

```
1 ---1.Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset:---
2
3 ---1.1.Data type of all columns in the "customers" table.---
4 select * from Target.INFORMATION_SCHEMA.COLUMNS where TABLE_NAME='orders';
5 select * from Target.INFORMATION_SCHEMA.COLUMNS where TABLE_NAME='customers';
6 select * from Target.INFORMATION_SCHEMA.COLUMNS where TABLE_NAME='geolocation';
7 select * from Target.INFORMATION_SCHEMA.COLUMNS where TABLE_NAME='order_items';
8 select * from Target.INFORMATION_SCHEMA.COLUMNS where TABLE_NAME='order_reviews';
9 select * from Target.INFORMATION_SCHEMA.COLUMNS where TABLE_NAME='payments';
10 select * from Target.INFORMATION_SCHEMA.COLUMNS where TABLE_NAME='products';
11 select * from Target.INFORMATION_SCHEMA.COLUMNS where TABLE_NAME='sessions';
```

Press Alt+F1 for accessibility options.

Query results

SAVE RESULTS EXPLORE DATA

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	EXECUTION GRAPH	
Row	table_catalog	table_schema	table_name	column_name	ordinal_position	is_nullable
1	my-target-project-390115	Target	customers	customer_id	1	YES
2	my-target-project-390115	Target	customers	customer_unique_id	2	YES
3	my-target-project-390115	Target	customers	customer_zip_code_prefix	3	YES
4	my-target-project-390115	Target	customers	customer_city	4	YES
5	my-target-project-390115	Target	customers	customer_state	5	YES

Activate Windows
Go to Settings to activate Windows.

PERSONAL HISTORY PROJECT HISTORY

REFRESH

```
13 ---1.2 Get the time range between which the orders were placed.---
14 select max(order_purchase_timestamp) as end_date,min(order_purchase_timestamp) as start_date
15 from Target.orders
16
```

Press Alt+F1 for accessibility options.

Query results

SAVE RESULTS EXPLORE DATA

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	EXECUTION GRAPH
Row	end_date	start_date			
1	2018-10-17 17:30:18 UTC	2016-09-04 21:15:19 UTC			

Activate Windows
Go to Settings to activate Windows.

PERSONAL HISTORY PROJECT HISTORY

REFRESH

```
17 ---1.3 Count the number of Cities and States in our dataset.---
18 select count(distinct geolocation_city) as city_count, count(distinct geolocation_state) as state_count
19 from 'Target.geolocation'
20
```

Query results

[SAVE RESULTS](#) [EXPLORE DATA](#)

JOB INFORMATION RESULTS JSON EXECUTION DETAILS EXECUTION GRAPH

Row	city_count	state_count
1	8011	27

```
21
22 ---2.In-depth Exploration:---
23 ---2.1 Is there a growing trend in the no. of orders placed over the past years?---
24 SELECT extract(year from(order_purchase_timestamp)) as year,
25         extract(month from(order_purchase_timestamp)) as month,
26         count(*) as no_of_orders
27 from 'Target.orders'
28 where order_status='delivered'
29 group by year, month
30 order by year, month asc;
31
```

Query results

[SAVE RESULTS](#) [EXPLORE DATA](#)

JOB INFORMATION RESULTS JSON EXECUTION DETAILS EXECUTION GRAPH

Row	year	month	no_of_orders
1	2016	9	1
2	2016	10	265
3	2016	12	1
4	2017	1	750
5	2017	2	1653
6	2017	3	2546
7	2017	4	2303

Results per page: 50 1 - 23 of 23

“YES”, there is a growing trend in the number of orders as seeing by the month on month numbers.

---2.2 Can we see some kind of monthly seasonality in terms of the no. of orders being placed?---

- “YES”, It is observed that the winter season has the maximum sales(Oct.,Nov.,Dec.).

```

33
34 ---2.3 During what time of the day, do the Brazilian customers mostly place their orders? (Dawn, Morning, Afternoon or Night)
35 0-6 hrs : Dawn, 7-12 hrs : Mornings,13-18 hrs : Afternoon, 19-23 hrs : Night
36
37 SELECT
38 SUM(CASE
39     WHEN hour BETWEEN 0 AND 6 THEN volume
40     ELSE
41         0
42     END
43 ) AS Dawn,
44 SUM(CASE
45     WHEN hour BETWEEN 7 AND 12 THEN volume
46     ELSE
47         0
48     END
49 ) AS Morning,
50 SUM(CASE
51     WHEN hour BETWEEN 13 AND 18 THEN volume
52     ELSE
53         0
54     END
55 ) AS Afternoon,
56 SUM(CASE
57     WHEN hour BETWEEN 19 AND 23 THEN volume
58     ELSE
59         0
60     END
61 ) AS Night

