BUSINESS CASE STUDY

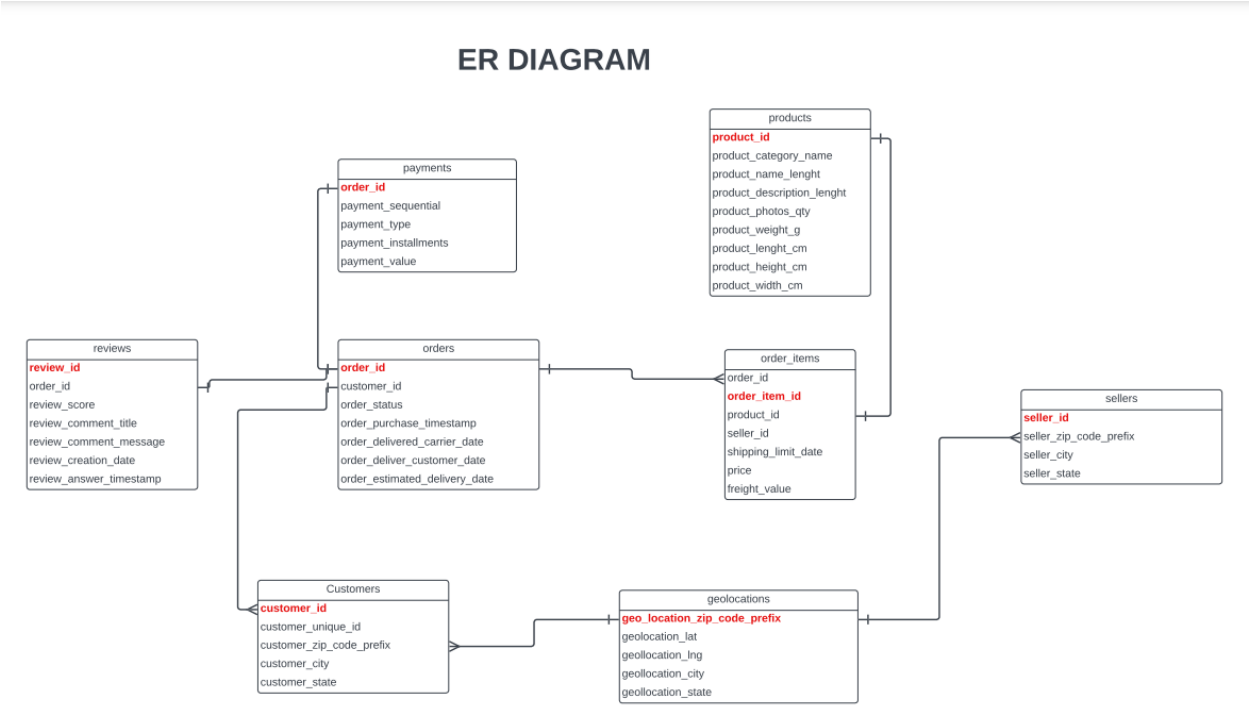
CONTEXT:

The below Business Case Study is related to globally renowned brand and a prominent retailer in the United States . It mainly focuses on the operations of Target in Brazil and provides insightful information about 100,000 orders placed between 2016 and 2018.

The Dataset consists of 8 tables of data, they are: customers, sellers, order_items, geolocation, payments, reviews, orders, products .

After analyzing the dataset we can gain some possible information like:
Analysis based on sales, payments ,Impact on economy, price, monthly sales etc.

ER DIAGRAM:



PROBLEM STATEMENT:

Assuming you are a data analyst/ scientist at Target, you have been assigned the task of analyzing the given dataset to extract valuable insights and provide actionable recommendations.

1. Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset:
1. Data type of all columns in the "customers" table.

QUERY:

```
Select column_name,data_type
from `level-landing-387114.Business_CaseStudy.INFORMATION_SCHEMA.COLUMNS`
where table_name='customers';
```

RESULT:

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

COMMENT:

From the above query we can gain the information about the customers table column names and the data types of the column names as customer_id datatype is STRING customer-zip_code_prefix as INTEGER etc.

