Target SQL – Business Case

Q 1.1

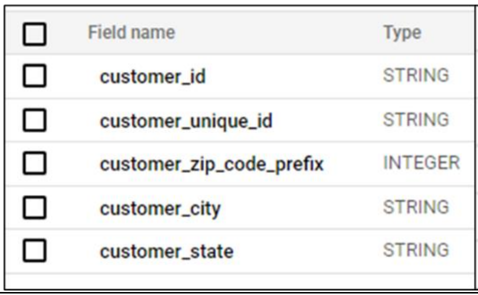
Data type of all columns in the Customers table

Query:

```
SELECT column_name, data_type
FROM `theta-style-410310.Target_SQL.INFORMATION_SCHEMA.COLUMNS`
WHERE table_name = "customers"
```

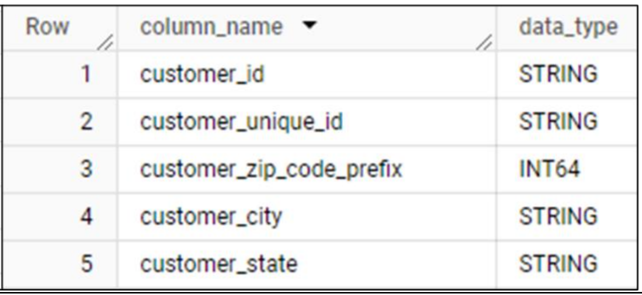
Output:

Method 1



□	Field name	Type
□	customer_id	STRING
□	customer_unique_id	STRING
□	customer_zip_code_prefix	INTEGER
□	customer_city	STRING
□	customer_state	STRING

Method 2



Row	column_name	data_type
1	customer_id	STRING
2	customer_unique_id	STRING
3	customer_zip_code_prefix	INT64
4	customer_city	STRING
5	customer_state	STRING

Solution:

Datatype can be found using two ways -

1) Simply by clicking on the respective table in the side tab. This will open up a new tab with datatypes listed for all the columns.
2) Using the above query that uses the concept of Information schema.

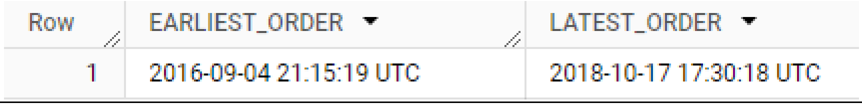
Q 1.2

Get the time range between which the orders were placed.

Query:

```
SELECT MIN(order_purchase_timestamp) AS EARLIEST_ORDER,
MAX(order_purchase_timestamp) AS LATEST_ORDER
FROM `Target_SQL.orders`
```

Output:



Row	EARLIEST_ORDER	LATEST_ORDER
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC

Insights:

1) First order from Target in Brazil was placed on 4th Sept 2016 at 9pm UTC (Brazil time zone is UTC – 3:00:00).
2) Latest order was placed on 17th Oct 2018 at 5.30pm UTC.

Recommendations:

None

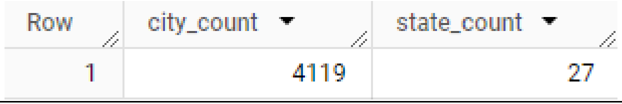
Q 1.3

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

Query:

```
SELECT COUNT(DISTINCT C.customer_city) as city_count,
COUNT(DISTINCT c.customer_state) as state_count
FROM `Target_SQL.customers` C
INNER JOIN
`Target_SQL.orders` O
on C.customer_id = O.customer_id
```

Output:



Row	city_count	state_count
1	4119	27

Insight:

Between 2016 and 2018, Target received orders from 4119 cities that are spread across 27 states.

Recommendations:

None

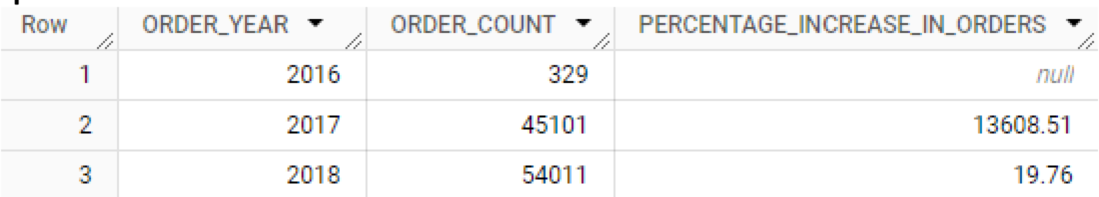
Q 2.1

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

Query:

```
SELECT ORDER_YEAR, ORDER_COUNT,
ROUND(100*(ORDER_COUNT - (LAG(ORDER_COUNT, 1) OVER(ORDER BY
ORDER_YEAR)))/(LAG(ORDER_COUNT, 1) OVER(ORDER BY ORDER_YEAR)),2)
AS PERCENTAGE_INCREASE_IN_ORDERS
FROM
(
  SELECT EXTRACT(YEAR FROM order_purchase_timestamp) as
ORDER_YEAR, COUNT(ORDER_ID) AS ORDER_COUNT,
  FROM `Target_SQL.orders`
  GROUP BY 1
  ORDER BY 1
)
ORDER BY 1
```

Output:



Row	ORDER_YEAR	ORDER_COUNT	PERCENTAGE_INCREASE_IN_ORDERS
1	2016	329	null
2	2017	45101	13608.51
3	2018	54011	19.76

Insights:

1) The order count has increased consistently year on year from 2016 to 2018.

2) From 2016 to 2017, the order count increased 137 times. This slightly unusual number can be attributed to the fact that the first ever order to be placed in 2016 was only in the 1st week of September.
3) The order count increased by 19.76% from 2017 to 2018. That is indeed a good improvement.
4) The order count includes those orders that not just delivered, but also created, shipped, approved, cancelled, invoiced, under process and in some cases are unavailable.

