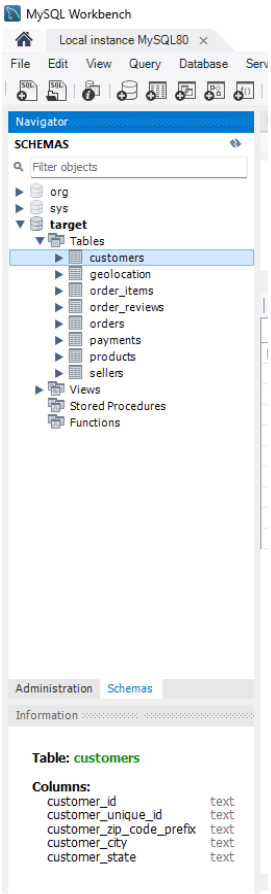
SQL Case Study No:1

Topic:
Business Case: Target Data

Creating db named target
Adding data in tables using bulk import command

```
set global local_infile=true;
LOAD DATA LOCAL INFILE
'C:/Users/k/Desktop/New folder (3)/sellers.csv'
INTO TABLE target.sellers
FIELDS TERMINATED BY ','
ENCLOSED BY ""
LINES TERMINATED BY '\n'
IGNORE 1 ROWS
(seller_id, seller_zip_code_prefix, seller_city, seller_state);
```

Did this for all tables



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

All the columns in all tables was either integer or text data even date were given in text
SELECT * FROM target.customers limit 10;

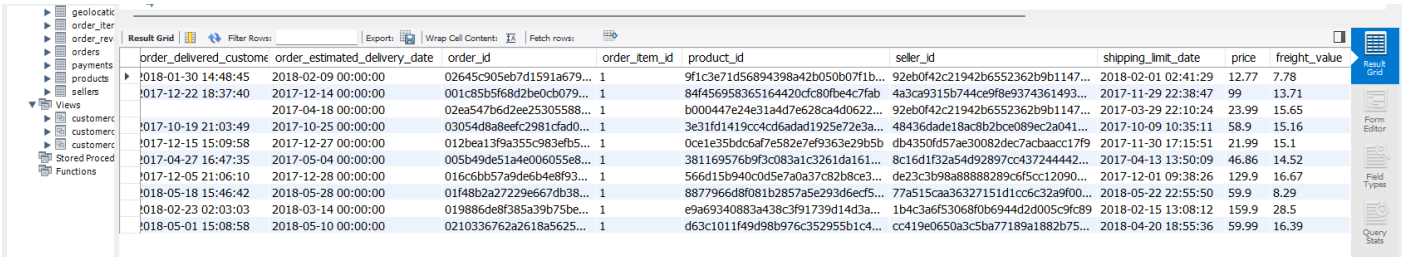
customer_id	customer_unique_id	customer_zip_code_prefix	customer_city	customer_state
06b8999e2fba1a1fbc88172c00ba8bc7	861eff4711a542e4b93843c6dd7febb0	14409	franca	SP
18955e83d337fd6b2def6b18a428ac...	290c77bc529b7ac935b93aa66c333...	09790	sao bernardo do campo	SP
4e7b3e00288586ebd08712fdd0374...	060e732b5b29e8181a18229c7b0b2...	01151	sao paulo	SP
b2b6027bc5c5109e529d4dc6358b1...	259dac757896d24d7702b9acbbff3f3c	08775	mogi das cruzes	SP
4f2d8ab171c80ec8364f7c12e35b23ad	345ecd01c38d18a9036ed96c73b8d...	13056	campinas	SP
879864dab9bc3047522c92c82e121...	4c937445166667ad3b8f1fb645a3116...	89254	jaragua do sul	SC
fd826e7cf63160e536e0908c76c3f441	addec96d2e059c80c30fe6871d30d1...	04534	sao paulo	SP
5e274e7a0c3809e14aba7ad5aae0d4...	57b2a98a409812fe9618067b6b8eb...	35182	timoteo	MG
5adf08e34b2e993982a47070956c5...	1175e95fb47ddff9de6b2b06188f7e0d	81560	curitiba	PR
4b7139f34592b3a31687243a302fa7...	9afe194fb833f79e300e37e580171f22	30575	belo horizonte	MG

SELECT * FROM target.customers as c
inner join target.orders as o on c.customer_id = o.customer_id limit 10;