2. Get the time range between which the orders were placed.

QUERY:

```
Select order_id,order_purchase_timestamp ,
lead(order_purchase_timestamp) over(order by order_purchase_timestamp) as new_time,
TIMESTAMP_DIFF(lead(order_purchase_timestamp) over(order by order_purchase_timestamp
),order_purchase_timestamp,DAY) as time_stamp_days
from `Business_CaseStudy.orders`
where order_status='created'
order by order_purchase_timestamp;
```

RESULT:

Row	order_id	order_purchase_timestamp	new_time	time_stamp_days
1	90ab3e7d52544ec7bc3363c82...	2017-11-06 13:12:34 UTC	2017-11-25 11:10:33 UTC	18
2	7a4df5d8cff4090e541401a20a...	2017-11-25 11:10:33 UTC	2017-12-05 01:07:52 UTC	9
3	b5359909123fa03c50bdb0cfe...	2017-12-05 01:07:52 UTC	2017-12-05 01:07:58 UTC	0
4	35de4050331c6c644cddc86f4...	2017-12-05 01:07:58 UTC	2018-02-09 17:21:04 UTC	66
5	dba5062fbda3af4fb6c33b1e04...	2018-02-09 17:21:04 UTC	*null*	*null*

COMMENT:
From the above query we gain information like the time range between the orders placed from the orders table.To gain the information here we use TIMESTAMP_DIFF() to get the difference between the two days.In this way we can get know how the orders are being placed by the customers.

3. Count the number of Cities and States in our dataset.

QUERY:

```
select count(distinct(geolocation_city)) as cities_count,count(distinct
geolocation_state) as states_count
from `Business_CaseStudy.geolocation`
```

RESULT:

Row	cities_count	states_count
1	8011	27

COMMENT:
From the above query we gain information about the count of cities and the count of states is 8011 and 27. To get this information here we used a COUNT() function on cities and states from geolocations.

2. **In-depth Exploration:**
1.Is there a growing trend in the no. of orders placed over the past years?

QUERY:

```
with cte as
(Select (EXTRACT(YEAR FROM timestamp_trunc(order_purchase_timestamp,year))) AS
year,count(*) as order_count,  from `Business_CaseStudy.orders`
group by year
order by year)
```

```
Select *,order_count- lag(order_count) over(order by year) as order_growth
from cte
order by year ;
```

RESULT:

Row	year	order_count	order_growth
1	2016	329	*null*
2	2017	45101	44772
3	2018	54011	8910

COMMENT:

From the above query we can come to the conclusion that there is a rapid growth in orders from the year 2016 to 2017 is 44772 extra orders placed by the customers in this year.To gain this information we used EXTRACT() for year,LAG(),COUNT().

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

QUERY:

```
with cte as
(Select FORMAT_DATE('%b ', order_purchase_timestamp) AS Month,count(*) as order_count
from `Business_CaseStudy.orders`
group by Month
order by order_count desc)
Select * from
(Select *, rank() over (order by order_count desc)as order_rank ,
Case
  when Month like '%Aug%'
  then 'de Cachaça- Drink, Dance & Be Merry'
  when Month like '%May%'
  then 'Brasil Sabor'
  else 'Enflor & Garden Fair'
end as Festivals
from cte
order by order_count desc)
where order_rank<=3;
```

RESULT:

Row	Month	order_count	order_rank	Festivals
1	Aug	10843	1	de Cachaça- Drink, Dance & Be ...
2	May	10573	2	Brasil Sabor
3	Jul	10318	3	Enflor & Garden Fair

COMMENT:

The result of the above query tells about the seasonality based on months. We've got to know about the sudden hike in their sales in that particular month. There are three main festivals in Brazil which affect our sales in those particular months.

3. During what time of the day, do the Brazilian customers mostly place their orders? (Dawn, Morning, Afternoon or Night)

- 0-6 hrs : Dawn
- 7-12 hrs : Mornings
- 13-18 hrs : Afternoon
- 19-23 hrs : Night

QUERY:

```
with cte as
(select extract(hour from order_purchase_timestamp) as hours
from `Business_CaseStudy.orders`)
SELECT *,rank() over(order by order_count desc) as order_rank from
(select
case
  when hours between  0 and 6
  then 'Dawn'
  when hours between 7 and 12
  then 'Mornings'
  when hours between 13 and 18
  then 'Afternoon'
  else 'Night'
end as Timings,
count(*) as order_count,
from cte
```

```
group by Timings)
order by order_rank;
```

RESULT:

Row	Timings	order_count	order_rank
1	Afternoon	38135	1
2	Night	28331	2
3	Mornings	27733	3
4	Dawn	5242	4

COMMENT:
From the query I can conclude that the customers place more orders in Afternoon timings i.e in between 13 to 18 . and the least orders were placed in the Dawn timings i.e. o to 6.
To obtain this information we use concepts like CASE,EXTRACT(),RANK() etc.

