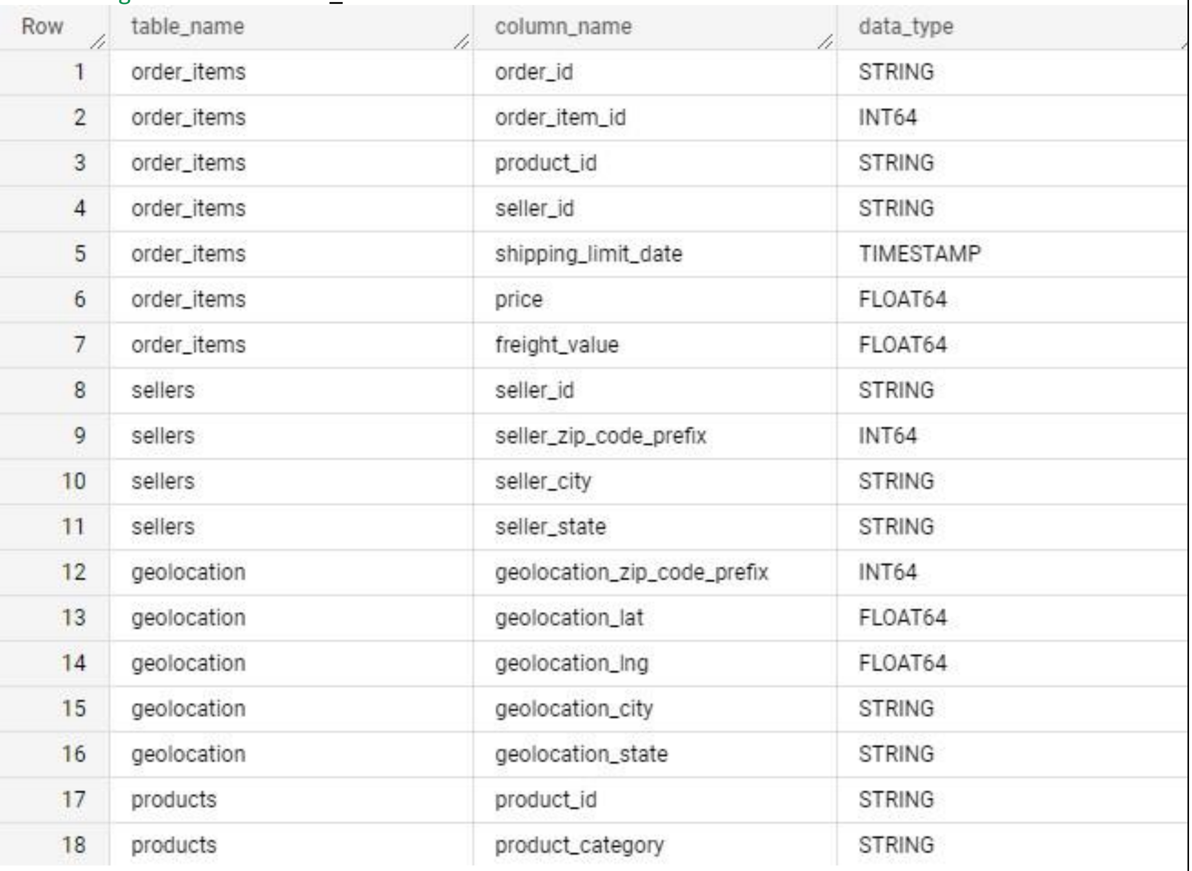
Business Case: Target SQL

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

1. Data type of columns in a table

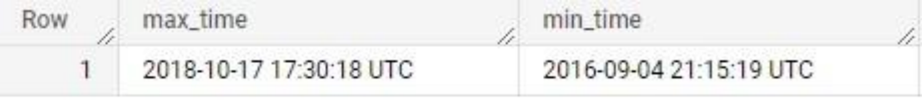
```
SELECT table_name, column_name, data_type
FROM `target`.INFORMATION_SCHEMA.COLUMNS
```



Row	table_name	column_name	data_type
1	order_items	order_id	STRING
2	order_items	order_item_id	INT64
3	order_items	product_id	STRING
4	order_items	seller_id	STRING
5	order_items	shipping_limit_date	TIMESTAMP
6	order_items	price	FLOAT64
7	order_items	freight_value	FLOAT64
8	sellers	seller_id	STRING
9	sellers	seller_zip_code_prefix	INT64
10	sellers	seller_city	STRING
11	sellers	seller_state	STRING
12	geolocation	geolocation_zip_code_prefix	INT64
13	geolocation	geolocation_lat	FLOAT64
14	geolocation	geolocation_lng	FLOAT64
15	geolocation	geolocation_city	STRING
16	geolocation	geolocation_state	STRING
17	products	product_id	STRING
18	products	product_category	STRING

2. Time Period for which the data is given

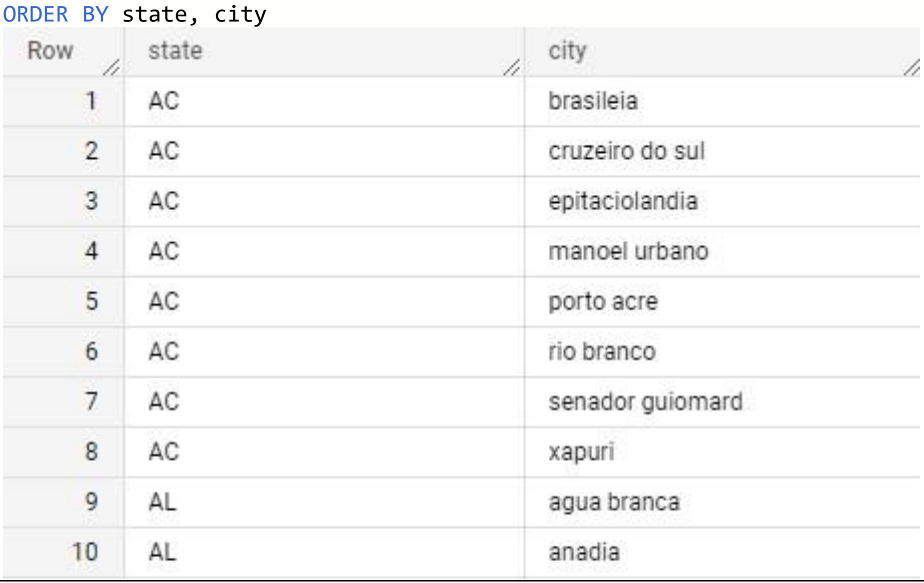
```
SELECT DISTINCT MAX(order_purchase_timestamp) AS max_time, MIN(order_purchase_timestamp)
 AS min_time
FROM `target.orders`
```



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

3. Cities and States Covered in dataset

```
WITH v1 AS
(SELECT DISTINCT customer_state AS state, customer_city AS city
FROM `target-sql-368610.target.customers`

UNION ALL

SELECT DISTINCT seller_state AS state, seller_city AS city
FROM `target-sql-368610.target.sellers`)

SELECT DISTINCT * FROM v1
ORDER BY state, city
```