Recommendations:

No recommendations based on the orders table alone. More recommendations provided in future questions.

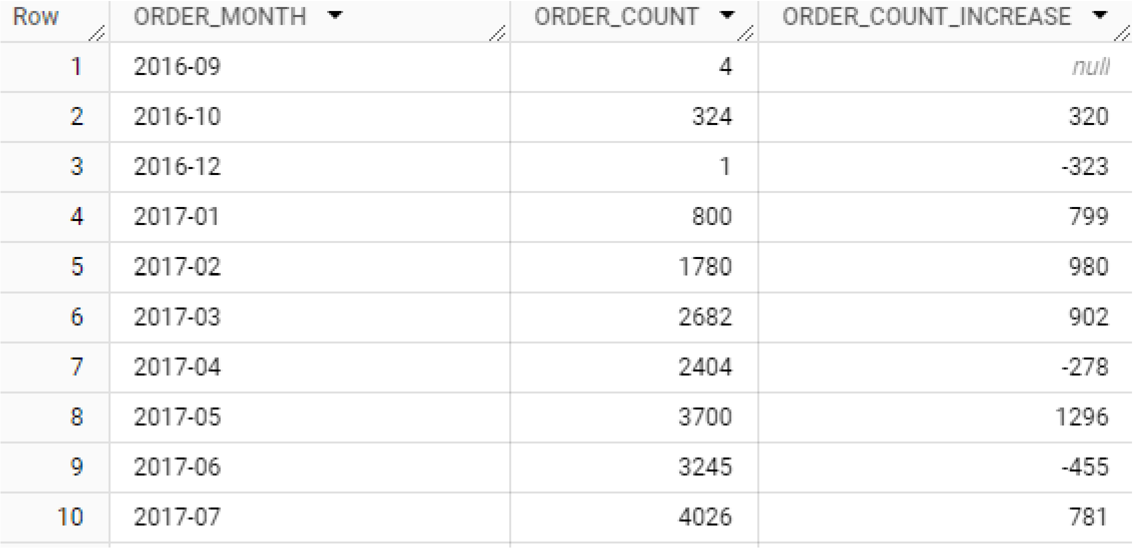
Q 2.2

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

Query:

```
SELECT ORDER_MONTH, ORDER_COUNT,
(ORDER_COUNT - LAG(ORDER_COUNT, 1) OVER(ORDER BY ORDER_MONTH)) AS
ORDER_COUNT_INCREASE
FROM
(
  SELECT FORMAT_DATE("%Y-%m", order_purchase_timestamp) as
ORDER_MONTH, COUNT(ORDER_ID) AS ORDER_COUNT,
  FROM `Target_SQL.orders`
  GROUP BY 1
  ORDER BY 1
)
ORDER BY 1
```

Output (showing only 10 rows):



Row	ORDER_MONTH	ORDER_COUNT	ORDER_COUNT_INCREASE
1	2016-09	4	null
2	2016-10	324	320
3	2016-12	1	-323
4	2017-01	800	799
5	2017-02	1780	980
6	2017-03	2682	902
7	2017-04	2404	-278
8	2017-05	3700	1296
9	2017-06	3245	-455
10	2017-07	4026	781

Insights:

1) The order count in general has been consistently increasing month on month.

2) No order has been placed in Nov 2016 which seems like an anomaly after 324 orders in previous month (Oct 2016).
3) The highest order count has been observed around Christmas and New year (Nov 2017, Jan 2018)
4) Summer months in 2018 have good order count (consistently more than 6000 from March to August).
5) Month on Month order count has mostly been positive except in some cases.
6) Sept 2018 and Oct 2018 has been the worst in terms of order count as it decreased by 400 times.

Recommendation:
Even though December months seem to be having a good order count, there has been a dip in both 2016 and 2017 when compared with November. This could be attributed to customers wanting to spend more on travel and vacation and not much on retail. In order to make the most of the festive season, a seasonal discount would probably help drive the orders up. (The total amount spent has also decreased - referenced from Payments table).

Q 2.3

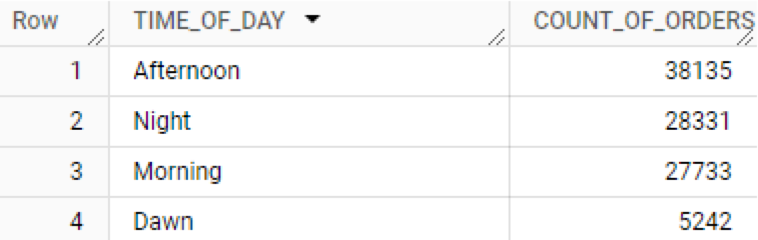
During what time of the day, do the Brazilian customers mostly place their orders? (Dawn, Morning, Afternoon or Night)
(0-6 hrs : Dawn) (7-12 hrs : Mornings) (13-18 hrs : Afternoon) (19-23 hrs : Night)

Assumption:
Brazil is assumed to be in the same time zone as UTC. That is, it is not 3 hours behind UTC.

