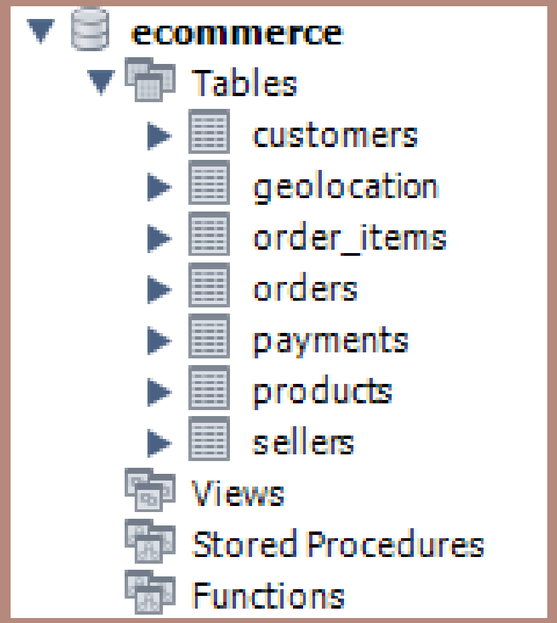
ECOMMERCE SALES ANALYSIS USING SQL

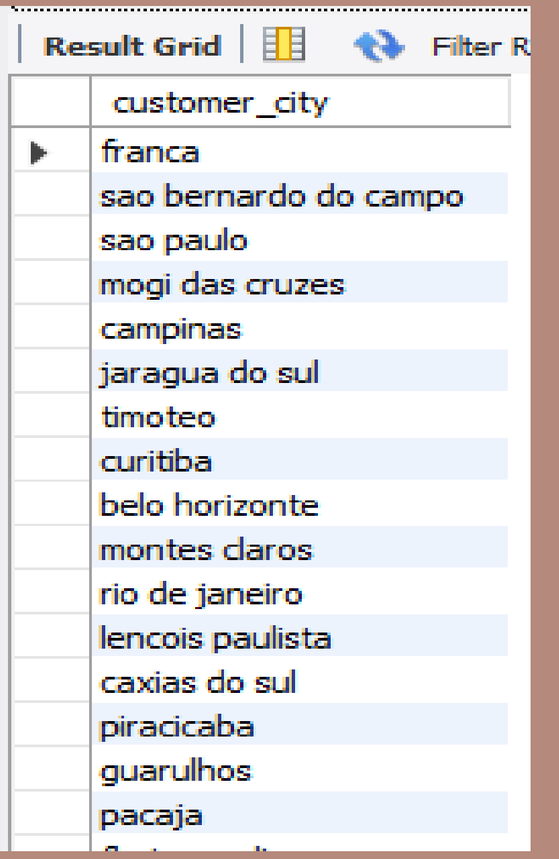
ECOMMERCE SALES DATASET




```
create database ecommerce;
```

LIST ALL UNIQUE CITIES WHERE CUSTOMERS ARE LOCATED

```sql
select distinct(customer_city) from customers;
```

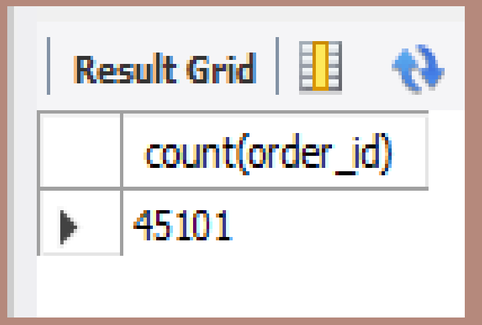


Result Grid | Filter R

customer_city
franca
sao bernardo do campo
sao paulo
mogi das cruzes
campinas
jaragua do sul
timoteo
curitiba
belo horizonte
montes claros
rio de janeiro
lencois paulista
caxias do sul
piracicaba
guarulhos
pacaja

COUNT THE NUMBER OF ORDERS PLACED IN 2017

```
select count(order_id)
from orders
where year(order_purchase_timestamp)=2017;
```



FIND THE TOTAL SALES PER CATEGORY

```
select products.product_category as category,round(sum(payments.payment_value),2) as sales
from products join order_items
on products.product_id = order_items.product_id
join payments
on payments.order_id=order_items.order_id
group by category;
```