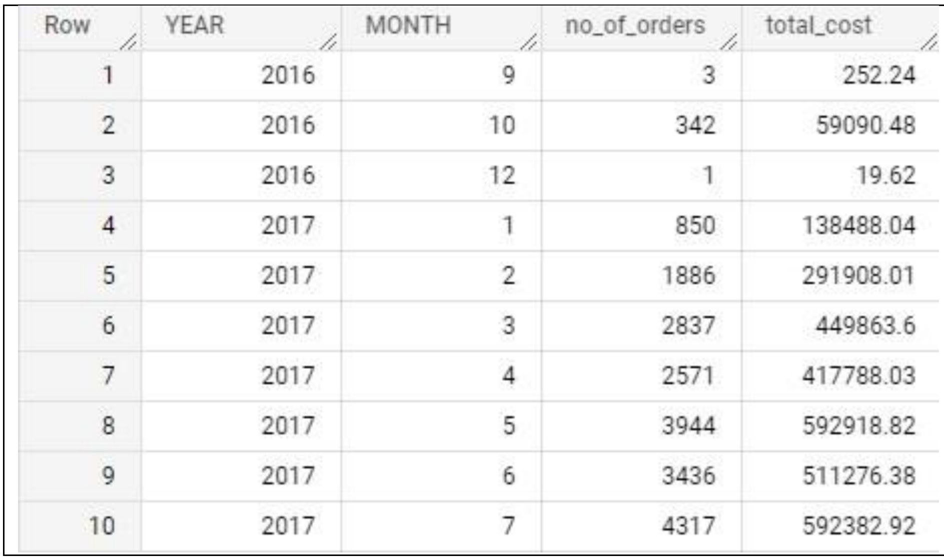


Row	state	city
1	AC	brasileia
2	AC	cruzeiro do sul
3	AC	epitaciolandia
4	AC	manoel urbano
5	AC	porto acre
6	AC	rio branco
7	AC	senador guiomard
8	AC	xapuri
9	AL	agua branca
10	AL	anadia

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?

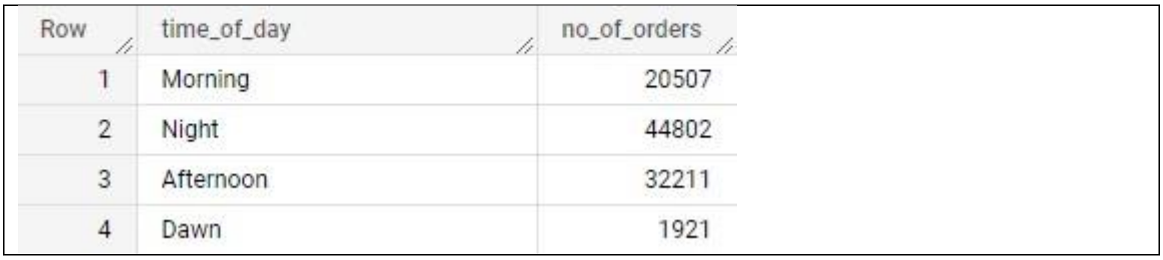
```
SELECT EXTRACT(YEAR FROM order_purchase_timestamp) AS YEAR, EXTRACT(MONTH FROM order_purchase_timestamp) AS MONTH, COUNT(*) AS no_of_orders, ROUND(SUM(payment_value), 2)
AS total_cost
FROM `target.orders` o JOIN `target.payments` p ON o.order_id = p.order_id
GROUP BY 1, 2
ORDER BY 1, 2
```



Row	YEAR	MONTH	no_of_orders	total_cost
1	2016	9	3	252.24
2	2016	10	342	59090.48
3	2016	12	1	19.62
4	2017	1	850	138488.04
5	2017	2	1886	291908.01
6	2017	3	2837	449863.6
7	2017	4	2571	417788.03
8	2017	5	3944	592918.82
9	2017	6	3436	511276.38
10	2017	7	4317	592382.92

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

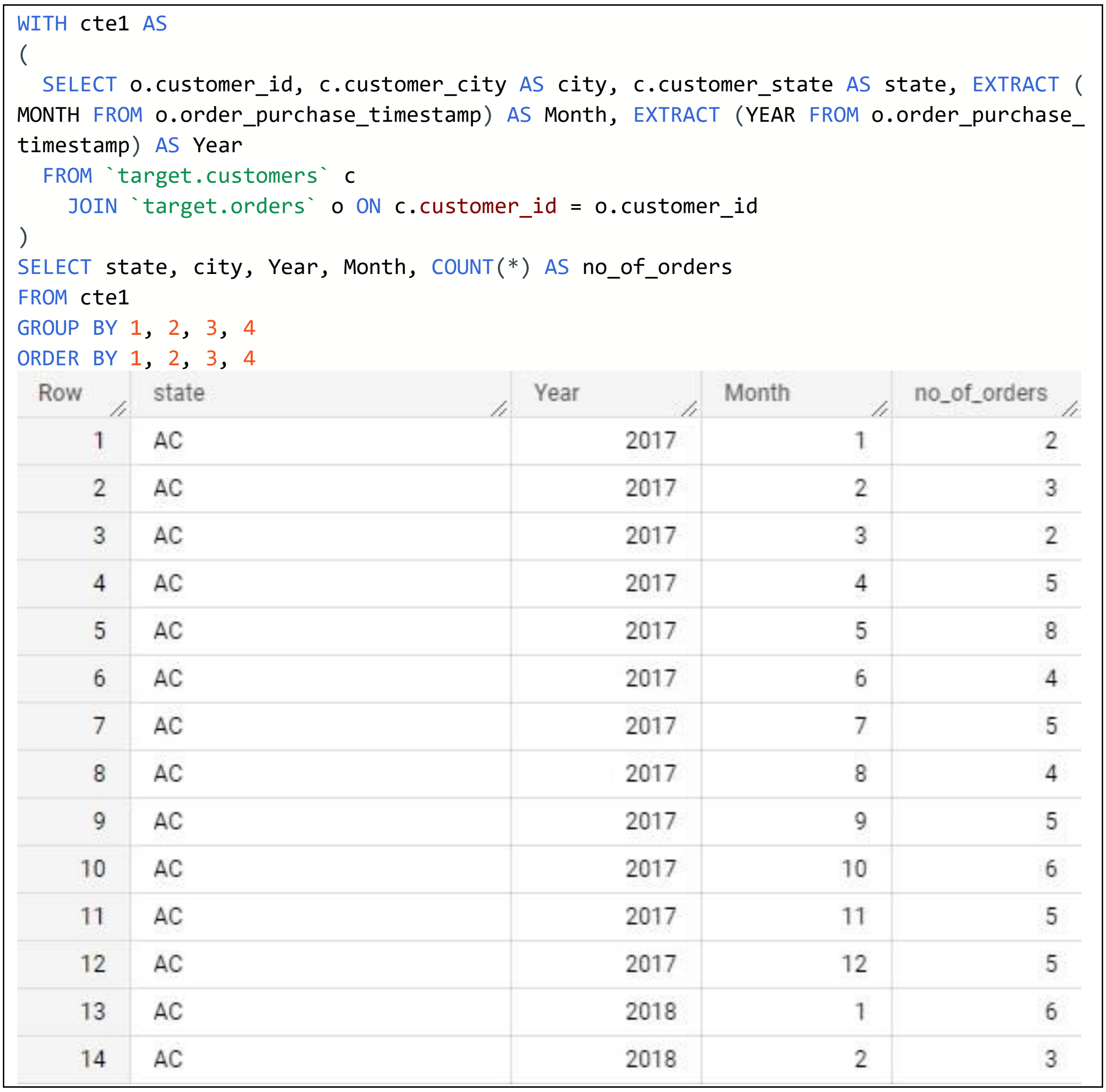
```sql
WITH cte1 AS
(
  SELECT customer_id, order_id, order_purchase_timestamp, EXTRACT(HOUR FROM order_purchase_timestamp) AS Hour
  FROM `target.orders`
),
cte2 AS
(
 SELECT *,
  CASE
    WHEN Hour >= 5 AND Hour <8 THEN 'Dawn'
    WHEN Hour >= 8 AND Hour <12 THEN 'Morning'
    WHEN Hour >= 12 AND Hour <17 THEN 'Afternoon'
    WHEN Hour >= 17 AND Hour <=23 THEN 'Night'
    WHEN Hour >= 0 AND Hour <5  THEN 'Night'
    END AS time_of_day
 FROM cte1
),
cte3 AS
(
SELECT customer_id, order_id, Hour, time_of_day
FROM cte2
ORDER BY 1, 2
)

SELECT time_of_day, COUNT(*) AS no_of_orders
FROM cte3
GROUP BY 1
```



