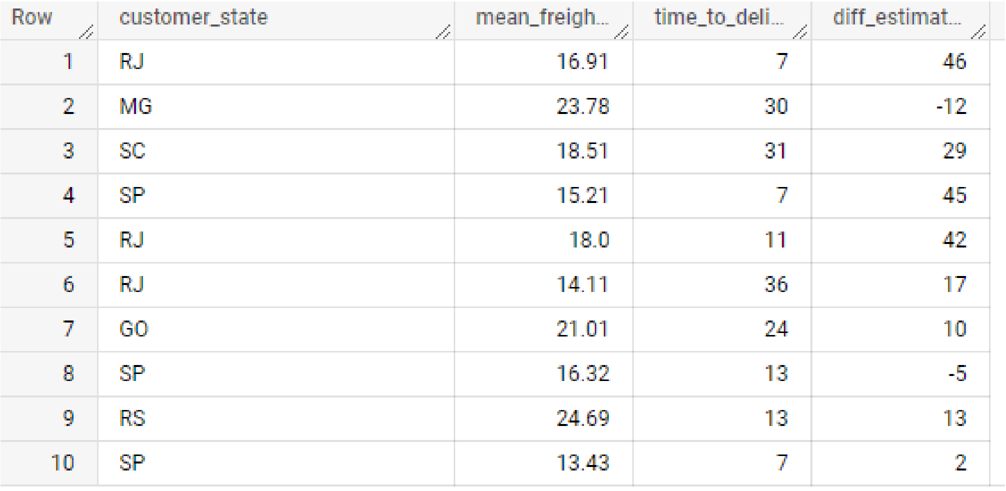
```
SELECT order_id,    EXTRACT(date from
order_delivered_customer_date) as
actual_dalivery_date,EXTRACT(date from
order_estimated_delivery_date) as
estimated_delivery_date, DATE_DIFF( EXTRACT(date from
order_estimated_delivery_date),EXTRACT(date from
order_delivered_customer_date), DAY) AS
diff_estimated_delivery              FROM
`target_sql.orders`
WHERE order_delivered_customer_date IS NOT NUL
LIMIT 10
```



Row	order_id	actual_dalivery_date	estimated_delivery_date	diff_estimated_delivery
1	770d331c84e5b214bd9dc70a...	2016-10-14	2016-11-29	46
2	1950d777989f6a877539f5379...	2018-03-21	2018-03-09	-12
3	2c45c33d2f9cb8ff8b1c86cc28...	2016-11-09	2016-12-08	29
4	dabf2b0e35b423f94618bf965f...	2016-10-16	2016-11-30	45
5	8beb59392e21af5eb9547ae1a...	2016-10-19	2016-11-30	42
6	65d1e226dfaeb8cdc42f66542...	2016-11-08	2016-11-25	17
7	c158e9806f85a33877bdfd4f60...	2017-05-08	2017-05-18	10
8	b60b53ad0bb7dacacf2989fe2...	2017-05-23	2017-05-18	-5
9	c830f223aae08493ebecb52f2...	2017-05-05	2017-05-18	13
10	a8aa2cd070eeac7e4368cae3d...	2017-05-16	2017-05-18	2

3. Group data by state, take mean of freight_value, time_to_delivery, diff_estimated_delivery

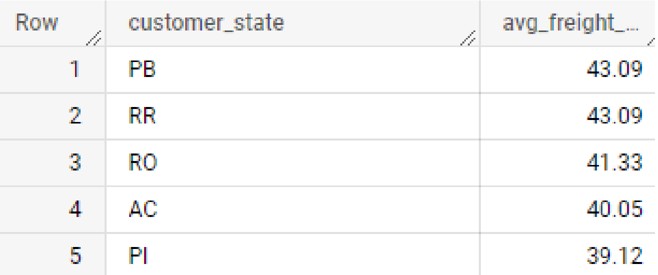
```
WITH cte AS (SELECT  o.order_id, c.customer_id,
c.customer_state, ot.price, ot.freight_value, DATE_DIFF(
EXTRACT(date from o.order_estimated_delivery_date),EXTRACT(date
from o.order_delivered_customer_date), DAY) AS
diff_estimated_delivery,       DATE_DIFF( EXTRACT(date from
order_delivered_customer_date),EXTRACT(date from
order_purchase_timestamp), DAY) AS time_to_delivery
FROM `target_sql.customers` as c
JOIN `target_sql.orders` as o
ON c.customer_id = o.customer_id
JOIN `target_sql.order_items` as ot
ON o.order_id = ot.order_id
WHERE order_delivered_customer_date IS NOT NULL)
SELECT  cte.customer_state, ROUND(Avg(cte.freight_value),2) as
mean_freight_value,cte.time_to_delivery
,cte.diff_estimated_delivery
FROM cte
GROUP BY cte.customer_state,cte.time_to_delivery
,cte.diff_estimated_delivery
LIMIT 10
```