3. **Evolution of E-commerce orders in the Brazil region:**
1. Get the month on month no. of orders placed in each state.

QUERY:

```
SELECT c.customer_state,FORMAT_DATE('%b ', o.order_purchase_timestamp) AS
month,count(*) as order_count
from `Business_CaseStudy.customers` c
join `Business_CaseStudy.orders` o
on c.customer_id=o.customer_id
group by month,c.customer_state
order by c.customer_state ,month
```

RESULT:

Row	customer_state	month	order_count
1	AC	Apr	9
2	AC	Aug	7
3	AC	Dec	5
4	AC	Feb	6
5	AC	Jan	8
6	AC	Jul	9
7	AC	Jun	7
8	AC	Mar	4
9	AC	May	10
10	AC	Nov	5
11	AC	Oct	6
12	AC	Sep	5
13	AL	Apr	51

COMMENT:

From this query we obtain the information that orders getting placed in each state on monthly basis for example in state AC the orders placed on month April is 9 , August is 7 and so on .

2. How are the customers distributed across all the states?

QUERY:

```
with cte as
(Select customer_state,count(*) as customer_count from `Business_CaseStudy.customers`
group by customer_state
order by customer_state)

Select *,dense_rank() over( order by customer_count desc) as state_rank from cte
order by state_rank
```

RESULT:

Row	customer_state	customer_count	state_rank
1	SP	41746	1
2	RJ	12852	2
3	MG	11635	3
4	RS	5466	4
5	PR	5045	5
6	SC	3637	6
7	BA	3380	7
8	DF	2140	8
9	ES	2033	9
10	GO	2020	10
11	PE	1652	11
12	CE	1336	12

COMMENT:
From this output I came to know that customers in the state SP ranks 1 all over the 27 states and the count of customers is 41746 and the least the customers are from RR is 46.

4. **Impact on Economy: Analyze the money movement by e-commerce by looking at order prices, freight and others.**

1. Get the % increase in the cost of orders from year 2017 to 2018 (include months between Jan to Aug only).
You can use the "payment_value" column in the payments table to get the cost of orders.

QUERY:

```
with cte as
(Select extract(month from o.order_purchase_timestamp) as
Months,round(sum(p.payment_value)) as total_cost from `Business_CaseStudy.payments` p
join
`Business_CaseStudy.orders` o
on o.order_id=p.order_id
where extract(month from o.order_purchase_timestamp) between 1 and 8
group by Months
order by  Months )
```

```
Select *,round(100*total_cost/lag(total_cost) over (order by Months))-100 as
growth_percentage
from cte
order by Months
```

RESULT:

Row	Months	total_cost	growth_percentage
1	1	1253492.0	null
2	2	1284371.0	2.0
3	3	1609516.0	25.0
4	4	1578574.0	-2.0
5	5	1746901.0	11.0
6	6	1535157.0	-12.0
7	7	1658924.0	8.0
8	8	1696822.0	2.0

COMMENT:
From this query we can see the growth percentage month wise in the year from Jan to Aug. Here we can see 2% growth in between the months January and February , likewise there is 2% decline in March and April etc . In this way we can get a monthly growth percentage .

