PROJECT – TARGET

Q 1.1 Data type of columns in a table

```
select column_name, data_type
from `scaler-dsml-380816.Target.INFORMATION_SCHEMA.COLUMNS`
where table_name = "Orders";
```

Row	column_name	data_type
1	order_id	STRING
2	customer_id	STRING
3	order_status	STRING
4	order_purchase_timestamp	TIMESTAMP
5	order_approved_at	TIMESTAMP
6	order_delivered_carrier_date	TIMESTAMP
7	order_delivered_customer_date	TIMESTAMP
8	order_estimated_delivery_date	TIMESTAMP

Q 1.2 Time period for which the data is given

```
select min(order_purchase_timestamp) as start_date, max(order_purchase_timestamp) as end_date
from `scaler-dsml-380816.Target.Orders`;
```

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

Assumption:

Time period considered here is for purchase date

Q 1.3 Cities and States of customers ordered during the given period

```
select distinct customer_city as city, customer_state as state
from `scaler-dsml-380816.Target.Customers`
order by city;
```

Row	city	state
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

Q 2.1 Is there a growing trend on e-commerce in Brazil? How can we describe a complete scenario? Can we see some seasonality with peaks at specific months?

```
select order_status, count(*) as total_orders,
round((count(*)/(select count(*) from `scaler-dsml-
380816.Target.Orders` ))*100,3) as percentage_of_orders
from `scaler-dsml-380816.Target.Orders`
group by order_status
order by total_orders desc;
```

Row	order_status	total_orders	percentage_of_orders
1	delivered	96478	97.02
2	shipped	1107	1.113
3	canceled	625	0.629
4	unavailable	609	0.612
5	invoiced	314	0.316
6	processing	301	0.303
7	created	5	0.005
8	approved	2	0.002

Actionable Insight:

Above table provides an overview of total orders and their respective order status. We can look into 0.6% of orders which are cancelled and understand the reasons behind it.

```
select extract(month from order_purchase_timestamp) as month,extract( year from order_purchase
_timestamp) as year,
count(order_id) as num_of_orders
from `scaler-dsml-380816.Target.Orders`
group by year, month
order by year, month;
```

Row	month	year	num_of_orders
1	9	2016	4
2	10	2016	324
3	12	2016	1
4	1	2017	800
5	2	2017	1780
6	3	2017	2682
7	4	2017	2404
8	5	2017	3700
9	6	2017	3245
10	7	2017	4026
11	8	2017	4331
12	9	2017	4285

Actionable Insight:

Above table provides evolution of total orders month on month. Gradual increase in orders from January 2017 to January 2018 and then gradual decrease in orders. No specific trend observed. Orders peaked in November 2017 & dipped back to lowest sales orders in October 2018.

```
select extract(month from order_purchase_timestamp) as month,
extract(year from order_purchase_timestamp ) as year,
count(*) as orders
from `scaler-dsml-380816.Target.Orders`
group by month,year
order by month,year;
```

Row	month	year	num_of_orders
1	1	2017	800
2	1	2018	7269
3	2	2017	1780
4	2	2018	6728
5	3	2017	2682
6	3	2018	7211
7	4	2017	2404
8	4	2018	6939
9	5	2017	3700
10	5	2018	6873

Actionable Insight:

Above table compares month vs month of each year based on orders placed

```
select p.product_category, count(o.order_id) as num_of_orders
from `scaler-dsml-380816.Target.Products` p
join `scaler-dsml-380816.Target.Order_items` o
on p.product_id = o.product_id
where p.product_category is not null
group by p.product_category
order by num_of_orders desc;
```

Row	product_category	num_of_orders
1	bed table bath	11115
2	HEALTH BEAUTY	9670
3	sport leisure	8641
4	Furniture Decoration	8334
5	computer accessories	7827
6	housewares	6964
7	Watches present	5991
8	telephony	4545
9	Garden tools	4347
10	automotive	4235
11	toys	4117
12	Cool Stuff	3796

Actionable Insight:

Above table provides sales based on product categories. We can find out what product categories are sold most and prioritize inventory accordingly

Recommendations:

Discount offers on product categories with lower number of orders could attract customers & increase revenue from those product categories

Q 2.2 What time do Brazilian customers tend to buy (Dawn, Morning, Afternoon or Night)?

```
select
case when hour between 0 and 6 then "Dawn"
when hour between 6 and 12 then "Morning"
when hour between 12 and 17 then "Afternoon"
else "Evening"
end as time_of_the_day,
count(*) as no_of_orders
from
(
```

```
select *,
extract(hour from order_purchase_timestamp) as hour
from `scaler-dsml-380816.Target.Orders`
) t
group by time_of_the_day;
```

