Business Case: Target SQL

24 October 2022 19:27

1. Initial exploration of dataset like checking the characteristics of data

1. data type of columns in a table

Field name	Type	Mode	C
order_id	STRING	NULLABLE	
customer_id	STRING	NULLABLE	
order_status	STRING	NULLABLE	
order_purchase_timestamp	TIMESTAMP	NULLABLE	
order_approved_at	TIMESTAMP	NULLABLE	
order_delivered_carrier_date	TIMESTAMP	NULLABLE	
order_delivered_customer_date	TIMESTAMP	NULLABLE	
order_estimated_delivery_date	TIMESTAMP	NULLABLE	

2. Time period for which the data is given

```
select min(order_purchase_timestamp) as start_date, max(order_purchase_timestamp) as last_date,
date_diff(max(order_purchase_timestamp),min(order_purchase_timestamp), day) as time_period_in_days
from target.orders;
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH | PREVIEW

Row	start_date	last_date	time_period_in_days
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC	772

3. Cities and States covered in the dataset

```
select distinct c.customer_city as cities,
c.customer_state as states from target.orders o join target.customers c
on o.customer_id = c.customer_id
order by cities , states;
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECU

Row	cities	states
1	abadia dos dourados	MG
2	abadiania	GO
3	abaete	MG
4	abaetetuba	PA
5	abaiara	CE
6	abaira	BA
7	abare	BA
8	abatia	PR
9	abdon batista	SC
10	abelardo luz	SC
11	abrantes	BA
12	abre campo	MG
13	abreu e lima	PE
14	acaiaca	MG
15	acailandia	MA

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

```
6  select count(order_id) as number_of_orders,
7  extract( month from order_purchase_timestamp) as order_month,
8  extract(year from order_purchase_timestamp) as order_year
9  from target.orders
10 group by order_year, order_month
11 order by order_year;
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	number_of_...	order_month	order_year
1	324	10	2016
2	4	9	2016
3	1	12	2016
4	7544	11	2017
5	5673	12	2017
6	2404	4	2017
7	4026	7	2017
8	4631	10	2017
9	3245	6	2017
10	4285	9	2017
11	1780	2	2017
12	800	1	2017
13	4331	8	2017
14	2682	3	2017
15	3700	5	2017

Exploring the resultant data in excel, we see that there is a increase trend in number of orders year on year basis.

SUM of number_of_orders	*order_year*		
order_month	2016	2017	2018
1		800	7269
2		1780	6728
3		2682	7211
4		2404	6939
5		3700	6873
6		3245	6167
7		4026	6292
8		4331	6512
9	4	4285	16
10	324	4631	4
11		7544	
12	1	5673	
sum of orders	329	45101	54011

	A	B	C	D
1	*SUM of number_of_orders*	*order_year*		
2	*order_month*	2016	2017	2018
3	1		800	7269
4	2		1780	6728
5	3		2682	7211
6	4		2404	6939
7	5		3700	6873
8	6		3245	6167
9	7		4026	6292
10	8		4331	6512
11	9	4	4285	16
12	10	324	4631	4
13	11		7544	
14	12	1	5673	
15	sum of orders	329	45101	54011
16				

Judging from the data available , we see highest peak of number of orders in month of October-November 2017 to January 2018 (towards er

2. What time do Brazilian customers tend to buy (Dawn, Morning, Afternoon or Night)?
   Mostly in afternoon and followed by night, morning and least at dawn respectively

```sql
select sum(orders) as num_of_orders, time_of_day from (
  select count(*) as orders,order_purchase_timestamp,
  extract(hour from order_purchase_timestamp) as hour_purchased,
  CASE
    WHEN extract(hour from order_purchase_timestamp) between 0 and 6 then 'dawn'
    when extract(hour from order_purchase_timestamp) between 6 and 11 then 'morning'
    when extract(hour from order_purchase_timestamp) between 12 and 17 then 'afternoon'
    else 'night'
  end
  as time_of_day
  from orders
  where order_purchase_timestamp is not null
  group by time_of_day,order_purchase_timestamp
  order by  orders
)x
group by x.time_of_day
```

100% 23:52 4 errors found

Result Grid Filter Rows: Search Export:

num_of_orders	time_of_day
21738	morning
38361	afternoon
34100	night
5242	dawn

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

1. Get month on month orders by region, states

