1. Data type of all columns in the "customers" table.
    1.

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

2. select max(order_purchase_timestamp) as Last_date_of_order_,
   min(order_purchase_timestamp) as
   Date_of_first_order,date_diff(max(order_purchase_timestamp),min(order_purcha
   se_timestamp),day) as Time_span_in_days_between_orders from
   `customers.orders`
    1.

select customer_city,count(customer_city) as
Total_no_of_cities_based_on_order_placed, customer_state,count

(customer_state) as Total_no_of_states_based_on_order_placed
from `customers.customers` where customer_id in (select
customer_id from `customers.orders` where
order_purchase_timestamp between '2017-11-25 11:10:33' and
'2018-01-13 15:41:34') group by customer_city,customer_state
Qn:
Qy: select 'No_of_cities' as Category,count(distinct
customer_city)as Stastistics from `customers.customers`

union all

select 'No_of_states' as Category,count(distinct
customer_state) as Stastistics from `customers.customers`

2.1
select count(order_id) as No_of_orders,(extract(year FROM
order_purchase_timestamp)) as Year_of_orders from `customers.orders` group by
Year_of_orders order by Year_of_orders

```
1  select count(order_id) as No_of_orders,(extract(year FROM order_purchase_timestamp)) as Year_of_orders  from
   `customers.orders` group by Year_of_orders  order by Year_of_orders
```

Press Alt+F1 for Accessibili

Query results ⬇ SAVE RESULTS ▾   📊 EXPLORE DATA ▾

| JOB INFORMATION | **RESULTS** | JSON | EXECUTION DETAILS | CHART **PREVIEW** | EXECUTION GRAPH |

Row	No_of_orders ▾	Year_of_orders ▾
1	329	2016
2	45101	2017
3	54011	2018

ENG   03:

2.2

```
select extract(month FROM order_purchase_timestamp) as Month,count(order_id) as
No_of_orders
 from `customers.orders` group by Month
order  by Month
```

```
1  select extract(month FROM order_purchase_timestamp) as Month,count(order_id) as No_of_orders
2  | from `customers.orders` group by Month
3  order  by Month
```

Press Alt+F1 for Accessibility Options.

Query results ⬇ SAVE RESULTS ▾   📊 EXPLORE DATA ▾ ↕

| JOB INFORMATION | **RESULTS** | JSON | EXECUTION DETAILS | CHART **PREVIEW** | EXECUTION GRAPH |

Row	Month ▾	No_of_orders ▾
3	3	9893
4	4	9343
5	5	10573
6	6	9412
7	7	10318
8	8	10843
9	9	4305
10	10	4959
11	11	7544
12	12	5674

Results per page: 50 ▾   1 – 12 of 12 |< < > >|

```
select  case when (extract(hour FROM order_purchase_timestamp)) betwee
 when (extract(hour FROM order_purchase_timestamp)) between 18 and 23
when (extract(hour FROM order_purchase_timestamp)) between 6 and 11 th
when (extract(hour FROM order_purchase_timestamp)) between 12 and 17 t
end as Time_of_the_day_when_orders_were_placed,count(order_id) as No_o
group by Time_of_the_day_when_orders_were_placed order by No_of_orders
 desc
```

query results

JOB INFORMATION | **RESULTS** | JSON | EXECUTION DETAILS | CHA

Time_of_the_day_when_orders_were_placed ▼	No_of_orders_placed ▼	
1	Afternoon	38361
2	Night	34100
3	Morning	22240
4	Dawn	4740

3.2

```
select customer_state as Name_of_the_state,count(customer_unique_id) as No_of_customers from `customers.customers`
group by  customer_state
```

Press Alt+F1 for Accessibility Option

Query results   🖫 SAVE RESULTS ▾   📊 EXPLORE DATA ▾   ↕

B INFORMATION   **RESULTS**   JSON   EXECUTION DETAILS   CHART PREVIEW   EXECUTION GRAPH

	Name_of_the_state ▾	No_of_customers ▾
1	RN	485
2	CE	1336
3	RS	5466
4	SC	3637
5	SP	41746
6	MG	11635
7	BA	3380
8	RJ	12852
9	GO	2020
0	MA	747

Results per page: 50 ▾   1 – 27 of 27 |< < > >|

```
1  --Get the month on month no. of orders placed in each state.
2  select count(o.order_id) as No_of_orders, c.customer_state as State , extract(month from o.
   order_purchase_timestamp)  as Month from `customers.customers` c join `customers.orders` o using(customer_id)
```

Press Alt+F1 for Accessibility Options.