Row	time_of_day	no_of_orders
1	Morning	20507
2	Night	44802
3	Afternoon	32211
4	Dawn	1921

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

1. Get month on month orders by states

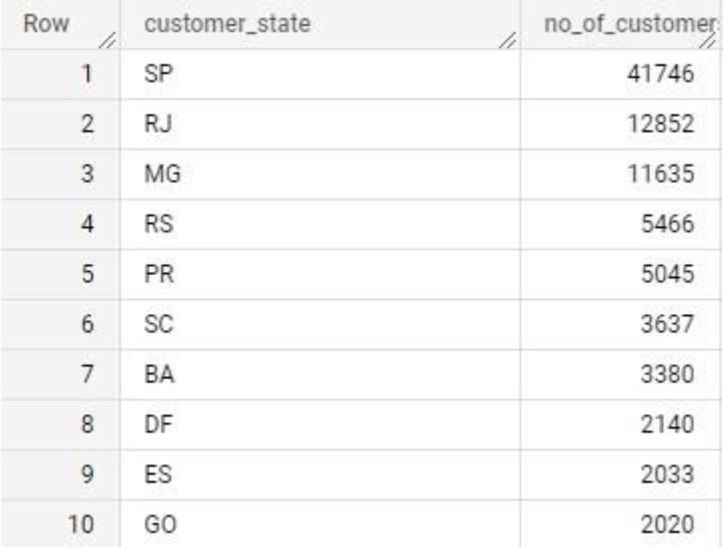


```
WITH cte1 AS
(
  SELECT o.customer_id, c.customer_city AS city, c.customer_state AS state, EXTRACT (
MONTH FROM o.order_purchase_timestamp) AS Month, EXTRACT (YEAR FROM o.order_purchase_
timestamp) AS Year
  FROM `target.customers` c
    JOIN `target.orders` o ON c.customer_id = o.customer_id
)
SELECT state, city, Year, Month, COUNT(*) AS no_of_orders
FROM cte1
GROUP BY 1, 2, 3, 4
ORDER BY 1, 2, 3, 4
```

Row	state	Year	Month	no_of_orders
1	AC	2017	1	2
2	AC	2017	2	3
3	AC	2017	3	2
4	AC	2017	4	5
5	AC	2017	5	8
6	AC	2017	6	4
7	AC	2017	7	5
8	AC	2017	8	4
9	AC	2017	9	5
10	AC	2017	10	6
11	AC	2017	11	5
12	AC	2017	12	5
13	AC	2018	1	6
14	AC	2018	2	3

2. Distribution of customers across the states in Brazil

```
SELECT customer_state, COUNT(*) AS no_of_customers
FROM `target.customers`
GROUP BY 1
ORDER BY 2 DESC
```

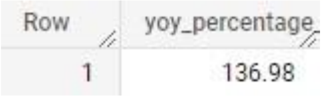


Row	customer_state	no_of_customer
1	SP	41746
2	RJ	12852
3	MG	11635
4	RS	5466
5	PR	5045
6	SC	3637
7	BA	3380
8	DF	2140
9	ES	2033
10	GO	2020

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 to Aug only) - You can use “payment_value” column in payments table

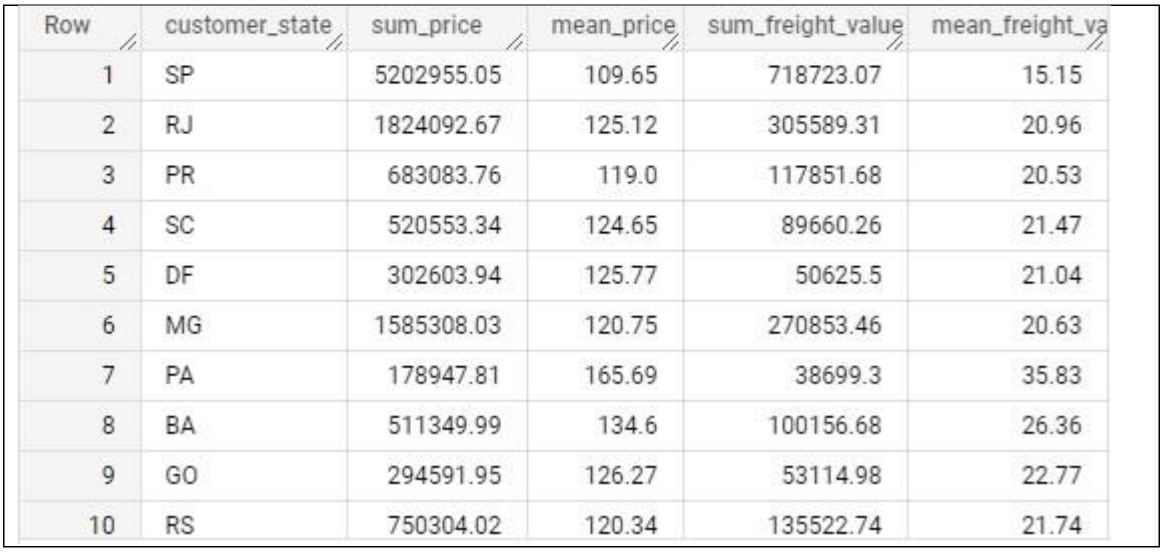
```
WITH cte1 AS
(
  SELECT order_id, EXTRACT(YEAR FROM order_purchase_timestamp) AS Year
  FROM `target.orders`
  WHERE EXTRACT(MONTH FROM order_purchase_timestamp) in (1, 2, 3, 4, 5, 6, 7, 8)
  ORDER BY 2
),
cte2 AS
(
  SELECT order_id, payment_value FROM `target.payments`
),
cte3 AS
(
  SELECT *
  FROM cte1 AS o
    JOIN cte2 AS p ON o.order_id = p.order_id
),
cte4 AS
(
  SELECT Year, ROUND(SUM(payment_value), 2) AS total_payment_value
  FROM cte3
  GROUP BY 1
),
cte5 AS
(
  SELECT *,
  LAG(cte4.total_payment_value) OVER(ORDER BY Year DESC) AS yoy_change
  FROM cte4
  ORDER BY Year DESC
)
SELECT yoy_percentage_change FROM
(SELECT ABS(ROUND(((total_payment_value - yoy_change)/total_payment_value)*100, 2)) A
S yoy_percentage_change
FROM cte5) A
WHERE A.yoy_percentage_change IS NOT NULL
```



