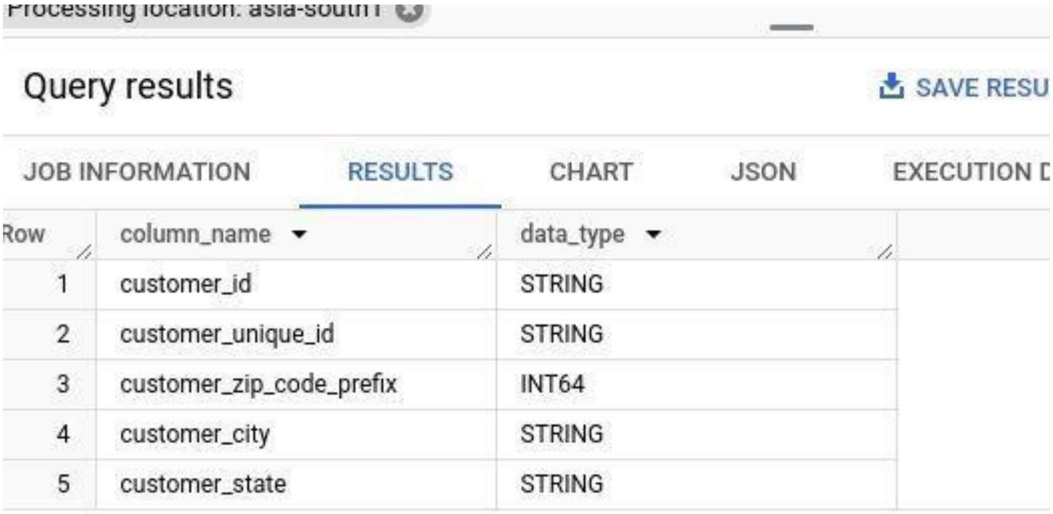
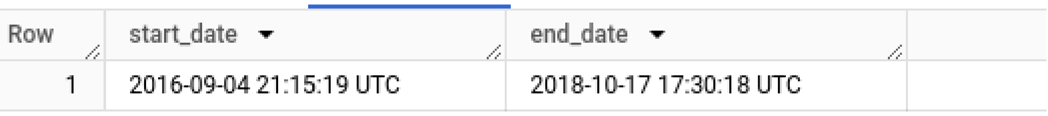
Q1

```
1.SELECT
column_name,
data_type
FROM
`scaler-dsml-sql-437510.target_sql_business_case.INFORMATION_SCHEMA.COLUMNS`
WHERE
table_name='customers_csv';
```



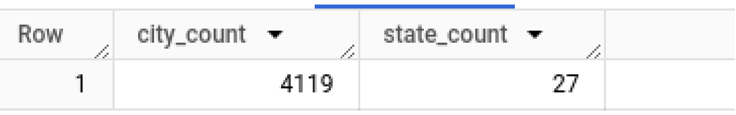
Row	column_name	data_type
1	customer_id	STRING
2	customer_unique_id	STRING
3	customer_zip_code_prefix	INT64
4	customer_city	STRING
5	customer_state	STRING

```
2.selectmin(order_purchase_timestamp)asstart_date,
max(order_purchase_timestamp) as end_date
from`target_sql_business_case.orders_csv`
```



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

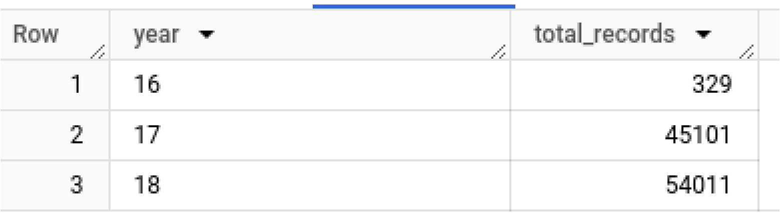
```
3.selectcount(DISTINCTcustomer_city)ascity_count,
count( DISTINCT customer_state) as state_count
from`target_sql_business_case.customers_csv`
```