```sql
select customer_state,sum(num_of_orders)as total_orders,order_year,order_month from (
select c.customer_state, extract( month from o.order_purchase_timestamp) as order_month,
extract(year from o.order_purchase_timestamp) as order_year,
count(*)  as num_of_orders
from orders o left join customers c
on o.customer_id = c.customer_id
group by c.customer_state, order_year,order_month,o.order_id
)x
group by order_year,order_month,customer_state
order by order_year,order_month,customer_state
```

100% 1:64 4 errors found

Result Grid Filter Rows: Search Export:

customer_state	total_orders	order_year	order_month
RR	1	2016	9
RS	1	2016	9
SP	2	2016	9
AL	2	2016	10
BA	4	2016	10
CE	8	2016	10
DF	6	2016	10
ES	4	2016	10
GO	9	2016	10
MA	4	2016	10
MG	40	2016	10
MT	3	2016	10
PA	4	2016	10
PB	1	2016	10
PE	7	2016	10
PI	1	2016	10
PR	19	2016	10
RJ	56	2016	10
RN	4	2016	10
RR	1	2016	10
RS	24	2016	10
SC	11	2016	10
SE	3	2016	10

2. How are customers distributed in Brazil?

State wise distribution of customers in Brazil:

```sql
select count(*) as total_customers, customer_state
from customers
group by customer_state
order by customer_state
```

100% 24:70 4 errors found

Result Grid Filter Rows: Search Export:

total_custom...	customer_state
81	AC
413	AL
148	AM
68	AP
3380	BA
1336	CE
2140	DF
2033	ES
2020	GO
747	MA
11635	MG
715	MS
907	MT
975	PA
536	PB
1652	PE
495	PI
5045	PR
12852	RJ
485	RN
253	RO
46	RR
5466	RS
3637	SC
350	SE
41746	SP
280	TO

4. Impact on 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)

The price of orders from 2017 to 2018 from Jan to Aug is 57.85

```
select total_price, order_year, total_price_2017, round((((total_price - total_price_2017)/total_price)*100),2) as per_price_increase from (
 select round(sum(oi.price),2) as total_price, extract( year from o.order_purchase_timestamp) as order_year,
 lag(round(sum(oi.price),2),1) over( order by extract( year from o.order_purchase_timestamp)) as total_price_2017
 from orders o left join order_items oi
 on o.order_id=oi.order_id
 where extract( year from o.order_purchase_timestamp) between 2017 and 2018 and extract( month from o.order_purchase_timestamp) between 1 and 8
 group by order_year
 )x
 where x.total_price_2017 is not null
```

1:81 4 errors found

Result Grid Filter Rows: Export:

total_price	order_year	total_price_2017	per_price_increase
7385905.8	2018	3113000.32	57.85

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

```
select c.customer_state,
round(sum(oi.price),2) as total_price,
round(sum(oi.freight_value),2) as total_frieght,
round(avg(oi.price),2) as mean_price,
round(avg(oi.freight_value),2) as mean_frieght
from orders o  left join order_items oi on o.order_id = oi.order_id
join customers c on o.customer_id = c.customer_id
group by c.customer_state
order by c.customer_state
```

100% 1:99 4 errors found

Result Grid Filter Rows: Search Export:

customer_state	total_price	total_frieght	mean_price	mean_frieght
AC	15982.95	3686.75	173.73	40.07
AL	80314.81	15914.59	180.89	35.84
AM	22356.84	5478.89	135.5	33.21
AP	13474.3	2788.5	164.32	34.01
BA	511349.99	100156.68	134.6	26.36
CE	227254.71	48351.59	153.76	32.71
DF	302603.94	50625.5	125.77	21.04
ES	275037.31	49764.6	121.91	22.06
GO	294591.95	53114.98	126.27	22.77
MA	119648.22	31523.77	145.2	38.26
MG	1585308....	270853.46	120.75	20.63
MS	116812.64	19144.03	142.63	23.37
MT	156453.53	29715.43	148.3	28.17
PA	178947.81	38699.3	165.69	35.83
PB	115268.08	25719.73	191.48	42.72
PE	262788.03	59449.66	145.51	32.92
PI	86914.08	21218.2	160.36	39.15
PR	683083.76	117851.68	119	20.53
RJ	1824092....	305589.31	125.12	20.96
RN	83034.98	18860.1	156.97	35.65
RO	46140.64	11417.38	165.97	41.07
RR	7829.43	2235.19	150.57	42.98
RS	750304.02	135522.74	120.34	21.74
SC	520553.34	89660.26	124.65	21.47
SE	58920.85	14111.47	153.04	36.65
SP	5202955....	718723.07	109.65	15.15
TO	49621.74	11732.68	157.53	37.25

5. Analysis on sales, freight and delivery time
   1. Calculate days between purchasing, delivering and estimated delivery