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

customer_id	customer_unique_id	custome	customer_city	custome	order_id	customer_id	order_status	order_purchase_timestamp	order_approved_
41ce2a54c0b03bf3443c3d931a367...	3a653a41f6f9fc3d2a113cf8398680e8	75265	vianopolis	GO	47770eb9100c2d0c44946...	41ce2a54c0b03bf...	delivered	2018-08-08 08:38:49	2018-08-08 08:5
3b6828a50ffe546942b7a473d70ac0fc	ccafc1c3f270410521c3c6f3b249870f	74820	goiania	GO	dcb36b511fcac050b97cd...	3b6828a50ffe546...	delivered	2018-06-07 19:03:12	2018-06-12 23:3
d3e3b74c766bc6214e0c830b17ee2...	e97109680b052ee858d93a539597b...	35400	ouro preto	MG	82566a660a982b15fb86e...	d3e3b74c766bc6...	delivered	2018-06-07 10:06:19	2018-06-09 03:1
ae8db0691449a44352e7d535ddf78...	ad28944afc91824e30366a595654aa...	11440	quaruja	SP	f11b36b3bc7bacf06deef8...	ae8db0691449a4...	delivered	2018-05-15 09:10:20	2018-05-16 03:1
0489975a325480c9e385e9f135bb1...	b577af9a54b023b4bcc51f69bb937f1c	14406	franca	SP	1d067305b599c1e0dceb3...	0489975a325480...	delivered	2018-02-14 13:05:17	2018-02-14 13:1
7f8c8b9c2ae27bf3300f670c3d478be8	634f09f6075fe9032e6c19609ffe995a	44024	feira de santana	BA	83018ec114eee8641c97e...	7f8c8b9c2ae27bf...	delivered	2017-10-26 15:54:26	2017-10-26 16:0
d2b091571da224a1b36412c18bc3b...	d699688533772c15a061e8ce81cb5...	04001	sao paulo	SP	203096f03d82e0dffbc41e...	d2b091571da224...	delivered	2017-09-18 14:31:30	2017-09-19 04:0
569cf68214806a39acc0f39344aea67f	c2551ea089b7ebbc67a2ea8757152...	44380	cruz das almas	BA	d17dc4a904426827ca80f...	569cf68214806a...	delivered	2017-05-14 20:28:25	2017-05-14 20:4
5ad934ace4646e3dad70148a823eb...	795ec127906a9e161109399a70f58...	39480	januaria	MG	22ff7c206b8858435452...	5ad934ace4646e...	delivered	2017-04-27 10:24:22	2017-04-27 10:3
021e84751ba0ead75b6d314a6ead8...	9a58c55c10b4eb6c69fb0874d8beb...	23025	rio de janeiro	RJ	38b7efdf33dd5561f4f5d4f...	021e84751ba0ea...	invoiced	2017-08-01 18:17:41	2017-08-01 18:3

SELECT * FROM target.customers as c
inner join target.orders as o on c.customer_id = o.customer_id
left join target.order_items as i on o.order_id = i.order_id limit 10;



Counting important data

```
SELECT count(*) FROM target.sellers;
```

count(*)
3095

select count(customer_id) from target.customers;

count(customer_id)
99441

```
SELECT count(distinct c.customer_id) FROM target.customers as c
inner join target.orders as o on c.customer_id = o.customer_id
left join target.order_items as i on o.order_id = i.order_id limit 10;
```

count(distinct c.customer_id)
99441

2. Time period for which the data is given

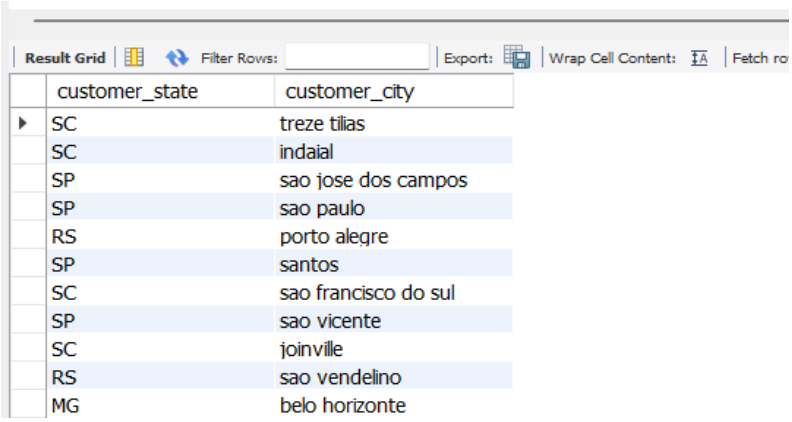
Order exact time period (to get datetime range of data)

```
SELECT min(order_purchase_timestamp),max(order_purchase_timestamp) FROM
target.customers as c
inner join target.orders as o on c.customer_id = o.customer_id
left join target.order_items as i on o.order_id = i.order_id limit 10;
```

min(order_purchase_timestamp)	max(order_purchase_timestamp)
2016-09-04 21:15:19	2018-10-17 17:30:18

3. Cities and States of customers ordered during the given period

```
SELECT customer_state, customer_city  FROM target.customers as c
inner join target.orders as o on c.customer_id = o.customer_id
left join target.order_items as i on o.order_id = i.order_id
group by customer_state, customer_city ;
```



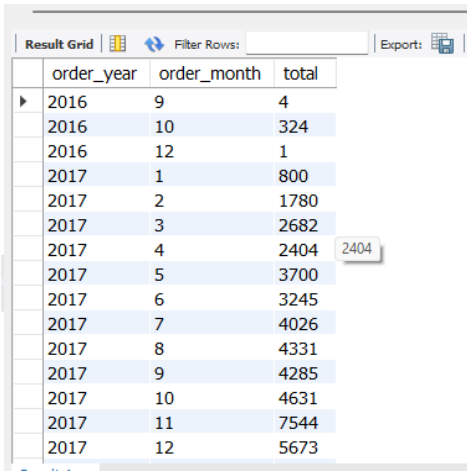
customer_state	customer_city
SC	treze tilias
SC	indaial
SP	sao jose dos campos
SP	sao paulo
RS	porto alegre
SP	santos
SC	sao francisco do sul
SP	sao vicente
SC	joinville
RS	sao vendelino
MG	belo horizonte

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?