```

Query results

PERSONAL HISTORY

PROJECT HISTORY

```

53 0
54 END
55 ) AS Afternoon,
56 SUM(CASE
57     WHEN hour BETWEEN 19 AND 23 THEN volume
58     ELSE
59         0
60     END
61 ) AS Night,|
62 FROM(
63     SELECT
64         EXTRACT (hour
65         FROM (order_purchase_timestamp)) AS hour,
66         count(DISTINCT order_id) AS volume
67     FROM
68         Target.orders
69     GROUP BY
70         hour
71     ORDER BY
72         hour);

```

Query results

JOB INFORMATION RESULTS JSON EXECUTION DETAILS EXECUTION GRAPH

Row	Dawn	Morning	Afternoon	Night
1	5242	27733	38135	28331

PERSONAL HISTORY

PROJECT HISTORY

The maximum number of transactions have happened in the afternoon (38135).

```

75 ---- 3. Evolution of E-commerce orders in the Brazil region:----
76 -- 3.1 Get the month on month no. of orders placed in each state?--
77
78 SELECT extract(year from(o.order_purchase_timestamp)) as year,
79         extract(month from(o.order_purchase_timestamp)) as month,
80         count(*) as no_of_orders,o.customer_state as state
81 from Target.orders as o
82 join Target.customers as c
83 on c.customer_id = o.customer_id
84 where o.order_status='delivered'
85 group by year,month,state
86 order by year, month,state asc;
87

```

```
75 |---- 3. Evolution of E-commerce orders in the Brazil region:-----
76 -- 3.1 Get the month on month no. of orders placed in each state?---
77
```

Press Alt+F1 for accessibility options

Query results

SAVE RESULTS

EXPLORE DATA



JOB INFORMATION RESULTS JSON EXECUTION DETAILS EXECUTION GRAPH

Row	year	month	no_of_orders	state
1	2016	9	1	SP
2	2016	10	1	AL
3	2016	10	3	BA
4	2016	10	6	CE
5	2016	10	6	DF
6	2016	10	3	ES
7	2016	10	7	GO
8	2016	10	4	MA
9	2016	10	35	MG
10	2016	10	1	MT
11	2016	10	4	PA
12	2016	10	1	PB

Results per page: 50 1 - 50 of 556

PERSONAL HISTORY

PROJECT HISTORY

REFRESH

```
87
88 ----- 3.2 How are the customers distributed across all the states?-----
89
90 SELECT count(*) as no_of_customers, customer_state
91 FROM Target.customers
92 GROUP BY customer_state
93 ORDER BY customer_state asc;
94
```

Press Alt+F1 for accessibility options

Query results

SAVE RESULTS

EXPLORE DATA



JOB INFORMATION RESULTS JSON EXECUTION DETAILS EXECUTION GRAPH

Row	no_of_customers	customer_state
1	81	AC
2	413	AL
3	148	AM
4	68	AP
5	3380	BA
6	1336	CE
7	2140	DF
8	2033	ES

Load more

Results per page: 50 1 - 27 of 27

PERSONAL HISTORY

PROJECT HISTORY

REFRESH

```

94
95 ---4. Impact on Economy: Analyze the money movement by e-commerce by looking at order prices, freight and others---
96 ---4.1 Get the % increase in the cost of orders from year 2017 to 2018 (include months between Jan to Aug only).
97 You can use the "payment_value" column in the payments table to get the cost of orders,----
98

```

```

99 WITH
100 yearly_revenue AS (
101   SELECT
102     EXTRACT (year
103     FROM
104       o.order_purchase_timestamp) AS year,
105     SUM(p.payment_value) AS revenue,
106   FROM
107     'Target.orders' AS o
108   JOIN
109     'Target.payments' AS p
110   ON
111     o.order_id = p.order_id
112   WHERE
113     o.order_status = 'delivered'
114     AND EXTRACT (month
115     FROM
116       o.order_purchase_timestamp) BETWEEN 0
117     AND 8
118   GROUP BY
119     year),
120 prev_year_revenue AS (
121   SELECT
122     *