Query results   🖫 SAVE RESULTS ▾   📊 EXPLORE DATA ▾   ↕

JOB INFORMATION   **RESULTS**   JSON   EXECUTION DETAILS   CHART PREVIEW   EXECUTION GRAPH

ow	No_of_orders ▾	State ▾	Month ▾
1	8	AC	1
2	39	AL	1
3	12	AM	1
4	11	AP	1
5	264	BA	1
6	99	CE	1
7	151	DF	1
8	159	ES	1
9	164	GO	1
10	66	MA	1

Results per page: 50 ▾   1 – 50 of 322 |< < > >|

PERSONAL HISTORY   PROJECT HISTORY   🔄 REFRESH ∧

3:2

How are the customers distributed across all the states?
96136,96096,99441,99441,99441,99441

```
select count(distinct c.customer_unique_id) as  No_of_unique_customers,
c.customer_state as State  from `customers.customers`  as c join `customers.orders`
as o on c.customer_id=o.customer_id   group by  c.customer_state order  by
c.customer_state
```

```
1  --How are the customers distributed across all the states? 96136,96096,99441,99441,99441,99441
2  select count(distinct c.customer_unique_id) as  No_of_unique_customers, c.customer_state as State  from `customers.
```

Press Alt+F1 for Accessibility Options.

Query results

⬇ SAVE RESULTS ▾   📊 EXPLORE DATA ▾   ↕

JOB INFORMATION   **RESULTS**   JSON   EXECUTION DETAILS   CHART **PREVIEW**   EXECUTION GRAPH

Row	No_of_unique_customers ▾	State ▾
1	77	AC
2	401	AL
3	143	AM
4	67	AP
5	3277	BA
6	1313	CE
7	2075	DF
8	1964	ES
9	1952	GO
10	726	MA

Results per page: 50 ▾   1 – 27 of 27   I< < > >I

4:1

```
select o.order_id,payment_value,lead(payment_value) over (order by payment_value
asc) as cost_of_order_lead_value , case when (lead(payment_value) over (order by
payment_value asc)-payment_value) >1 then ((lead(payment_value) over (order by
payment_value asc)-payment_value)/payment_value)*.01
when (lead(payment_value) over (order by payment_value asc)-payment_value) =0 then
0
when payment_value=0 then 0
else -((lead(payment_value) over (order by payment_value asc)-
payment_value)/payment_value)*.01  end as Percentage_increase from
`customers.payments` p join `customers.orders` o on o.order_id=p.order_id where
o.order_purchase_timestamp between '2017-01-01' and '2018-08-31' order  by
payment_value,cost_of_order_lead_value
```

RUN SAVE ▾ SHARE ▾ SCHEDULE MORE ▾

```
1  select o.order_id,payment_value,lead(payment_value) over (order by payment_value asc) as
   cost_of_order_lead_value , case when (lead(payment_value) over (order by payment_value asc)-payment_value) >1 then
```

Press Alt+F1 for Accessibility Options.

Query results

SAVE RESULTS ▾   EXPLORE DATA ▾

| | JOB INFORMATION | **RESULTS** | JSON | EXECUTION DETAILS | CHART PREVIEW | EXECUTION GRAPH |

Row	order_id ▾	payment_value ▾	cost_of_order_lead_v	Percentage_increase
103510	8dbc85d1447242f3b127dda39...	4681.78	4764.34	0.000176343185...
103511	199af31afc78c699f0dbf71fb1...	4764.34	4809.44	9.466159006284...
103512	b4c4b76c642808cbe472a32b8...	4809.44	4950.34	0.000292965501...
103513	a96610ab360d42a2e5335a399...	4950.34	6081.54	0.002285095569...
103514	2cc9089445046817a7539d90...	6081.54	6726.66	0.001060783946...
103515	f5136e38d1a14a4dbd87dff67d...	6726.66	6922.21	0.000290708910...
103516	fefacc66af859508bf1a7934ea...	6922.21	6929.31	1.025683993984...
103517	0812eb902a67711a1cb742b3c...	6929.31	7274.88	0.000498707663...
103518	736e1922ae60d0d6a89247b8...	7274.88	13664.08	0.008782550365...
103519	03caa2c082116e1d31e67e9ae...	13664.08	*null*	*null*