Query:

```
WITH A AS
(SELECT "Dawn" as TIME_OF_DAY, COUNT(ORDER_ID) AS COUNT_OF_ORDERS
FROM `Target_SQL.orders`
WHERE FORMAT_DATE("%H:%m", order_purchase_timestamp) BETWEEN
"00:00:00" AND "06:59:59"),

B AS
(SELECT "Morning" as TIME_OF_DAY, COUNT(ORDER_ID) AS
COUNT_OF_ORDERS
FROM `Target_SQL.orders`
WHERE FORMAT_DATE("%H:%m", order_purchase_timestamp) BETWEEN
"07:00:00" AND "12:59:59"),

C AS
(SELECT "Afternoon" as TIME_OF_DAY, COUNT(ORDER_ID) AS
COUNT_OF_ORDERS
FROM `Target_SQL.orders`
WHERE FORMAT_DATE("%H:%m", order_purchase_timestamp) BETWEEN
"13:00:00" AND "18:59:59"),

D AS
(SELECT "Night" as TIME_OF_DAY, COUNT(ORDER_ID) AS COUNT_OF_ORDERS
FROM `Target_SQL.orders`
WHERE FORMAT_DATE("%H:%m", order_purchase_timestamp) BETWEEN
"19:00:00" AND "23:59:59")

SELECT *
FROM `A`
UNION ALL
SELECT *
FROM `B`
UNION ALL
SELECT *
FROM `C`
UNION ALL
SELECT *
FROM `D`
ORDER BY 2 DESC
```

Output:



Row	TIME_OF_DAY ▾	COUNT_OF_ORDERS
1	Afternoon	38135
2	Night	28331
3	Morning	27733
4	Dawn	5242

Insights:

1) The order count is highest during afternoon and next highest during night and morning.
2) The order count is least during dawn (as expected).

Recommendation:

Since customers are usually asleep during dawn, this time could be utilised for website and server maintenance activities.

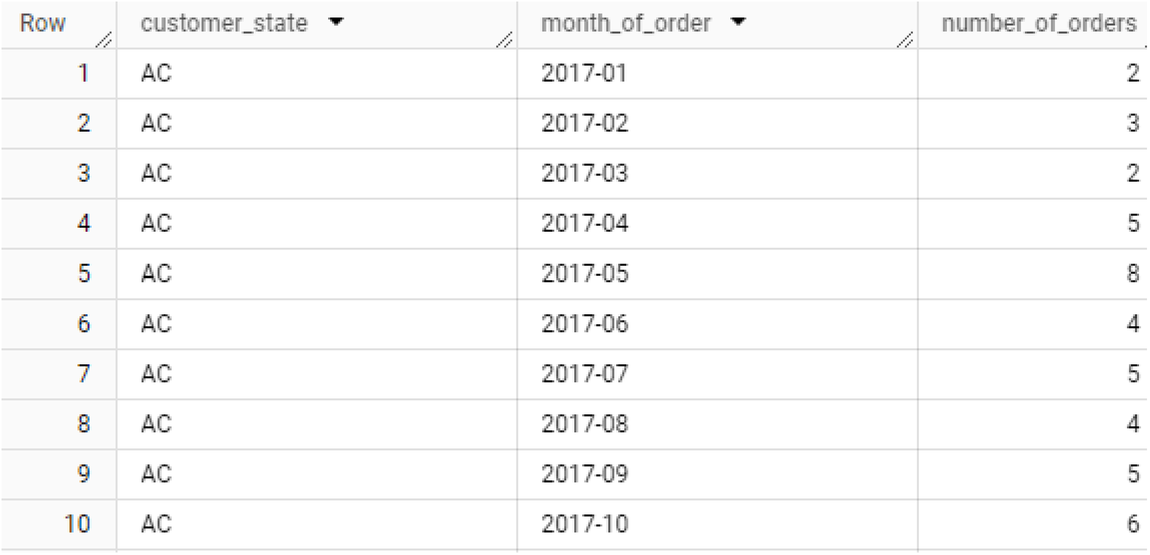
Q 3.1

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

Query:

```
SELECT c.customer_state, FORMAT_DATE("%Y-%m",
order_purchase_timestamp) as month_of_order,
COUNT(o.order_id) as number_of_orders
FROM `Target_SQL.orders` o
JOIN
`Target_SQL.customers` c
ON o.customer_id = c.customer_id
GROUP BY  1, 2
ORDER BY 1, 2
```

Output:



Row	customer_state	month_of_order	number_of_orders
1	AC	2017-01	2
2	AC	2017-02	3
3	AC	2017-03	2
4	AC	2017-04	5
5	AC	2017-05	8
6	AC	2017-06	4
7	AC	2017-07	5
8	AC	2017-08	4
9	AC	2017-09	5
10	AC	2017-10	6

Insights:

1) SP state has the highest order count in 2017 and 2018. In fact, top 16 positions belong to the state. SP has 21% of Brazil's total population living in it, hence the large number of orders.

2) States like PI, PB, RR, AM, TO, PR, AP, MS, SC have only one order placed in several months.

Recommendation:

Target can focus more on those states where the orders have been less than 3 in a month. There are 14 such states and Target can take up campaigns, increase advertising budgets and open more stores in these states.

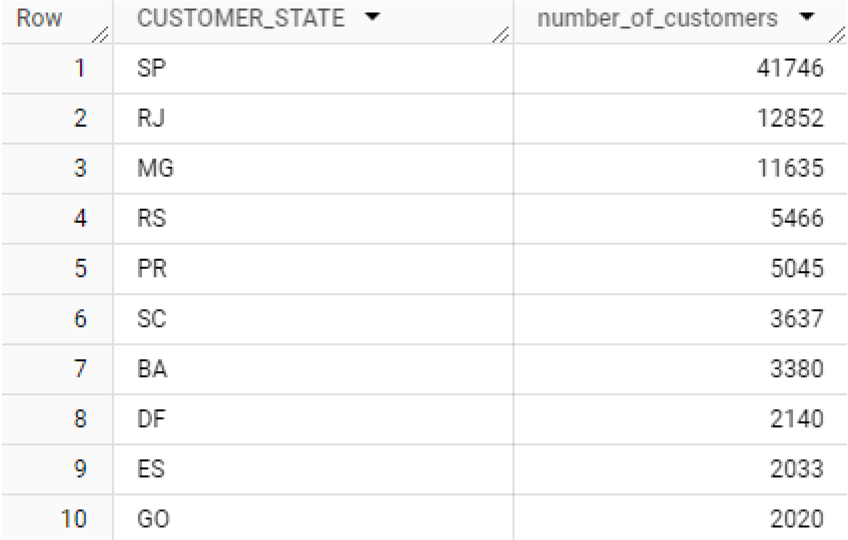
Q 3.2

How are the customers distributed across all the states?