Row	city_count	state_count
1	4119	27

Q2

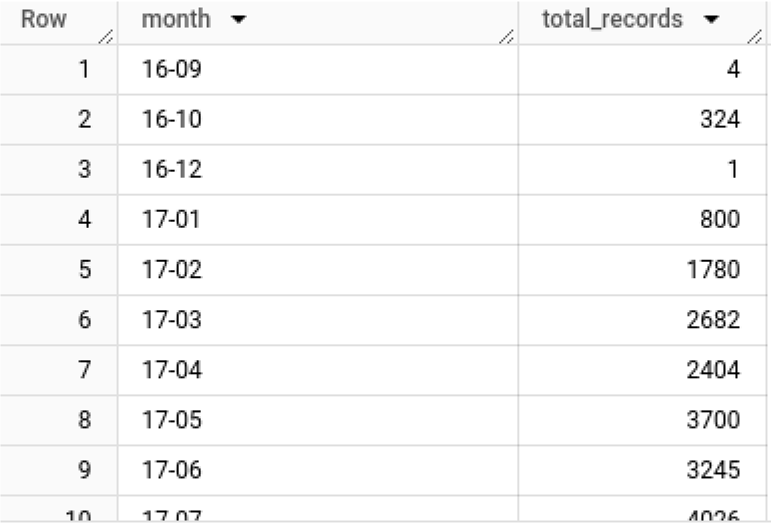
```
1. SELECT
FORMAT_DATETIME('%y',order_purchase_timestamp)ASyear,
COUNT(*)AStotal_records
from`target_sql_business_case.orders_csv`
GROUP BY
year
ORDERBY
year
```



Row	year	total_records
1	16	329
2	17	45101
3	18	54011

Thebusinessgrowthfrom2016to2017showsincreasewhichcould indicateabusniessexpansion oramaket demand mayincreaseandtheyaresuccesfulintheir marketingstartegiesthat continued to 2018 if this continues might be theyget further growth in 2019 and beyond

```
2. SELE
CT
FORMAT_DATETIME('%y-%m',order_purchase_timestamp)ASmonth,
COUNT(*)AStotal_records
from`target_sql_business_case.orders_csv`
GROUP BY
month
ORDERBY
month
```

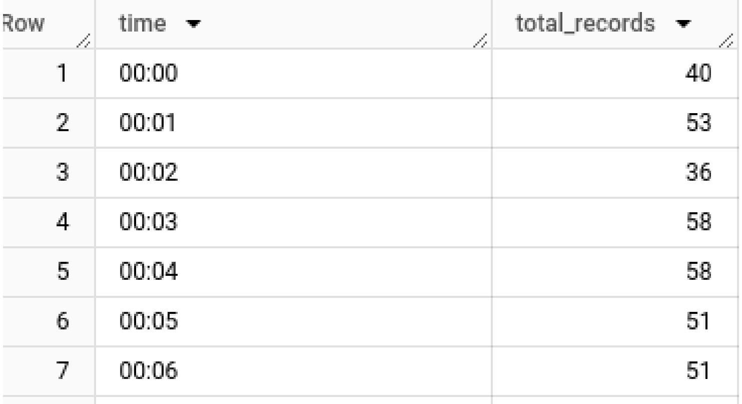


Row	month	total_records
1	16-09	4
2	16-10	324
3	16-12	1
4	17-01	800
5	17-02	1780
6	17-03	2682
7	17-04	2404
8	17-05	3700
9	17-06	3245
10	17-07	4026

Investigate what factors may have contributed to the low order volume at the end of2016. This could havebeenduetoalate launch, ineffective marketingcampaigns,orthebusinessstillscaling up.

3.

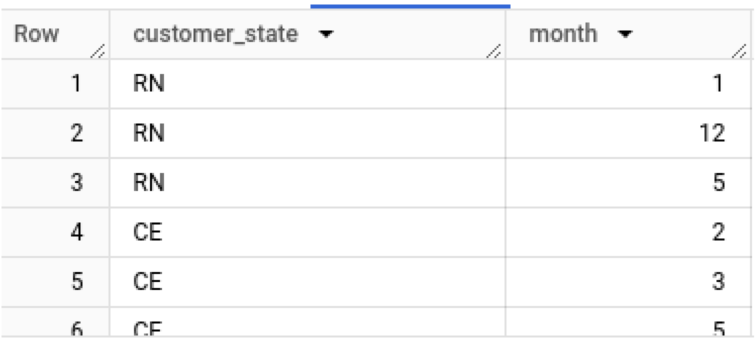
```
SELECT
FORMAT_timestamp('%H:%M',order_purchase_timestamp)AStime,
COUNT(*)AStotal_records
from`target_sql_business_case.orders_csv`
GROUP BY
time
order bytime
```