Row	customer_state	mean_freigh...	time_to_deli...	diff_estimat...
1	RJ	16.91	7	46
2	MG	23.78	30	-12
3	SC	18.51	31	29
4	SP	15.21	7	45
5	RJ	18.0	11	42
6	RJ	14.11	36	17
7	GO	21.01	24	10
8	SP	16.32	13	-5
9	RS	24.69	13	13
10	SP	13.43	7	2

4. Sort the data to get the following:
    1. Top 5 states with highest/lowest average freight value - sort in desc/asc limit 5

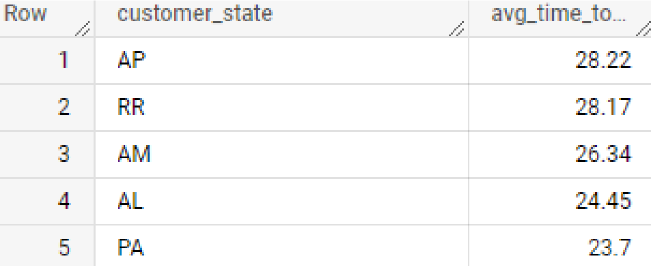
```
WITH cte AS (SELECT  o.order_id, c.customer_id,
c.customer_state, ot.price, ot.freight_value, DATE_DIFF(
EXTRACT(date from o.order_estimated_delivery_date),EXTRACT(date
from o.order_delivered_customer_date), DAY) AS
diff_estimated_delivery,       DATE_DIFF( EXTRACT(date from
order_delivered_customer_date),EXTRACT(date from
order_purchase_timestamp), DAY) AS time_to_delivery
FROM `target_sql.customers` as c
JOIN `target_sql.orders` as o
ON c.customer_id = o.customer_id
JOIN `target_sql.order_items` as ot
ON o.order_id = ot.order_id
WHERE order_delivered_customer_date IS NOT NULL)
SELECT  cte.customer_state, ROUND(Avg(cte.freight_value),2)  as
avg_freight_value
FROM cte
GROUP BY cte.customer_state
ORDER BY avg_freight_value  DESC
LIMIT 5
```



Row	customer_state	avg_freight_...
1	PB	43.09
2	RR	43.09
3	RO	41.33
4	AC	40.05
5	PI	39.12

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

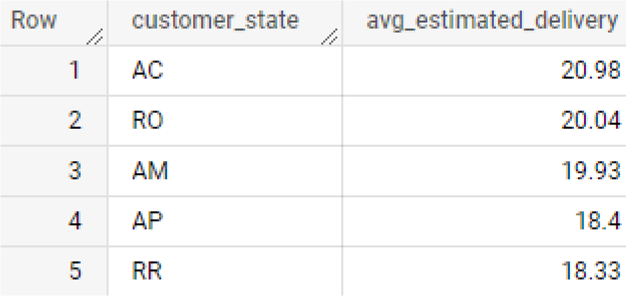
```
 WITH cte AS (SELECT  o.order_id, c.customer_id,
c.customer_state, ot.price, ot.freight_value, DATE_DIFF(
EXTRACT(date from o.order_estimated_delivery_date),EXTRACT(date
from o.order_delivered_customer_date), DAY) AS
diff_estimated_delivery,       DATE_DIFF( EXTRACT(date from
order_delivered_customer_date),EXTRACT(date from
order_purchase_timestamp), DAY) AS time_to_delivery
FROM `target_sql.customers` as c
JOIN `target_sql.orders` as o
ON c.customer_id = o.customer_id
JOIN `target_sql.order_items` as ot
ON o.order_id = ot.order_id
WHERE order_delivered_customer_date IS NOT NULL)
SELECT  cte.customer_state, ROUND(Avg(cte.time_to_delivery),2)
as avg_time_to_delivery
FROM cte
GROUP BY cte.customer_state
ORDER BY avg_time_to_delivery   DESC
LIMIT 5
```



Row	customer_state	avg_time_to...
1	AP	28.22
2	RR	28.17
3	AM	26.34
4	AL	24.45
5	PA	23.7

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

```
WITH cte AS (SELECT  o.order_id, c.customer_id,
c.customer_state, ot.price, ot.freight_value, DATE_DIFF(
EXTRACT(date from o.order_estimated_delivery_date),EXTRACT(date
from o.order_delivered_customer_date), DAY) AS
diff_estimated_delivery,       DATE_DIFF( EXTRACT(date from
order_delivered_customer_date),EXTRACT(date from
order_purchase_timestamp), DAY) AS time_to_delivery
FROM `target_sql.customers` as c
JOIN `target_sql.orders` as o
ON c.customer_id = o.customer_id
JOIN `target_sql.order_items` as ot
ON o.order_id = ot.order_id
WHERE order_delivered_customer_date IS NOT NULL)
SELECT  cte.customer_state,
ROUND(Avg(diff_estimated_delivery),2)  as
avg_estimated_delivery    FROM cte
GROUP BY cte.customer_state
ORDER BY avg_estimated_delivery  DESC
LIMIT 5
```