Row	yoy_percentage_
1	136.98

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

```
WITH cte1 AS
(
  SELECT order_id, price, freight_value
  FROM `target.order_items`
),
cte2 AS
(
  SELECT oi.order_id, o.customer_id, price, freight_value
  FROM `target.orders` o
    RIGHT JOIN `cte1` oi ON oi.order_id = o.order_id
),
cte3 AS
(
  SELECT a.order_id, a.customer_id, c.customer_state, price, freight_value
  FROM cte2 a
    JOIN `target.customers` c ON a.customer_id = c.customer_id
)
SELECT customer_state,
  ROUND(SUM(price), 2) AS sum_price, ROUND(SUM(price)/COUNT(*), 2) AS mean_price,
  ROUND(SUM(freight_value), 2) AS sum_freight_value, ROUND(SUM(freight_value)/COUNT(*
), 2) AS mean_freight_value
FROM cte3
GROUP BY customer_state
```

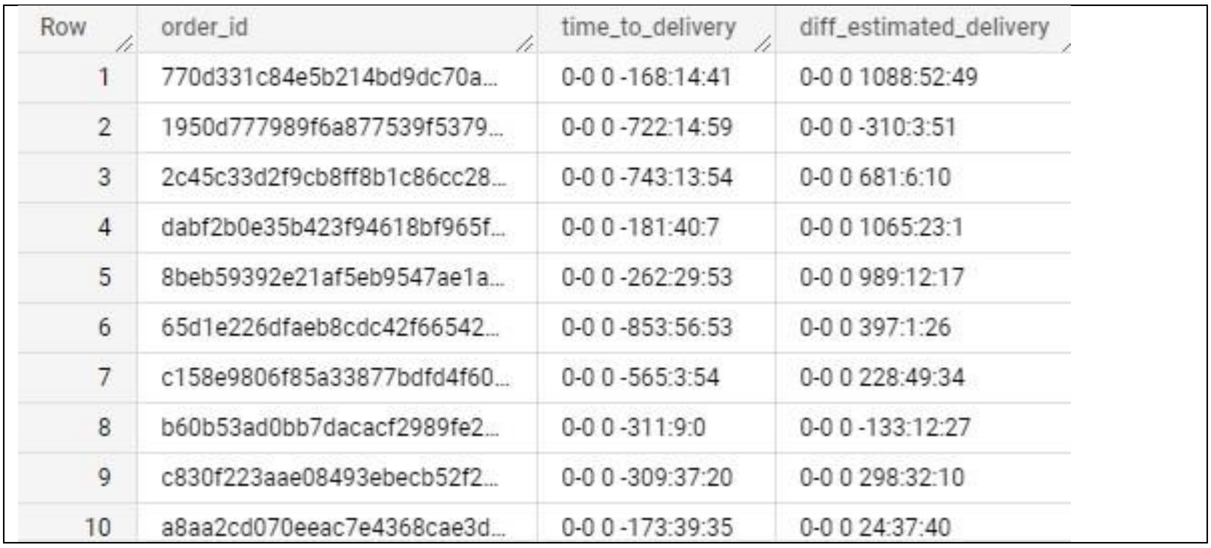


Row	customer_state	sum_price	mean_price	sum_freight_value	mean_freight_va
1	SP	5202955.05	109.65	718723.07	15.15
2	RJ	1824092.67	125.12	305589.31	20.96
3	PR	683083.76	119.0	117851.68	20.53
4	SC	520553.34	124.65	89660.26	21.47
5	DF	302603.94	125.77	50625.5	21.04
6	MG	1585308.03	120.75	270853.46	20.63
7	PA	178947.81	165.69	38699.3	35.83
8	BA	511349.99	134.6	100156.68	26.36
9	GO	294591.95	126.27	53114.98	22.77
10	RS	750304.02	120.34	135522.74	21.74

5. Analysis on sales, freight and delivery time

1. Calculate days between purchasing, delivering and estimated delivery
2. Find time_to_delivery & diff_estimated_delivery. Formula for the same given below:
   - time_to_delivery = order_purchase_timestamp-order_delivered_customer_date
   - diff_estimated_delivery = order_estimated_delivery_date-order_delivered_customer_date

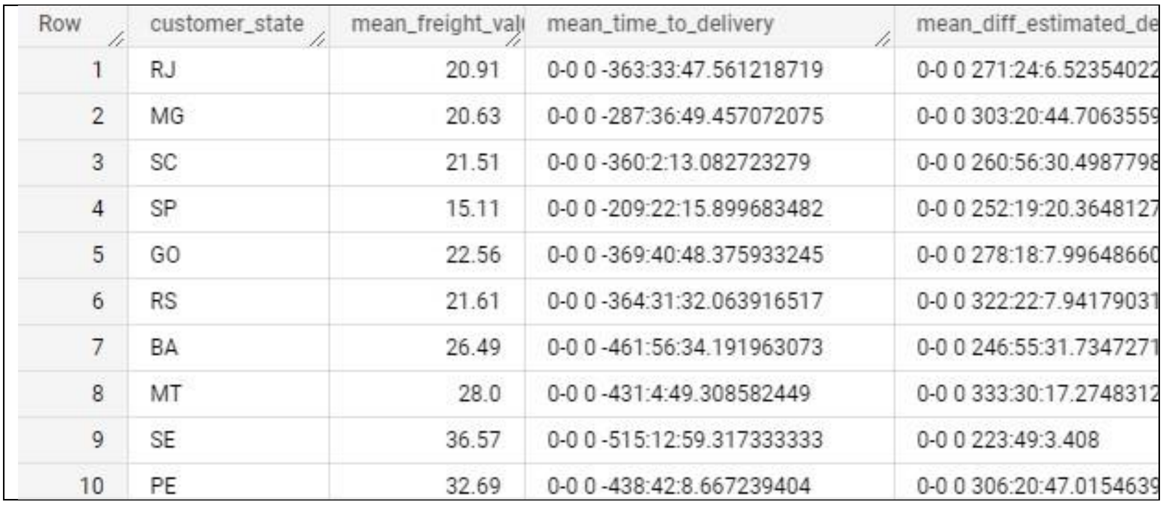
```
WITH cte1 AS
(
  SELECT *,
  order_purchase_timestamp - order_delivered_customer_date AS time_to_delivery,
  order_estimated_delivery_date - order_delivered_customer_date AS diff_estimated_delive
ry
  FROM `target.orders`
),
cte2 AS
(
SELECT order_id, time_to_delivery, diff_estimated_delivery FROM cte1
WHERE time_to_delivery is NOT NULL
AND diff_estimated_delivery is NOT NULL
)
SELECT * FROM cte2
```