Row	time	total_records
1	00:00	40
2	00:01	53
3	00:02	36
4	00:03	58
5	00:04	58
6	00:05	51
7	00:06	51

Q3

```
1.selectDISTINCT(customer_state),extract(MONTHfromorder_purchase_timestamp)asmonth,
from `target_sql_business_case.customers_csv` c
join`target_sql_business_case.orders_csv`o on
c.customer_id = o.customer_iD
```

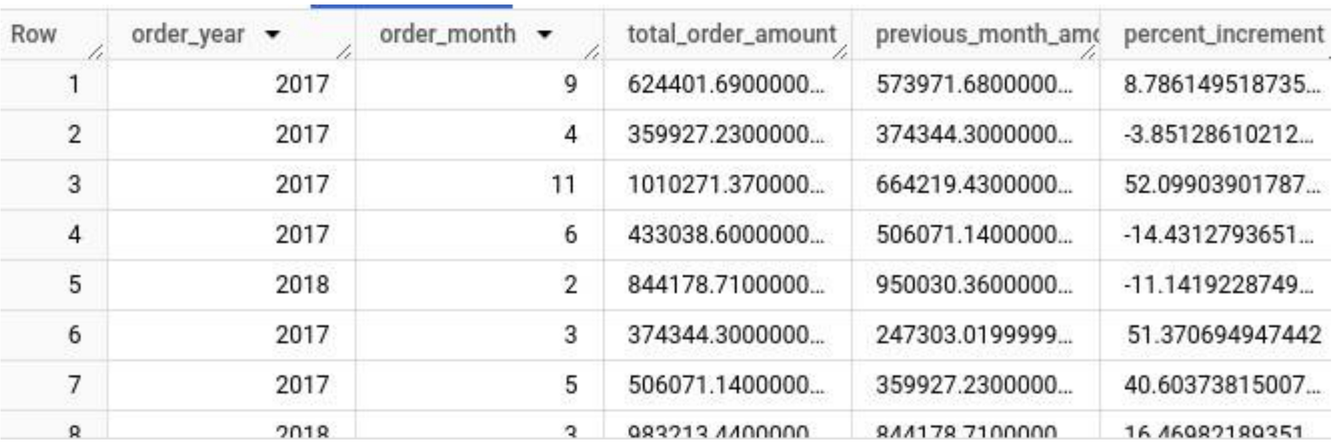


Row	customer_state	month
1	RN	1
2	RN	12
3	RN	5
4	CE	2
5	CE	3
6	CE	5

Q4

1.

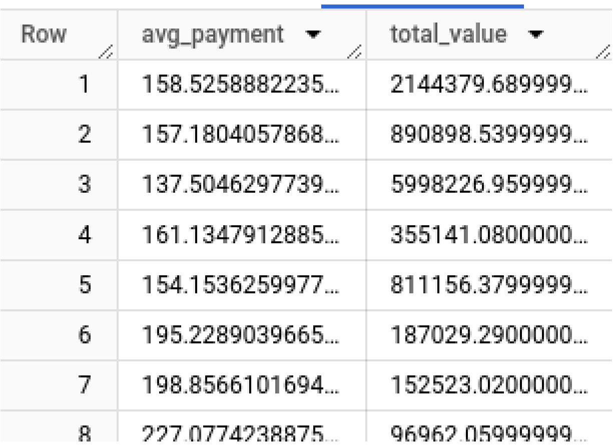
```
WITHcus_detailsAS(
  SELECT
    EXTRACT(YEAR FROM oo.order_purchase_timestamp) AS order_year,
    EXTRACT(MONTHFROMoo.order_purchase_timestamp)ASorder_month,
    SUM(o.price) AS total_order_amount
  FROM
    `target_sql_business_case.order_items` o
  JOIN
    `target_sql_business_case.orders_csv`oo
    ON o.order_id = oo.order_id
  WHERE
    oo.order_purchase_timestampBETWEEN'2017-01-31'AND'2018-08-30' GROUP
  BY
    order_year,order_month
  ORDER BY
    order_year,order_month
)
SELECT
  order_year,
  order_month,
  total_order_amount,
  LAG(total_order_amount)OVER(ORDERBYorder_year,order_month)AS
previous_month_amount,


  SAFE_DIVIDE(
    (total_order_amount-LAG(total_order_amount)OVER(ORDERBYorder_year, order_month)),
    LAG(total_order_amount)OVER(ORDERBYorder_year,order_month)
  )*100ASpercent_increment
FROM cus_details;
```