```
select order_id, order_purchase_timestamp as purchased,
order_estimated_delivery_date as est_delivery, order_delivered_customer_date as cust_delivered,
datediff(order_delivered_customer_date,order_purchase_timestamp)as "diff(purchase,delivered)",
datediff(order_estimated_delivery_date,order_purchase_timestamp)as "diff(purchase,est.delivery)"
from orders
```

100% 98:112 4 errors found

Result Grid Filter Rows: Search Export: Fetch rows:

order_id	purchased	est_delivery	cust_delivered	diff(purchase,delivered)	diff(purchase,est.delivery)
e481f51cbdc54678b7cc49136f2d6af7	2017-10-02 10:56:33	2017-10-18 00:00:00	2017-10-10 21:25:13	8	16
53cdb2fc8bc7dce0b6741e2150273451	2018-07-24 20:41:37	2018-08-13 00:00:00	2018-08-07 15:27:45	14	20
47770eb9100c2d0c44946d9cf07ec65d	2018-08-08 08:38:49	2018-09-04 00:00:00	2018-08-17 18:06:29	9	27
949d5b44dbf5de918fe9c16f97b45f8a	2017-11-18 19:28:06	2017-12-15 00:00:00	2017-12-02 00:28:42	14	27
ad21c59c0840e6cb83a9ceb5573f8159	2018-02-13 21:18:39	2018-02-26 00:00:00	2018-02-16 18:17:02	3	13
a4591c265e18cb1dcee52889e2d8acc3	2017-07-09 21:57:05	2017-08-01 00:00:00	2017-07-26 10:57:55	17	23
136cce7faa42fdb2cefd53fdc79a6098	2017-04-11 12:22:08	2017-05-09 00:00:00		NULL	28
6514b8ad8028c9f2cc2374ded245783f	2017-05-16 13:10:30	2017-06-07 00:00:00	2017-05-26 12:55:51	10	22
76c6e866289321a7c93b82b54852dc33	2017-01-23 18:29:09	2017-03-06 00:00:00	2017-02-02 14:08:10	10	42
e69bfb5eb88e0ed6a785585b27e16dbf	2017-07-29 11:55:02	2017-08-23 00:00:00	2017-08-16 17:14:30	18	25
e6ce16cb79ec1d90b1da9085a6118aeb	2017-05-16 19:41:10	2017-06-07 00:00:00	2017-05-29 11:18:31	13	22
34513ce0c4fab462a55830c0989c7edb	2017-07-13 19:58:11	2017-08-08 00:00:00	2017-07-19 14:04:48	6	26
82566a660a982b15fb86e904c8d32918	2018-06-07 10:06:19	2018-07-18 00:00:00	2018-06-19 12:05:52	12	41
5ff96c15d0b717ac6ad1f3d77225a350	2018-07-25 17:44:10	2018-08-08 00:00:00	2018-07-30 15:52:25	5	14
432aaf21d85167c2c86ec954d8c4a42cc	2018-03-01 14:14:28	2018-03-21 00:00:00	2018-03-12 23:36:28	11	20
dcb36b511fcac050b97cd5c05de84dc3	2018-06-07 19:03:12	2018-07-04 00:00:00	2018-06-21 15:34:32	14	27
403b97836b0c04a622354cf531062e5f	2018-01-02 19:00:43	2018-02-06 00:00:00	2018-01-20 01:38:59	18	35
116f0b09343b49556bbad5f35beee0cf	2017-12-26 23:41:31	2018-01-29 00:00:00	2018-01-08 22:36:36	13	34
85ce859fd6dc634de8e2f1e290444043	2017-11-21 00:03:41	2017-12-11 00:00:00	2017-11-27 18:28:00	6	20
83018ec114eee8641c97e09f7b4e928f	2017-10-26 15:54:26	2017-11-23 00:00:00	2017-11-08 22:22:00	13	28
203096f03d82e0dffbc41ebc2e2bcfb7	2017-09-18 14:31:30	2017-09-28 00:00:00	2017-10-09 22:23:46	21	10
f848643eec1d69395095eb3840d2051e	2018-03-15 08:52:40	2018-03-29 00:00:00	2018-03-19 18:08:32	4	14
2807d0e504d6d4894d41672727bc139f	2018-02-03 20:37:35	2018-02-21 00:00:00	2018-02-08 16:13:46	5	18
95266cb57a203540abd0796fdac78d5	2018-01-08 07:55:29	2018-02-21 00:00:00	2018-01-26 17:32:38	18	44
f3e7c359154d9658273559f39d6b1fdac	2018-08-09 11:44:40	2018-08-17 00:00:00	2018-08-13 18:24:27	4	8
fbf9ac61453ac646ce8ad9783d7d0af6	2018-02-20 23:46:53	2018-03-12 00:00:00	2018-03-21 22:03:54	29	20
acce194856392f074dbf9dada14d8d82	2018-06-04 00:00:13	2018-07-18 00:00:00	2018-06-16 15:20:55	12	44
dd78f560c270f1909639c11b925620ea	2018-03-12 01:50:26	2018-03-28 00:00:00	2018-03-21 14:41:50	9	16

The average difference between the dates year on year decreased over time , the difference is more visible between 2016 and 2(
Specially in difference between estimated delivery date and actual delivered date suggesting that the estimated delivery date is n
Being auto generated more optimally and with better accuracy