Result Grid | Filter Rows:

category	sales
perfumery	506738.66
Furniture Decoration	1430176.39
telephony	486882.05
bed table bath	1712553.67
automotive	852294.33
computer accessories	1585330.45
housewares	1094758.13
babies	539845.66
toys	619037.69
Furniture office	646826.49
Cool Stuff	779698
HEALTH BEAUTY	1657373.12
pet Shop	311268.97
General Interest Books	64694.76
stationary store	317440.07
Watches present	1429216.68
PCs	279121.55
climatization	91170.66
Construction Tools C...	241475.63
sport leisure	1392127.56

CALCULATE THE PERCENTAGE OF ORDERS THAT WERE PAID IN INSTALLMENTS

```
select ((sum(case when payment_installments>=1 then 1
                  else 0 end))/count(*))*100 as orders_in_installments
 from payments;
```

Result Grid | Filter

	orders_in_installments
▶	99.9981

COUNT THE NUMBER OF CUSTOMERS FROM EACH STATE

```sql
select customer_state, count(customer_id)
from customers
group by customer_state;
```

Result Grid | Filter Rows:

customer_state	count(customer_id)
SP	41746
SC	3637
MG	11635
PR	5045
RJ	12852
RS	5466
PA	975
GO	2020
ES	2033
BA	3380
MA	747
MS	715
CE	1336
DF	2140
RN	485
PE	1652

CALCULATE THE NUMBER OF ORDERS PER MONTH IN 2018

```
select monthname(order_purchase_timestamp) as months,count(order_id) as order_count
from orders
where year(order_purchase_timestamp)=2018
group by months;
```

Result Grid | Filter Row

months	order_count
July	6292
August	6512
February	6728
June	6167
March	7211
January	7269
May	6873
April	6939
September	16
October	4

FIND THE AVERAGE NUMBER OF PRODUCTS PER ORDER, GROUPED BY CUSTOMER CITY

```sql
with count_per_order as (select orders.order_id, orders.customer_id,
count(order_items.order_id) as oc
from orders join order_items
on orders.order_id=order_items.order_id
group by orders.order_id, orders.customer_id)

select customers.customer_city,round(avg(count_per_order.oc),2) average_orders
from customers join count_per_order
on customers.customer_id=count_per_order.customer_id
group by customers.customer_city;
```

Result Grid | Filter Rows:

customer_city	average_orders
sao paulo	1.16
sao jose dos campos	1.14
porto alegre	1.17
indaial	1.12
treze tilias	1.27
rio de janeiro	1.15
mario campos	1.33
guariba	1.00
cuiaba	1.20
franca	1.25
tocos	1.00
januaria	1.18
campinas	1.16
embu-guacu	1.24
cerquilho	1.17
belo horizonte	1.14
taubate	1.13
arcos	1.12

CALCULATE THE PERCENTAGE OF TOTAL REVENUE CONTRIBUTED BY EACH PRODUCT CATEGORY

```
select products.product_category as category,
round(((sum(payments.payment_value)/(select sum(payment_value) from payments))*100),2) as sales
from products join order_items
on products.product_id = order_items.product_id
join payments
on payments.order_id=order_items.order_id
group by category
order by sales desc;
```

Result Grid | Filter Rows:

category	sales
bed table bath	10.7
HEALTH BEAUTY	10.35
computer accessories	9.9
Furniture Decoration	8.93
Watches present	8.93
sport leisure	8.7
housewares	6.84
automotive	5.32
Garden tools	5.24
Cool Stuff	4.87
Furniture office	4.04
toys	3.87
babies	3.37
perfumery	3.17
telephony	3.04
stationary store	1.98
pet Shop	1.94
PCs	1.74

IDENTIFY THE CORRELATION BETWEEN PRODUCT PRICE AND THE NUMBER OF TIMES A PRODUCT HAS BEEN PURCHASED

```
select products.product_category, count(order_items.product_id) as order_count,
round(avg(order_items.price),2) as price
from products join order_items
on products.product_id=order_items.product_id
group by products.product_category;
```

Result Grid | Filter Rows:

product_category	order_count	price
HEALTH BEAUTY	9670	130.16
sport leisure	8641	114.34
Cool Stuff	3796	167.36
computer accessories	7827	116.51
Watches present	5991	201.14
housewares	6964	90.79
electronics	2767	57.91
NULL	1603	112
toys	4117	117.55
bed table bath	11115	93.3
Games consoles	1137	138.49
automotive	4235	139.96
Furniture Decoration	8334	87.56
home appliances	771	103.98
telephony	4545	71.21
Construction Tools C...	929	155.73
perfumery	3419	116.74
stationary store	2517	91.75