Results per page: 50 ▾   103501 – 103519 of 103519 |< < > >|

```
  5:1 b
select order_id,
abs(date_diff(order_purchase_timestamp,order_delivered_custome
r_date,day)) as
No_of_days_taken_for_the_order_to_be_delivered  from
`customers.orders`
```

RUN   SAVE ▾   SHARE ▾   SCHEDULE   MORE ▾

```
1 | select order_id, abs(date_diff(order_purchase_timestamp,order_delivered_customer_date,day)) as
    No_of_days_taken_for_the_order_to_be_delivered  from `customers.orders`
```

Press Alt+F

Query results

SAVE RESULTS ▾   EXPL(

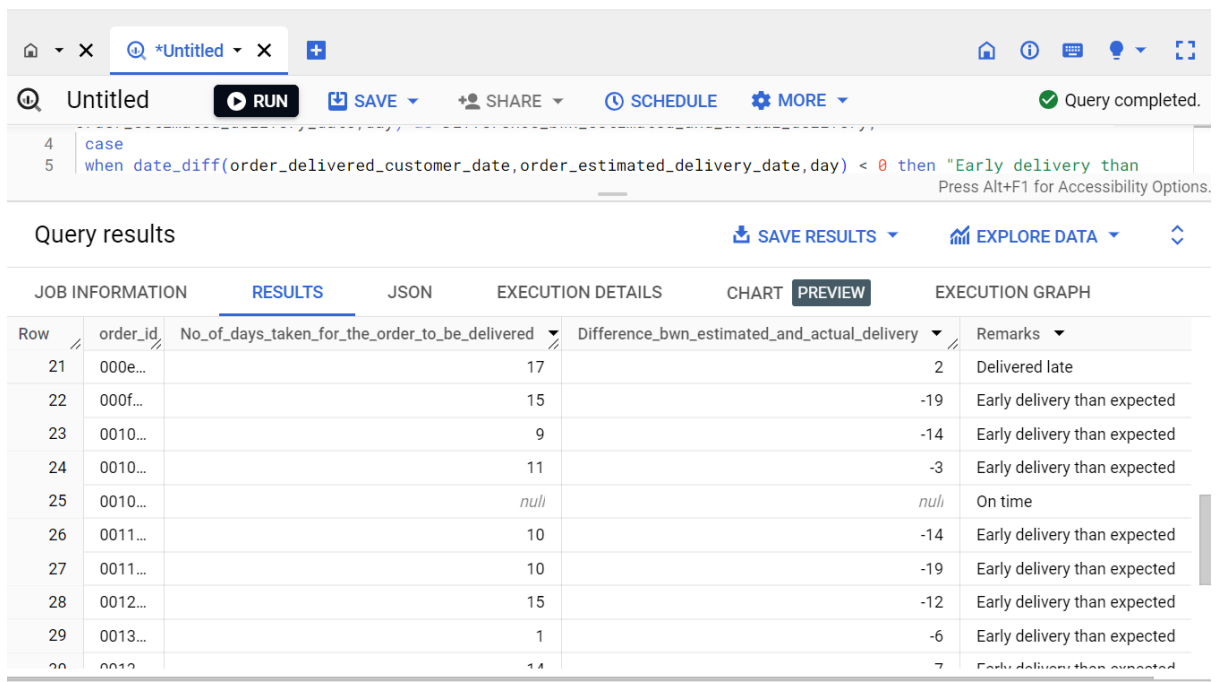| | JOB INFORMATION | **RESULTS** | JSON | EXECUTION DETAILS | CHART PREVIEW | EXECUTION |