Row	time_of_the_day	no_of_orders
1	Morning	27733
2	Dawn	5242
3	Afternoon	32366
4	Evening	34100

Actionable Insight:

Maximum orders are made during the evening time post 5'o clock till the midnight, followed by afternoon, morning and dawn

Recommendations:

Most people are active during the evening, so therefore, offering discounts during evening would increase sales and revenue

Q 3.1 Evolution of E-commerce orders in the Brazil region: Get month on month orders by states

```
select extract(month from o.order_purchase_timestamp) as month, c.customer_state,
count(*) as no_of_orders
from `scaler-dsml-380816.Target.Orders` o
join `scaler-dsml-380816.Target.Customers` c
on o.customer_id = c.customer_id
group by  month, c.customer_state
order by month, no_of_orders desc;
```

Row	month	customer_state	no_of_orders
1	1	SP	3351
2	1	RJ	990
3	1	MG	971
4	1	PR	443
5	1	RS	427
6	1	SC	345
7	1	BA	264
8	1	GO	164
9	1	ES	159
10	1	DF	151

Q 3.2 Evolution of E-commerce orders in the Brazil region: Distribution of customers across the states in Brazil

```
select customer_state, count(distinct customer_id) as no_of_customers
from `scaler-dsml-380816.Target.Customers`
group by customer_state
order by no_of_customers desc;
```

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

Actionable Insight:

States SP, RJ and MG have got highest number of customers in the country in the same order. Also, these major states SP, RJ and MG has the highest orders in the same order. Higher number of customers leads to higher number of orders

Q 4.1 Impact on Economy: Analyze the money movement by e-commerce by looking at order prices, freight and others: 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 x.year,
round(((x.sum_of_sales - lag(x.sum_of_sales) over(order by x.year))/x.sum_of_sales)*100,2) as perc
entage_increase
from
(select extract(year from o.order_purchase_timestamp) as year, sum(p.payment_value) as sum_of_s
ales
from `scaler-dsml-380816.Target.Payments` p
join `scaler-dsml-380816.Target.Orders` o
on o.order_id = p.order_id
where extract(month from o.order_purchase_timestamp ) in (1,2,3,4,5,6,7,8)
group by year
order by year) x;
```

year	percentage_increase
2017	null
2018	57.8

Actionable Insight:

Revenue has increased by 57% from 2017 to 2018

Q 4.2 Impact on Economy: Analyze the money movement by e-commerce by looking at order prices, freight and others: Mean & Sum of price and freight value by customer state

```
select c.customer_state, round(avg(x.price),1) as avg_price, round(sum(x.price),1) as sum_of_price,
round(avg(x.freight_value),1) as avg_freight_value, round(sum(x.freight_value),1) as sum_of_freight
_value
from `scaler-dsml-380816.Target.Order_items` x
join `scaler-dsml-380816.Target.Orders` o
on x.order_id = o.order_id
join `scaler-dsml-380816.Target.Customers` c
on o.customer_id = c.customer_id
group by c.customer_state
order by avg_price, avg_freight_value;
```

Row	customer_state	avg_price	sum_of_price	avg_freight_value	sum_of_freight_value
1	SP	109.7	5202955.1	15.1	718723.1
2	PR	119.0	683083.8	20.5	117851.7
3	RS	120.3	750304.0	21.7	135522.7
4	MG	120.7	1585308.0	20.6	270853.5
5	ES	121.9	275037.3	22.1	49764.6
6	SC	124.7	520553.3	21.5	89660.3
7	RJ	125.1	1824092.7	21.0	305589.3
8	DF	125.8	302603.9	21.0	50625.5
9	GO	126.3	294591.9	22.8	53115.0
10	BA	134.6	511350.0	26.4	100156.7

Actionable Insight:

State SP has the lowest average product value and lowest freight value, and therefore, has the highest revenue

Recommendations:

Optimize the product cost & freight value to increase revenue from low revenue states

Q 5.1 Analysis on sales, freight and delivery time: Calculate days between purchasing, delivering and estimated delivery