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

QUERY:

```
Select customer_state,round(sum(price),2) as total_value ,round(avg(price),2) as
avg_value
from `Business_CaseStudy.orders` o
join `Business_CaseStudy.order_items` oi
on o.order_id=oi.order_id
join `Business_CaseStudy.customers` c
on c.customer_id=o.customer_id
group by customer_state
order by customer_state
```

RESULT:

Row	customer_state	total_value	avg_value
1	AC	15982.95	173.73
2	AL	80314.81	180.89
3	AM	22356.84	135.5
4	AP	13474.3	164.32
5	BA	511349.99	134.6
6	CE	227254.71	153.76
7	DF	302603.94	125.77
8	ES	275037.31	121.91
9	GO	294591.95	126.27
10	MA	119648.22	145.2
11	MG	1585308.03	120.75
12	MS	116812.64	142.63

COMMENT:
From this query we can come to know about state wise total price and average price .
For getting this information here we need to join orders,order_items and customers table to get customer_state, total price and average

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

QUERY:

```
Select *,dense_rank() over(order by avg_freight_value desc) as rank from
(Select  customer_state  as  state,round(sum(freight_value),2)  as  total_freight_value
,round(avg(freight_value),2) as avg_freight_value
from  `Business_CaseStudy.orders` o
join `Business_CaseStudy.order_items` oi
on o.order_id=oi.order_id
join `Business_CaseStudy.customers` c
on c.customer_id=o.customer_id
group by customer_state)
order by rank
```

RESULT:

Row	state	total_freight_value	avg_freight_value	rank
1	RR	2235.19	42.98	1
2	PB	25719.73	42.72	2
3	RO	11417.38	41.07	3
4	AC	3686.75	40.07	4
5	PI	21218.2	39.15	5
6	MA	31523.77	38.26	6
7	TO	11732.68	37.25	7
8	SE	14111.47	36.65	8
9	AL	15914.59	35.84	9
10	PA	38699.3	35.83	10
11	RN	18860.1	35.65	11
12	AP	2788.5	34.01	12
13	AM	5478.89	33.21	13

COMMENT:

From the above the query we can get to know about total freight value and average freight value of all 27 states . from that the highest average freight value state is RR and the lowest average freight value is SP.

5. **Analysis based on sales, freight and delivery time.**

1.Find the no. of days taken to deliver each order from the order's purchase date as delivery time. Also, calculate the difference (in days) between the estimated & actual delivery date of an order. Do this in a single query.

QUERY:

```
Select order_id,
date_diff(order_delivered_customer_date,order_purchase_timestamp,day)as
time_to_deliver,
date_diff(order_estimated_delivery_date,order_delivered_customer_date,day)as
diff_estimated_delivery
from `Business_CaseStudy.orders`
order by order_id;
```

RESULT:

Row	order_id	time_to_deliver	diff_estimated_deliv
1	00010242fe8c5a6d1ba2dd792...	7	8
2	00018f77f2f0320c557190d7a1...	16	2
3	000229ec398224ef6ca0657da...	7	13
4	00024acbcdf0a6daa1e931b03...	6	5
5	00042b26cf59d7ce69dfabb4e...	25	15
6	00048cc3ae777c65dbb7d2a06...	6	14
7	00054e8431b9d7675808bcb8...	8	16
8	000576fe39319847cbb9d288c...	5	15
9	0005a1a1728c9d785b8e2b08...	9	0
10	0005f50442cb953dcd1d21e1f...	2	18
11	00061f2a7bc09da83e415a52d...	4	10
12	00063b381e2406b52ad42947...	10	0

COMMENT:

From the above result of the query, we can conclude that most of the orders are getting delayed to reach customers and we've given very less estimating delivery time but failed to achieve. So we should take this into consideration to give appropriate estimated delivery time and then we can achieve them as before the estimated delivery time. It'll boost our customer support and faith in us.

