FAANG Questions - Part I

Advanced SQL Queries

Author: - Azarudhin Usman Reference: LeetCode

Introduction:

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

-- Intresting trick on Numbers

```
CREATE TABLE IF NOT EXISTS public.numbers
(
   id integer
)
INSERT INTO public.numbers(
   id)
   VALUES (1),(2),(3),(6),(7),(9);
```

Display the output in the below format:

id integer	â	bid integer	â
	1		3
	6		7
	9		9

```
select id,bid from ( with a as (
select * from numbers
where id-1 not in (select distinct id from numbers)),
b as ( select * from numbers
where id+1 not in (select distinct id from numbers))
select a.id ,b.id as bid, row_number() over(partition by a.id order by a.id)
as r from a,b
where a.id<=b.id) x where r =1</pre>
```

```
1 select id, bid from ( with a as (
 2 select * from numbers
 3 where id-1 not in (select distinct id from numbers)),
 4 b as ( select * from numbers
 5 where id+1 not in (select distinct id from numbers))
 7 select a.id ,b.id as bid, row_number()over(partition by a.id order by a.id) as r from a,b
 8 where a.id<=b.id) x where r =1</pre>
Data Output Messages Notifications
                         ₹ ~
=+ | • | • | • |
                    5
              bid
           â
              integer
           1
                     3
                     7
2
           6
           9
                     9
```

-- Find the employees who earn more than their managers

-- Query

```
with a as (
select a.employee_id, a.salary "empsalary", b.employee_id as "managerid",
b.salary as bsalary
from employees a join employees b
on a.employee_id = b.manager_id)
select * from a where empsalary > bsalary;
```

Output:

```
1 with a as (
 2 select a.employee_id, a.salary "empsalary", b.employee_id as "managerid", b.salary as bsalary
 3 from employees a join employees b
 4 on a.employee_id = b.manager_id)
 5 select * from a where empsalary > bsalary ;
Data Output Messages Notifications
₹ 1 2 1 3 3 4
     employee_id empsalary | PK] integer numeric (8,
                                          bsalary
                               managerid
                 numeric (8,2)
                                           numeric (8,2)
                              integer
                                                 9000.00
                      17000.00
                                      103
             102
             103
                       9000.00
                                      104
                                                 6000.00
```

-- Find the last person to enter in to the lift. Lift maximum capacity is $1000 \ \text{kgs}$

```
create table lift (name character varying(20), weight integer, turn integer)
```

```
insert into lift
values('John',200,1),('Adam',200,2),('Smith',232,3),('Mike',300,4),('B
lake',250,5),('Alex',120,6)
```

```
with a as (
          select name, weight, turn, sum(weight) over(order by turn) as Capacity
          from lift)
select name, weight, turn, Capacity from a where Capacity<=1000
order by capacity desc</pre>
```

1 2 3	<pre>with a as (select name,we from lift)</pre>	eight,turn	n, sum (we	ight) over	(order by turn) as Capacit				
4	<pre>select name, weight, turn, Capacity from a where Capacity<=1000 order by capacity desc</pre>								
5									
6	, , , , , , , , , , , , , , , , , , , ,								
Data	Output Messages 1	Notifications							
=+		<u>*</u> *							
=+	name character varying (20)	weight integer	turn integer	capacity bigint					
	name	weight	turn	bigint					
1	name character varying (20)	weight integer	turn integer	bigint 932					
1 2 3	name character varying (20)	weight integer 300	turn integer	932 632					

-- Customer_Orders, Products

create table Customer_orders (customer_id integer, products character varying (20)) $\,$

create table products (product_name character varying(20))

```
insert into Customer_orders values
(3,'Boost'),(3,'Boost'),
(2,'Horlicks'),(2,'Boost'),(2,'Bournvita'),
(1,'Horlicks'),(1,'Boost')
```

insert into products values ('Bournvita'),('Boost'),('Horlicks')

