1.1 Data type of all columns in the "customers" table.

```
SELECT column_name, data_type
FROM `Target.INFORMATION_SCHEMA.COLUMNS`
WHERE table_name = 'customers';
```

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

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

```
select
min(order_purchase_timestamp) as min_time,
max(order_purchase_timestamp) as max_time
from `Target.orders`
```

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

1.3 Count the Cities & States of customers who ordered during the given period.

```
  select
  count(distinct c.customer_city) as no_of_city,
  count(distinct c.customer_state) as no_of_states
  from `Target.orders` o
left join `Target.customers` c on o.customer_id =
c.customer_id
```

Row	no_of_city	no_of_states
1	4119	27

2.1 Is there a growing trend in the no. of orders placed over the past years?

```
select
extract(year from order_purchase_timestamp) as year,
count(distinct extract(month from order_purchase_timestamp))
as no_ofmonths,
count(order_id) as no_of_orders
from `Target.orders`
group by year
order by year;
```

Row	year	no_ofmonths	no_of_orders
1	2016	3	329
2	2017	12	45101
3	2018	10	54011

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

```
select
extract(year from order_purchase_timestamp) as year,
extract(month from order_purchase_timestamp) as month,
```

```
count(*)
from `Target.orders`
group by year,month
order by year,month
```

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

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

```
select
case when extract(hour from order_purchase_timestamp)
between 0 and 6 then "Dawn"
when extract(hour from order_purchase_timestamp) between 7
and 12 then "Mornings"
when extract(hour from order_purchase_timestamp) between 13
and 18 then "Afternoon"
```

```
   else "Night"
   end as Time_of_day,
   count(order_id) as total_orders
from `Target.orders`
group by Time_of_day
```

Row	Time_of_day	total_orders
1	Mornings	27733
2	Dawn	5242
3	Afternoon	38135
4	Night	28331

3.1 Get the month-on-month no. of orders placed in each state.

```
select
extract(year from order_purchase_timestamp) as year,
extract(month from order_purchase_timestamp) as month,
customer_state,
count(order_id)
from `Target.customers` c
left join `Target.orders` o on c.customer_id= o.customer_id
group by year,month,customer_state
order by year,month,customer_state
```

Row	year	month	customer_state	no_of_orders
1	2016	9	RR	1
2	2016	9	RS	1
3	2016	9	SP	2
4	2016	10	AL	2
5	2016	10	BA	4
6	2016	10	CE	8
7	2016	10	DF	6
8	2016	10	ES	4
9	2016	10	GO	9
10	2016	10	MA	4

3.2 How are the customers distributed across all the states?

```
select
customer_state,
count(distinct customer_id) as total_customers
from `Target.customers`
group by customer_state
order by total_customers
```

Row	customer_state	total_customers
1	RR	46
2	AP	68
3	AC	81
4	AM	148
5	RO	253
6	TO	280
7	SE	350
8	AL	413
9	RN	485
10	PI	495

4 . 1Get the % increase in the cost of orders from year 2017 to 2018 (include months between Jan to Aug only).

```
with cte as (select
sum(case when year = 2018 then payment_value else 0 end) as
total2018,
sum(case when year = 2017 then payment_value else 0 end) as
total2017
from (select
o.customer_id,
p.order_id,
extract(month from o.order_purchase_timestamp) as month,
extract(year from o.order_purchase_timestamp) as year,
p.payment_value as payment_value
```

```
from `Target.payments` p
join `Target.orders` o
on p.order_id = o.order_id
where extract(month from o.order_purchase_timestamp) in
(1,2,3,4,5,6,7,8)))

select
round(total2018*100/total2017,2) as percentage_increase
from cte
```

Row	percentage_increase
1	236.98

4.2Calculate the Total & Average value of order price for each state.

```
select
customer_state,
round(sum(oi.price),2) as total_order_price,
round(avg(oi.price),2) as avg_order_price
from `Target.customers` c
join `Target.orders`o  on c.customer_id = o.customer_id
join `Target.order_items` oi on o.order_id=oi.order_id
group by customer_state
```

Row	customer_state	total_order_price	avg_order_price
1	RN	83034.98	156.97
2	CE	227254.71	153.76
3	RS	750304.02	120.34
4	SC	520553.34	124.65
5	SP	5202955.05	109.65
6	MG	1585308.03	120.75
7	BA	511349.99	134.6
8	RJ	1824092.67	125.12
9	GO	294591.95	126.27

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

```
select
customer_state,
round(sum(oi.freight_value),2) as total_freight_value,
round(avg(oi.freight_value),2) as avg_freight_value
from `Target.customers` c
join `Target.orders`o  on c.customer_id = o.customer_id
join `Target.order_items` oi on o.order_id=oi.order_id
group by customer_state
```

Row	customer_state	total_freight_value	avg_freight_value
1	MT	29715.43	28.17
2	MA	31523.77	38.26
3	AL	15914.59	35.84
4	SP	718723.07	15.15
5	MG	270853.46	20.63
6	PE	59449.66	32.92
7	RJ	305589.31	20.96
8	DF	50625.5	21.04
9	RS	135522.74	21.74
10	SE	14111.47	36.65