Yes, there a growing trend on e-commerce in Brazil as the every month on month sales are increasing and after every few months the the sales peak mostly during year end or the during the winters

```
select order_year, order_month, count(distinct order_id) total   from customerOrderShort group by order_year, order_month order by order_year, order_month ;
```

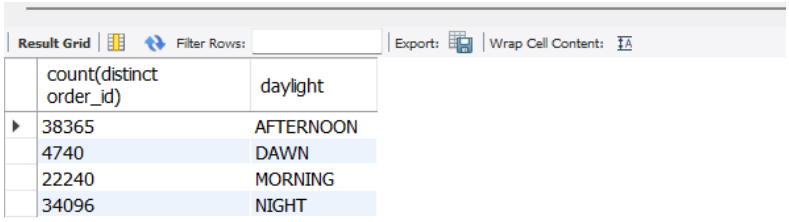


order_year	order_month	total
2016	9	4
2016	10	324
2016	12	1
2017	1	800
2017	2	1780
2017	3	2682
2017	4	2404
2017	5	3700
2017	6	3245
2017	7	4026
2017	8	4331
2017	9	4285
2017	10	4631
2017	11	7544
2017	12	5673

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

Afternoon and night time brazilian customers did the most shopping

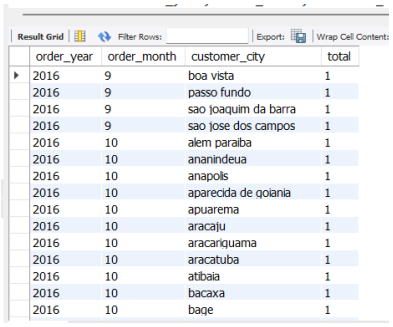
```
select count(distinct order_id),
case
when order_timestamp between '00:00:00' and  '06:00:00' then 'DAWN'
when order_timestamp >= '06:00:00' and order_timestamp <='12:00:00' then 'MORNING'
when order_timestamp >= '12:00:00' and order_timestamp <= '18:00:00' then 'AFTERNOON'
when order_timestamp >= '18:00:00' and order_timestamp <= '24:00:00' then 'NIGHT'
else 'day'
end as daylight from customerordershortdaytime group by daylight;
```



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

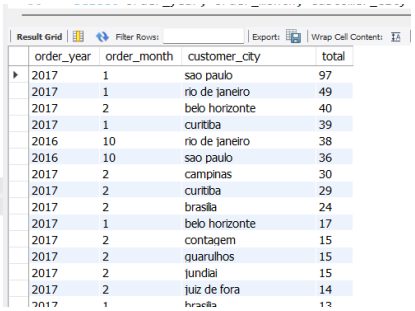
count(distinct order_id)	daylight
38365	AFTERNOON
4740	DAWN
22240	MORMING
34096	NIGHT

1. **Evolution of E-commerce orders in the Brazil region:**
    1. **Get month on month orders by states**



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

order_year	order_month	customer_city	total
2016	9	boa vista	1
2016	9	passo fundo	1
2016	9	sao joaquim da barra	1
2016	9	sao jose dos campos	1
2016	10	alem paraiba	1
2016	10	ananindeua	1
2016	10	anapolis	1
2016	10	aparecida de goiania	1
2016	10	apuarema	1
2016	10	aracaju	1
2016	10	aracariguama	1
2016	10	aracatuba	1
2016	10	atibaia	1
2016	10	bacaxa	1
2016	10	bage	1



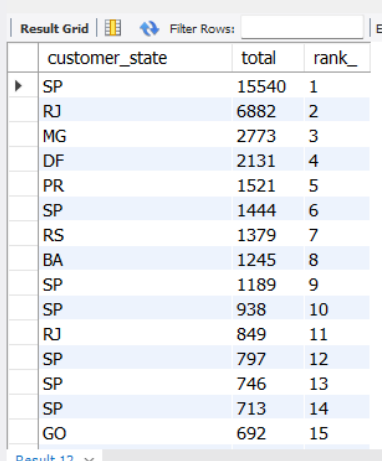
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

order_year	order_month	customer_city	total
2017	1	sao paulo	97
2017	1	rio de janeiro	49
2017	2	belo horizonte	40
2017	1	curitiba	39
2016	10	rio de janeiro	38
2016	10	sao paulo	36
2017	2	campinas	30
2017	2	curitiba	29
2017	2	brasilia	24
2017	1	belo horizonte	17
2017	2	contagem	15
2017	2	guarulhos	15
2017	2	jundiai	15
2017	2	juiz de fora	14

2. **Distribution of customers across the states in Brazil**

Sao paulo had max orders

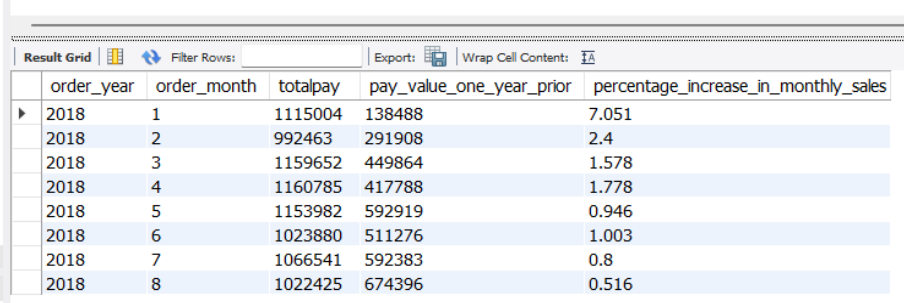
```
select *, rank() over ( order by q1.total desc) as rank_ from
(select customer_city, count(distinct customer_id) total  from customerOrderShort_view group
by customer_city
order by total desc ) q1;
```



customer_state	total	rank_
SP	15540	1
RJ	6882	2
MG	2773	3
DF	2131	4
PR	1521	5
SP	1444	6
RS	1379	7
BA	1245	8
SP	1189	9
SP	938	10
RJ	849	11
SP	797	12
SP	746	13
SP	713	14
GO	692	15