-- Find the customer who has purchased all the products

```
select customer_id, count(distinct products)
from customer_orders
group by customer_id
having count(distinct products) = (select count(distinct product_name) from
products)
order by customer_id asc
```

```
select customer_id, count(distinct products)
from customer_orders
group by customer_id
having count(distinct products) = (select count(distinct product_name) from products)
order by customer_id asc

Data Output Messages Notifications

customer_id count integer count i
```

-- Find the customer who haven't purchased all the products

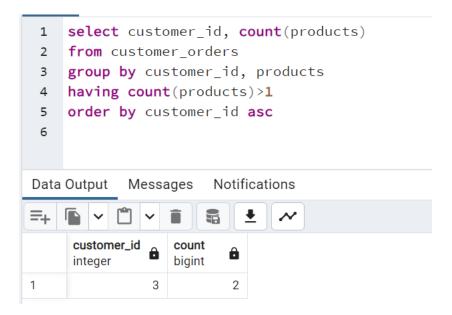
```
select customer_id, count(distinct products)
from customer_orders
group by customer_id
having count(distinct products) != (select count(distinct product_name)
from products)
order by customer id asc
```

Output:



-- Find the customer who have purchased same product more than once

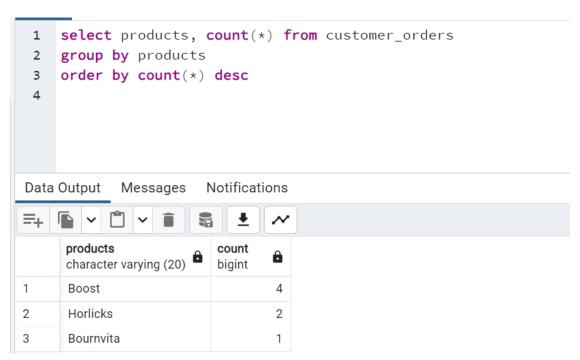
```
select customer_id, count(products)
from customer_orders
group by customer_id, products
having count(products)>1
order by customer id asc
```



-- Display the highest selling health drink

select products, count(*) from customer_orders
group by products
order by count(*) desc

Output:



Find the percentage change in revenue for companies

create table companies (year date, company_name character varying(10), sales
integer)

```
insert into companies values
(2007,'A',100),(2008,'B',80),(2007,'C',120),
(2008,'A',120),(2008,'B',100),(2008,'B',160)
```

-- Step 1

select *, (sales-lag(sales) over(partition by company_name order by year))
as revenue_diff
from companies

```
1 select *, (sales-lag(sales) over(partition by company_name order by year)) as revenue_diff
2 from companies
Data Output Messages Notifications
=+ | • | • | • | • |
                     $ ± ~
                                            revenue_diff
     year
               company_name
                                  sales
               character varying (10)
                                         â
     integer
                                  integer
                                            integer
          2007 A
                                        100
          2008 A
                                        120
                                                     20
3
          2008 B
                                         80
          2008 B
                                        100
                                                     20
          2007 C
5
                                        120
          2008 C
6
                                        160
                                                     40
```

```
-- step 2
```

```
with a as (
select *, lag(sales) over(partition by company_name order by year) as
revenue_diff
from companies)
select
company_name, round(((sales::decimal-revenue_diff::decimal)/revenue_dif
f::decimal)*100,2) from a
where revenue_diff is not null
```