Query:

```
SELECT CUSTOMER_STATE, COUNT(CUSTOMER_ID) AS number_of_customers
FROM `Target_SQL.customers`
GROUP BY 1
ORDER BY 2 DESC
```

Output:



Row	CUSTOMER_STATE	number_of_customers
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

Insights:

1) The top 3 states with highest number of customers are SP, RJ and MG.
2) Incidentally, the above 3 states have had instances less where the number of orders placed in a single month is less than 5 (that too in the year 2018)
3) States RR, AP, AC AND AM have fewer than 150 customers.

Recommendations:

1) Target needs to heavily focus on a state like SP, which contributes 34% to Brazil's GDP, especially with a population of 44 million. It's a state where a lot of urban population resides, hence online social media campaigns can be run to gain customer base.
2) Target needs to focus more on gaining a good customer base in those states where the number of customers is very less compared to its population. Even though a state like RR has much of its territory as part of the amazon rain forest, the population is 5 lacs. There is a good scope for increasing customers.

Q 4.1

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

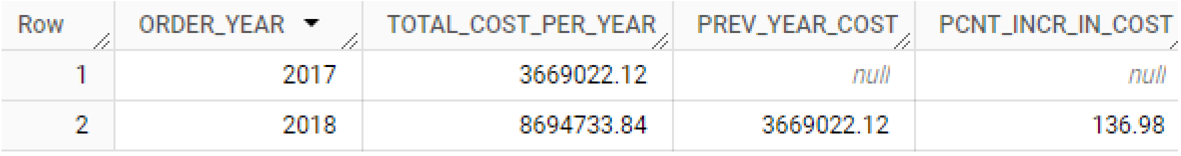
Assumption:

Cost of order calculated as total of months Jan to Aug and the yearly costs are calculated and compared.

Query:

```
WITH CTE AS (
  SELECT EXTRACT(YEAR FROM O.order_purchase_timestamp) as
ORDER_YEAR,
  ROUND(SUM(P.PAYMENT_VALUE),2) as TOTAL_COST_PER_YEAR
  FROM `Target_SQL.payments` AS P
  INNER JOIN
  `Target_SQL.orders` AS O
  ON P.ORDER_ID = O.ORDER_ID
  WHERE EXTRACT(MONTH FROM O.order_purchase_timestamp) BETWEEN 1
AND 8
  GROUP BY 1
  ORDER BY 1
)

SELECT ORDER_YEAR, TOTAL_COST_PER_YEAR,
LAG(TOTAL_COST_PER_YEAR, 1) OVER(ORDER BY ORDER_YEAR) AS
PREV_YEAR_COST,
ROUND(100*((TOTAL_COST_PER_YEAR)-(LAG(TOTAL_COST_PER_YEAR, 1)
OVER(ORDER BY ORDER_YEAR)))/(LAG(TOTAL_COST_PER_YEAR, 1)
OVER(ORDER BY ORDER_YEAR)), 2) AS PCNT_INCR_IN_COST
FROM CTE
ORDER BY 1
```

Output:



Row	ORDER_YEAR ▼	TOTAL_COST_PER_YEAR	PREV_YEAR_COST	PCNT_INCR_IN_COST
1	2017	3669022.12	null	null
2	2018	8694733.84	3669022.12	136.98

Insights:

1) The percentage increase in order cost from 2017 to 2018 is close to 137%.

2) The huge increase is clearly because 2017 was Target's early months and since then the orders cost has shot up.

Recommendation:

None.

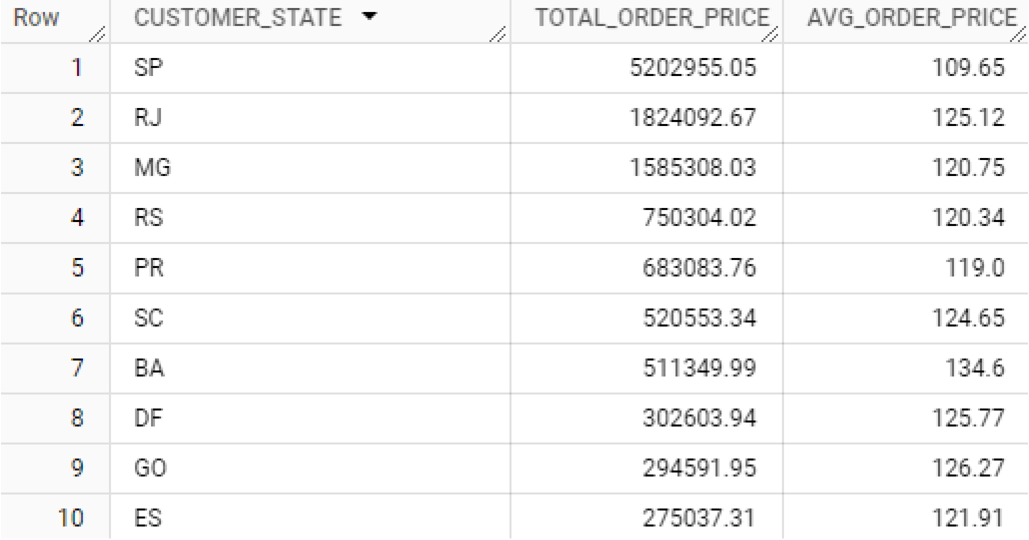
Q 4.2

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

Query:

```
SELECT C.CUSTOMER_STATE, ROUND(SUM(I.PRICE),2) AS
TOTAL_ORDER_PRICE, ROUND(AVG(I.PRICE),2) AS AVG_ORDER_PRICE
FROM `Target_SQL.orders` O
JOIN
`Target_SQL.order_items` I
ON O.ORDER_ID = I.ORDER_ID
JOIN
`Target_SQL.customers` C
ON O.CUSTOMER_ID = C.CUSTOMER_ID
GROUP BY 1
ORDER BY 2 DESC
```