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

QUERY:

```
WITH low_average_freight_vlaue AS
(SELECT *
FROM
(
SELECT *,
DENSE_RANK() OVER(ORDER BY lowest_average_freight_vlaue) as dense_rank,
ROW_NUMBER() OVER(ORDER BY lowest_average_freight_vlaue) as row_num
FROM
(
SELECT
c.customer_state,
ROUND(SUM(ot.freight_value),2) lowest_total_freight_vlaue,
```

```
ROUND(AVG(ot.freight_value),2) lowest_average_freight_vlaue
FROM
`Business_CaseStudy.customers` c
INNER JOIN `Business_CaseStudy.orders` o
ON c.customer_id = o.customer_id
INNER JOIN `Business_CaseStudy.order_items`ot
ON o.order_id = ot.order_id
GROUP BY c.customer_state
ORDER BY 2 DESC
)
ORDER BY lowest_average_freight_vlaue
) WHERE dense_rank <= 5
)

,

 high_average_freight_vlaue AS
(
  SELECT *
FROM
(
SELECT *,
DENSE_RANK() OVER(ORDER BY highest_average_freight_vlaue DESC) as denserank,
ROW_NUMBER() OVER(ORDER BY highest_average_freight_vlaue DESC) as row_num
FROM
(
SELECT
c.customer_state,
ROUND(sum(ot.freight_value),2) highest_total_freight_vlaue,
ROUND(AVG(ot.freight_value),2) highest_average_freight_vlaue
FROM
`Business_CaseStudy.customers` c
INNER JOIN `Business_CaseStudy.orders` o
ON c.customer_id = o.customer_id
INNER JOIN `Business_CaseStudy.order_items`ot
ON o.order_id = ot.order_id
GROUP BY c.customer_state
ORDER BY 2 DESC
)
ORDER BY highest_average_freight_vlaue DESC
) WHERE dense_rank <= 5
)
```

```
SELECT
h.customer_state AS high_avg_state,
h.highest_total_freight_vlaue,
h.highest_average_freight_vlaue,
h.dense_rank,
l.customer_state AS low_avg_state,
l.lowest_total_freight_vlaue,
l.lowest_average_freight_vlaue,
l.dense_rank
FROM
low_average_freight_vlaue AS l
INNER JOIN high_average_freight_vlaue h
ON l.row_num = h.row_num
ORDER BY 4 ASC;
```

RESULT:

Row	high_avg_state	highest_total_freight	highest_average_frej	denserank	low_avg_state	lowest_total_freight_	lowest_average_freic	denserank_1
1	RR	2235.19	42.98	1	SP	718723.07	15.15	1
2	PB	25719.73	42.72	2	PR	117851.68	20.53	2
3	RO	11417.38	41.07	3	MG	270853.46	20.63	3
4	AC	3686.75	40.07	4	RJ	305589.31	20.96	4
5	PI	21218.2	39.15	5	DF	50625.5	21.04	5

COMMENT:

From the above result of the query , the freight value varies the number of orders and number of customers in that particular state. High freight value states having less orders and low freight value having more orders and customers. So, if we try to decrease the freight value of those particular states then it'll help to increase customers and sales.

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

QUERY:

```
with cte1 as
(Select * from
(SELECT customer_state,lowest_total_time,lowest_average_time ,
DENSE_RANK() OVER(ORDER BY lowest_average_time) as denserank,
ROW_NUMBER() OVER(ORDER BY lowest_average_time) as row_num
FROM
```

```sql
(Select customer_state,
ROUND(SUM(time_to_deliver),2) lowest_total_time,
ROUND(AVG(time_to_deliver),2) lowest_average_time from
(Select customer_state,
date_diff(order_delivered_customer_date , order_purchase_timestamp,day) as
time_to_deliver
from `Business_CaseStudy.orders` o
join `Business_CaseStudy.customers` c
on o.customer_id=c.customer_id)
group by customer_state)
order by lowest_average_time)
where denserank<=5)
,
cte2 as
(Select * from
(SELECT customer_state,high_total_time,high_average_time ,
DENSE_RANK() OVER(ORDER BY high_average_time desc) as denserank,
ROW_NUMBER() OVER(ORDER BY high_average_time desc) as row_num
FROM
(Select customer_state,
ROUND(SUM(time_to_deliver),2) high_total_time,
ROUND(AVG(time_to_deliver),2) high_average_time from
(Select customer_state,
date_diff(order_delivered_customer_date , order_purchase_timestamp,day) as
time_to_deliver
from `Business_CaseStudy.orders` o
join `Business_CaseStudy.customers` c
on o.customer_id=c.customer_id)
group by customer_state)
order by high_average_time desc)
where denserank<=5)

Select
c2.customer_state as slo_delivery_state,
c2.high_total_time,
c2.high_average_time,
c2.denserank as slow_delivery_rank,
c1.customer_state as fast_delivery_state,
c1.lowest_total_time,
c1.lowest_average_time,
c1.denserank as fast_delivery_rank
from cte1 as c1
```

```
join
cte2 as c2
on c1.row_num=c2.row_num
order by 4
```

RESULT:

Row	slo_delivery_state	high_total_time	high_average_time	slow_delivery_rank	fast_delivery_state	lowest_total_time	lowest_average_time	fast_delivery_rank
1	RR	1188.0	28.98	1	SP	336030.0	8.3	1
2	AP	1791.0	26.73	2	PR	56746.0	11.53	2
3	AM	3768.0	25.99	3	MG	131080.0	11.54	3
4	AL	9544.0	24.04	4	DF	26019.0	12.51	4
5	PA	22057.0	23.32	5	SC	51359.0	14.48	5

COMMENT:

From the result of the above query we get to know about which states are getting fast deliveries and which states are getting slow deliveries. By observing previous query and present query freight value and delivery time are interrelated. These two factors affect the number of customers and number of sales of those states. We can observe that RR states have high average time, so it has slow delivery. On the other hand, state SP has the lowest average time, so it has the fastest delivery. Based on the delivery speed indirectly the orders count getting affected.

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

You can use the difference between the averages of actual & estimated delivery date to figure out how fast the delivery was for each state.

QUERY:

```
with cte as
(Select customer_state as state,
date_diff(order_delivered_customer_date , order_purchase_timestamp,day) as
time_to_deliver,
date_diff(order_estimated_delivery_date , order_delivered_customer_date ,day) as
diff_estimated_delivery
from `Business_CaseStudy.orders` o
join
`Business_CaseStudy.customers` c
```

```
on o.customer_id=c.customer_id
order by state)
Select state,rank_del from
(Select *,dense_rank() over(order by avg_del) as rank_del from
(Select * , round(avg_days-avg_diff,2) as avg_del from
(Select distinct state ,
round(avg(time_to_deliver) over(partition by state),2) as avg_days,
round(avg(cte.diff_estimated_delivery) over (partition by state),2) as avg_diff
from cte)))
where rank_del<=5
order by  rank_del
```

RESULT:

Row	state	rank_del
1	SP	1
2	PR	2
3	MG	3
4	RO	4
5	AC	5

COMMENT:

From the result of the above query we're getting to know about states of highest delivery speed when compared to estimated delivery time. From this observation these states are having more customers and orders. To improve the orders we should focus on the delivery system.

6. **Analysis based on the payments:**

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

QUERY:

```
Select  *  ,dense_rank()  over(partition  by  payment_type  order  by  order_count  desc)
order_rank from
(SELECT  payment_type,FORMAT_DATE('%b  ',  o.order_purchase_timestamp)  AS  month,count(*)
as order_count
from `Business_CaseStudy.payments` p
join `Business_CaseStudy.orders` o
```

```
on p.order_id=o.order_id
group by month,payment_type)
order by payment_type,order_rank
```

RESULT:

Row	payment_type	month	order_count	order_rank
1	UPI	Aug	2077	1
2	UPI	Jul	2074	2
3	UPI	May	2035	3
4	UPI	Mar	1942	4
5	UPI	Jun	1807	5
6	UPI	Apr	1783	6
7	UPI	Feb	1723	7
8	UPI	Jan	1715	8
9	UPI	Nov	1509	9
10	UPI	Dec	1160	10
11	UPI	Oct	1056	11
12	UPI	Sep	903	12
13	credit_card	May	8350	1
14	credit_card	Aug	8269	2
15	credit_card	Jul	7841	3
16	credit_card	Mar	7707	4

COMMENT:

From the result of the above query we've concluded that most of the orders are getting paid by using a credit card which tells that most of the customers rely on credit cards to shop. If we introduce attractive offers on credit cards, this will help us to boost our orders and customers

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

QUERY:

```
Select * , dense_rank() over(order by order_placed desc) as installments_rank from
(Select payment_installments,count(*) as order_placed ,
from `Business_CaseStudy.payments`
group by payment_installments)
order by payment_installments
```

RESULT:

Row	payment_installment	order_placed	installments_rank
1	0	2	22
2	1	52546	1
3	2	12413	2
4	3	10461	3
5	4	7098	4
6	5	5239	6
7	6	3920	8
8	7	1626	9
9	8	4268	7
10	9	644	10
11	10	5328	5
12	11	23	14
13	12	133	11
14	13	16	17
15	14	15	18
16	15	74	12
17	16	5	20

COMMENT:

By observing the above result we can conclude that most of the users bought the things in single installments and very few users bought the things without any installments. So we can provide a limited period of installments.