```
select order_id, order_purchase_timestamp, order_delivered_customer_date,
order_estimated_delivery_date,
timestamp_diff(order_delivered_customer_date,order_purchase_timestamp,day) as time_to_delive
ry,
timestamp_diff(order_estimated_delivery_date, order_delivered_customer_date, day) as diff_estim
ated_delivery
from `scaler-dsml-380816.Target.Orders`
where order_delivered_customer_date is not null
order by order_purchase_timestamp;
```

Row	order_id	order_purchase_timestamp	order_delivered_customer_date	order_estimated_delivery_date	time_to_delivery	diff_estimated_delivery
1	bfbd0f9bdef84302105ad712db...	2016-09-15 12:16:38 UTC	2016-11-09 07:47:38 UTC	2016-10-04 00:00:00 UTC	54	-36
2	3b697a20d9e427646d925679...	2016-10-03 09:44:50 UTC	2016-10-26 14:02:13 UTC	2016-10-27 00:00:00 UTC	23	0
3	be5bc2f0da14d8071e2d45451...	2016-10-03 16:56:50 UTC	2016-10-27 18:19:38 UTC	2016-11-07 00:00:00 UTC	24	10
4	65d1e226dfaeb8cdc42f66542...	2016-10-03 21:01:41 UTC	2016-11-08 10:58:34 UTC	2016-11-25 00:00:00 UTC	35	16
5	a41c8759fbe7aab36ea07e038...	2016-10-03 21:13:36 UTC	2016-11-03 10:58:07 UTC	2016-11-29 00:00:00 UTC	30	25
6	d207cc272675637bfed0062ed...	2016-10-03 22:06:03 UTC	2016-10-31 11:07:42 UTC	2016-11-23 00:00:00 UTC	27	22
7	cd3b8574c82b42fc8129f6d50...	2016-10-03 22:31:31 UTC	2016-10-14 16:08:00 UTC	2016-11-23 00:00:00 UTC	10	39
8	ae8a60e4b03c5a4ba9ca0672c...	2016-10-03 22:44:10 UTC	2016-11-03 14:04:50 UTC	2016-12-01 00:00:00 UTC	30	27
9	ef1b29b591d31d57c0d733746...	2016-10-03 22:51:30 UTC	2016-11-01 15:14:45 UTC	2016-11-25 00:00:00 UTC	28	23
10	0a0837a5eee9e7a9ce2b1fa83...	2016-10-04 09:06:10 UTC	2016-10-22 14:51:18 UTC	2016-11-24 00:00:00 UTC	18	32

Q 5.3 Analysis on sales, freight and delivery time: Group data by state, take mean of freight_value, time_to_delivery, diff_estimated_delivery

```
select c.customer_state, round(avg(x.freight_value),1) as mean_freight_value,
round(avg(timestamp_diff(o.order_delivered_customer_date,o.order_purchase_timestamp,day)),1)
as time_to_delivery,
round(avg(timestamp_diff(o.order_estimated_delivery_date, o.order_delivered_customer_date, da
y)),1) as diff_estimated_delivery
from `scaler-dsml-380816.Target.Order_items` x
join `scaler-dsml-380816.Target.Orders` o
on x.order_id = o.order_id
join `scaler-dsml-380816.Target.Customers` c
on o.customer_id = c.customer_id
where order_delivered_customer_date is not null
group by c.customer_state
order by c.customer_state;
```

Row	customer_state	mean_freight_value	time_to_delivery	diff_estimated_delivery
1	AC	40.0	20.3	20.0
2	AL	35.9	24.0	8.0
3	AM	33.3	26.0	19.0
4	AP	34.2	27.8	17.4
5	BA	26.5	18.8	10.1
6	CE	32.7	20.5	10.3
7	DF	21.1	12.5	11.3
8	ES	22.0	15.2	9.8
9	GO	22.6	14.9	11.4
10	MA	38.5	21.2	9.1

Q 5.5 Analysis on sales, freight and delivery time: Top 5 states with highest/lowest average freight value - sort in desc/asc limit 5

```
select c.customer_state, round(avg(x.freight_value),1) as mean_freight_value,
from `scaler-dsml-380816.Target.Order_items` x
join `scaler-dsml-380816.Target.Orders` o
on x.order_id = o.order_id
join `scaler-dsml-380816.Target.Customers` c
on o.customer_id = c.customer_id
```

```
where order_delivered_customer_date is not null
group by c.customer_state
order by mean_freight_value desc
limit 5;
```

Row	customer_state	mean_freight_value
1	PB	43.1
2	RR	43.1
3	RO	41.3
4	AC	40.0
5	PI	39.1

```
select c.customer_state, round(avg(x.freight_value),1) as mean_freight_value,
from `scaler-dsml-380816.Target.Order_items` x
join `scaler-dsml-380816.Target.Orders` o
on x.order_id = o.order_id
join `scaler-dsml-380816.Target.Customers` c
on o.customer_id = c.customer_id
where order_delivered_customer_date is not null
group by c.customer_state
order by mean_freight_value asc
limit 5;
```

Row	customer_state	mean_freight_value
1	SP	15.1
2	PR	20.5
3	MG	20.6
4	RJ	20.9
5	DF	21.1

Q 5.6 Analysis on sales, freight and delivery time: Top 5 states with highest/lowest average time to delivery