Row	order_year	order_month	total_order_amount	previous_month_amo	percent_increment
1	2017	9	624401.6900000...	573971.6800000...	8.786149518735...
2	2017	4	359927.2300000...	374344.3000000...	-3.85128610212...
3	2017	11	1010271.370000...	664219.4300000...	52.09903901787...
4	2017	6	433038.6000000...	506071.1400000...	-14.4312793651...
5	2018	2	844178.7100000...	950030.3600000...	-11.1419228749...
6	2017	3	374344.3000000...	247303.0199999...	51.370694947442
7	2017	5	506071.1400000...	359927.2300000...	40.60373815007...
8	2018	3	983213.4400000	844178.7100000	16.46982189351

2.

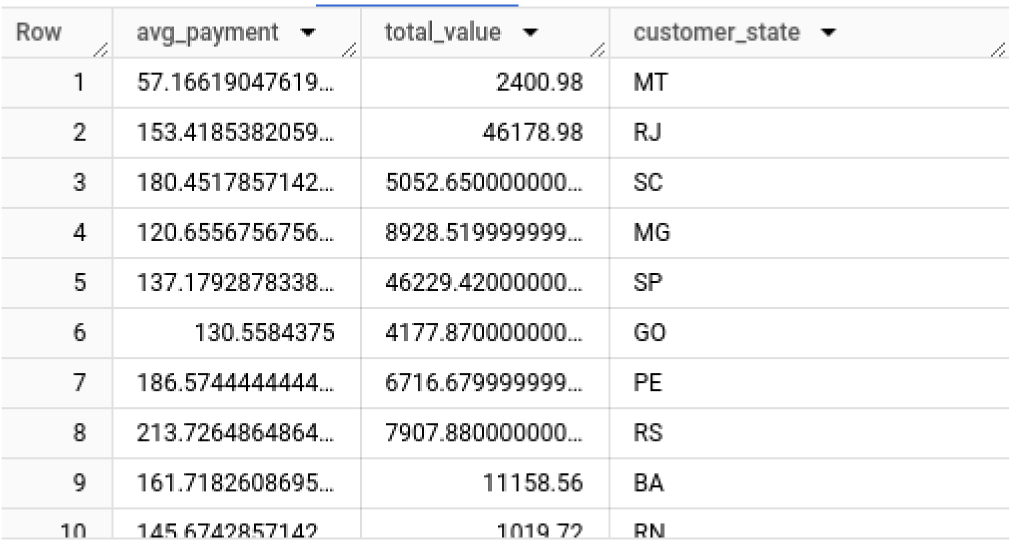
```
SELECTAVG(payment_value)ASavg_payment,sum(payment_value)astotal_value from
`target_sql_business_case.payments_csv`P
join `target_sql_business_case.orders_csv`o
on p.order_id = o.order_id
join`target_sql_business_case.customers_csv`c on
o.customer_id = c.customer_id
groupbyc.customer_state
```



Row	avg_payment	total_value
1	158.5258882235...	2144379.689999...
2	157.1804057868...	890898.5399999...
3	137.5046297739...	5998226.959999...
4	161.1347912885...	355141.0800000...
5	154.1536259977...	811156.3799999...
6	195.2289039665...	187029.2900000...
7	198.8566101694...	152523.0200000...
8	227.0774238875...	96962.05999999...