1. **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**

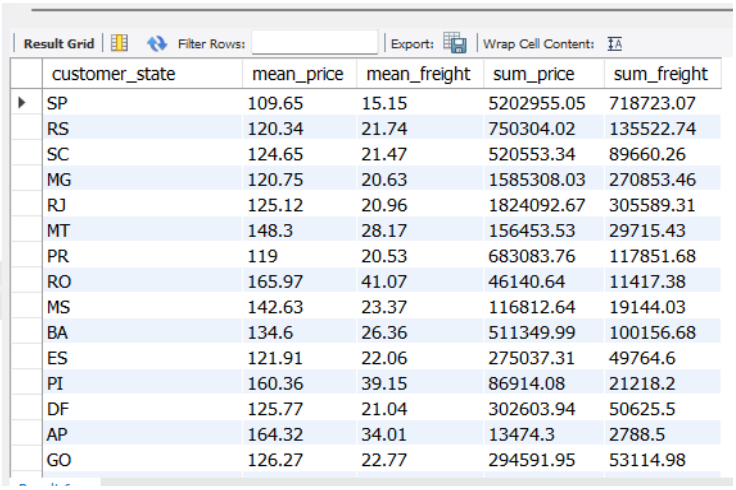
```
select *,round((totalpay-pay_value_one_year_prior)/pay_value_one_year_prior,3) as
percentage_increase_in_monthly_sales from (

select *, lag(totalpay,8) over (order by order_year, order_month) pay_value_one_year_prior from (

select order_year, order_month, round(sum(payment_value)) totalpay from customerOrderShort_view as
a

left join payments as p on a.order_id = p.order_id

where order_year >=2017 and order_month<=8

group by order_year, order_month

order by order_year, order_month)as q1

limit 8 offset 8 ) q2;
```



order_year	order_month	totalpay	pay_value_one_year_prior	percentage_increase_in_monthly_sales
2018	1	1115004	138488	7.051
2018	2	992463	291908	2.4
2018	3	1159652	449864	1.578
2018	4	1160785	417788	1.778
2018	5	1153982	592919	0.946
2018	6	1023880	511276	1.003
2018	7	1066541	592383	0.8
2018	8	1022425	674396	0.516

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

```
select customer_state , round(avg(price),2 )mean_price, round(avg(freight_value),2)
mean_freight, round(sum(price),2) sum_price, round(sum(freight_value),2) sum_freight
from orders as o left join customers as c on o.customer_id = c.customer_id
left join order_items as i on o.order_id = i.order_id
group by customer_state ;
```



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

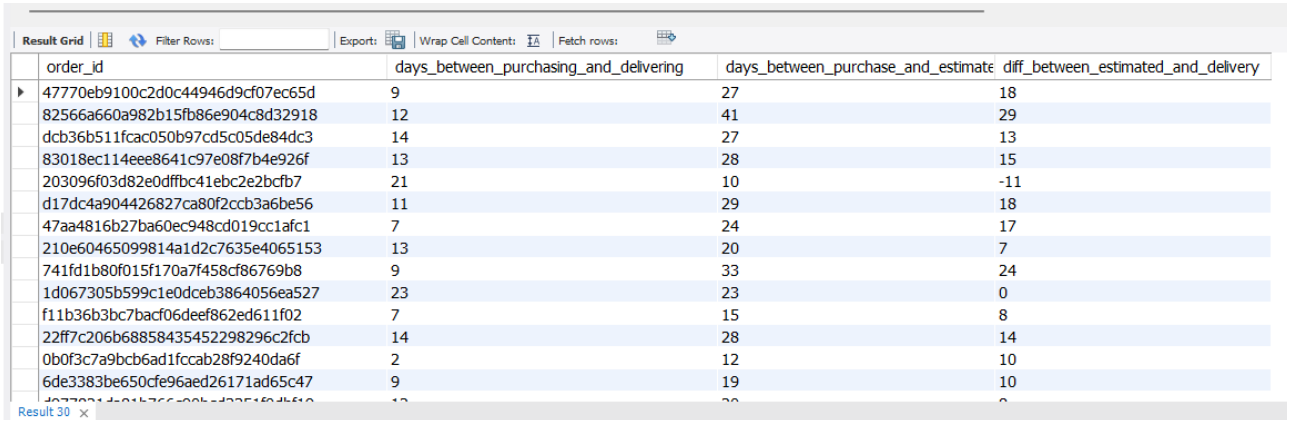
customer_state	mean_price	mean_freight	sum_price	sum_freight
SP	109.65	15.15	5202955.05	718723.07
RS	120.34	21.74	750304.02	135522.74
SC	124.65	21.47	520553.34	89660.26
MG	120.75	20.63	1585308.03	270853.46
RJ	125.12	20.96	1824092.67	305589.31
MT	148.3	28.17	156453.53	29715.43
PR	119	20.53	683083.76	117851.68
RO	165.97	41.07	46140.64	11417.38
MS	142.63	23.37	116812.64	19144.03
BA	134.6	26.36	511349.99	100156.68
ES	121.91	22.06	275037.31	49764.6
PI	160.36	39.15	86914.08	21218.2
DF	125.77	21.04	302603.94	50625.5
AP	164.32	34.01	13474.3	2788.5
GO	126.27	22.77	294591.95	53114.98

