Advanced SQL Queries

Author: - Azarudhin Usman Reference: Youtube- @AnalystAdithya

Introduction:

This document covers SQL topics like Aggregate functions, Joins, Windows Functions and joins.

```
-- Display the palindrome which has least character
```

```
with a as (SELECT word
FROM (
    VALUES
          ('Madam'),
          ('Bob'),
          ('Wow'),
          ('deed'),
                ('rotator'),
                 ('Tech')
) AS word_table(word))
select word, reverse(word),
case when lower(word) = lower(reverse(word)) then 'Palindrome' else 'No'
end as Palin
from a
where length(word) = (select min(length(word)) from a )
```

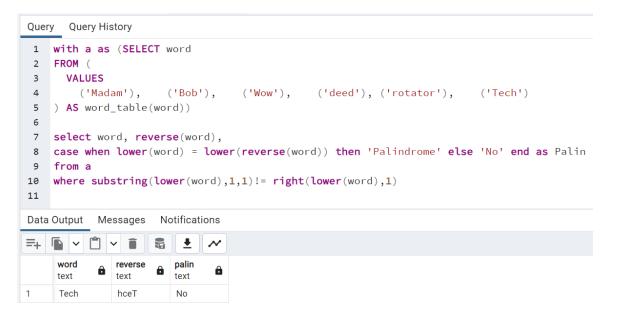
Output:

```
1 with a as (SELECT word
 2 FROM (
     VALUES
 3
                                  ('Wow'), ('deed'), ('rotator'), ('Tech')
        ('Madam'),
                    ('Bob'),
5 ) AS word_table(word))
 7 select word, reverse(word),
8 case when lower(word) = lower(reverse(word)) then 'Palindrome' else 'No' end as Palin,
9 count(word) from a
10 group by word
having length(word) = (select min(length(word)) from a )
Data Output Messages Notifications
=+ | • | • | • | • |
                         <u>*</u>
     word
              reverse
                        palin
                                  count
                               â
                                         â
     text
              text
                                         1
     Rob
              hoB
                        Palindrome
2
                        Palindrome
     Wow
              woW
                                         1
```

 $\ensuremath{\mathsf{--}}$ Display the word which are not palindrome without creating more than 1 cte

```
--Query
```

Output:

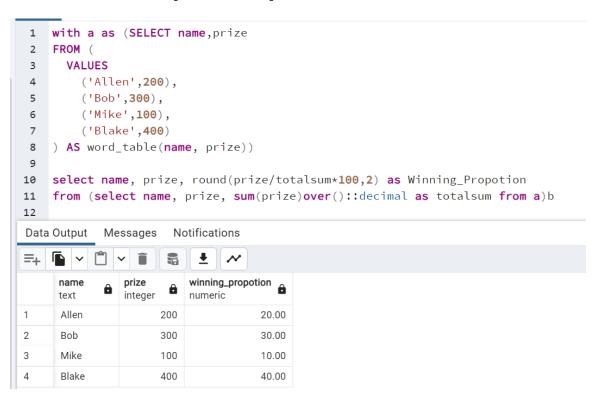


-- Find the relative propotion of winning money

```
with a as (SELECT name, prize
FROM (
   VALUES
      ('Allen', 200),
      ('Bob', 300),
      ('Mike', 100),
```

```
('Blake',400)
) AS word table(name, prize))
```

select name, prize, round(prize/totalsum*100,2) as Winning_Propotion
from (select name, prize, sum(prize)over()::decimal as totalsum from a)b



-- Leet code problem

Find the latest price change of a product on or before a particular date. (example - 2020-08-23)

Sample Output:

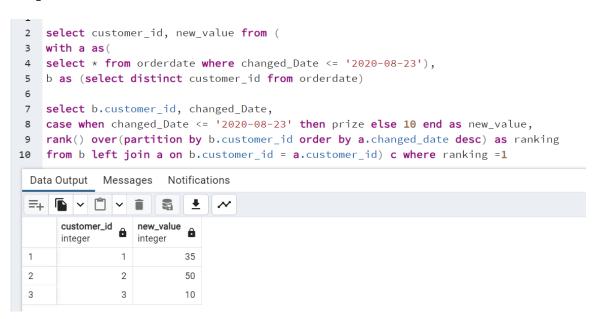
	customer_id integer	a	new_value integer	•
1		1	35	5
2	:	2	50)
3	:	3	10)

If no price change, then give *10 for that product

create table orderdate (customer_id integer, prize integer, changed_date
date)

```
Insert into orderdate
 VALUES
   (1,20,'2020-08-21'),
   (2,50,'2020-08-21'),
   (1,30,'2020-08-22'),
   (1,35,'2020-08-23'),
     (2,65,'2020-08-24'),
     (3,20,'2020-08-24')
-- Query
select customer id, new value from (
with a as (
select * from orderdate where changed Date <= '2020-08-23'),</pre>
b as (select distinct customer id from orderdate)
select b.customer id, changed Date, case when changed Date <= '2020-08-23'
then prize else 10 end as new value,
rank() over(partition by b.customer id order by a.changed date desc) as
ranking
from b left join a on b.customer id = a.customer id) c where ranking =1
```