3.

```
SELECTAVG(payment_value)ASavg_payment,sum(payment_value)as
total_value,c.customer_state
from `target_sql_business_case.payments_csv`P
join `target_sql_business_case.orders_csv`o
onp.order_id= o.order_id
join`target_sql_business_case.customers_csv`c on
o.customer_id = c.customer_id
whereorder_status='shipped' group
by c.customer_state
```

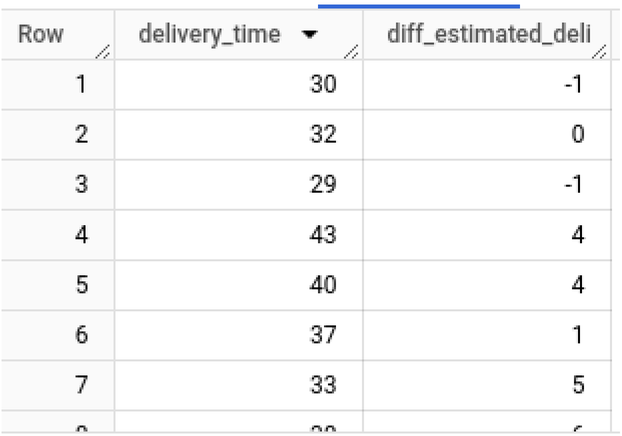


Row	avg_payment	total_value	customer_state
1	57.16619047619...	2400.98	MT
2	153.4185382059...	46178.98	RJ
3	180.4517857142...	5052.650000000...	SC
4	120.6556756756...	8928.519999999...	MG
5	137.1792878338...	46229.42000000...	SP
6	130.5584375	4177.870000000...	GO
7	186.5744444444...	6716.679999999...	PE
8	213.7264864864...	7907.880000000...	RS
9	161.7182608695...	11158.56	BA
10	145.6742857142	1019.72	RN

Q5

1.

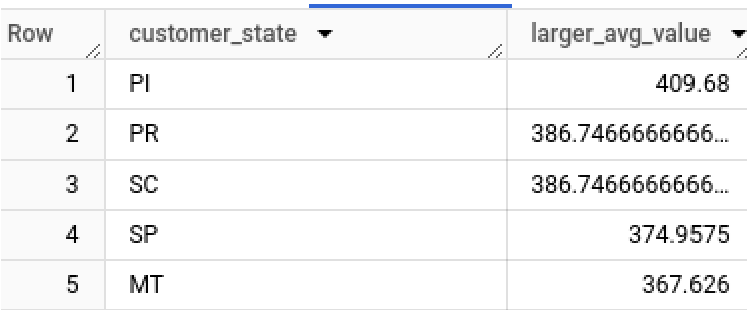
```
selectTIMESTAMP_DIFF(order_delivered_customer_date,order_purchase_timestamp,day)as
delivery_time ,
TIMESTAMP_DIFF(order_delivered_customer_date,order_estimated_delivery_date,day)as
diff_estimated_deli
from`target_sql_business_case.orders_csv`
where order_status = 'delivered'
```



Row	delivery_time	diff_estimated_deli
1	30	-1
2	32	0
3	29	-1
4	43	4
5	40	4
6	37	1
7	33	5

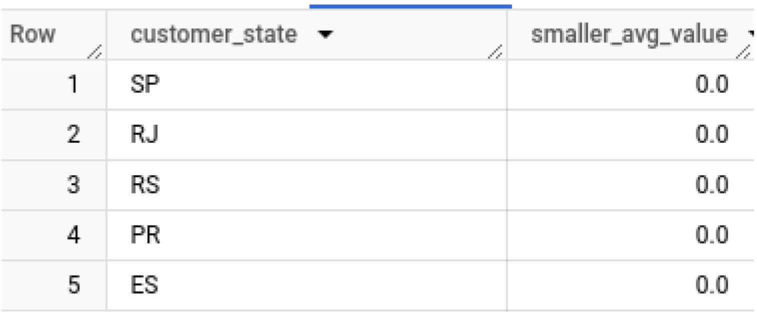
2.

```
selectDISTINCT(customer_state),
avg(freight_value)over(orderbyfreight_valuedesc)aslarger_avg_value from
`target_sql_business_case.order_items` io
join`target_sql_business_case.orders_csv`o on
io.order_id = o.order_id
join`target_sql_business_case.customers_csv`c on
c.customer_id = o.customer_id
limit5
```