CALCULATE THE TOTAL REVENUE GENERATED BY EACH SELLER, AND RANK THEM BY REVENUE

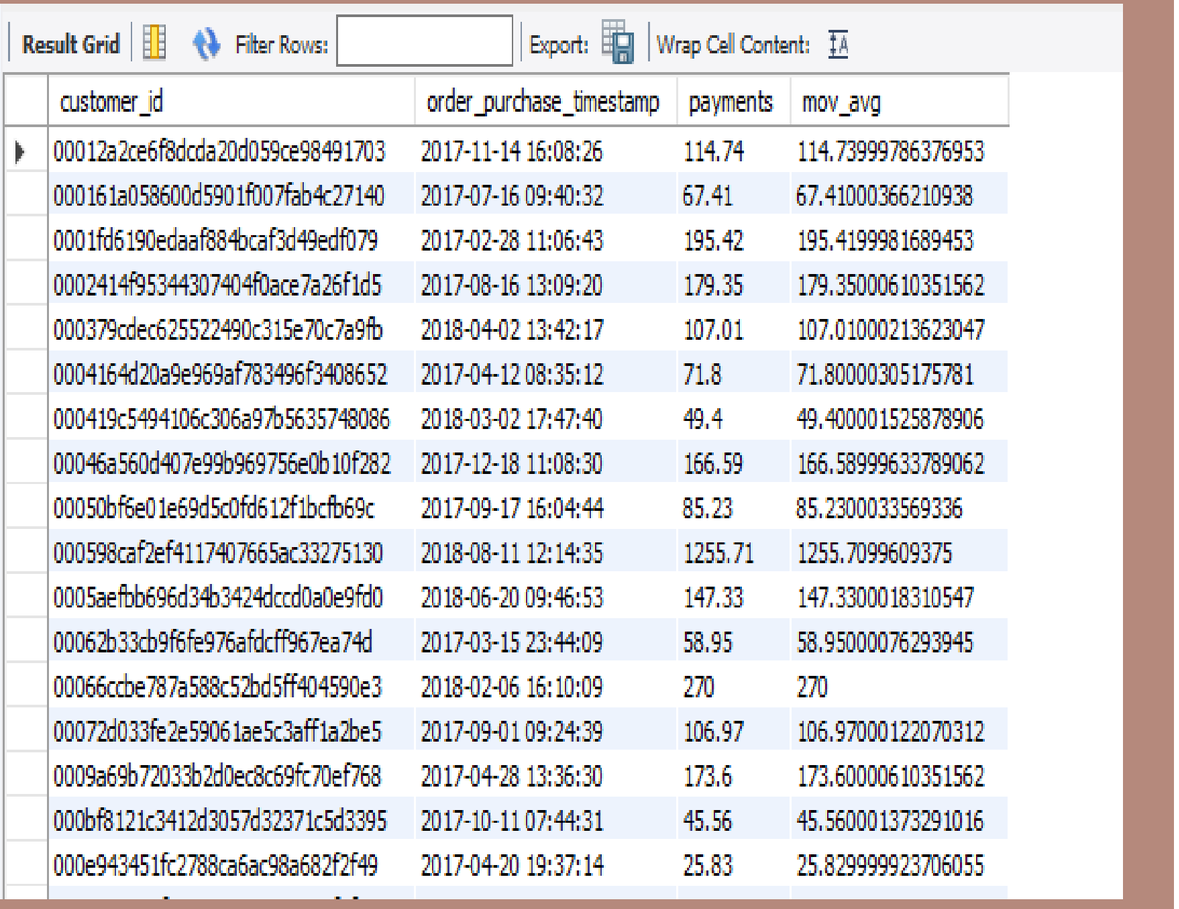
```sql
select *,dense_rank() over(order by revenue desc) as rn
from
(select order_items.seller_id,round(sum(payments.payment_value),2) as revenue
from order_items join payments
on order_items.order_id=payments.order_id
group by order_items.seller_id) as a;
```