OUR INSIGHTS:

1.Get the count of review score and determine the how many orders are liked by the user and get insights from the reviews table to improve the business

QUERY:.

```
Select review_score,
case
  when review_score = 5
  then 'Excellent'
  when review_score = 4
```

```
  then 'Very good'
  when review_score = 3
  then 'Satisfactory'
  when review_score = 2
  then 'Average'
  else 'Not as expected'
end as Rating,
count(review_score) as count,
rank() over(order by count(review_score) desc) as rank_ranting
from `Business_CaseStudy.order_reviews`
group by review_score
order by 1 desc
```

RESULT:

Row	review_score	Rating	count	rank_ranting
1	5	Excellent	57328	1
2	4	Very good	19142	2
3	3	Satisfactory	8179	4
4	2	Average	3151	5
5	1	Not as expected	11424	3

COMMENT:

The above table clearly represents the data about review scores of the users. It shows that most of the users are giving Excellent rating which is a good sign, but 3rd most rank_rating is 1, so we need to take care of '1' review score users to improve our shopping experience.

2.Get the best selling items of TOP 2 cities of each state.

QUERY:

```
with cte as
(Select seller_state ,seller_city,count(*) as orders,
dense_rank() over(partition by seller_state order by count(*)desc) as rank_order_city
from
`Business_CaseStudy.sellers` s
inner join
`Business_CaseStudy.order_items` o
on s.seller_id=o.seller_id
```

```
group by seller_city,seller_state)

Select seller_state as state,seller_city,orders,rank_order_city from cte
where  rank_order_city<=2
group by seller_city,seller_state,rank_order_city,orders
order by seller_state,rank_order_city
```

RESULT:

Row	state	seller_city	orders	rank_order_city
1	SP	sao paulo	27983	1
2	SP	ibitinga	7750	2
3	SE	neopolis	5	1
4	SE	aracaju	5	1
5	SC	joinville	670	1
6	SC	blumenau	483	2
7	RS	porto alegre	799	1
8	RS	caxias do sul	225	2
9	RO	ji parana	8	1
10	RO	porto velho	6	2
11	RN	natal	39	1
12	RN	parnamirim	14	2
13	RJ	rio de janeiro	2436	1
14	RJ	petropolis	840	2
15	PR	curitiba	2944	1

COMMENT:

The result of the above query tells about the count of orders in each state and their corresponding city. From my observation, SP has more orders when compared to the remaining states. As per my previous observation SP has also more customers and less delivery time so, we can consider SP is our main state that impacts our sales.

3.Get the product category with their order's count and give ranking based on their number of orders

QUERY:

```
SELECT distinct product_category,count(order_id) as count_orders,
rank() over(order by count(order_id)desc) as rank_orders
from `Business_CaseStudy.products` p
```

```
inner join
`Business_CaseStudy.order_items` o
on p.product_id=o.product_id
group by product_category
order by rank_orders
```

RESULT:

Row	product_category	count_orders	rank_orders
1	bed table bath	11115	1
2	HEALTH BEAUTY	9670	2
3	sport leisure	8641	3
4	Furniture Decoration	8334	4
5	computer accessories	7827	5
6	housewares	6964	6
7	Watches present	5991	7
8	telephony	4545	8
9	Garden tools	4347	9
10	automotive	4235	10
11	toys	4117	11
12	Cool Stuff	3796	12
13	perfumery	3419	13
14	babies	3065	14
15	electronics	2767	15
16	stationary store	2517	16

COMMENT:

From the result of the above query we get to know about which product category has more sales. It clearly shows that “bed bath table” is the most selling category, so we need to keep the item in stock.

4.Get average price of each product category and arrange them in descending order

QUERY:

```
SELECT distinct product_category,round(avg(price) over(partition by product_category))
as avg_price
from `Business_CaseStudy.products` p
inner join `Business_CaseStudy.order_items` o
```

```
on p.product_id=o.product_id
group by product_category,price
order by 2 desc
```

RESULT:

Row	product_category	avg_price
1	PCs	1478.0
2	ELECTRICES 2	514.0
3	HOUSE PASTALS OVEN AND C...	443.0
4	Agro Industria e Comercio	425.0
5	Watches present	415.0
6	musical instruments	380.0
7	electrostile	345.0
8	Kitchen portable and food coach	289.0
9	CONSTRUCTION SECURITY TO...	273.0
10	Industry Commerce and Busine...	254.0
11	fixed telephony	251.0
12	Garden tools	251.0
13	Cool Stuff	243.0
14	babies	229.0

COMMENT:

The result of the above query tells us about the average price of each product category. This result will make us aware of which product category produces more income than other product categories. So that we'll keep an eye on these categories to keep in stock and ready to ship whenever the order comes.