Output:



Row	CUSTOMER_STATE	TOTAL_ORDER_PRICE	AVG_ORDER_PRICE
1	SP	5202955.05	109.65
2	RJ	1824092.67	125.12
3	MG	1585308.03	120.75
4	RS	750304.02	120.34
5	PR	683083.76	119.0
6	SC	520553.34	124.65
7	BA	511349.99	134.6
8	DF	302603.94	125.77
9	GO	294591.95	126.27
10	ES	275037.31	121.91

Insights:

1) The state with highest total order price is SP. The 2nd and 3rd places are taken by RJ and MJ respectively. This is the exact same ranking that these 3 states have even in "Number of customers" (as seen in Q3.2). So, it's natural to expect these states to have highest order prices. The same principle applies to the lowest ranked states as well (AC, AP, RR).

2) The state with highest average order price is PB and that with lowest order price is SP.

3) Here is an interesting stat - Even though SP had highest total order price, it was placed at the bottom in the average order price. This is typically because, in urban states like SP, where there are huge number of orders (47450), there are bound to be a lot of orders whose prices are high and a lot many more orders whose prices are very low. In contrast, the state RP has only 600 orders. Hence the difference.

Recommendations:

1) Insight No. 3) provides a very interesting information where in a state like SP, the majority of orders' prices are low. One way in which Target can fix this problem is by introducing clauses like "Min order price for free delivery" and provide discounts above a minimum

order value. This could help Target drive up the order prices and also make customers spend more with each purchase.
2) From Insight No. 1), it's clear that there is a direct co-relation between number of customers and the total order price for the state. Hence Target must try to expand their customer base in every state and that will ensure the increase in revenue.

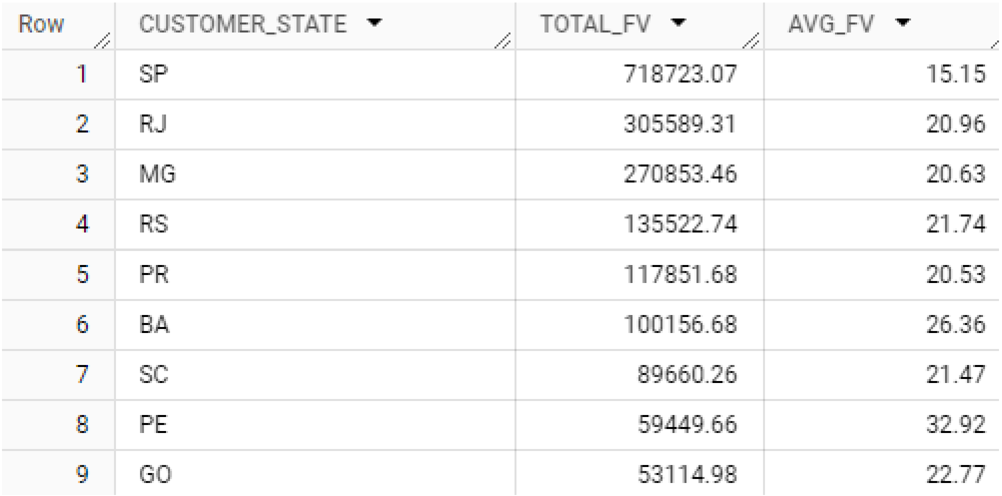
Q 4.3

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

Query:

```
SELECT C.CUSTOMER_STATE, ROUND(SUM(I.FREIGHT_VALUE),2) AS
TOTAL_FV, ROUND(AVG(I.FREIGHT_VALUE),2) AS AVG_FV
FROM `Target_SQL.orders` O
JOIN
`Target_SQL.order_items` I
ON O.ORDER_ID = I.ORDER_ID
JOIN
`Target_SQL.customers` C
ON O.CUSTOMER_ID = C.CUSTOMER_ID
GROUP BY 1
ORDER BY 2 DESC
```

Output:



Row	CUSTOMER_STATE	TOTAL_FV	AVG_FV
1	SP	718723.07	15.15
2	RJ	305589.31	20.96
3	MG	270853.46	20.63
4	RS	135522.74	21.74
5	PR	117851.68	20.53
6	BA	100156.68	26.36
7	SC	89660.26	21.47
8	PE	59449.66	32.92
9	GO	53114.98	22.77

Insights:

1) States SP, RJ and MG have the highest Freight values given their large customer count. The same principle applies to the lowest ranked states as well (AC, AP, RR). It is a highlight that SP has the least average freight cost.
2) I tried comparing top 10 freight avg cost and bottom 10 orders cost to see if there are any common states, but there was not a single match. This shows that the states with high average freight cost have good order costs as well.

Recommendations:

1) Freight cost of RJ and MG make up 17% of the total order cost of their respective states. This figure is only 13% for SP. So, Target can focus on reducing their freight costs for RJ and

MG as this is taking a huge chunk of their revenue. This can be done by taking some measures such as -

a) Planning deliveries better through routing orders for maximum efficiency.

b) Offering small discounts for bulk orders (e.g.: maximum orders can be delivered on a single delivery route)

c) RJ has 2nd highest Freight value, but has only 171 sellers (info from sellers table). Target needs to identify more such states where Freight values are very high and have more local sellers which will eventually reduce the freight costs.