```

Press Alt+F1 for accessibility options.

```

118   GROUP BY
119     year),
120 prev_year_revenue AS (
121   SELECT
122     *,
123     LAG(revenue) OVER (ORDER BY year ASC) AS prev_revenue
124   FROM
125     yearly_revenue)
126   SELECT
127     *,
128     (revenue-prev_revenue)/prev_revenue*100 AS per_increase
129   FROM
130     prev_year_revenue;
131

```

Press Alt+F1 for accessibility options.

Query results

SAVE RESULTS

EXPLORE DATA



JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS		EXECUTION GRAPH
Row	year	revenue	prev_revenue	per_increase		
1	2018	8452975.199999...	3473862.760000...	143.3307181081...		
2	2017	3473862.760000...	null	null		

Activate Windows

Go to Settings to activate Windows.

PERSONAL HISTORY

PROJECT HISTORY

REFRESH



```

131 |
132 | ---4.2.Calculate the Total & Average value of order price for each state.
133 | SELECT sum(OI.price) as total, AVG(OI.price) as Average, c.customer_state
134 | FROM 'Target.order_items' as OI
135 | join 'Target.orders' as O
136 | on OI.order_id = O.order_id
137 | join 'Target.customers' as C
138 | on C.customer_id = O.customer_id
139 | GROUP BY C.customer_state
140 | ORDER BY C.customer_state asc;

```

Press Alt+F1 for accessibility options.

Query results

SAVE RESULTS

EXPLORE DATA



JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	EXECUTION GRAPH
Row	total	Average	customer_state		
1	15982.94999999...	173.7277173913...	AC		
2	80314.80999999...	180.8892117117...	AL		
3	22356.84000000...	135.4959999999...	AM		
4	13474.29999999...	164.3207317073...	AP		
5	511349.9900000...	134.6012082126...	BA		
6	227254.7099999...	153.7582611637...	CE		
7	302603.9399999...	125.7705486284...	DF		
8	275037.3099999...	121.9137012411...	ES		

Results per page: 50 1 - 27 of 27

Activate Windows
Go to Settings to activate Windows.

PERSONAL HISTORY

PROJECT HISTORY

REFRESH

```

142 | ---4.3 Calculate the Total & Average value of order freight for each state.---
143 |
144 | SELECT sum(OI.freight_value) as total, AVG(OI.freight_value) as Average, C.customer_state
145 | FROM 'Target.order_items' as OI
146 | JOIN 'Target.orders' as O
147 | ON OI.order_id = O.order_id
148 | JOIN 'Target.customers' as C
149 | ON C.customer_id = O.customer_id
150 | GROUP BY C.customer_state
151 | ORDER BY C.customer_state asc;

```

Press Alt+F1 for accessibility option

Query results

SAVE RESULTS

EXPLORE DATA



JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	EXECUTION GRAPH
Row	total	Average	customer_state		
1	3686.750000000...	40.07336956521...	AC		
2	15914.58999999...	35.84367117117...	AL		
3	5478.890000000...	33.20539393939...	AM		
4	2788.500000000...	34.00609756097...	AP		
5	100156.6799999...	26.36395893656...	BA		
6	48351.58999999...	32.71420162381...	CE		
7	50625.49999999...	21.04135494596...	DF		
8	49764.59999999...	22.05877659574...	ES		
9	53114.97999999...	22.76681525932...	GO		

Results per page: 50 1 - 27 of 27

Activate Windows
Go to Settings to activate Windows.

```

152
153 --- 5. Analysis based on sales, freight and delivery time ---
154 --- 5.1 Find the no. of days taken to deliver each order from the order's purchase date as delivery time.
155 Also, calculate the difference (in days) between the estimated & actual delivery date of an order.
156 Do this in a single query.
157
158 SELECT DATE_DIFF(order_purchase_timestamp, order_delivered_customer_date, DAY) AS time_to_delivery,
159        | | DATE_DIFF(order_estimated_delivery_date, order_delivered_customer_date, DAY) AS diff_estimated_delivery
160 FROM Target.orders
161
162

```

Press Alt+F1 for accessibility options

Query results

SAVE RESULTS

EXPLORE DATA



JOB INFORMATION RESULTS JSON EXECUTION DETAILS EXECUTION GRAPH

Row	time_to_delivery	diff_estimated_delivery
1	-30	-12
2	-30	28
3	-35	16
4	-30	1
5	-32	0
6	-29	1
7	-43	-4
8	-40	-4

Load more

Activate Windows

Go to Settings to activate Windows.