```
select extract( year from order_purchase_timestamp) as order_year,
#datediff(order_delivered_customer_date,order_purchase_timestamp)as "diff(purchase,delivered)",
#datediff(order_estimated_delivery_date,order_purchase_timestamp)as "diff(purchase,est.delivery)",
avg(datediff(order_delivered_customer_date,order_purchase_timestamp))as "avg_diff(purchase,delivered)" ,
avg(datediff(order_estimated_delivery_date,order_purchase_timestamp))as "avg_diff(purchase,est.delivery)"
from orders
group by order_year
order by order_year
```

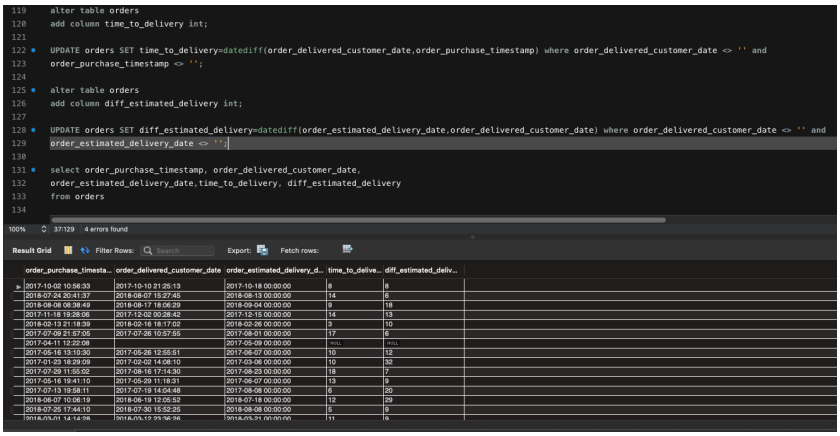
100% 1:117 4 errors found

Result Grid Filter Rows: Search Export:

order_year	avg_diff(purchase,deliver...	avg_diff(purchase,est.deliv...
2016	19.6875	55.5471
2017	12.9790	25.3527
2018	12.0639	23.4220

   2. Create columns:

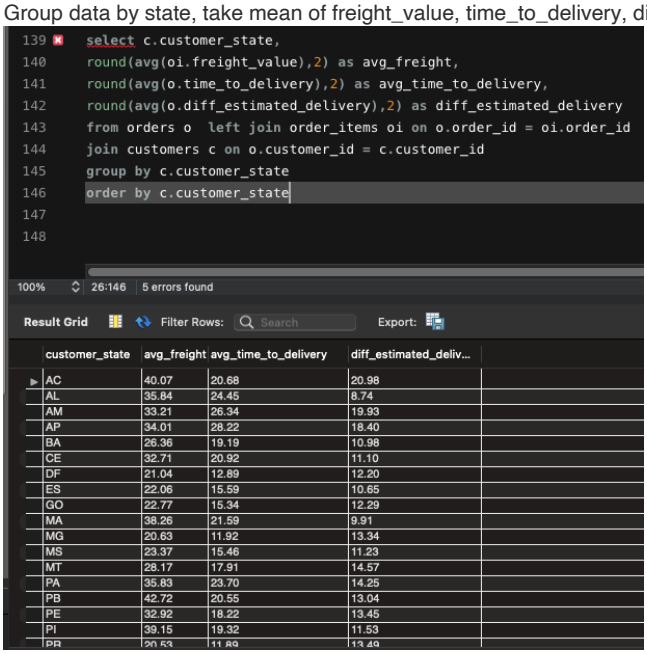
- time_to_delivery = order_purchase_timestamp-order_delivered_customer_date
- diff_estimated_delivery = order_estimated_delivery_date-order_delivered_customer_date