Row	order_id ▾	No_of_days_taken_for_the_order_to_be_delivered ▾
1	1950d777...	30
2	2c45c33d...	30
3	65d1e226...	35
4	635c894d...	30
5	3b97562c...	32
6	68f47f50f...	29
7	276e9ec3...	43
8	54e1a3c2...	40
9	fd04fa410...	37
10	302bb810...	33

Results per page: 50 ▾   1 – 50 of 99441 |<

```sql
select
order_id,date_diff(order_delivered_customer_date,order_purchase_timestamp,day) as
No_of_days_taken_for_the_order_to_be_delivered
,date_diff(order_delivered_customer_date,order_estimated_delivery_date,day) as
Difference_bwn_estimated_and_actual_delivery,
 case
 when date_diff(order_delivered_customer_date,order_estimated_delivery_date,day) <
0 then "Early delivery than expected"
 when date_diff(order_delivered_customer_date,order_estimated_delivery_date,day) >
0 then "Delivered late"
 else
 "On time"
end as Remarks from `customers.orders` order by order_id
```

Row	order_id	No_of_days_taken_for_the_order_to_be_delivered	Difference_bwn_estimated_and_actual_delivery	Remarks
21	000e...	17	2	Delivered late
22	000f...	15	-19	Early delivery than expected
23	0010...	9	-14	Early delivery than expected
24	0010...	11	-3	Early delivery than expected
25	0010...	null	null	On time
26	0011...	10	-14	Early delivery than expected
27	0011...	10	-19	Early delivery than expected
28	0012...	15	-12	Early delivery than expected
29	0013...	1	-6	Early delivery than expected
30	0013...	14	7	Early delivery than expected

5:2
```sql
select  count(o.order_id)as No_of_orders,sum(freight_value)as
Total_freight_value,sum(freight_value)/count(o.order_id) as
Increasing_order_of_the_avg_freight_value,c.customer_state as State,row_number()
over(order by  sum(freight_value)/count(o.order_id)) as Order_of_the_states from
`customers.order_items` as i join `customers.orders` as o on i.order_id =o.order_id
join `customers.customers` as c on c.customer_id=o.customer_id group by
c.customer_state
```

| | Untitled | ▶ RUN | 🖫 SAVE ▾ | +⚹ SHARE ▾ | ⏱ SCHEDULE | ⚙ MORE ▾ | | ✓ Query completed. |

```
1  select  count(o.order_id)as No_of_orders,sum(freight_value)as Total_freight_value,sum(freight_value)/count(o.
   order_id) as Increasing_order_of_the_avg_freight_value,c.customer_state as State,row_number() over(order by  sum
```

Query results ⬇ SAVE RESULTS ▾   📊 EXPLORE DATA ▾   ↕

| JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | CHART **PREVIEW** | EXECUTION GRAPH |

Row	No_of_orders ▾	Total_freight_value	Increasing_order_of_the_avg_freight_value	State ▾	Order_of_the_states
1	47449	718723.0699999...	15.147275390418836	SP	1
2	5740	117851.6800000...	20.531651567944493	PR	2
3	13129	270853.4600000...	20.630166806306921	MG	3
4	14579	305589.3100000...	20.960923931682579	RJ	4
5	2406	50625.49999999...	21.041354945968333	DF	5
6	4176	89660.26000000...	21.470368773946465	SC	6
7	6235	135522.7400000...	21.735804330393282	RS	7
8	2256	49764.59999999...	22.058776595744632	ES	8
9	2333	53114.97999999...	22.7668152593227	GO	9
10	819	19144.03000000...	23.374884004884013	MS	10