Results per page: 50 1 - 50 of 99441

```

163 --- 5.2 Find out the top 5 states with the highest & lowest average freight value. ---
164
165 SELECT c.customer_state, AVG(OI.freight_value) as mean_freight_value
166 FROM Target.orders as O
167 join Target.order_items as OI
168 ON O.order_id = OI.order_id
169 join Target.customers as C
170 ON C.customer_id = O.customer_id
171 GROUP BY c.customer_state
172 ORDER BY c.customer_state, mean_freight_value desc
173 LIMIT 5
174

```

Press Alt+F1 for accessibility options

Query results

SAVE RESULTS

EXPLORE DATA



JOB INFORMATION RESULTS JSON EXECUTION DETAILS EXECUTION GRAPH

Row	customer_state	mean_freight_value
1	AC	40.07336956521...
2	AL	35.84367117117...
3	AM	33.20539393939...
4	AP	34.00609756097...
5	BA	26.36395893656...

2023-06-18 21:11:29

RUN

SAVE

SHARE

SCHEDULE

MORE

```

175 SELECT c.customer_state, AVG(OI.freight_value) as mean_freight_value
176 FROM Target.orders as O
177 join Target.order_items as OI
178 ON O.order_id = OI.order_id
179 join Target.customers as C
180 ON C.customer_id = O.customer_id
181 GROUP BY c.customer_state
182 ORDER BY c.customer_state, mean_freight_value asc
183 LIMIT 5

```

Press Alt+F1 for accessibility options

Query results

SAVE RESULTS

EXPLORE DATA



JOB INFORMATION RESULTS JSON EXECUTION DETAILS EXECUTION GRAPH

Row	customer_state	mean_freight_value
1	AC	40.07336956521...
2	AL	35.84367117117...
3	AM	33.20539393939...
4	AP	34.00609756097...
5	BA	26.36395893656...


```

185 --- 5.3 Find out the top 5 states with the highest & lowest average delivery time.---|
186
187
188 SELECT c.customer_state,
189        AVG(DATE_DIFF(order_purchase_timestamp,order_delivered_customer_date, DAY)) AS time_to_delivery,
190        AVG(DATE_DIFF(order_estimated_delivery_date,order_delivered_customer_date,DAY)) AS diff_estimated_delivery
191 FROM   `Target.orders` as O
192 join   `Target.order_items` as OI
193 ON OI.order_id = OI.order_id
194 join   `Target.customers` as C
195 on C.customer_id = O.customer_id
196 GROUP BY c.customer_id,c.customer_state
197 ORDER BY c.customer_state,time_to_delivery desc
198 LIMIT 5

```

Press Alt+F1 for accessibility options.

Query results

SAVE RESULTS

EXPLORE DATA



JOB INFORMATION RESULTS JSON EXECUTION DETAILS EXECUTION GRAPH

Row	customer_state	time_to_delivery	diff_estimated_delivery	
1	AC	-7.0	26.0	
2	AC	-9.0	28.0	
3	AC	-10.0	33.0	
4	AC	-11.0	22.0	
5	AC	-11.0	28.0	

Activate Windows

2023-06-18 21:11:29

RUN

SAVE

SHARE

SCHEDULE

MORE

```

199
200 SELECT c.customer_state,
201        AVG(DATE_DIFF(order_purchase_timestamp,order_delivered_customer_date, DAY)) AS time_to_delivery,
202        AVG(DATE_DIFF(order_estimated_delivery_date,order_delivered_customer_date,DAY)) AS diff_estimated_delivery
203 FROM   `Target.orders` as O
204 join   `Target.order_items` as OI
205 ON OI.order_id = OI.order_id
206 join   `Target.customers` as C
207 on C.customer_id = O.customer_id
208 GROUP BY c.customer_id,c.customer_state
209 ORDER BY c.customer_state,time_to_delivery asc
210 LIMIT 5
211

```

Press Alt+F1 for accessibility options.