Row	customer_state	larger_avg_value
1	PI	409.68
2	PR	386.7466666666...
3	SC	386.7466666666...
4	SP	374.9575
5	MT	367.626

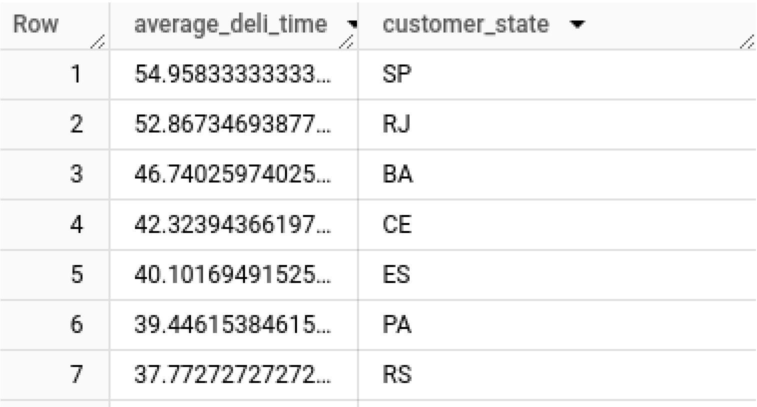
```
selectDISTINCT(customer_state),
avg(freight_value)over(orderbyfreight_valueasc)assmaller_avg_value from
`target_sql_business_case.order_items` io
join`target_sql_business_case.orders_csv`o on
io.order_id = o.order_id
join`target_sql_business_case.customers_csv`c on
c.customer_id = o.customer_id
limit5
```



Row	customer_state	smaller_avg_value
1	SP	0.0
2	RJ	0.0
3	RS	0.0
4	PR	0.0
5	ES	0.0

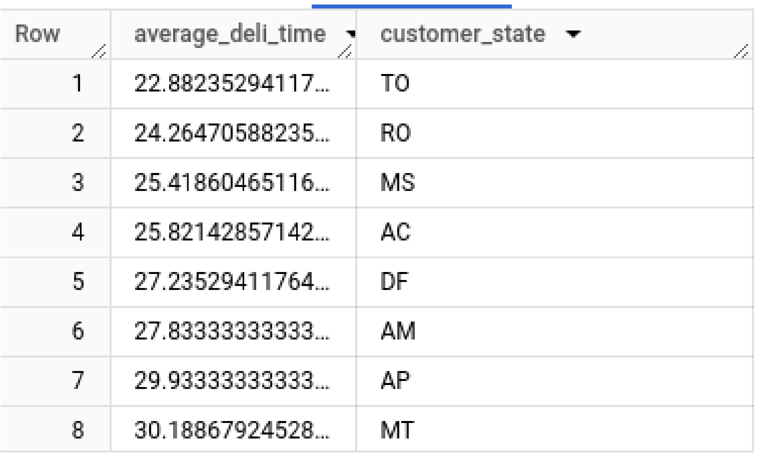
3.

```
#for highest
withreach_time
as
(select DISTINCT
( customer_state) ,TIMESTAMP_DIFF(order_delivered_customer_date,order_purchase_timestamp,
day) as delivery_time
from`target_sql_business_case.customers_csv`c
join `target_sql_business_case.orders_csv` o
onc.customer_id=o.customer_id
where order_status = 'delivered'
order by delivery_time asc)

selectavg(reach_time.delivery_time)asaverage_deli_time,customer_state
from reach_time
groupbycustomer_state
order byaverage_deli_timedesc
```



Row	average_deli_time	customer_state
1	54.95833333333...	SP
2	52.86734693877...	RJ
3	46.74025974025...	BA
4	42.32394366197...	CE
5	40.10169491525...	ES
6	39.44615384615...	PA
7	37.77272727272...	RS

```
#forlowestavg

withreach_time
as
(select DISTINCT
( customer_state) ,TIMESTAMP_DIFF(order_delivered_customer_date,order_purchase_timestamp,
day) as delivery_time
from`target_sql_business_case.customers_csv`c
join `target_sql_business_case.orders_csv` o
onc.customer_id=o.customer_id
where order_status = 'delivered'
order by delivery_time asc)

selectavg(reach_time.delivery_time)asaverage_deli_time,customer_state
from reach_time
groupbycustomer_state
orderbyaverage_deli_timeasc
```