Row	order_id	time_to_delivery	diff_estimated_delivery
1	770d331c84e5b214bd9dc70a...	0-0 0 -168:14:41	0-0 0 1088:52:49
2	1950d777989f6a877539f5379...	0-0 0 -722:14:59	0-0 0 -310:3:51
3	2c45c33d2f9cb8ff8b1c86cc28...	0-0 0 -743:13:54	0-0 0 681:6:10
4	dabf2b0e35b423f94618bf965f...	0-0 0 -181:40:7	0-0 0 1065:23:1
5	8beb59392e21af5eb9547ae1a...	0-0 0 -262:29:53	0-0 0 989:12:17
6	65d1e226dfaeb8cdc42f66542...	0-0 0 -853:56:53	0-0 0 397:1:26
7	c158e9806f85a33877bdfd4f60...	0-0 0 -565:3:54	0-0 0 228:49:34
8	b60b53ad0bb7dacacf2989fe2...	0-0 0 -311:9:0	0-0 0 -133:12:27
9	c830f223aae08493ebecb52f2...	0-0 0 -309:37:20	0-0 0 298:32:10
10	a8aa2cd070eeac7e4368cae3d...	0-0 0 -173:39:35	0-0 0 24:37:40

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

```
WITH cte1 AS
(
  SELECT C.customer_state, A.customer_id, A.order_id, B.freight_value,
  order_purchase_timestamp - order_delivered_customer_date AS time_to_delivery,
  order_estimated_delivery_date - order_delivered_customer_date AS diff_estimated_delivery
  FROM `target.orders` A
    LEFT JOIN `target.order_items` B ON A.order_id = B.order_id
    JOIN `target.customers` C ON A.customer_id = C.customer_id
),
cte2 AS
(
SELECT * FROM cte1
WHERE time_to_delivery is NOT NULL
AND diff_estimated_delivery is NOT NULL
)
SELECT customer_state,
  ROUND(SUM(freight_value)/COUNT(*), 2) AS mean_freight_value,
  SUM(time_to_delivery)/COUNT(*) AS mean_time_to_delivery,
  SUM(diff_estimated_delivery)/COUNT(*) AS mean_diff_estimated_delivery
FROM cte2
GROUP BY customer_state
```



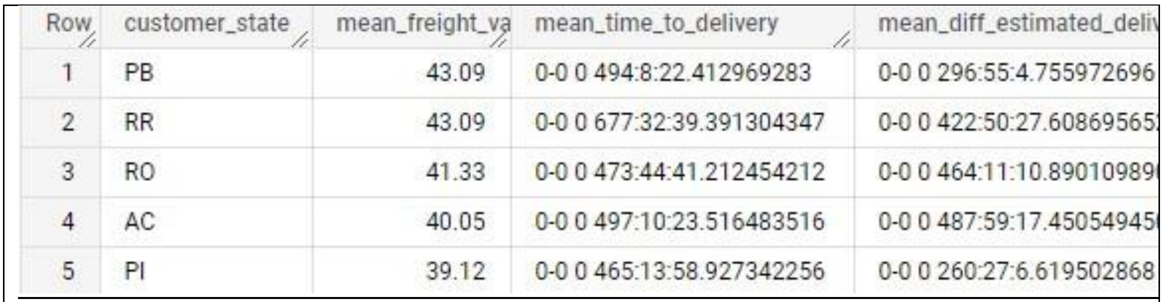
Row	customer_state	mean_freight_valı	mean_time_to_delivery	mean_diff_estimated_de
1	RJ	20.91	0-0 0 -363:33:47.561218719	0-0 0 271:24:6.52354022
2	MG	20.63	0-0 0 -287:36:49.457072075	0-0 0 303:20:44.7063559
3	SC	21.51	0-0 0 -360:2:13.082723279	0-0 0 260:56:30.4987798
4	SP	15.11	0-0 0 -209:22:15.899683482	0-0 0 252:19:20.3648127
5	GO	22.56	0-0 0 -369:40:48.375933245	0-0 0 278:18:7.99648660
6	RS	21.61	0-0 0 -364:31:32.063916517	0-0 0 322:22:7.94179031
7	BA	26.49	0-0 0 -461:56:34.191963073	0-0 0 246:55:31.7347271
8	MT	28.0	0-0 0 -431:4:49.308582449	0-0 0 333:30:17.2748312
9	SE	36.57	0-0 0 -515:12:59.317333333	0-0 0 223:49:3.408
10	PE	32.69	0-0 0 -438:42:8.667239404	0-0 0 306:20:47.0154639

4. Sort the data to get the following:

5. Top 5 states with highest/lowest average freight value - sort in desc/asc limit 5

Descending

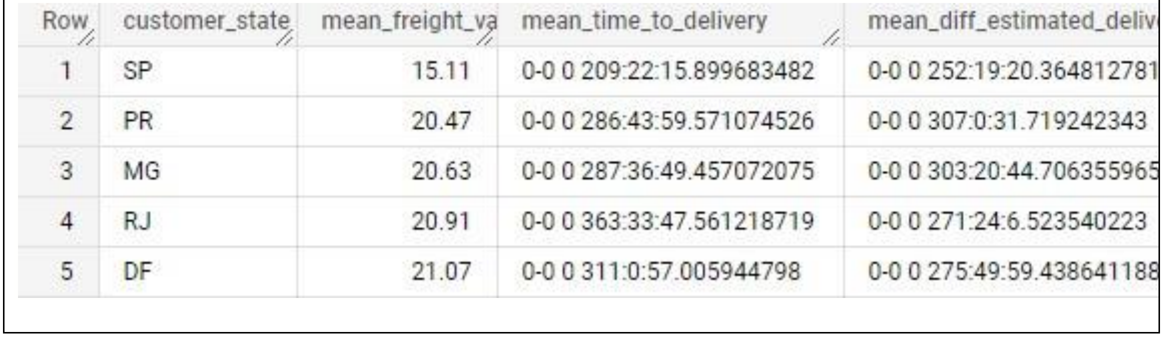
```
WITH cte1 AS
(
  SELECT C.customer_state, A.customer_id, A.order_id, B.freight_value,
  ((-
1)*(order_purchase_timestamp - order_delivered_customer_date)) AS time_to_delivery,
  order_estimated_delivery_date - order_delivered_customer_date AS diff_estimated_del
ivery
  FROM `target.orders` A
    LEFT JOIN `target.order_items` B ON A.order_id = B.order_id
    JOIN `target.customers` C ON A.customer_id = C.customer_id
),
cte2 AS
(
SELECT *
FROM cte1
WHERE time_to_delivery is NOT NULL
AND diff_estimated_delivery is NOT NULL
)
SELECT customer_state,
  ROUND(SUM(freight_value)/COUNT(*), 2) AS mean_freight_value,
  SUM(time_to_delivery)/COUNT(*) AS mean_time_to_delivery,
  SUM(diff_estimated_delivery)/COUNT(*) AS mean_diff_estimated_delivery
FROM cte2
GROUP BY customer_state
ORDER BY 2 DESC
LIMIT 5
```