5.Get the growth rate of customer spending on e-commerce during 2016 to 2018

QUERY:

```
with cte as
(Select extract(year from o.order_purchase_timestamp)as
Year,round(sum(p.payment_value)) as total_cost from `Business_CaseStudy.payments` p
join `Business_CaseStudy.orders` o
on o.order_id=p.order_id
group by Year
order by  Year )
```

```
Select *,round(100*total_cost/lag(total_cost)over(order by Year)) as growth_percentage
from cte
order by Year
```

RESULT:

Row	Year	total_cost	growth_percentage
1	2016	59362.0	null
2	2017	7249747.0	12213.0
3	2018	8699763.0	120.0

COMMENT:

The result of the above query provides us to analyze the average spending of consumers on each year. It'll give an idea to proceed with any cost increments or discounts in future. We can see a rapid growth in total cost from 2016 to 2017. So, here we can see a peak of the growth of average spending.

6.Get the most selling product category of each city in each state.

QUERY:

```
with cte as
(Select * , dense_rank()  over(partition by customer_city order by oo desc) from
(Select    distinct    customer_state,    customer_city,product_category    ,count(*)
over(partition by customer_city) as oo

from `Business_CaseStudy.products` p
join
`Business_CaseStudy.order_items` oi
on p.product_id=oi.product_id
join
`Business_CaseStudy.orders` o
on o.order_id=oi.order_id
join `Business_CaseStudy.customers` c
on c.customer_id=o.customer_id)
where oo=1)
```

```
Select customer_state as state,customer_city as city,product_category from cte
order by state
```

RESULT:

Row	state	city	product_category
1	AC	manoel urbano	sport leisure
2	AC	porto acre	Fashion Calcados
3	AC	epitaciolandia	sport leisure
4	AL	uniao dos palmares	*null*
5	AL	murici	housewares
6	AL	monteiropolis	Furniture Decoration
7	AL	luziapolis	Watches present
8	AL	marechal deodoro	HEALTH BEAUTY
9	AL	sao jose da tapera	Fashion Bags and Accessories
10	AL	atalaia	automotive
11	AL	coite do noia	stationary store
12	AL	traipu	toys
13	AL	cacimbinhas	computer accessories
14	AL	porto de pedras	computer accessories
15	AL	girau do ponciano	Games consoles
16	AL	canapi	Furniture Decoration

Load more

COMMENT:

From this result we get to know about the most selling product category in each state and city. So, from my observation we should implement warehouses in each city with their most selling products to deliver them quickly. This will help to boost up our orders.

7.Get the highest and least selling product category.

QUERRY:

```
Select product_category,count_orders from
(SELECT  product_category,count(order_id) as count_orders,
rank() over(order by count(order_id)desc) as rank_desc,
rank() over(order by count(order_id)asc) as rank_asc,
from `Business_CaseStudy.products` p
inner join
`Business_CaseStudy.order_items` o
on p.product_id=o.product_id
group by product_category)
where rank_desc=1 or rank_asc=1
```

RESULT:

Row	product_category	count_orders
1	bed table bath	11115
2	insurance and services	2

COMMENT:

From the result of the above query we get to know which product is most selling and which product is least selling item. From this we have known the reason behind the least item and, Is there any way to improve their sales.

Recommendations:

From the above case study I would recommend some implementations to help in the growth of the Business

- Firstly we should enlarge our marketing media over the whole country to reach more customers
- We should focus on logistics to improve delivery system for better and fast deliveries
- We have observed that most customers use credit cards for their payments. So, we should give some interesting offers to attract the credit card users
- After analyzing the most selling items in each city we have to implement some warehouses with the most selling products in their corresponding cities for fast deliveries.
- Based on the seasonality we should maintain our stocks to meet the requirements of the customers .

.