5. Analysis on sales, freight and delivery time

1. Calculate days between purchasing, delivering and estimated delivery

```
select * from (
select o.order_id,
 datediff (str_to_date(order_delivered_customer_date,'%Y-%m-%d %H:%i:%s') ,
 str_to_date(order_purchase_timestamp,'%Y-%m-%d %H:%i:%s'))
days_between_purchasing_and_delivering,
datediff(str_to_date (order_estimated_delivery_date,'%Y-%m-%d %H:%i:%s'),
 str_to_date(order_purchase_timestamp,'%Y-%m-%d %H:%i:%s')) as
days_between_purchase_and_estimated_delivery,
datediff(str_to_date (order_estimated_delivery_date,'%Y-%m-%d %H:%i:%s'),
```

```
 str_to_date(order_delivered_customer_date,'%Y-%m-%d %H:%i:%s')) as
diff_between_estimated_and_delivery

from target.orders as o

inner join target.customers as c on

o.customer_id = c.customer_id ) q1

where q1.days_between_purchasing_and_delivering is not null   ;
```



Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

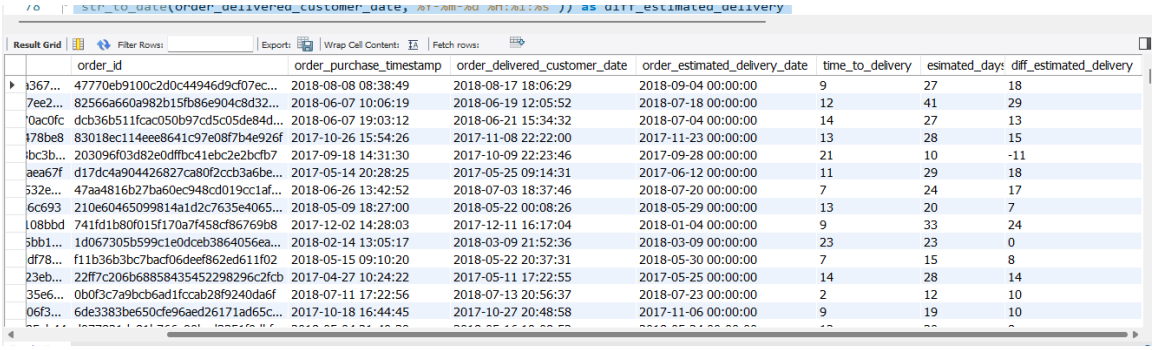
order_id	days_between_purchasing_and_delivering	days_between_purchase_and_estimate	diff_between_estimated_and_delivery
47770eb9100c2d0c44946d9cf07ec65d	9	27	18
82566a660a982b15fb86e904c8d32918	12	41	29
dcb36b511fcac050b97cd5c05de84dc3	14	27	13
83018ec114eee8641c97e08f7b4e926f	13	28	15
203096f03d82e0dffbc41ebc2e2bcfb7	21	10	-11
d17dc4a904426827ca80f2ccb3a6be56	11	29	18
47aa4816b27ba60ec948cd019cc1afc1	7	24	17
210e60465099814a1d2c7635e4065153	13	20	7
741fd1b80f015f170a7f458cf86769b8	9	33	24
1d067305b599c1e0dceb3864056ea527	23	23	0
f11b36b3bc7bacf06deef862ed611f02	7	15	8
22ff7c206b68858435452298296c2fcb	14	28	14
0b0f3c7a9bcb6ad1fccab28f9240da6f	2	12	10
6de3383be650cfe96aed26171ad65c47	9	19	10

Result 30

2. **Find time_to_delivery & diff_estimated_delivery.**

```
select * from (

select
c.customer_id,o.order_id,o.order_purchase_timestamp,o.order_delivered_customer_date,o.ord
er_estimated_delivery_date,

 datediff (str_to_date(order_delivered_customer_date,'%Y-%m-%d %H:%i:%s') ,

 str_to_date(order_purchase_timestamp,'%Y-%m-%d %H:%i:%s')) time_to_delivery,

datediff(str_to_date (order_estimated_delivery_date,'%Y-%m-%d %H:%i:%s'),

 str_to_date(order_purchase_timestamp,'%Y-%m-%d %H:%i:%s')) as
esimated_days_to_deliver,

datediff(str_to_date (order_estimated_delivery_date,'%Y-%m-%d %H:%i:%s'),

 str_to_date(order_delivered_customer_date,'%Y-%m-%d %H:%i:%s')) as
diff_estimated_delivery

from target.orders as o
```

```
inner join target.customers as c on

o.customer_id = c.customer_id ) q1

where q1.time_to_delivery is not null   ;
```