Row	customer_state	mean_freight_va	mean_time_to_delivery	mean_diff_estimated_deliv
1	PB	43.09	0-0 0 494:8:22.412969283	0-0 0 296:55:4.755972696
2	RR	43.09	0-0 0 677:32:39.391304347	0-0 0 422:50:27.60869565
3	RO	41.33	0-0 0 473:44:41.212454212	0-0 0 464:11:10.89010989
4	AC	40.05	0-0 0 497:10:23.516483516	0-0 0 487:59:17.45054945
5	PI	39.12	0-0 0 465:13:58.927342256	0-0 0 260:27:6.619502868

Ascending

```
WITH cte1 AS
(
  SELECT C.customer_state, A.customer_id, A.order_id, B.freight_value,
  ((-
1)*(order_purchase_timestamp - order_delivered_customer_date)) AS time_to_delivery,
  order_estimated_delivery_date - order_delivered_customer_date AS diff_estimated_del
ivery
  FROM `target.orders` A
    LEFT JOIN `target.order_items` B ON A.order_id = B.order_id
    JOIN `target.customers` C ON A.customer_id = C.customer_id
),
cte2 AS
(
SELECT *
FROM cte1
WHERE time_to_delivery is NOT NULL
AND diff_estimated_delivery is NOT NULL
)
SELECT customer_state,
  ROUND(SUM(freight_value)/COUNT(*), 2) AS mean_freight_value,
  SUM(time_to_delivery)/COUNT(*) AS mean_time_to_delivery,
  SUM(diff_estimated_delivery)/COUNT(*) AS mean_diff_estimated_delivery
FROM cte2
GROUP BY customer_state
ORDER BY 2
LIMIT 5
```

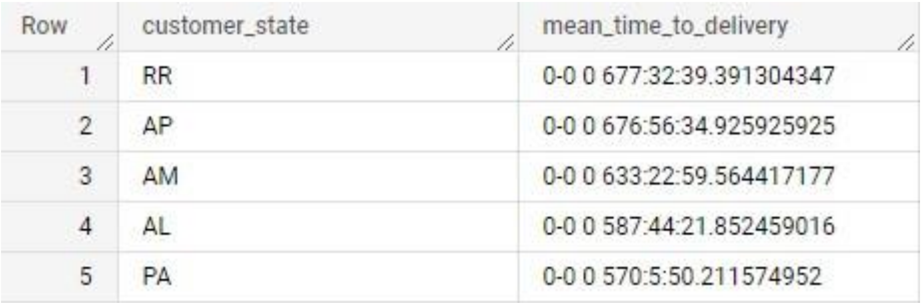


Row	customer_state	mean_freight_va	mean_time_to_delivery	mean_diff_estimated_deliv
1	SP	15.11	0-0 0 209:22:15.899683482	0-0 0 252:19:20.364812781
2	PR	20.47	0-0 0 286:43:59.571074526	0-0 0 307:0:31.719242343
3	MG	20.63	0-0 0 287:36:49.457072075	0-0 0 303:20:44.706355965
4	RJ	20.91	0-0 0 363:33:47.561218719	0-0 0 271:24:6.523540223
5	DF	21.07	0-0 0 311:0:57.005944798	0-0 0 275:49:59.438641188

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

Highest

```
WITH cte1 AS
(
  SELECT C.customer_state, A.customer_id, A.order_id, B.freight_value,
  ((-
1)*(order_purchase_timestamp - order_delivered_customer_date)) AS time_to_delivery,
  order_estimated_delivery_date - order_delivered_customer_date AS diff_estimated_del
ivery
  FROM `target.orders` A
    LEFT JOIN `target.order_items` B ON A.order_id = B.order_id
    JOIN `target.customers` C ON A.customer_id = C.customer_id
),
cte2 AS
(
SELECT *
FROM cte1
WHERE time_to_delivery is NOT NULL
AND diff_estimated_delivery is NOT NULL
)
SELECT customer_state,
  SUM(time_to_delivery)/COUNT(*) AS mean_time_to_delivery
FROM cte2
GROUP BY customer_state
ORDER BY 2 DESC
LIMIT 5
```

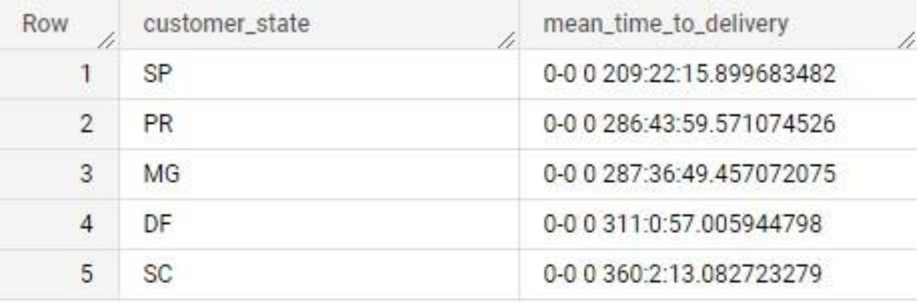


Row	customer_state	mean_time_to_delivery
1	RR	0-0 0 677:32:39.391304347
2	AP	0-0 0 676:56:34.925925925
3	AM	0-0 0 633:22:59.564417177
4	AL	0-0 0 587:44:21.852459016
5	PA	0-0 0 570:5:50.211574952

Lowest

```
WITH cte1 AS
(
  SELECT C.customer_state, A.customer_id, A.order_id, B.freight_value,
  ((-
1)*(order_purchase_timestamp - order_delivered_customer_date)) AS time_to_delivery,
  order_estimated_delivery_date - order_delivered_customer_date AS diff_estimated_del
ivery
  FROM `target.orders` A
    LEFT JOIN `target.order_items` B ON A.order_id = B.order_id
    JOIN `target.customers` C ON A.customer_id = C.customer_id
),
cte2 AS
(
SELECT *
FROM cte1
WHERE time_to_delivery is NOT NULL
AND diff_estimated_delivery is NOT NULL
)
SELECT customer_state,
  SUM(time_to_delivery)/COUNT(*) AS mean_time_to_delivery
FROM cte2
GROUP BY customer_state
ORDER BY 2
```

```
LIMIT 5
```

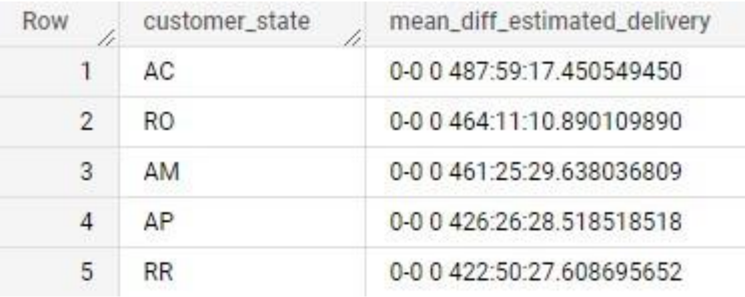


Row	customer_state	mean_time_to_delivery
1	SP	0-0 0 209:22:15.899683482
2	PR	0-0 0 286:43:59.571074526
3	MG	0-0 0 287:36:49.457072075
4	DF	0-0 0 311:0:57.005944798
5	SC	0-0 0 360:2:13.082723279