Results per page: 50 ▾   1 – 27 of 27 |< < > >|

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

```
select count(p.order_id) as No_of_orders , p.payment_type,
extract(month FROM o.order_purchase_timestamp) as month from
`customers.orders` o join `customers.payments`
p  using(order_id) group by extract(month FROM
o.order_purchase_timestamp),p.payment_type

--select count(p.order_id) as No_of_orders, p.payment_type,
FORMAT_date("%B",o.order_purchase_timestamp) as Month  from
`customers.orders` o join `customers.payments`
p  using(order_id) group
by  FORMAT_date("%B",o.order_purchase_timestamp),p.
payment_type order by order_purchase_timestamp
select date_diff(order_delivered_customer_date,order_estimated_delivery_date,day)
as Difference_bwn_estimated_and_actual_delivery,c.customer_state as State from
`customers.customers` c join `customers.orders` o on
o.customer_id=c.customer_id  where
date_diff(order_delivered_customer_date,order_estimated_delivery_date,day) is not
null group by State,Difference_bwn_estimated_and_actual_delivery order by
Difference_bwn_estimated_and_actual_delivery limit 5

5:3 select count(o.order_id)
Orders,c.customer_state,o.order_delivered_customer_date
,date_diff(o.order_delivered_customer_date,order_purchase_timestamp,day) as
Delivery_time,sum(date_diff(o.order_delivered_customer_date,order_purchase_timestam
p,day))/count(o.order_id) as Average_delivery_time from `customers.customers` c
join `customers.orders` o using(customer_id) group by c.customer_state,
date_diff(o.order_delivered_customer_date,order_purchase_timestamp,day),o.order_del
ivered_customer_date
```

Untitled ▶ RUN 💾 SAVE ▾ +⊙ SHARE ▾ ⏱ SCHEDULE ⚙ MORE ▾ ✅ Query completed.

```
1  select date_diff(order_delivered_customer_date,order_estimated_delivery_date,day) as
   Difference_bwn_estimated_and_actual_delivery,c.customer_state as State from `customers.customers` c join
   `customers.orders` o on o.customer_id=c.customer_id  where date_diff(order_delivered_customer_date,
   order_estimated_delivery_date,day) is not null group by State,Difference_bwn_estimated_and_actual_delivery order
   by Difference_bwn_estimated_and_actual_delivery limit 5
```

Press Alt+F1 for Accessibility Options.

Query results

⬇ SAVE RESULTS ▾   📊 EXPLORE DATA ▾   ⇕

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | CHART PREVIEW | EXECUTION GRAPH

Row	Difference_bwn_esti	State ▾
1	-146	SP
2	-139	MA
3	-134	RS
4	-123	SP
5	-108	RJ

Untitled ▶ RUN 💾 SAVE ▾ +⊙ SHARE ▾ ⏱ SCHEDULE ⚙ MORE ▾ ✅ Query completed.

```
1  select count(p.order_id) as No_of_orders , p.payment_type, extract(month FROM o.order_purchase_timestamp) as month
   from `customers.orders` o join `customers.payments` p  using(order_id) group by extract(month FROM o.
```

Press Alt+F1 for Accessibility Options.

Query results

⬇ SAVE RESULTS ▾   📊 EXPLORE DATA ▾   ⇕

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | CHART PREVIEW | EXECUTION GRAPH

Row	No_of_orders ▾	payment_type ▾	month ▾
1	1509	UPI	11
2	4378	credit_card	12
3	1723	UPI	2
4	5897	credit_card	11
5	572	voucher	4
6	7841	credit_card	7
7	2074	UPI	7
8	8350	credit_card	5
9	3778	credit_card	10
10	6103	credit_card	1

Results per page: 50 ▾   1 – 50 of 50 |< < > >|

PERSONAL HISTORY   PROJECT HISTORY   ↻ REFRESH ⌃

4:2
```
select c.customer_state, count(o.order_id)as
No_of_orders,round(sum(price),2) as
Total_cost_of_all_orders_made,
round(sum(price)/count(o.order_id),2) as Average_price from
`customers.order_items` as i join `customers.orders` as o on
i.order_id =o.order_id join `customers.customers` as c on
c.customer_id=o.customer_id group by c.customer_state order by
c.customer_state
```

Untitled ▶ RUN 💾 SAVE ▾ ➕ SHARE ▾ 🕐 SCHEDULE ⚙ MORE ▾ ✅ Query complete

```
1  select c.customer_state, count(o.order_id)as No_of_orders,round(sum(price),2) as Total_cost_of_all_orders_made,
      round(sum(price)/count(o.order_id),2) as Average_price from `customers.order_items` as i join `customers.orders`
```
Press Alt+F1 for Accessibility Optio