Q 5.1

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

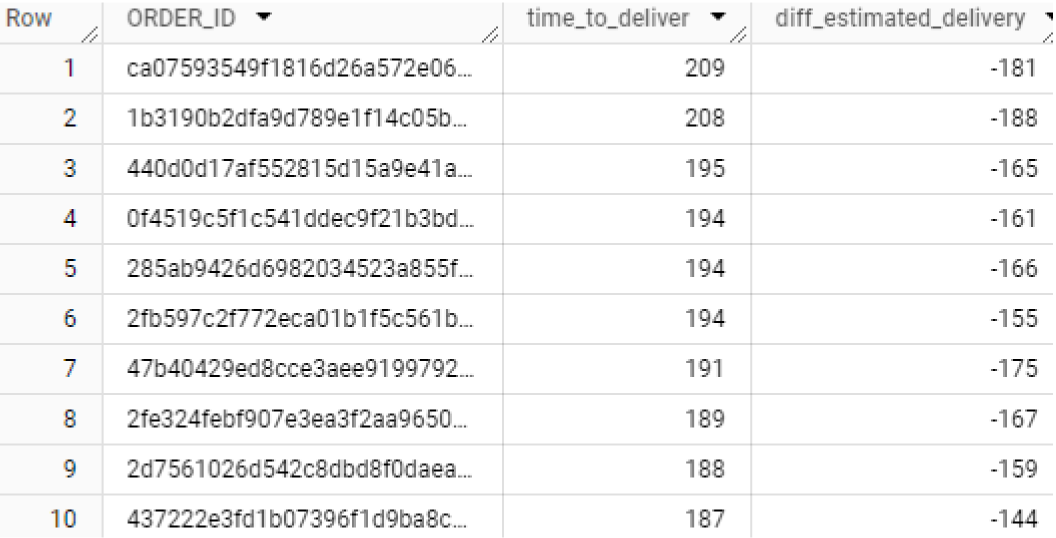
Query:

```
SELECT ORDER_ID,
DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp,
DAY) AS time_to_deliver,
DATE_DIFF(order_estimated_delivery_date,
order_delivered_customer_date, DAY ) AS diff_estimated_delivery
FROM `Target_SQL.orders`
WHERE order_delivered_customer_date IS NOT NULL
ORDER BY 2 DESC
```

Assumption:

Here I interpret the question to consider every order that was delivered, regardless of whether it was cancelled later or not. Hence, I have used IS NOT NULL and not order_status = "delivered".

Output:



Row	ORDER_ID	time_to_deliver	diff_estimated_delivery
1	ca07593549f1816d26a572e06...	209	-181
2	1b3190b2dfa9d789e1f14c05b...	208	-188
3	440d0d17af552815d15a9e41a...	195	-165
4	0f4519c5f1c541ddec9f21b3bd...	194	-161
5	285ab9426d6982034523a855f...	194	-166
6	2fb597c2f772eca01b1f5c561b...	194	-155
7	47b40429ed8cce3aee9199792...	191	-175
8	2fe324febf907e3ea3f2aa9650...	189	-167
9	2d7561026d542c8dbd8f0daea...	188	-159
10	437222e3fd1b07396f1d9ba8c...	187	-144

Insights:

1) Nearly 3000 orders delivery dates are missing. In these cases, most orders were cancelled or approved processed, invoiced and shipped but never delivered.

2) Orders have long delivery lead time. Out of 99k orders, 62k of them took more than a week's time.
3) 6535 orders did not deliver as per the estimated delivery date.

Recommendations:
1) A root cause analysis has to be performed as to why 3000 orders were not delivered after the customer has ordered it (Negative values in diff_estimated_delivery). It could be cancelled by customer, but apart from that there could be several reasons like stock unavailability or processing errors or logistics issue. Target needs to address these issues so as to reduce the losses.
2) In the current times where orders are delivered by competitors within 24 hours, these high delivery lead times won't help Target grow. Target needs to make most deliveries happen within a week and for that it needs to invest in more local warehouses and tie up with more local suppliers, that way even the freight charges will go down.
3) Target needs to perform a more accurate estimation of order delivery time or find root cause for delayed deliveries.

Q 5.2

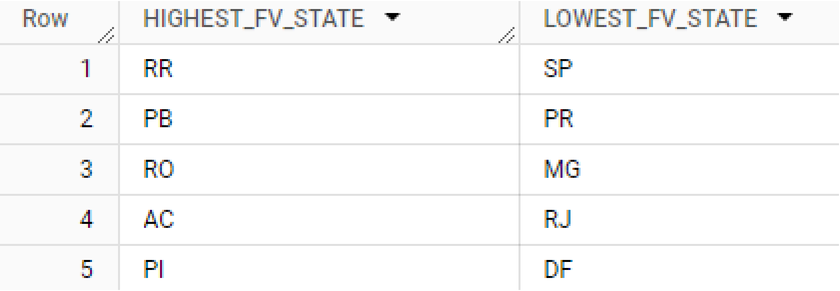
Find out the top 5 states with the highest & lowest average freight value.

Assumption:
Writing query to show only the states details, and not the actual average freight value as it’s not specifically asked in the question.

Query:

```
WITH CTE AS (
  SELECT C.CUSTOMER_STATE, ROUND(AVG(I.FREIGHT_VALUE),2) AS AVG_FV
  FROM `Target_SQL.orders` o
  JOIN
  `Target_SQL.order_items` I
  ON O.ORDER_ID = I.ORDER_ID
  JOIN
  `Target_SQL.customers` c
  ON O.CUSTOMER_ID = C.CUSTOMER_ID
  GROUP BY 1
  ORDER BY 2 DESC
),

A AS (
  SELECT CUSTOMER_STATE AS HIGHEST_FV_STATE,
  ROW_NUMBER() OVER(ORDER BY AVG_FV DESC) AS ROW_NUMBERS
  FROM CTE
  ORDER BY AVG_FV DESC
  LIMIT 5
),

B AS (
  SELECT CUSTOMER_STATE AS LOWEST_FV_STATE,
  ROW_NUMBER() OVER(ORDER BY AVG_FV ASC) AS ROW_NUMBERS
  FROM CTE
  ORDER BY AVG_FV ASC
  LIMIT 5
)

SELECT A.HIGHEST_FV_STATE, B.LOWEST_FV_STATE
FROM A
JOIN
B
ON A.ROW_NUMBERS = B.ROW_NUMBERS
```

Output:



Row	HIGHEST_FV_STATE	LOWEST_FV_STATE
1	RR	SP
2	PB	PR
3	RO	MG
4	AC	RJ
5	PI	DF

Insights:

1) States like SP, PR and MG which have highest total order prices have lowest average freight value.