Query results

SAVE RESULTS

EXPLORE DATA



JOB INFORMATION RESULTS JSON EXECUTION DETAILS EXECUTION GRAPH

Row	customer_state	time_to_delivery	diff_estimated_delivery	
1	AC	null	null	
2	AC	-72.0	-31.0	
3	AC	-66.0	-24.0	
4	AC	-42.0	3.0	
5	AC	-41.0	-1.0	


```

212 --- 5.4 Find out the top 5 states where the order delivery is really fast as compared to the estimated date of delivery.
213 You can use the difference between the averages of actual & estimated delivery date to figure out how fast the delivery was for each state.
214
215 SELECT
216   C.customer_state,
217   count(DATE_DIFF(o.order_estimated_delivery_date,o.order_delivered_customer_date, DAY)) AS fast_delivery
218 FROM
219   'Target.orders' AS o
220 JOIN
221   'Target.customers' AS c
222 ON
223   c.customer_id = o.customer_id
224 group by c.customer_state
225 ORDER BY
226   fast_delivery desc
227 LIMIT 5;

```

Press Alt+F1 for accessibility options.

Query results

[SAVE RESULTS](#) [EXPLORE DATA](#)

JOB INFORMATION RESULTS JSON EXECUTION DETAILS EXECUTION GRAPH

Row	customer_state	fast_delivery
1	SP	40495
2	RJ	12353
3	MG	11355
4	RS	5344
5	PR	4923

Activate Windows
Go to Settings to activate Windows.

```

228
229
230 SELECT
231   C.customer_state,
232   count(DATE_DIFF(o.order_estimated_delivery_date,o.order_delivered_customer_date, DAY)>=0) AS fast_delivery
233 FROM
234   'Target.orders' AS o
235 JOIN
236   'Target.customers' AS c
237 ON
238   c.customer_id = o.customer_id
239 group by c.customer_state
240 ORDER BY
241   fast_delivery desc
242 LIMIT 5;

```

Press Alt+F1 for accessibility options.

Query results

[SAVE RESULTS](#) [EXPLORE DATA](#)

JOB INFORMATION RESULTS JSON EXECUTION DETAILS EXECUTION GRAPH

Row	customer_state	fast_delivery
1	SP	40495
2	RJ	12353
3	MG	11355
4	RS	5344
5	PR	4923

Activate Windows
Go to Settings to activate Windows.

```

245 ---6. Analysis based on the payments---
246 --- 6.1 Find the month on month no. of orders placed using different payment types ---
247
248 SELECT extract(year from(o.order_purchase_timestamp)) as year,
249        extract(month from(o.order_purchase_timestamp)) as month,
250        count(*) as no_of_orders,p.payment_type as payment_type
251 FROM   'Target.orders' as o
252 JOIN   'Target.payments' as p
253 ON     p.order_id=o.order_id
254 WHERE  o.order_status='delivered'
255 GROUP BY year,month,payment_type
256 ORDER BY year,month,payment_type asc;
257

```

Press Alt+F1 for accessibility options.

Query results

SAVE RESULTS

EXPLORE DATA



JOB INFORMATION RESULTS JSON EXECUTION DETAILS EXECUTION GRAPH

Row	year	month	no_of_orders	payment_type
1	2016	10	51	UPI
2	2016	10	209	credit_card
3	2016	10	2	debit_card
4	2016	10	20	voucher
5	2016	12	1	credit_card
6	2017	1	188	UPI
7	2017	1	542	credit_card
8	2017	1	9	debit_card

Activate Windows
Go to Settings to activate Windows.

Results per page: 50 1 - 50 of 85 |< < > >|

```

257
258 --- 6.2 Find the no. of orders placed on the basis of the payment installments that have been paid. ---
259 SELECT count(*) as no_of_orders,p.payment_installments as payment_installments
260 FROM   'Target.orders' as o
261 JOIN   'Target.payments' as p
262 ON     p.order_id=o.order_id
263 GROUP BY p.payment_installments
264 ORDER BY p.payment_installments asc;
265

```

Press Alt+F1 for accessibility options.

Query results

SAVE RESULTS

EXPLORE DATA



JOB INFORMATION RESULTS JSON EXECUTION DETAILS EXECUTION GRAPH

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

Activate Windows
Go to Settings to activate Windows.

Results per page: 50 1 - 24 of 24 |< < > >|