Query results

⬇ SAVE RESULTS ▾   📊 EXPLORE DATA ▾   ⇕

JOB INFORMATION | **RESULTS** | JSON | EXECUTION DETAILS | CHART PREVIEW | EXECUTION GRAPH

Row	customer_state ▾	No_of_orders ▾	Total_cost_of_all_ord ▾	Average_price ▾	
1	AC	92	15982.95	173.73	
2	AL	444	80314.81	180.89	
3	AM	165	22356.84	135.5	
4	AP	82	13474.3	164.32	
5	BA	3799	511349.99	134.6	
6	CE	1478	227254.71	153.76	
7	DF	2406	302603.94	125.77	
8	ES	2256	275037.31	121.91	
9	GO	2333	294591.95	126.27	
10	MA	824	119648.22	145.2	

Results per page: 50 ▾   1 – 27 of 27 |< < > >|

4:3 select c.customer_state, count(o.order_id)as No_of_orders,round(sum(freight_value),2) as Total_freight_value, round(sum(freight_value)/count(o.order_id),2) as Average_freight_value from `customers.order_items` as i join `customers.orders` as o on i.order_id =o.order_id join `customers.customers` as c on c.customer_id=o.customer_id group by c.customer_state order by c.customer_state

Untitled ▶ RUN 💾 SAVE ▾ ➕ SHARE ▾ 🕐 SCHEDULE ⚙ MORE ▾ ✅ Query completed

```
1  select c.customer_state, count(o.order_id)as No_of_orders,round(sum(freight_value),2) as Total_freight_value, round
      (sum(freight_value)/count(o.order_id),2) as Average_freight_value from `customers.order_items` as i join
      `customers orders` as o on i order_id =o order_id join `customers customers` as c on c customer_id=o customer_id
```
Press Alt+F1 for Accessibility Optio

Query results

⬇ SAVE RESULTS ▾   📊 EXPLORE DATA ▾   ⇕

JOB INFORMATION | **RESULTS** | JSON | EXECUTION DETAILS | CHART PREVIEW | EXECUTION GRAPH

Row	customer_state ▾	No_of_orders ▾	Total_freight_value ▾	Average_freight_value ▾	
1	AC	92	3686.75	40.07	
2	AL	444	15914.59	35.84	
3	AM	165	5478.89	33.21	
4	AP	82	2788.5	34.01	
5	BA	3799	100156.68	26.36	
6	CE	1478	48351.59	32.71	
7	DF	2406	50625.5	21.04	
8	ES	2256	49764.6	22.06	
9	GO	2333	53114.98	22.77	
10	MA	824	31523.77	38.26	

Results per page: 50 ▾   1 – 27 of 27 |< < > >|

4:2select c.customer_state, count(o.order_id)as No_of_orders,round(sum(price),2) as Total_cost_of_all_orders_made, round(sum(price)/count(o.order_id),2) as Average_price from `customers.order_items` as i join `customers.orders` as o on i.order_id =o.order_id join `customers.customers` as c on c.customer_id=o.customer_id group by c.customer_state order by c.customer_state

select count(o.order_id)as No_of_orders,round(avg(date_diff(order_delivered_customer_date ,order_purchase_timestamp,day)),2) as Average_delivery_time,customer_state as State from `customers.customers` c join `customers.orders` o on c.customer_id=o.customer_id group by customer_state order by c.customer_state

<table>
<thead>
<tr><th>Row</th><th>No_of_orders ▼</th><th>Average_delivery_tim</th><th>State ▼</th></tr>
</thead>
<tbody>
<tr><td>1</td><td>81</td><td>20.64</td><td>AC</td></tr>
<tr><td>2</td><td>413</td><td>24.04</td><td>AL</td></tr>
<tr><td>3</td><td>148</td><td>25.99</td><td>AM</td></tr>
<tr><td>4</td><td>68</td><td>26.73</td><td>AP</td></tr>
<tr><td>5</td><td>3380</td><td>18.87</td><td>BA</td></tr>
<tr><td>6</td><td>1336</td><td>20.82</td><td>CE</td></tr>
<tr><td>7</td><td>2140</td><td>12.51</td><td>DF</td></tr>
<tr><td>8</td><td>2033</td><td>15.33</td><td>ES</td></tr>
<tr><td>9</td><td>2020</td><td>15.15</td><td>GO</td></tr>
<tr><td>10</td><td>747</td><td>21.12</td><td>MA</td></tr>
</tbody>
</table>