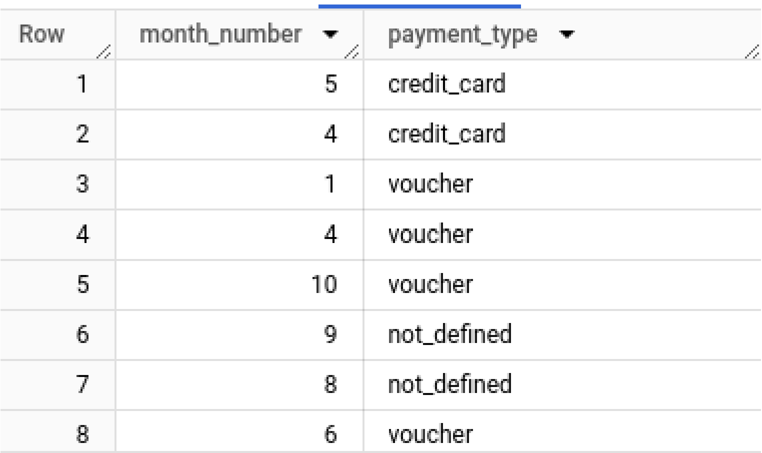


Row	average_deli_time	customer_state
1	22.88235294117...	TO
2	24.26470588235...	RO
3	25.41860465116...	MS
4	25.82142857142...	AC
5	27.23529411764...	DF
6	27.83333333333...	AM
7	29.93333333333...	AP
8	30.18867924528...	MT

Q6

1.

```
SELECT
EXTRACT(MONTHFROMo.order_purchase_timestamp)ASmonth_number,
p.payment_type
FROM`target_sql_business_case.payments_csv`p JOIN
`target_sql_business_case.orders_csv` o
ONp.order_id=o.order_id
GROUPBYmonth_number,p.payment_type;
```



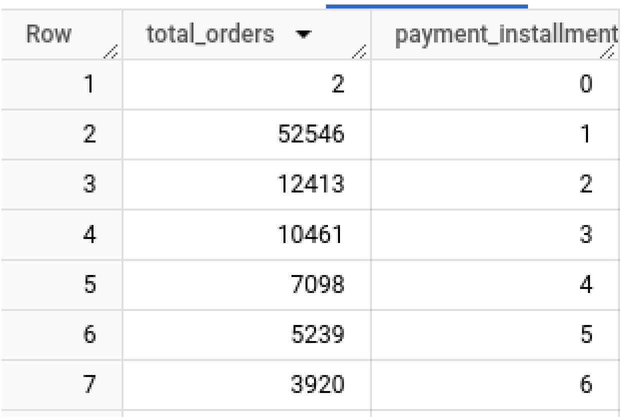
Row	month_number	payment_type
1	5	credit_card
2	4	credit_card
3	1	voucher
4	4	voucher
5	10	voucher
6	9	not_defined
7	8	not_defined
8	6	voucher

Identiftywhichpayment type isusedbymost ofthecustomersand lookforseasonablepayment pattern so we can promote that payment method accordingly .

Encourage payment method review from customers Investigate if different customer segments preferspecificpaymenttypes.Thiscould help indesigningtargetedmarketingcampaignsorloyalty programs.

2.

```
selectCOUNT(order_id)astotal_orders,payment_installments from
`target_sql_business_case.payments_csv`
groupbypayment_installments
```



Row	total_orders	payment_installment
1	2	0
2	52546	1
3	12413	2
4	10461	3
5	7098	4
6	5239	5
7	3920	6

Recommendation

*Offering discounts or rewards for customers who pay installments earlier

*and Collect feedback from customers about their payment experience to identify points and areas for improvement

*Enhance communication with customers regarding their payment statuses ,upcoming due dates, and reminders to payments