order_id	order_purchase_timestamp	order_delivered_customer_date	order_estimated_delivery_date	time_to_delivery	esimated_days	diff_estimated_delivery
a367...	47770eb9100c2d0c44946d9cf07ec...	2018-08-08 08:38:49	2018-08-17 18:06:29	2018-09-04 00:00:00	9	27	18
7ee2...	82566a660a982b15fb86e904c8d32...	2018-06-07 10:06:19	2018-06-19 12:05:52	2018-07-18 00:00:00	12	41	29
'0ac0fc	dcb36b511fcac050b97cd5c05de84d...	2018-06-07 19:03:12	2018-06-21 15:34:32	2018-07-04 00:00:00	14	27	13
478be8	83018ec114eee8641c97e08f7b4e926f	2017-10-26 15:54:26	2017-11-08 22:22:00	2017-11-23 00:00:00	13	28	15
ibc3b...	203096f03d82e0dffbc41ebc2e2bcfb7	2017-09-18 14:31:30	2017-10-09 22:23:46	2017-09-28 00:00:00	21	10	-11
aea67f	d17dc4a904426827ca80f2ccb3a6be...	2017-05-14 20:28:25	2017-05-25 09:14:31	2017-06-12 00:00:00	11	29	18
532e...	47aa4816b27ba60ec948cd019cc1af...	2018-06-26 13:42:52	2018-07-03 18:37:46	2018-07-20 00:00:00	7	24	17
6c693	210e60465099814a1d2c7635e4065...	2018-05-09 18:27:00	2018-05-22 00:08:26	2018-05-29 00:00:00	13	20	7
108bbd	741fd1b80f015f170a7f458cf86769b8	2017-12-02 14:28:03	2017-12-11 16:17:04	2018-01-04 00:00:00	9	33	24
5bb1...	1d067305b599c1e0dceb3864056ea...	2018-02-14 13:05:17	2018-03-09 21:52:36	2018-03-09 00:00:00	23	23	0
df78...	f11b36b3bc7bacf06deef862ed611f02	2018-05-15 09:10:20	2018-05-22 20:37:31	2018-05-30 00:00:00	7	15	8
23eb...	22ff7c206b68858435452298296c2fcb	2017-04-27 10:24:22	2017-05-11 17:22:55	2017-05-25 00:00:00	14	28	14
35e6...	0b0f3c7a9bcb6ad1fccab28f9240da6f	2018-07-11 17:22:56	2018-07-13 20:56:37	2018-07-23 00:00:00	2	12	10
06f3...	6de3383be650cfe96aed26171ad65c...	2017-10-18 16:44:45	2017-10-27 20:48:58	2017-11-06 00:00:00	9	19	10

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

   Sort the data to get the following:

1. **Top 5 states with highest/lowest average freight value**

```
select customer_state, avg(i.freight_value) mean_fright

        from target.orders o

        left join target.customers c  on o.customer_id = c.customer_id

   left join target.order_items i  on i.order_id = o.order_id

   group by customer_state

   order by  mean_fright desc limit 5;
```

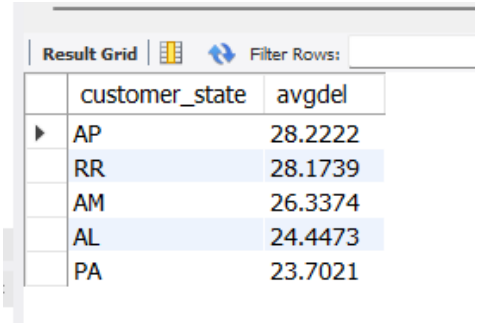
customer_state	mean_fright
RR	42.98442307692309
PB	42.723803986710934
RO	41.069712230215835
AC	40.073369565217384
PI	39.1479704797048

customer_state	mean_fright
SP	15.14727539041905
PR	20.531651567944433
MG	20.630166806306978
RJ	20.960923931682654
DF	21.041354945968294

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

Top 5 (highest avg time to delivery)

```
select customer_state , avg(time_to_delivery) as avgdel from (

        select customer_state,  datediff
(str_to_date(order_delivered_customer_date,'%Y-%m-%d %H:%i:%s') ,

        str_to_date(order_purchase_timestamp,'%Y-%m-%d %H:%i:%s')) time_to_delivery

        from orders o

        left join target.customers c  on o.customer_id = c.customer_id

        left join target.order_items i  on i.order_id = o.order_id

) q1

group by customer_state

order by avgdel desc limit 5;
```

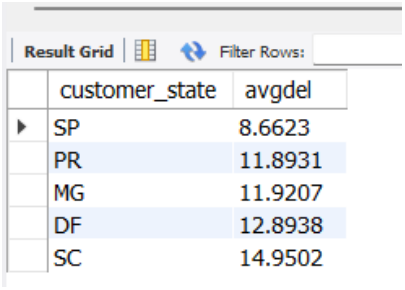


customer_state	avgdel
AP	28.2222
RR	28.1739
AM	26.3374
AL	24.4473
PA	23.7021

Bottom 5 (Lowest time to delivery)

```
select customer_state , avg(time_to_delivery) as avgdel from (

        select customer_state,  datediff
(str_to_date(order_delivered_customer_date,'%Y-%m-%d %H:%i:%s') ,

        str_to_date(order_purchase_timestamp,'%Y-%m-%d %H:%i:%s')) time_to_delivery

        from orders o

        left join target.customers c  on o.customer_id = c.customer_id

        left join target.order_items i  on i.order_id = o.order_id

) q1
```

```
group by customer_state

order by avgdel asc limit 5;
```



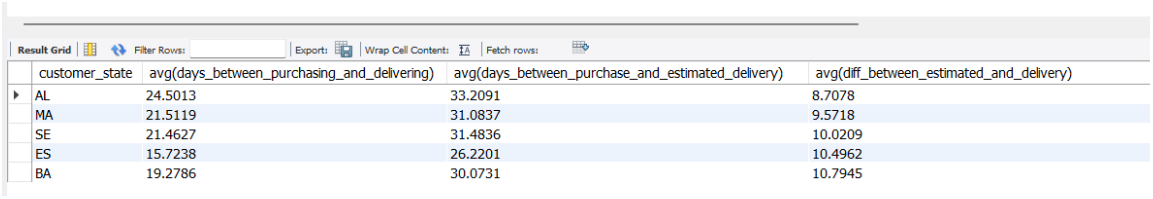
Result Grid | Filter Rows:

customer_state	avgdel
SP	8.6623
PR	11.8931
MG	11.9207
DF	12.8938
SC	14.9502

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

Slower states compared to delivery estimates

```
select customer_state,
avg(days_between_purchasing_and_delivering),avg(days_between_purchase_and_estimated_
delivery),avg(diff_between_estimated_and_delivery) from (

select o.order_id,customer_state,

 datediff (str_to_date(order_delivered_customer_date,'%Y-%m-%d %H:%i:%s') ,

 str_to_date(order_purchase_timestamp,'%Y-%m-%d %H:%i:%s'))
days_between_purchasing_and_delivering,

datediff(str_to_date (order_estimated_delivery_date,'%Y-%m-%d %H:%i:%s'),

 str_to_date(order_purchase_timestamp,'%Y-%m-%d %H:%i:%s')) as
days_between_purchase_and_estimated_delivery,

datediff(str_to_date (order_estimated_delivery_date,'%Y-%m-%d %H:%i:%s'),

 str_to_date(order_delivered_customer_date,'%Y-%m-%d %H:%i:%s')) as
diff_between_estimated_and_delivery

from target.orders as o

inner join target.customers as c on

o.customer_id = c.customer_id ) q1

where q1.days_between_purchasing_and_delivering is not null

group by customer_state order by avg(diff_between_estimated_and_delivery) limit 5;
```

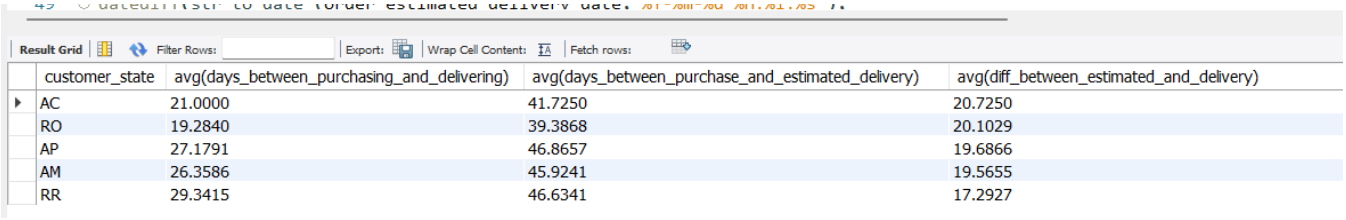


Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

customer_state	avg(days_between_purchasing_and_delivering)	avg(days_between_purchase_and_estimated_delivery)	avg(diff_between_estimated_and_delivery)
AL	24.5013	33.2091	8.7078
MA	21.5119	31.0837	9.5718
SE	21.4627	31.4836	10.0209
ES	15.7238	26.2201	10.4962
BA	19.2786	30.0731	10.7945

Faster states compared to delivery estimates

```
select customer_state,
avg(days_between_purchasing_and_delivering),avg(days_between_purchase_and_estimated_delivery),avg(diff_between_estimated_and_delivery) from (

select o.order_id,customer_state,

 datediff (str_to_date(order_delivered_customer_date,'%Y-%m-%d %H:%i:%s') ,

 str_to_date(order_purchase_timestamp,'%Y-%m-%d %H:%i:%s'))
days_between_purchasing_and_delivering,

datediff(str_to_date (order_estimated_delivery_date,'%Y-%m-%d %H:%i:%s'),

 str_to_date(order_purchase_timestamp,'%Y-%m-%d %H:%i:%s')) as
days_between_purchase_and_estimated_delivery,

datediff(str_to_date (order_estimated_delivery_date,'%Y-%m-%d %H:%i:%s'),

 str_to_date(order_delivered_customer_date,'%Y-%m-%d %H:%i:%s')) as
diff_between_estimated_and_delivery

from target.orders as o

inner join target.customers as c on

o.customer_id = c.customer_id ) q1

where q1.days_between_purchasing_and_delivering is not null

group by customer_state order by avg(diff_between_estimated_and_delivery) desc limit
5;
```



customer_state	avg(days_between_purchasing_and_delivering)	avg(days_between_purchase_and_estimated_delivery)	avg(diff_between_estimated_and_delivery)
AC	21.0000	41.7250	20.7250
RO	19.2840	39.3868	20.1029
AP	27.1791	46.8657	19.6866
AM	26.3586	45.9241	19.5655
RR	29.3415	46.6341	17.2927

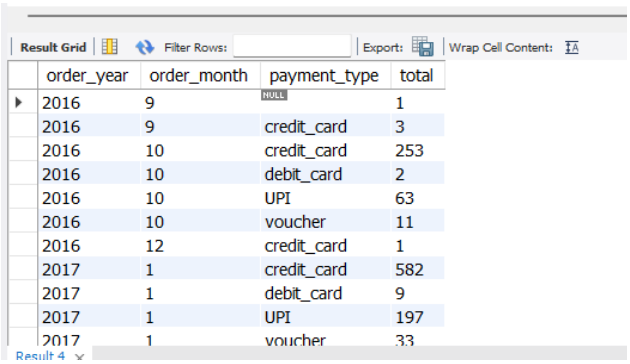
6. Payment type analysis:

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

Most of the orders were paid using credit card but upi is steadily getting popular in brazil.