2) Similarly states with highest average freight values have lower total order prices.

Recommendations:

Refer recommendations from Q4.3 on average freight values and how Target can work on it.

Q 5.3

Find out the top 5 states with the highest & lowest average delivery time.

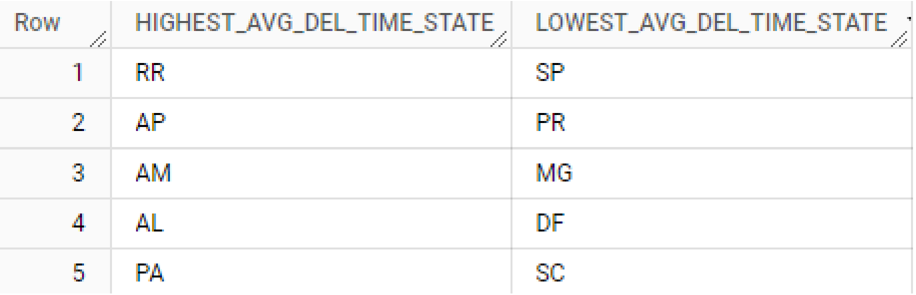
Assumption:

Writing query to show only the states details, and not the actual average delivery time as it's not specifically asked in the question.

Query:

```
WITH CTE AS (
  SELECT C.CUSTOMER_STATE,
  ROUND(AVG(DATE_DIFF(O.order_delivered_customer_date,
O.order_purchase_timestamp, DAY)),2) AS AVG_DELIVERY_DAYS
  FROM `Target_SQL.orders` O
  JOIN
  `Target_SQL.customers` C
  ON C.CUSTOMER_ID = O.CUSTOMER_ID AND
O.order_delivered_customer_date IS NOT NULL
  GROUP BY 1
  ORDER BY 2 DESC
),

A AS (
  SELECT CUSTOMER_STATE AS HIGHEST_AVG_DEL_TIME_STATE,
  ROW_NUMBER() OVER(ORDER BY AVG_DELIVERY_DAYS DESC) AS
ROW_NUMBERS
  FROM CTE
  ORDER BY AVG_DELIVERY_DAYS DESC
  LIMIT 5
),

B AS (
  SELECT CUSTOMER_STATE AS LOWEST_AVG_DEL_TIME_STATE,
  ROW_NUMBER() OVER(ORDER BY AVG_DELIVERY_DAYS ASC) AS ROW_NUMBERS
  FROM CTE
  ORDER BY AVG_DELIVERY_DAYS ASC
  LIMIT 5
)

SELECT A.HIGHEST_AVG_DEL_TIME_STATE, B.LOWEST_AVG_DEL_TIME_STATE
FROM A
JOIN
B
ON A.ROW_NUMBERS = B.ROW_NUMBERS
```

Output:



Row	HIGHEST_AVG_DEL_TIME_STATE	LOWEST_AVG_DEL_TIME_STATE
1	RR	SP
2	AP	PR
3	AM	MG
4	AL	DF
5	PA	SC

Insights:

1) All states in the top 5 lowest avg. delivery time have highest seller counts as well (referenced from Sellers table)

2) All states in the top 5 highest avg. delivery time don’t have any sellers in them. This is the reason the avg. delivery times are so high.

3) Some of the states which have high avg. delivery times also have highest avg. freight values (referred from Q 4.3 output)

Recommendation:

Target needs to first focus on the 5 states which have highest avg. delivery times, especially because they don’t have a single seller in these states. This will lower their freight values and increase order count.

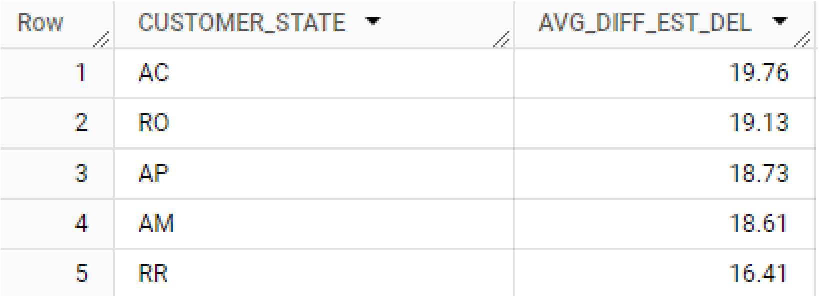
Q 5.4

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

Query:

```
SELECT C.CUSTOMER_STATE,
ROUND(AVG(DATE_DIFF(order_estimated_delivery_date,
order_delivered_customer_date, DAY)),2) AS AVG_DIFF_EST_DEL
FROM `Target_SQL.orders` O
JOIN
`Target_SQL.customers` C
ON C.CUSTOMER_ID = O.CUSTOMER_ID AND order_delivered_customer_date
IS NOT NULL
GROUP BY 1
ORDER BY 2 DESC
LIMIT 5
```

Output:



Row	CUSTOMER_STATE	AVG_DIFF_EST_DEL
1	AC	19.76
2	RO	19.13
3	AP	18.73
4	AM	18.61
5	RR	16.41

Insights:

1) An interesting stat here is that none of the top 5 states are part of the top 10 states with highest seller count (reference - Sellers table)

2) RR, which has the 5th fastest delivery compared to the estimated dates, doesn't even have a single seller located in the state.