Result Grid | Filter Rows: | Export:

seller_id	revenue	rn
7c67e1448b00f6e969d365cea6b010ab	507166.91	1
1025f0e2d44d7041d6cf58b6550e0bfa	308222.04	2
4a3ca9315b744ce9f8e9374361493884	301245.27	3
1f50f920176fa81dab994f9023523100	290253.42	4
53243585a1d6dc2643021fd1853d8905	284903.08	5
da8622b14eb17ae2831f4ac5b9dab84a	272219.32	6
4869f7a5dfa277a7dca6462dcf3b52b2	264166.12	7
955fee9216a65b617aa5c0531780ce60	236322.3	8
fa1c13f2614d7b5c4749cbc52fecda94	206513.23	9
7e93a43ef30c4f03f38b393420bc753a	185134.21	10
6560211a19b47992c3666cc44a7e94c0	179657.75	11
7a67c85e85bb2ce8582c35f2203ad736	169030.8	12
25c5c91f63607446a97b143d2d535d31	160534.74	13
a1043bafd471dff536d0c462352beb48	154356.91	14
46dc3b2cc0980fb8ec44634e21d2718e	148864.34	15
b37c4c02bda3161a7546a4e6d222d5b2	145319.04	16
620c87c171fb2a6dd6e8bb4dec959fc6	145267.95	17
cc419e0650a3c5ba77189a1882b7556a	141309.58	18

CALCULATE THE MOVING AVERAGE OF ORDER VALUES FOR EACH CUSTOMER OVER THEIR ORDER HISTORY

```
select customer_id,order_purchase_timestamp,payments,
avg(payments) over(partition by customer_id order by order_purchase_timestamp
rows between 2 preceding and current row) as mov_avg
from
(select orders.customer_id,orders.order_purchase_timestamp,
payments.payment_value as payments
from payments join orders
on payments.order_id= orders.order_id) as a;
```


Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customer_id	order_purchase_timestamp	payments	mov_avg
00012a2ce6f8dcda20d059ce98491703	2017-11-14 16:08:26	114.74	114.73999786376953
000161a058600d5901f007fab4c27140	2017-07-16 09:40:32	67.41	67.41000366210938
0001fd6190edaaf884bcaf3d49edf079	2017-02-28 11:06:43	195.42	195.4199981689453
0002414f95344307404f0ace7a26f1d5	2017-08-16 13:09:20	179.35	179.35000610351562
000379cdec625522490c315e70c7a9fb	2018-04-02 13:42:17	107.01	107.01000213623047
0004164d20a9e969af783496f3408652	2017-04-12 08:35:12	71.8	71.80000305175781
000419c5494106c306a97b5635748086	2018-03-02 17:47:40	49.4	49.400001525878906
00046a560d407e99b969756e0b10f282	2017-12-18 11:08:30	166.59	166.58999633789062
00050bf6e01e69d5c0fd612f1bcfb69c	2017-09-17 16:04:44	85.23	85.2300033569336
000598caf2ef4117407665ac33275130	2018-08-11 12:14:35	1255.71	1255.7099609375
0005aefbb696d34b3424dccd0a0e9fd0	2018-06-20 09:46:53	147.33	147.3300018310547
00062b33cb9f6fe976afdcff967ea74d	2017-03-15 23:44:09	58.95	58.95000076293945
00066ccbe787a588c52bd5ff404590e3	2018-02-06 16:10:09	270	270
00072d033fe2e59061ae5c3aff1a2be5	2017-09-01 09:24:39	106.97	106.97000122070312
0009a69b72033b2d0ec8c69fc70ef768	2017-04-28 13:36:30	173.6	173.60000610351562
000bf8121c3412d3057d32371c5d3395	2017-10-11 07:44:31	45.56	45.560001373291016
000e943451fc2788ca6ac98a682f2f49	2017-04-20 19:37:14	25.83	25.829999923706055

CALCULATE THE CUMULATIVE SALES PER MONTH FOR EACH YEAR

```
select years,months,payment,sum(payment) over(order by years,months) as cumulative_sales
from
(select year(orders.order_purchase_timestamp) as years,
month(orders.order_purchase_timestamp) as months,
round(sum(payments.payment_value),2) as payment
from orders join payments
on orders.order_id=payments.order_id
group by years,months
order by years,months) as a;
```