```
alter table orders
add column time_to_delivery int;

UPDATE orders SET time_to_delivery=datediff(order_delivered_customer_date,order_purchase_timestamp) where order_delivered_customer_date <> '' and
order_purchase_timestamp <> '';

alter table orders
add column diff_estimated_delivery int;

UPDATE orders SET diff_estimated_delivery=datediff(order_estimated_delivery_date,order_delivered_customer_date) where order_delivered_customer_date <> '' and
order_estimated_delivery_date <> '';

select order_purchase_timestamp, order_delivered_customer_date,
order_estimated_delivery_date,time_to_delivery, diff_estimated_delivery
from orders
```

order_purchase_timesta...	order_delivered_customer_date	order_estimated_delivery_d...	time_to_delive...	diff_estimated_deliv...
2017-10-02 10:56:33	2017-10-10 21:25:13	2017-10-18 00:00:00	8	8
2018-07-24 20:41:37	2018-08-07 15:27:45	2018-08-13 00:00:00	14	6
2018-08-08 08:38:49	2018-08-17 18:06:29	2018-09-04 00:00:00	9	18
2017-11-18 19:28:06	2017-12-02 00:28:42	2017-12-15 00:00:00	14	13
2018-02-13 21:18:39	2018-02-16 18:17:02	2018-02-26 00:00:00	3	10
2017-07-09 21:57:05	2017-07-26 10:57:55	2017-08-01 00:00:00	17	6
2017-04-11 12:22:08		2017-03-09 00:00:00	NULL	NULL
2017-05-16 13:10:30	2017-05-26 12:55:51	2017-06-07 00:00:00	10	12
2017-01-23 18:29:09	2017-02-02 14:08:10	2017-03-06 00:00:00	10	32
2017-07-29 11:55:02	2017-08-10 17:14:30	2017-08-23 00:00:00	18	7
2017-05-16 19:41:10	2017-05-29 11:18:31	2017-06-07 00:00:00	13	9
2017-07-13 19:58:11	2017-07-19 14:04:48	2017-08-08 00:00:00	6	20
2018-06-07 10:06:19	2018-06-19 13:05:52	2018-07-18 00:00:00	12	29
2018-07-25 17:44:10	2018-07-30 15:52:25	2018-08-08 00:00:00	5	9
2018-03-01 14:14:28	2018-03-12 23:38:36	2018-03-21 00:00:00	11	8

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

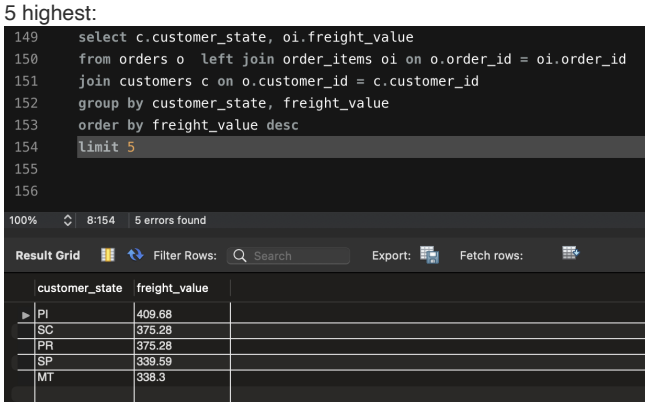