Output:



-- Find the total non-mac revenue made from customers who ordered the MAC Sample Output:

customer_id integer	nonrevenue bigint
1	140000
3	50000
4	0

Output:

```
1 with a as (select * from (
2 values
       (1, 'iphone', 50000),
3
4
        (1,'pc',90000),
5
        (1,'mac',80000),
 6
        (2, 'iphone', 50000),
7
        (3, 'iphone', 50000),
8
        (3, 'mac', 80000), (2, 'iphone', 50000), (4, 'mac', 80000)
9
   ) as apple_store(customer_id, products, rate))
10
11 select customer_id, sum(case when products = 'mac' then 0 else rate end) as nonrevenue
12 from a where customer_id in (select distinct customer_id from a where products='mac')
13 group by customer_id
14 order by customer_id
 Data Output Messages Notifications
 =+ | • | • | • |
                      customer_id
                  nonrevenue
      integer
                  bigint
 1
                1
                       140000
 2
                3
                        50000
                4
 3
                           0
```

-- Display the customer who purchased only Mac

Sample Output:

customer_id integer	products text	rate integer
4	mac	80000

-- Query

Output:

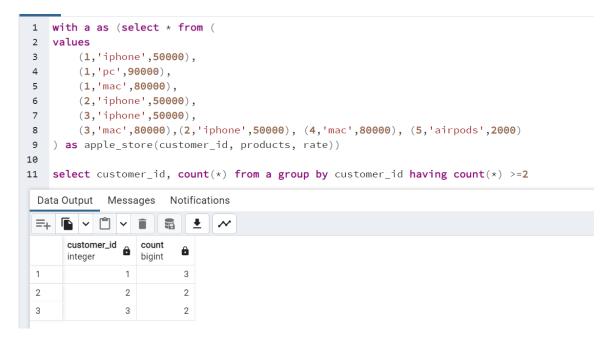
```
1 with a as (select * from (
2 values
        (1,'iphone',50000),
3
4
        (1,'pc',90000),
5
        (1, 'mac', 80000),
6
        (2, 'iphone', 50000),
        (3,'iphone',50000),
       (3,'mac',80000),(2,'iphone',50000), (4,'mac',80000), (5,'airpods',2000)
8
9 ) as apple_store(customer_id, products, rate))
10
11 select * from a where customer_id not in (select customer_id from a where products !='mac')
 Data Output Messages Notifications
 =+ □ ∨ □ ∨ i s
      customer_id products
                           rate
      integer
                           integer
                  text
                               80000
               4 mac
```

-- Display the customers who purchased more than 2 orders

	customer_id integer	â	count bigint	â
1		1		3
2		2		2
3		3		2

select customer_id, count(*) from a group by customer_id having count(*) >=2

Output:

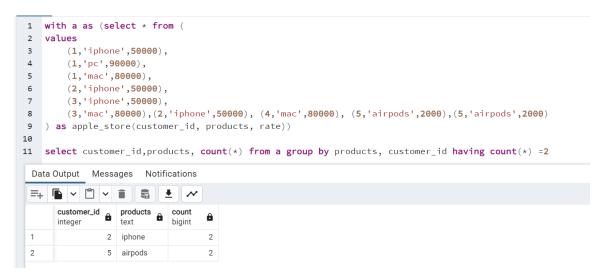


-- Display the customer who order same product more than once.

customer_id integer	â	count bigint	â
	2		2
	5		2

select customer_id, count(*) from a group by products, customer_id having
count(*) =2

Output:



-- Find the customer who purchased in the apple store which is $\geq 25\%$ of that revenue.

	customer_id integer	â	revernuepercent numeric
1		3	26.97
2		1	45.64

-- Ouerv

```
with a as (select * from (
values
      (1, 'iphone', 50000),
      (1, 'pc', 90000),
      (1, 'mac', 80000),
      (2, 'iphone', 50000),
      (3, 'iphone', 50000),
      (3, 'mac', 80000), (2, 'iphone', 50000), (4, 'mac', 80000),
(5, 'airpods', 2000), (5, 'airpods', 2000)
) as apple store(customer id, products, rate)),
b as (
select distinct customer id, (sum(rate) over(partition by
customer id)::decimal) as amountpercustomer,
     sum(rate) over()::decimal as totalrevenue from a
group by customer id, rate) --select * from b
select b.customer id, round((b.amountpercustomer/b.totalrevenue)*100,2)
from b
where round ((b.amountpercustomer/b.totalrevenue)*100,2) > 25;
```

Output:



-- Find the customer who took longest time between the first order and second order.



```
with c as (
with b as (
with a as (select * from (
values (1,101,'2023-12-11'),(2,102,'2023-12-12'),(3,103,'2023-12-13'),
      (4,101,'2023-12-19'), (5,102,'2023-12-20'), (6,103,'2023-12-21'),
     (7,101,'2023-12-12'),(8,102,'2023-12-18')
) as datedetails(id, customerid, orderdate))
select *, rank() over(partition by customerid order by orderdate) as ranking
from a)
select customerid, orderdate, lag(orderdate) over (partition by customerid)
as lag Details,
(orderdate::date-lag(orderdate) over(partition by customerid)::date) as
datedifference
from b where ranking <=2 )
select customerid, datedifference from c
where datedifference is not null
order by datedifference desc limit 1
```