And will take over within next few years

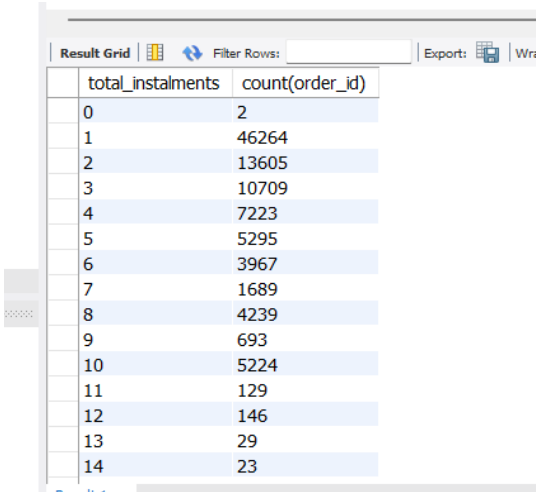
```
select order_year, order_month,payment_type, count(distinct a.order_id) total from
customerOrderShort_view as a

left join payments as p on a.order_id = p.order_id

group by order_year, order_month , payment_type

order by order_year, order_month ;
```



order_year	order_month	payment_type	total
2016	9	NULL	1
2016	9	credit_card	3
2016	10	credit_card	253
2016	10	debit_card	2
2016	10	UPI	63
2016	10	voucher	11
2016	12	credit_card	1
2017	1	credit_card	582
2017	1	debit_card	9
2017	1	UPI	197
2017	1	voucher	33

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

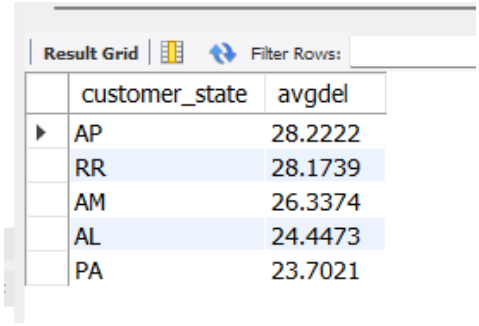
```
select total_instalments, count(order_id) from (
select o.order_id, sum(payment_installments) as total_instalments
        from orders o
        inner join target.customers c  on o.customer_id = c.customer_id
        left join target.payments p on o.order_id = p.order_id
   group by o.order_id)q1
group by total_instalments;
```



total_instalments	count(order_id)
0	2
1	46264
2	13605
3	10709
4	7223
5	5295
6	3967
7	1689
8	4239
9	693
10	5224
11	129
12	146
13	29
14	23

7. Actionable Insights

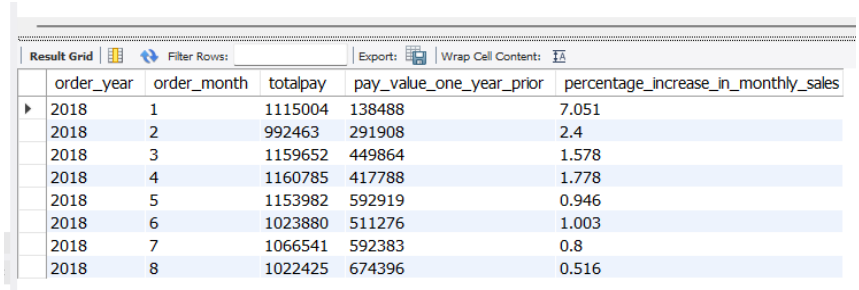
1.Delivery time is high in most states as long as 1 month needs to improved



customer_state	avgdel
AP	28.2222
RR	28.1739
AM	26.3374
AL	24.4473
PA	23.7021

2.Salses high during January

Sales growth is low during the time period February to August



order_year	order_month	totalpay	pay_value_one_year_prior	percentage_increase_in_monthly_sales
2018	1	1115004	138488	7.051
2018	2	992463	291908	2.4
2018	3	1159652	449864	1.578
2018	4	1160785	417788	1.778
2018	5	1153982	592919	0.946
2018	6	1023880	511276	1.003
2018	7	1066541	592383	0.8
2018	8	1022425	674396	0.516

3. Sales are low at places with high fright

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customer_state	mean_price	mean_freight	sum_price	sum_freight
RR	150.57	42.98	7829.43	2235.19
PB	191.48	42.72	115268.08	25719.73
RO	165.97	41.07	46140.64	11417.38
AC	173.73	40.07	15982.95	3686.75
PI	160.36	39.15	86914.08	21218.2
MA	145.2	38.26	119648.22	31523.77

4. Products with low rating needs to be removed

A Lot of products have low rating

Result Grid | Filter Rows: | Export:

review_score	count(distinct o.order_id)
1	11393
2	3148
3	8160
4	19098
5	57076

5. Performance of following Category have very little sales

```
select product_category, count(distinct o.order_id) total_order
from orders as o
inner join customers as c on c.customer_id=o.customer_id
left join order_items as i on i.order_id = o.order_id
left join products as p on i.product_id = p.product_id
group by product_category
order by total_order;
```

Result Grid | Filter Rows: | Export:

product_category	total_order
insurance and services	2
PC Gamer	8
Fashion Children's Clothing	8
cds music dvds	12
La Cuisine	13
Kitchen portable and food coach	14
Arts and Crafts	23
House Comfort 2	24
Hygiene diapers	27
Fashion Sport	27
flowers	29

Result 17

8. Recommendations

App seems to be getting popular among people of brazil the infrastructure and logistics needs to be improved in a lot of states

App needs to be promoted in states (AL,MA,SE,ES,BA) as they show low sales compared to others

Freight prices could be dropped in certain states to improve sales

Products with low rating needs to be promoted

States that have high freight expensive products could be promoted to maintain profitability