```
select c.customer_state,
round(avg(oi.freight_value),2) as avg_freight,
round(avg(o.time_to_delivery),2) as avg_time_to_delivery,
round(avg(o.diff_estimated_delivery),2) as diff_estimated_delivery
from orders o  left join order_items oi on o.order_id = oi.order_id
join customers c on o.customer_id = c.customer_id
group by c.customer_state
order by c.customer_state
```

customer_state	avg_freight	avg_time_to_delivery	diff_estimated_deliv...
AC	40.07	20.68	20.98
AL	35.84	24.45	8.74
AM	33.21	26.34	19.93
AP	34.01	28.22	18.40
BA	26.36	19.19	10.98
CE	32.71	20.92	11.10
DF	21.04	12.89	12.20
ES	22.06	15.59	10.65
GO	22.77	15.34	12.29
MA	38.26	21.59	9.91
MG	20.63	11.92	13.34
MS	23.37	15.46	11.23
MT	28.17	17.91	14.57
PA	35.83	23.70	14.25
PB	42.72	20.55	13.04
PE	32.92	18.22	13.45
PI	39.15	19.32	11.53
PR	20.53	11.89	13.49

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

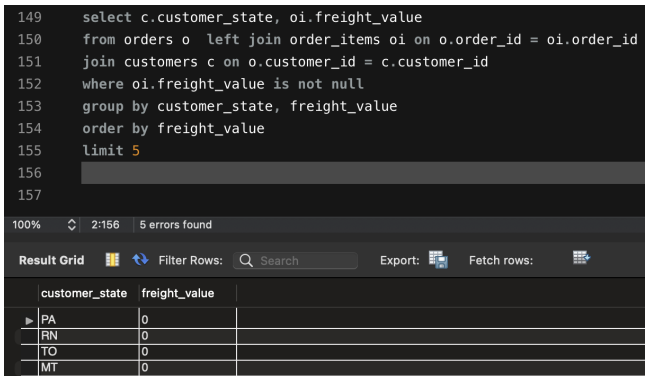
    5 highest:



```
select c.customer_state, oi.freight_value
from orders o  left join order_items oi on o.order_id = oi.order_id
join customers c on o.customer_id = c.customer_id
group by customer_state, freight_value
order by freight_value desc
limit 5
```

customer_state	freight_value
PI	409.68
SC	375.28
PR	375.28
SP	339.59
MT	338.3

    5 lowest:



```
select c.customer_state, oi.freight_value
from orders o  left join order_items oi on o.order_id = oi.order_id
join customers c on o.customer_id = c.customer_id
where oi.freight_value is not null
group by customer_state, freight_value
order by freight_value
limit 5
```

customer_state	freight_value
PA	0
RN	0
TO	0
MT	0

MA	0

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

   Top 5 state with lowest avg time to deliver:

```
149  select c.customer_state, avg(o.time_to_delivery)
150  from orders o  left join order_items oi on o.order_id = oi.order_id
151  join customers c on o.customer_id = c.customer_id
152  group by customer_state
153  order by avg(o.time_to_delivery)
154  limit 5
155
156
```

100% 2:155 5 errors found

Result Grid Filter Rows: Search Export: Fetch rows:

customer_state	avg(o.time_to_delive...
SP	8.6623
PR	11.8931
MG	11.9207
DF	12.8938
SC	14.9502

   Top 5 states with highest time_to_deliver:

```
149  select c.customer_state, avg(o.time_to_delivery)
150  from orders o  left join order_items oi on o.order_id = oi.order_id
151  join customers c on o.customer_id = c.customer_id
152  group by customer_state
153  order by avg(o.time_to_delivery) desc
154  limit 5
155
156
```

100% 8:154 5 errors found

Result Grid Filter Rows: Search Export: Fetch rows:

customer_state	avg(o.time_to_delive...
AP	28.2222
RR	28.1739
AM	26.3374
AL	24.4473
PA	23.7021

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

   5 states with really fast delivery:(on an avg over the years)

```
156  select c.customer_state, avg(o.diff_estimated_delivery)
157  from orders o  left join order_items oi on o.order_id = oi.order_id
158  join customers c on o.customer_id = c.customer_id
159  group by customer_state #,diff_estimated_delivery
160  order by avg(diff_estimated_delivery)
161  limit 5
162
163
```

100% 8:161 5 errors found

Result Grid Filter Rows: Search Export: Fetch rows:

customer_state	avg(o.diff_estimated_delivery)
AL	8.7354
MA	9.9063
SE	10.0027
ES	10.6463
BA	10.9826

   5 states with not so fast delivery: (on an avg over the years)

```
156  select c.customer_state, avg(o.diff_estimated_delivery)
157  from orders o  left join order_items oi on o.order_id = oi.order_id
158  join customers c on o.customer_id = c.customer_id
159  group by customer_state #,diff_estimated_delivery
160  order by avg(diff_estimated_delivery) desc
161  limit 5
162
163
```

100% 8:161 5 errors found

Result Grid Filter Rows: Search Export: Fetch rows:

customer_state	avg(o.diff_estimated_delivery)
AC	20.9780
RO	20.0403
AM	19.9325
AP	18.3951
RR	18.3261

   We can take a look at the logistics on the top 5 states with fastest delivery time on an average and can implement similar optimization c
   Portray the estimated delivery date as per actual delivery in the states with not so fast delivery.

6. Payment type analysis:
   1. Month over Month count of orders for different payment types

```
163  select p.payment_type, extract( year from order_purchase_timestamp) as order_year,
164  extract(month from o.order_purchase_timestamp) as order_month, count(*) as num_of_orders
165  from orders o left join payments p on o.order_id=p.order_id
166  where payment_type is not null
167  group by payment_type,order_year,  order_month
168  order by order_year,order_month
```

169

100% 33:168 5 errors found

Result Grid Filter Rows: Search Export:

payment_type	order_year	order_month	num_of_orders
credit_card	2016	9	3
debit_card	2016	10	2
voucher	2016	10	23
UPI	2016	10	63
credit_card	2016	10	254
credit_card	2016	12	1
voucher	2017	1	61
debit_card	2017	1	9
credit_card	2017	1	583
UPI	2017	1	197
credit_card	2017	2	1356
UPI	2017	2	398
debit_card	2017	2	13
voucher	2017	2	119
voucher	2017	3	200
credit_card	2017	3	2016
UPI	2017	3	590
debit_card	2017	3	31
credit_card	2017	4	1846
voucher	2017	4	202
debit_card	2017	4	27
UPI	2017	4	496

2. Distribution of payment instalments and count of orders

```
170  select payment_installments, count(*) as count_orders
171  from payments
172  group by payment_installments
173  order by payment_installments
174
```

100% 30:173 5 errors found

Result Grid Filter Rows: Search Export:

payment_installme...	count_orders
0	2
1	52546
2	12413
3	10461
4	7098
5	5239
6	3920
7	1626
8	4268
9	644
10	5328
11	23
12	133
13	16
14	15
15	74
16	5
17	8
18	27
20	17
21	3
22	1

Actionable Insights:

- The most orders comes around year end and new year. Target can scale up their infrastructure specially during this time and can sales even further
- Few states have considerable delta between estimated delivery date and actual delivery date. We can work with our logistics part network and reduce the time between order purchase and customer delivery. We can also tweak/optimize our estimated delivery delivery dates more accurately
- We see that there are many orders purchased via credit cards and we can partner with leading credit card provider in the country made via credit cards to even boost sales

Recommendations:

- To work with our delivery partners to reduce the time delta between purchase date and customer delivery date.
- For the states with highest freight charges , we can find alternate routes or work with freight institutions for exclusive partnerships time of delivery to customers
- Since there was a high YoY percentage of cost increase, we need to find ways to bring the costs down for a better sustainable bu on customer experience