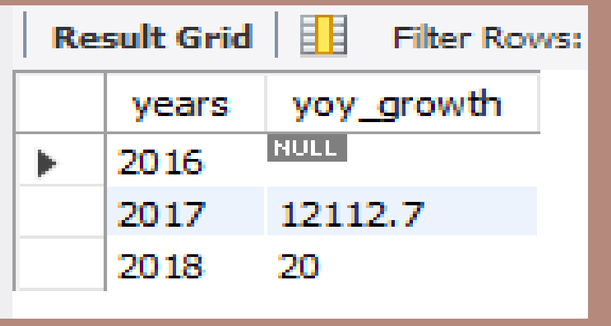
Result Grid | Filter Rows: | Export

years	months	payment	cumulative_sales
2016	9	252.24	252.24
2016	10	59090.48	59342.72
2016	12	19.62	59362.340000000004
2017	1	138488.04	197850.38
2017	2	291908.01	489758.39
2017	3	449863.6	939621.99
2017	4	417788.03	1357410.02
2017	5	592918.82	1950328.8399999999
2017	6	511276.38	2461605.2199999997
2017	7	592382.92	3053988.1399999997
2017	8	674396.32	3728384.4599999995
2017	9	727762.45	4456146.909999999
2017	10	779677.88	5235824.789999999
2017	11	1194882.8	6430707.589999999
2017	12	878401.48	7309109.069999998
2018	1	1115004.18	8424113.249999998
2018	2	992463.34	9416576.589999998
2018	3	1159652.12	10576228.709999997

CALCULATE THE YEAR-OVER-YEAR GROWTH RATE OF TOTAL SALES

```sql
with a as (select year(orders.order_purchase_timestamp) as years,
round(sum(payments.payment_value),2) as payment
from orders join payments
on orders.order_id=payments.order_id
group by years
order by years)

select years,
round((((payment-lag(payment,1) over(order by years))/lag(payment,1) over(order by years))*100),2) as yoy_growth
from a;
```

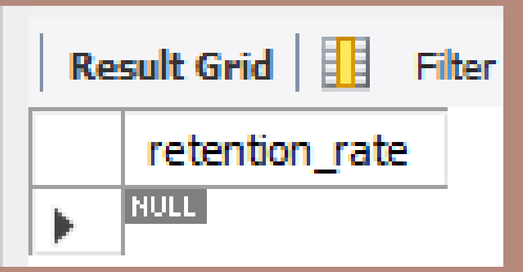


CALCULATE THE RETENTION RATE OF CUSTOMERS, DEFINED AS THE PERCENTAGE OF CUSTOMERS WHO MAKE ANOTHER PURCHASE WITHIN 6 MONTHS OF THEIR FIRST PURCHASE

```sql
with a as (select customers.customer_id,
min(orders.order_purchase_timestamp) as first_order
from customers join orders
on customers.customer_id=orders.customer_id
group by customers.customer_id),

b as (select a.customer_id,count(distinct orders.order_purchase_timestamp) as next_order
from a join orders
on a.customer_id=orders.customer_id
and orders.order_purchase_timestamp> first_order
and orders.order_purchase_timestamp< date_add(first_order, interval 6 month)
group by a.customer_id)

select 100*(count(distinct a.customer_id)/count(distinct b.customer_id)) as retention_rate
from a left join b
on a.customer_id=b.customer_id;
```



IDENTIFY THE TOP 3 CUSTOMERS WHO SPENT THE MOST MONEY IN EACH YEAR

```
with a as(select year(orders.order_purchase_timestamp) as years,
orders.customer_id,
sum(payments.payment_value) as payment,
dense_rank() over(partition by year(orders.order_purchase_timestamp)
order by sum(payments.payment_value) desc) as rn
from orders join payments
on orders.order_id=payments.order_id
group by years,orders.customer_id)

select years,customer_id,payment,rn
from a
where rn<=3;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

years	customer_id	payment	rn
2016	a9dc96b027d1252bbac0a9b72d837fc6	1423.550048828125	1
2016	1d34ed25963d5aae4cf3d7f3a4cda173	1400.739990234375	2
2016	4a06381959b6670756de02e07b83815f	1227.780029296875	3
2017	1617b1357756262bfa56ab541c47bc16	13664.080078125	1
2017	c6e2731c5b391845f6800c97401a43a9	6929.31005859375	2
2017	3fd6777bbce08a352fddd04e4a7cc8f6	6726.66015625	3
2018	ec5b2ba62e574342386871631fafd3fc	7274.879882812 5	1
2018	f48d464a0baaea338cb25f816991ab1f	6922.2099609375	2
2018	e0a2412720e9ea4f26c1ac985f6a7358	4809.43994140625	3

THANK YOU!!!