4:3

select c.customer_state, count(o.order_id)as No_of_orders,round(sum(freight_value),2) as Total_freight_value, round(sum(freight_value)/count(o.order_id),2) as Average_freight_value from `customers.order_items` as i join `customers.orders` as o on i.order_id =o.order_id join `customers.customers` as c on c.customer_id=o.customer_id group by c.customer_state order by c.customer_state

▶ RUN   🔖 SAVE ▾   👥 SHARE ▾   🕐 SCHEDULE   ⚙ MORE ▾   ✅ Query completed

```
1  select c.customer_state, count(o.order_id)as No_of_orders,round(sum(freight_value),2) as Total_freight_value, round
   (sum(freight_value)/count(o.order_id),2) as Average_freight_value from `customers.order_items` as i join
   `customers.orders` as o on i.order_id =o.order_id join `customers.customers` as c on c.customer_id=o.customer_id
   group by c.customer_state order by c.customer_state
```

Press Alt+F1 for Accessibility Optio

Query results

⬇ SAVE RESULTS ▾   📊 EXPLORE DATA ▾   ⇕

JOB INFORMATION   **RESULTS**   JSON   EXECUTION DETAILS   CHART **PREVIEW**   EXECUTION GRAPH

Row	customer_state ▾	No_of_orders ▾	Total_freight_value ▾	Average_freight_value ▾
1	AC	92	3686.75	40.07
2	AL	444	15914.59	35.84
3	AM	165	5478.89	33.21
4	AP	82	2788.5	34.01
5	BA	3799	100156.68	26.36
6	CE	1478	48351.59	32.71
7	DF	2406	50625.5	21.04

Results per page: 50 ▾   1 – 27 of 27 |<   <   >   >|

6:1
```
select p.payment_type as Payment_type, count(o.order_id) as No_of_orders  from
`customers.payments` p join `customers.orders` o on o.order_id=p.order_id group by
p.payment_type order by  No_of_orders
```

▶ RUN   🔖 SAVE ▾   👥 SHARE ▾   🕐 SCHEDULE   ⚙ MORE ▾   ✅ Query completed.

```
1  select p.payment_type as Payment_type, count(o.order_id) as No_of_orders  from `customers.payments` p join
   `customers.orders` o on o.order_id=p.order_id group by p.payment_type order by  No_of_orders
```

Press Alt+F1 for Accessibility Options

Query results

⬇ SAVE RESULTS ▾   📊 EXPLORE DATA ▾   ⇕

JOB INFORMATION   **RESULTS**   JSON   EXECUTION DETAILS   CHART **PREVIEW**   EXECUTION GRAPH

Row	Payment_type ▾	No_of_orders ▾
1	not_defined	3
2	debit_card	1529
3	voucher	5775
4	UPI	19784
5	credit_card	76795

6:2
```
select p.payment_installments as No_of_Installments, count(o.order_id) as
No_of_orders  from `customers.payments` p join `customers.orders` o on
o.order_id=p.order_id group by p.payment_installments order by  No_of_Installments
```

```
1  select p.payment_installments as No_of_Installments, count(o.order_id) as No_of_orders  from `customers.payments`
   p join `customers.orders`  o on o.order_id=p.order_id group by p.payment_installments order by  No_of_Installments
```

Press Alt+F1 for Accessibility Options

Query results

💾 SAVE RESULTS ▾   📊 EXPLORE DATA ▾   ↕

| JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | CHART PREVIEW | EXECUTION GRAPH |

Row	No_of_Installments	No_of_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

Results per page: 50 ▾   1 – 24 of 24 |< < > >|