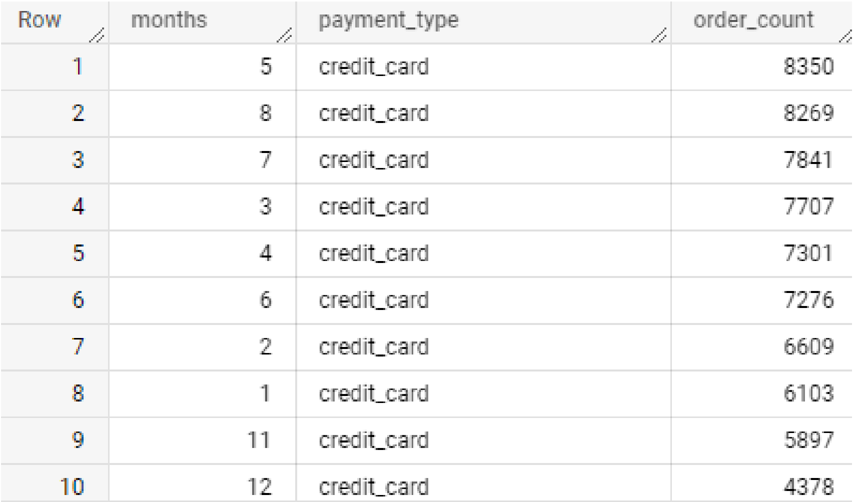
Row	customer_state	avg_estimated_delivery
1	AC	20.98
2	RO	20.04
3	AM	19.93
4	AP	18.4
5	RR	18.33

Q.6 Payment type analysis:

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

```
SELECT   EXTRACT(month from o.order_purchase_timestamp) as
months, p.payment_type, count(o.order_id) as order_count
FROM `target_sql.orders`as o
JOIN `target_sql.payments` as p
ON o.order_id = p.order_id
GROUP BY p.payment_type, months
```

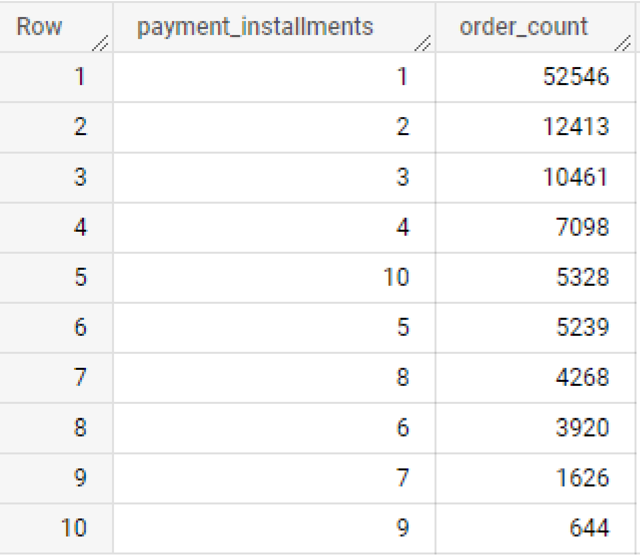
```
ORDER BY order_count DESC
LIMIT 15
```



Row	months	payment_type	order_count
1	5	credit_card	8350
2	8	credit_card	8269
3	7	credit_card	7841
4	3	credit_card	7707
5	4	credit_card	7301
6	6	credit_card	7276
7	2	credit_card	6609
8	1	credit_card	6103
9	11	credit_card	5897
10	12	credit_card	4378

2. Distribution of payment installments and count of orders

```
SELECT   payment_installments, count(o.order_id) as order_count
FROM `target_sql.orders`as o
JOIN `target_sql.payments` as p
ON o.order_id = p.order_id
GROUP BY payment_installments
ORDER BY order_count DESC
LIMIT 10
```



Row	payment_installments	order_count
1	1	52546
2	2	12413
3	3	10461
4	4	7098
5	10	5328
6	5	5239
7	8	4268
8	6	3920
9	7	1626
10	9	644

Actionable Insights

In Q.2

1. It can be seen from the output that the most orders (54011) were placed by customers in the year 2018 and the least number of orders were placed by customers in September(9) and October(10).
2. In the evening, the customer placed the maximum number of orders (40250).

In Q.3

1. The SP state of Brazil has the highest number of orders (41746), as well as the highest number of customers (41746).

In Q.4

1. Approx. An increase of 20% in the order cost between January and August of 2018 compared with 2017.

In Q.5

1. Acording to data, some product delivery takes more than a month.

In Q.6

1. Most of the time, customers use credit cards to make purchases and make one-time payments.

Recommendations:

1. The most important concern is delivery time, We will have to reduce the delivery time which has time taken more than a month.
2. We must change some selling strategies for SC, PI, RJ, and many more which have fewer orders
3. Somehow we have to reduce freight value to make more profit. It may happen because of delivering the product takes lots of time. For example, state PI has 409.68 mean freight value and the estimated delivery time is 20 days