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

Descending

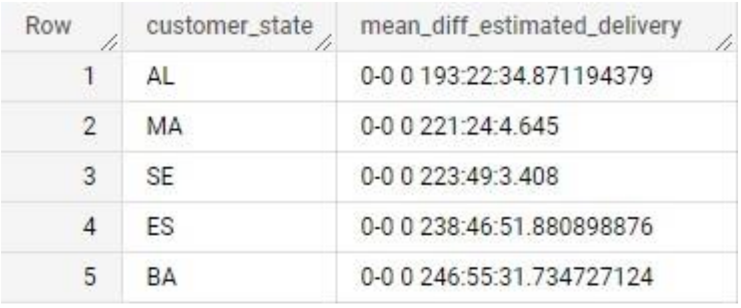
```
WITH cte1 AS
(
  SELECT C.customer_state, A.customer_id, A.order_id, B.freight_value,
  ((-
1)*(order_purchase_timestamp - order_delivered_customer_date)) AS time_to_delivery,
  order_estimated_delivery_date - order_delivered_customer_date AS diff_estimated_del
ivery
  FROM `target.orders` A
    LEFT JOIN `target.order_items` B ON A.order_id = B.order_id
    JOIN `target.customers` C ON A.customer_id = C.customer_id
),
cte2 AS
(
SELECT *
FROM cte1
WHERE time_to_delivery is NOT NULL
AND diff_estimated_delivery is NOT NULL
)
SELECT customer_state,
  SUM(diff_estimated_delivery)/COUNT(*) AS mean_diff_estimated_delivery
FROM cte2
GROUP BY customer_state
ORDER BY 2 DESC
LIMIT 5
```



Row	customer_state	mean_diff_estimated_delivery
1	AC	0-0 0 487:59:17.450549450
2	RO	0-0 0 464:11:10.890109890
3	AM	0-0 0 461:25:29.638036809
4	AP	0-0 0 426:26:28.518518518
5	RR	0-0 0 422:50:27.608695652

Ascending

```
WITH cte1 AS
(
  SELECT C.customer_state, A.customer_id, A.order_id, B.freight_value,
  ((-
1)*(order_purchase_timestamp - order_delivered_customer_date)) AS time_to_delivery,
  order_estimated_delivery_date - order_delivered_customer_date AS diff_estimated_del
ivery
  FROM `target.orders` A
    LEFT JOIN `target.order_items` B ON A.order_id = B.order_id
    JOIN `target.customers` C ON A.customer_id = C.customer_id
),
cte2 AS
(
SELECT *
FROM cte1
WHERE time_to_delivery is NOT NULL
AND diff_estimated_delivery is NOT NULL
)
SELECT customer_state,
  SUM(diff_estimated_delivery)/COUNT(*) AS mean_diff_estimated_delivery
FROM cte2
GROUP BY customer_state
ORDER BY 2
LIMIT 5
```

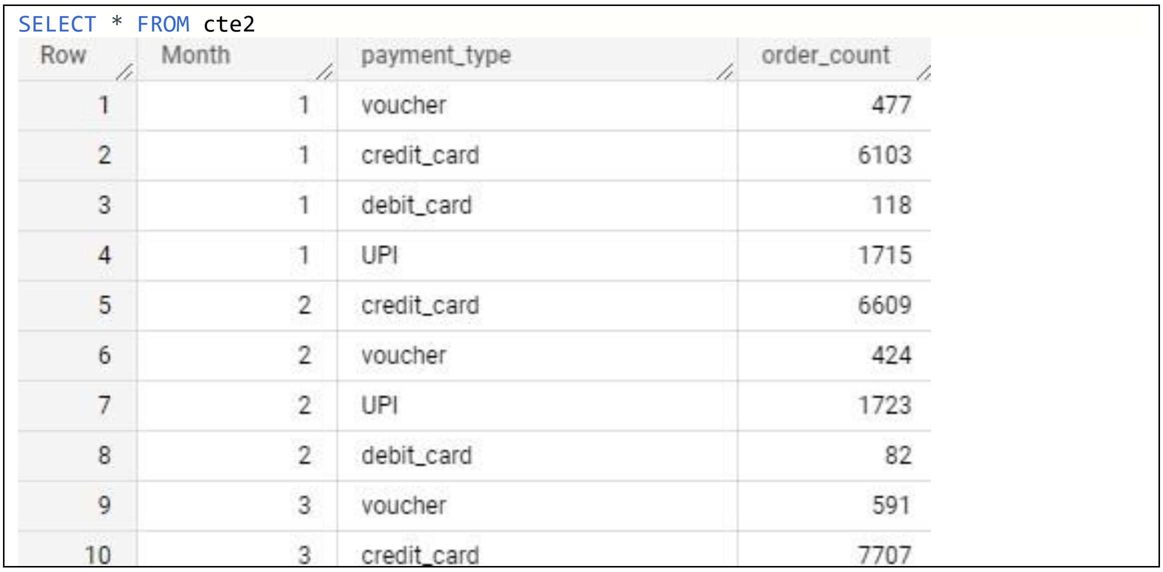


Row	customer_state	mean_diff_estimated_delivery
1	AL	0-0 0 193:22:34.871194379
2	MA	0-0 0 221:24:4.645
3	SE	0-0 0 223:49:3.408
4	ES	0-0 0 238:46:51.880898876
5	BA	0-0 0 246:55:31.734727124

6. Payment type analysis:

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

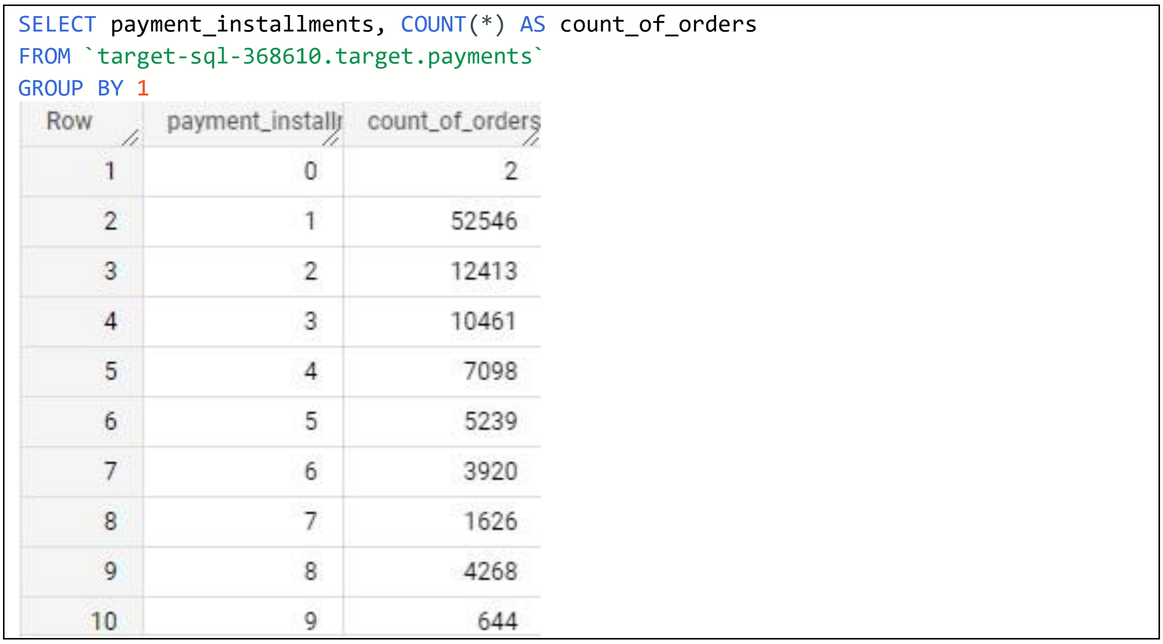
```
WITH cte1 AS
(
  SELECT *, EXTRACT(MONTH FROM order_purchase_timestamp) AS Month
  FROM `target.payments` p
    JOIN `target.orders` o ON o.order_id = p.order_id
),
cte2 AS
(
  SELECT Month, payment_type, COUNT(*) AS order_count
  FROM cte1
  GROUP BY 1, 2
  ORDER BY 1
)
```



```
SELECT * FROM cte2
```

