







6.1. Month over Month count of orders for different payment types

```
with order_updated as(
  select *,
    TO_CHAR(
      DATE_TRUNC('month', o.order_purchase_timestamp),
      'Month'
    ) as opt_month,
    extract(
      month
      from o.order_purchase_timestamp
    ) as opt_month_id
  from orders o
    join payments p using(order_id)
    where p.payment_type not like 'not_defined'
  order by o.order_purchase_timestamp
),
grouped_payment_details as(
  select payment_type,
    opt_month as month,
    count(distinct order_id) as order_quantity,
    sum(payment_value) as order_price,
    opt_month_id as rn
  from order_updated
  group by payment_type, opt_month, opt_month_id
  order by payment_type, opt_month_id
)
select gsd1.payment_type, gsd1.month, gsd1.order_quantity, gsd1.order_price,
  case
    when gsd2.order_quantity is not null
    and gsd2.order_quantity not in (0) then (
      (gsd1.order_quantity - gsd2.order_quantity) * 100 /
      gsd2.order_quantity
    )
    else 0
  end as quantity_mom,
  case
    when gsd2.order_price is not null
    and gsd2.order_price not in (0) then round(
      (gsd1.order_price - gsd2.order_price) * 100 / gsd2.order_price,
      2
    )
    else 0
  end as price_mom
from grouped_payment_details gsd1
  left join grouped_payment_details gsd2 on gsd1.payment_type =
    gsd2.payment_type
  and gsd1.rn = gsd2.rn + 1
order by gsd1.payment_type, gsd1.rn;
```

payment_type	month	order_quantity	order_price	quantity_mom	price_mom
 Filter...	 Filter...	 Filter...	 Filter...	 Filter...	 Filter...
credit_card	January	6093	978496.06	0	0
credit_card	February	6582	1005556.56	8	2.77
credit_card	March	7682	1288259.04	16	28.11
credit_card	April	7276	1256393.72	-5	-2.47
credit_card	May	8308	1372603.66	14	9.25
credit_card	June	7248	1199776.83	-12	-12.59
credit_card	July	7810	1258662.96	7	4.91
credit_card	August	8235	1319244.58	5	4.81
credit_card	September	3277	574736.39	-60	-56.43
credit_card	October	3763	662618.51	14	15.29
credit_card	November	5867	942856.67	55	42.29
credit_card	December	4364	682879.21	-25	-27.57
debit_card	January	118	12287.08	0	0
debit_card	February	82	8979.85	-30	-26.92
debit_card	March	109	11967.91	32	33.28
debit_card	April	124	13572.53	13	13.41
debit_card	May	81	12734.15	-34	-6.18
debit_card	June	208	38096.79	156	199.17

And so on.

Here, we have grouped records by Payment type, ordered by Month to find Month on Month Change of total quantity of product sold (**quantity_mom**) and total cost of product sold (**price_mom**) both are in percentage.

6.2. Count of orders based on the no. of payment installments:

```
select payment_installments,  
       count(order_id) as total_orders,  
       count(distinct order_id) as distinct_orders  
from payments  
group by payment_installments  
order by count(order_id) desc;
```

payment_install...	total_orders	distinct_orders
abc Filter...	abc Filter...	abc Filter...
1	52546	49060
2	12413	12389
3	10461	10443
4	7098	7088
10	5328	5315
5	5239	5234
8	4268	4253
6	3920	3916
7	1626	1623
9	644	644
12	133	133
15	74	74
18	27	27
11	23	23
24	18	18
20	17	17
13	16	16
14	15	15

And so on.