Recommendations:

1) Root cause analysis (Insight No.2) need not be done only in case of failures or issues; it should also be performed when a few states perform better than expected. This will help Target implement similar strategies in other states as well.

2) For these states, Target can start providing more realistic delivery estimates so as to increase the customer base.

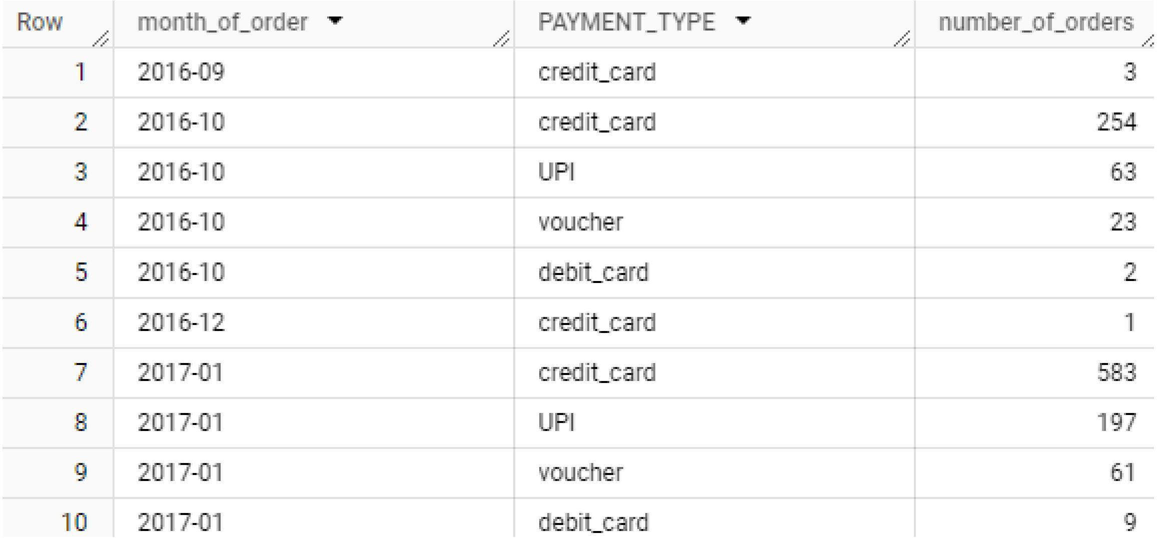
Q 6.1

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

Query:

```
SELECT FORMAT_DATE("%Y-%m", O.order_purchase_timestamp) as
month_of_order, P.PAYMENT_TYPE,
COUNT(O.order_id) as number_of_orders
FROM `Target_SQL.orders` O
JOIN
`Target_SQL.payments` P
ON O.ORDER_ID = P.ORDER_ID
GROUP BY 1, 2
ORDER BY 1, 3 DESC
```

Output:



Row	month_of_order	PAYMENT_TYPE	number_of_orders
1	2016-09	credit_card	3
2	2016-10	credit_card	254
3	2016-10	UPI	63
4	2016-10	voucher	23
5	2016-10	debit_card	2
6	2016-12	credit_card	1
7	2017-01	credit_card	583
8	2017-01	UPI	197
9	2017-01	voucher	61
10	2017-01	debit_card	9

Insights:

1) It is clear from the data that a majority of the payment modes are digital.
2) Credit card payment mode has the highest share of orders, followed by UPI.

Recommendations:

1) As the vast majority payments are coming through credit cards, Target can partner with a bank so that it can benefit from discounted transaction fees and the bank could also see an increase in number of transactions.
2) Additionally, they can come up with a new Target exclusive bank credit card which will give bonus reward points by shopping on Target.
3) Target can come up with a digital wallet and add it as a new payment mode. This will reduce the transaction fees for Target and also ease up the payment process for customers.

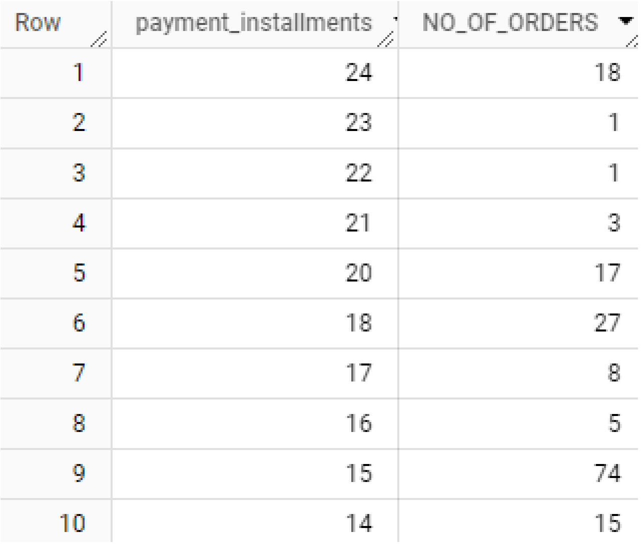
Q 6.2

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

Query:

```
SELECT payment_installments, COUNT(DISTINCT ORDER_ID) AS
NO_OF_ORDERS
FROM `Target_SQL.payments`
GROUP BY 1
ORDER BY 1 DESC
```

Output:



Row	payment_installments	NO_OF_ORDERS
1	24	18
2	23	1
3	22	1
4	21	3
5	20	17
6	18	27
7	17	8
8	16	5
9	15	74
10	14	15

Insights:

1) Majority of the orders were paid for within 1 to 3 instalments.

2) As it is clear from the Q 6.1, as the huge number customers have credit card as payment option, where they have sufficient time to pay off the dues (or the credit card itself might have EMI plans), it’s obvious that they don’t rely much on multiple instalments offered by Target, hence this explains the low number of customers opting for multiple payment instalments.

Recommendations:

Target can attract more customers by providing zero emi as instalment option. This will drive the customer order count as well.