-- Find the set of values that fits in a triangle

Note: Logic of drawing a triange is, Sum of two sides should be always greater than the 3rd side

```
x+y>z or Z+x>y or y+z>x

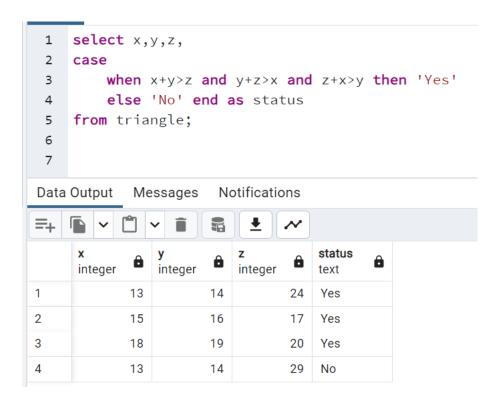
create table triangle (x integer, y integer, z integer)

insert into triangle values(13,14,24), (15,16,17),(18,19,20),(13,14,29)

-- query

select x,y,z,
case
    when x+y>z and y+z>x and z+x>y then 'Yes'
    else 'No' end as status

from triangle;
```

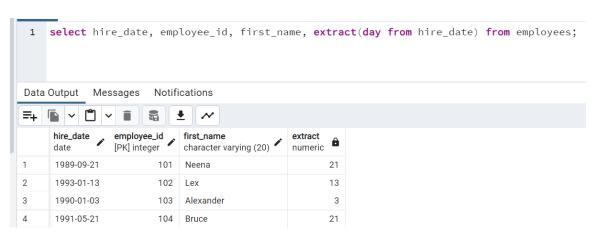


Date and Day calculations

Refer table: Employees

-- Display day from the Hire Date field

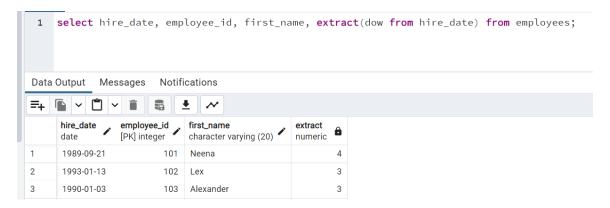
select hire_date, employee_id, first_name, extract(day from hire_date) from
employees;



-- Display day the week from the Hire date field

select hire_date, employee_id, first_name, extract(dow from hire_date) from
employees;

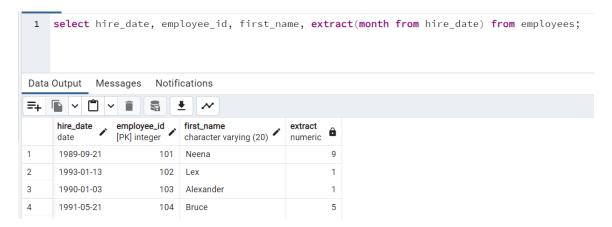
Output:



-- Dispaly month from the Hire_date field

select hire_date, employee_id, first_name, extract(month from hire_date)
from employees;

Output:



-- Display year from the Hire_date field

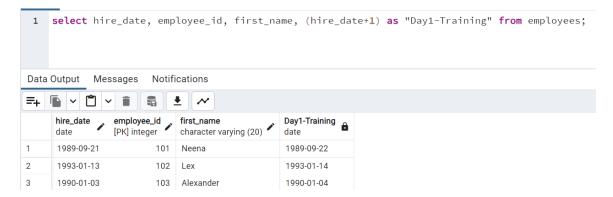
select hire_date, employee_id, first_name, extract(year from hire_date)
from employees;



-- Add 1 day from the Hire Date and rename as 'Day 1 - Training'

select hire_date, employee_id, first_name, (hire_date+1) as Day1-Training
from employees;

Output:



-- Add 6 months from the Hire date and raname as 'Permanent Employee'

select hire_date, employee_id, first_name, (hire_date+interval '180 days')
as Permanent Employee from employees;



-- Display the dayname from the Hire_date

```
select hire_date, employee_id,
first_name,Initcap(To_char(hire_date,'day'))from employees;
```

Output:

1	<pre>select hire_date, employee_id, first_name,Initcap(To_char(hire_date,'day'))from employees;</pre>							
Data	Output Mess	ages Notifi	cations					
=+								
		mployee_id PK] integer	first_name character varying (20)	initcap text				
1	1989-09-21	101	Neena	Thursday				
2	1993-01-13	102	Lex	Wednesday				
3	1990-01-03	103	Alexander	Wednesday				