5.1Find 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.

```
select
order_id,
date_diff(order_delivered_customer_date,order_purchase_timesta
mp,day) as delivery_time,
date_diff(order_estimated_delivery_date,order_delivered_custom
er_date,day) as diff_estimated_delivery
from `Target.orders`
```

Row	order_id	delivery_time	diff_estimated_delive
1	1950d777989f6a877539f5379...	30	-12
2	2c45c33d2f9cb8ff8b1c86cc28...	30	28
3	65d1e226dfaeb8cdc42f66542...	35	16
4	635c894d068ac37e6e03dc54e...	30	1
5	3b97562c3aee8bdedcb5c2e45...	32	0
6	68f47f50f04c4cb6774570cfde...	29	1
7	276e9ec344d3bf029ff83a161c...	43	-4
8	54e1a3c2b97fb0809da548a59...	40	-4
9	fd04fa4105ee8045f6a0139ca5...	37	-1
10	302bb8109d097a9fc6e9cefc5...	33	-5

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

```
with cte as (SELECT
c.customer_state,
dense_rank() over(ORDER BY AVG(oi.freight_value) desc) AS
highest_freight_value,
```

```
dense_rank() OVER (ORDER BY AVG(oi.freight_value)) AS
lowest_freight_value
from `Target.customers` c
JOIN`Target.orders` o ON c.customer_id = o.customer_id
JOIN`Target.order_items` oi ON o.order_id = oi.order_id
GROUP BY c.customer_state


)

SELECT
    a.customer_state AS five_highest_state,
    b.customer_state AS five_lowest_state
FROM
    cte a
JOIN
    cte b ON a.highest_freight_value = b.lowest_freight_value
    order by a.highest_freight_value
    limit 5
```

Row	five_highest_state	five_lowest_state
1	RR	SP
2	PB	PR
3	RO	MG
4	AC	RJ
5	PI	DF

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

```
with cte as (SELECT
c.customer_state,
dense_rank() over(ORDER BY
AVG(date_diff(order_delivered_customer_date,order_purchase_tim
estamp,day)) desc) AS highest_delivery_time,
dense_rank() OVER (ORDER BY
AVG(date_diff(order_delivered_customer_date,order_purchase_tim
estamp,day))) AS lowest_delivery_time
from `Target.customers` c
JOIN`Target.orders` o ON c.customer_id = o.customer_id
```

```
JOIN`Target.order_items` oi ON o.order_id = oi.order_id
GROUP BY c.customer_state


)

SELECT
    a.customer_state AS five_highest_state,
    b.customer_state AS five_lowest_state
FROM
    cte a
JOIN
    cte b ON a.highest_delivery_time = b.lowest_delivery_time
    order by a.highest_delivery_time
    limit 5
```

Row	five_highest_state	five_lowest_state
1	RR	SP
2	AP	PR
3	AM	MG
4	AL	DF
5	PA	SC

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

```
with cte as (select
c.customer_state,
avg(date_diff(o.order_delivered_customer_date,o.order_purchase
_timestamp,day)) as avg_delivery_time,
avg(date_diff(o.order_estimated_delivery_date,o.order_purchase
_timestamp,day))as avg_estimated_delivery_time,
avg(date_diff(o.order_estimated_delivery_date,o.order_purchase
_timestamp,day))-
avg(date_diff(o.order_delivered_customer_date,o.order_purchase
_timestamp,day)),
dense_rank()over (order by
avg(date_diff(o.order_estimated_delivery_date,o.order_purchase
```

```
_timestamp,day))-
avg(date_diff(o.order_delivered_customer_date,o.order_purchase
_timestamp,day)) desc) as fastest_delivery
from `Target.customers` c
join `Target.orders` o on c.customer_id = o.customer_id
group by c.customer_state
order by fastest_delivery
)
select
customer_state,
from cte
order by fastest_delivery
limit 5
```

Row	customer_state
1	AC
2	RO
3	AP
4	AM
5	RR

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

```
select
extract(year from order_purchase_timestamp) as year,
extract(month from order_purchase_timestamp) as month,
payment_type,
count(p.order_id) as no_of_orders
from `Target.orders` o
join `Target.payments` p  on o.order_id=p.order_id
group by year,month,payment_type
order by year,month,payment_type
```

Row	year	month	payment_type	no_of_orders
1	2016	9	credit_card	3
2	2016	10	UPI	63
3	2016	10	credit_card	254
4	2016	10	debit_card	2
5	2016	10	voucher	23
6	2016	12	credit_card	1
7	2017	1	UPI	197
8	2017	1	credit_card	583
9	2017	1	debit_card	9

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

```
select
payment_installments,
count(*) as total_orders
from `Target.payments`
group by payment_installments
```

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

ACTIONABLE INSIGHTS:

- The Delivery time of the state RR is the highest and as a result the no. of customer from the state is the lowest.
- State SP has the largest customer base and the fastest delivery time.
- 2018 gives the most number of orders in the 3 years but the no. of orders are decreasing in the end months of 2018.

RECOMMENDATOONS:

- Fasten the delivery in those states where no. of customers are less.