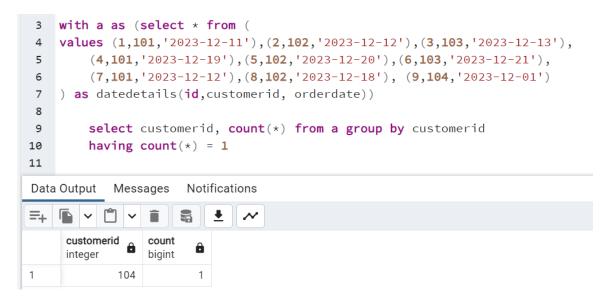
Output:

```
1 with c as (
2 with b as (
3 with a as (select * from (
 4 values (1,101,'2023-12-11'),(2,102,'2023-12-12'),(3,103,'2023-12-13'),
        (4,101,'2023-12-19'),(5,102,'2023-12-20'),(6,103,'2023-12-21'),
        (7,101,'2023-12-12'),(8,102,'2023-12-18')
 7 ) as datedetails(id,customerid, orderdate))
8 select *, rank() over(partition by customerid order by orderdate) as ranking from a)
9 select customerid, orderdate, lag(orderdate)over(partition by customerid) as lag_Details,
10
   (orderdate::date-lag(orderdate)over(partition by customerid)::date) as datedifference
11 from b where ranking <=2 )
12
13 select customerid, datedifference from c
14 where datedifference is not null
15 order by datedifference desc limit 1
16
Data Output Messages Notifications
타 [ V [ V ] V ] 등 [ 등 [ 보 ] 사
                datedifference
     integer
                integer
            103
                           8
```

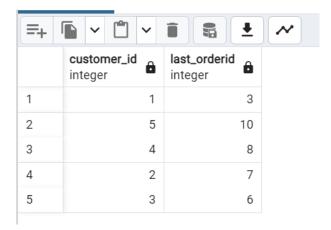
-- Display the customer who ordered only once.



Output:



-- Find the last product purchased by each customer without using rank function



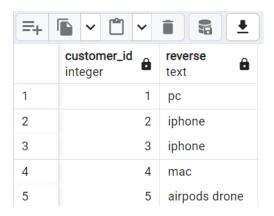
-- query

Output:

```
with a as (select * from (
2 values
3
       (1,1,'iphone',50000),
        (2,1,'pc',90000),
4
       (3,1,'mac',80000),
5
       (4,2,'iphone',50000),
        (5,3,'iphone',50000),
7
       (6,3,'mac',80000),(7,2,'iphone',50000), (8,4,'mac',80000), (9,5,'airpods',2000),(10,5,'airpods',2000)
9 ) as apple_store(orderid, customer_id, products, rate)),
10 b as (
select customer_id, max(orderid) as last_orderid from a
12 group by customer_id)
13 select * from b;
Data Output Messages Notifications
=+ 6 ~ 6 ~ 6
                        + ~
    customer_id
                last_orderid
                integer
2
                        10
              5
3
              4
                         8
4
```

-- Display along with the last ordered products

Sample Output:



-- Query

```
with a as (select * from (
values
     (1,1,'iphone',50000),
     (2,1,'pc',90000),
     (3,1,'mac',80000),
     (4,2,'iphone',50000),
     (5,3,'iphone',50000),
     (6,3,'mac',80000),(7,2,'iphone',50000), (8,4,'mac',80000),
(9,5,'airpods',2000)
) as apple store(orderid, customer id, products, rate)),
b as (
select customer id, string_agg(products, ',') as concatenated_order from
group by customer id)
select customer id, case
when length(reverse(split part(reverse(concatenated order),',',1)))>0
then reverse(split part(reverse(concatenated order),',',1))
else concatenated order end as lastorder
from b
order by customer id
```

Output:

```
1 with a as (select * from (
2 values
3
       (1,1,'iphone',50000),
4
       (2,1,'pc',90000),
       (3,1,'mac',80000),
5
       (4,2,'iphone',50000),
6
       (5,3,'iphone',50000),
       (6,3,'mac',80000),(7,2,'iphone',50000), (8,4,'mac',80000), (9,5,'airpods',2000),(10,5,'airpods drone',2000
8
9 ) as apple_store(orderid, customer_id, products, rate)),
b as (select customer_id, string_agg(products, ',') as concatenated_order from a
11 group by customer_id)
12 select customer_id,reverse(split_part(reverse(concatenated_order),',',1)) from b order by customer_id
                                                                                                          2
Data Output Messages Notifications
customer_id integer reverse text
    integer
             1 pc
2
             2 iphone
3
             3 iphone
4
             4 mac
5
             5 airpods drone
```