-- Display the Day of the month from the Hire date

select hire_date, employee_id, first_name, extract('dow' from hire_date)
as dayofweek,
to char(hire Date, 'dd') as dayofmonth from employees;

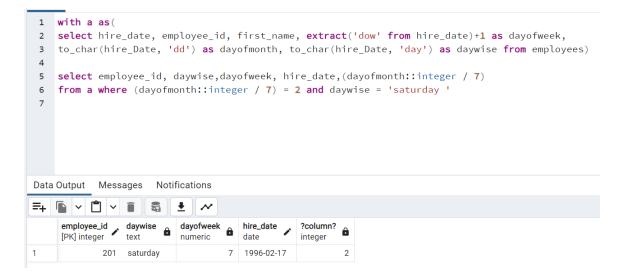
Output:



-- Display the employee who joined in the 2nd Saturday of the month.

```
with a as(
select hire_date, employee_id, first_name, extract('dow' from hire_date)+1
as dayofweek,
to_char(hire_Date, 'dd') as dayofmonth, to_char(hire_Date, 'day') as daywise
from employees)
select employee_id, daywise, dayofweek, hire_date, (dayofmonth::integer /
```

from a where (dayofmonth::integer / 7) = 2 and daywise = 'monday '



-- How to use a list of data without creating an actual table

```
SELECT word
FROM (
  VALUES
     ('apple'),
     ('banana'),
     ('orange'),
     ('grape')
) AS word table(word);
```

Note: Some queries are directly resolved using derived table without actually creating a physical table.

-- what is derived table?

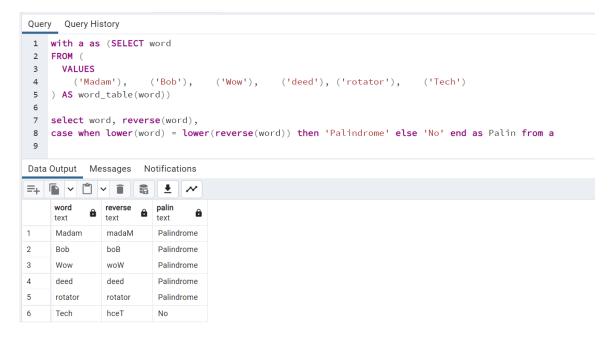
A derived table in SQL is a temporary table that is created on-the-fly within the scope of a query. It doesn't exist as a separate object in the database schema and is used only for the duration of the query. Derived tables are also known as inline views, subqueries in the FROM clause, or table expressions.

```
-- Find the Palindrome
--Query
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
```



-- Find the palindrome which has highest length

-- Query 1

```
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 max(length(word)) from a)
```

```
Query Query History
 1 with a as (SELECT word
 2 FROM (
  4
       ('Madam'),
                 ('Bob'), ('Wow'), ('deed'), ('rotator'), ('Tech')
  5 ) AS word_table(word))
  6 select word, reverse(word),
  7 case when lower(word) = lower(reverse(word)) then 'Palindrome' else 'No' end as Palin from a
  8 where length(word) = (select max(length(word)) from a )
 Data Output Messages Notifications
                 a ± ~
 =+ 🖺 🗸 📋 🗸 📋
                 palin
     text
            text
                   text
    rotator
            rotator
                   Palindrome
--Query 2
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,
count (word) from a
group by word
having length(word) = (select max(length(word)) from a )
```

```
Query Query History
  1 with a as (SELECT word
  2 FROM (
  3 VALUES
               ('Madam'), ('Bob'), ('Wow'), ('deed'), ('rotator'), ('Tech')
  4
 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 max(length(word)) from a )
Data Output Messages Notifications

    □
    √
    □
    √
    □
    √
    □
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓</
                             reverse a palin text
             text
        rotator
                                                            Palindrome
```