Row	Month	payment_type	order_count
1	1	voucher	477
2	1	credit_card	6103
3	1	debit_card	118
4	1	UPI	1715
5	2	credit_card	6609
6	2	voucher	424
7	2	UPI	1723
8	2	debit_card	82
9	3	voucher	591
10	3	credit_card	7707

2. Count of orders based on the no. of payment instalments

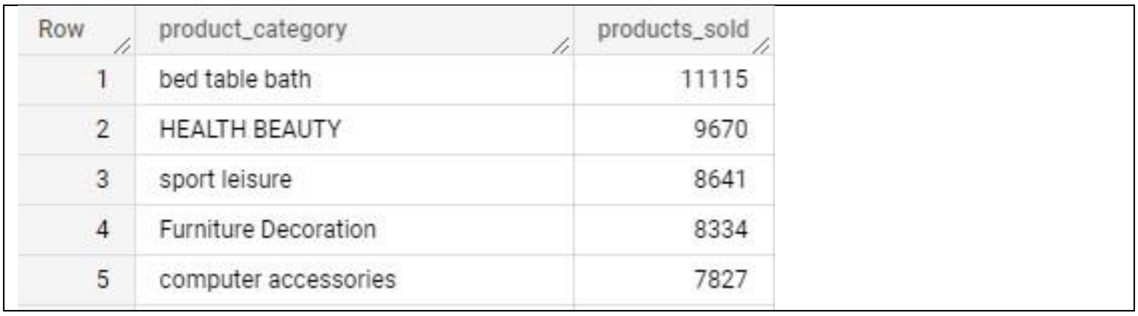


```
SELECT payment_installments, COUNT(*) AS count_of_orders
FROM `target-sql-368610.target.payments`
GROUP BY 1
```

Row	payment_install	count_of_orders
1	0	2
2	1	52546
3	2	12413
4	3	10461
5	4	7098
6	5	5239
7	6	3920
8	7	1626
9	8	4268
10	9	644

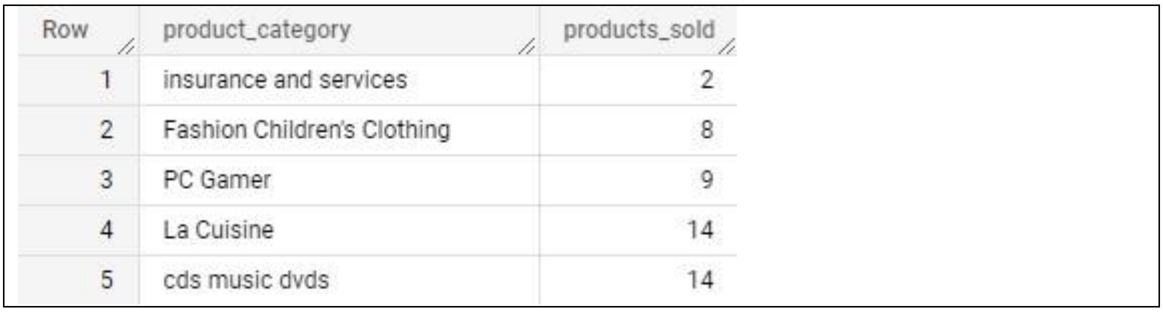
7. Insights

1. Top 5 product categories sold on Target



Row	product_category	products_sold
1	bed table bath	11115
2	HEALTH BEAUTY	9670
3	sport leisure	8641
4	Furniture Decoration	8334
5	computer accessories	7827

2. Least sold product categories on Target

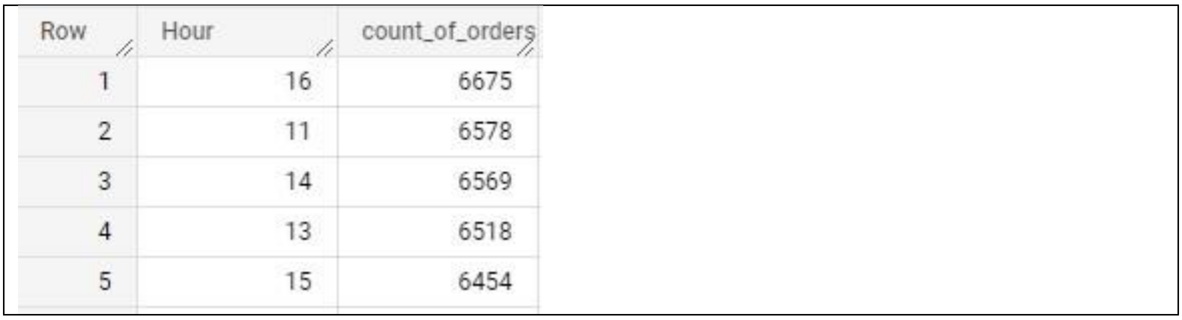


Row	product_category	products_sold
1	insurance and services	2
2	Fashion Children's Clothing	8
3	PC Gamer	9
4	La Cuisine	14
5	cds music dvds	14

3. The time to delivery is high in all cases which might lead to increased customer churn rate.
4. There is a wide gap between estimated delivery date and actual delivery date owing to which the customer might get uncertain about receiving the order leading to cancellations.
5. Most people who have paid in instalments have mostly paid 1 instalment. While the no of instalments more than 1 or 2 was opted by a fraction of people.

8. Recommendations

1. Ads spend on products belonging to top sold product categories can be increased while decreasing ad spend on least sold product categories
2. Delivery time should be reduced significantly to cater to customer satisfaction. The estimated time should also be calculated more accurately to give the customer a better picture about when their order will arrive.
3. Zero cost instalments can be introduced to encourage customers to buy more without worrying about a huge one-time cost, thereby increasing the company revenue and sales in longer term.
4. Increase footprint in states with highest sales in terms of physical store or online presence.
5. More ads can be shown during peak hours during the day. Peak hours as follows



Row	Hour	count_of_orders
1	16	6675
2	11	6578
3	14	6569
4	13	6518
5	15	6454