```
select c.customer_state,
round(avg(timestamp_diff(o.order_delivered_customer_date,o.order_purchase_timestamp,day)),1)
as time_to_delivery,
from `scaler-dsml-380816.Target.Order_items` x
join `scaler-dsml-380816.Target.Orders` o
on x.order_id = o.order_id
join `scaler-dsml-380816.Target.Customers` c
on o.customer_id = c.customer_id
where order_delivered_customer_date is not null
group by c.customer_state
order by time_to_delivery desc
limit 5;
```

Row	customer_state	time_to_delivery
1	AP	27.8
2	RR	27.8
3	AM	26.0
4	AL	24.0
5	PA	23.3

```
select c.customer_state,
round(avg(timestamp_diff(o.order_delivered_customer_date,o.order_purchase_timestamp,day)),1)
as time_to_delivery,
from `scaler-dsml-380816.Target.Order_items` x
join `scaler-dsml-380816.Target.Orders` o
on x.order_id = o.order_id
join `scaler-dsml-380816.Target.Customers` c
on o.customer_id = c.customer_id
where order_delivered_customer_date is not null
group by c.customer_state
order by time_to_delivery asc
limit 5;
```

Row	customer_state	time_to_delivery
1	SP	8.3
2	MG	11.5
3	PR	11.5
4	DF	12.5
5	SC	14.5

Q 5.7 Analysis on sales, freight and delivery time: Top 5 states where delivery is really fast/ not so fast compared to estimated date

```
select c.customer_state,
round(avg(timestamp_diff(o.order_estimated_delivery_date, o.order_delivered_customer_date, da
y)),1) as diff_estimated_delivery
from `scaler-dsml-380816.Target.Order_items` x
join `scaler-dsml-380816.Target.Orders` o
on x.order_id = o.order_id
join `scaler-dsml-380816.Target.Customers` c
on o.customer_id = c.customer_id
where order_delivered_customer_date is not null
group by c.customer_state
order by diff_estimated_delivery desc
limit 5;
```

Row	customer_state	diff_estimated_delivery
1	AC	20.0
2	RO	19.1
3	AM	19.0
4	AP	17.4
5	RR	17.4

```
select c.customer_state,
round(avg(timestamp_diff(o.order_estimated_delivery_date, o.order_delivered_customer_date, da
y)),1) as diff_estimated_delivery
from `scaler-dsml-380816.Target.Order_items` x
join `scaler-dsml-380816.Target.Orders` o
on x.order_id = o.order_id
join `scaler-dsml-380816.Target.Customers` c
on o.customer_id = c.customer_id
where order_delivered_customer_date is not null
group by c.customer_state
order by diff_estimated_delivery asc
limit 5;
```

Row	customer_state	diff_estimated_delivery
1	AL	8.0
2	MA	9.1
3	SE	9.2
4	ES	9.8
5	BA	10.1

Actionable Insight:

Q 6.1 Payment type analysis: Month over Month count of orders for different payment types

```
select extract(month from o.order_purchase_timestamp) as month,
payment_type, count(*) as no_of_orders
from `scaler-dsml-380816.Target.Orders` o
join `scaler-dsml-380816.Target.Payments` p
on o.order_id = p.order_id
group by month, payment_type
order by month, no_of_orders;
```

Row	month	payment_type	no_of_orders
1	1	debit_card	118
2	1	voucher	477
3	1	UPI	1715
4	1	credit_card	6103
5	2	debit_card	82
6	2	voucher	424
7	2	UPI	1723
8	2	credit_card	6609
9	3	debit_card	109
10	3	voucher	591

Actionable Insight:

More than 50% orders are payed by credit card, followed by UPI, voucher and debit card

Recommendations:

Since more than 50% customers pay through credit cards, credit card bank backed offers could boost the revenue

Q 6.2 Payment type analysis: Count of orders based on the no. of payment instalments

```
select payment_installments, count(*) as no_of_orders,
round(count(*)/(select count(*) from `scaler-dsml-380816.Target.Payments`)*100,2) as percentage
from `scaler-dsml-380816.Target.Payments`
group by payment_installments
order by payment_installments;
```

Row	payment_installments	no_of_orders	percentage
1	0	2	0.0
2	1	52546	50.58
3	2	12413	11.95
4	3	10461	10.07
5	4	7098	6.83
6	5	5239	5.04
7	6	3920	3.77
8	7	1626	1.57
9	8	4268	4.11
10	9	644	0.62

Actionable Insight:

50% of total orders are paid through single instalments, followed by two & three instalments.

Recommendations:

Order payment through Installments of more than 10 constitute to 1% of total payments. Therefore, optimizing interest rates for longer duration could attract more customers to opt for instalment payments of more than 10