

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	bid integer
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
```

Output:

```

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))
6
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

```

	id integer	bid integer
1	1	3
2	6	7
3	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 ;
6

```

	employee_id [PK] integer	empsalary numeric (8,2)	managerid integer	bsalary numeric (8,2)
1	102	17000.00	103	9000.00
2	103	9000.00	104	6000.00

-- Find the last person to enter in to the lift. Lift maximum capacity is 1000 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

```

Output:

```

1  with a as (
2      select name,weight,turn, sum(weight) over(order by turn) as Capacity
3      from lift)
4  select name, weight,turn, Capacity from a where Capacity<=1000
5  order by capacity desc
6

```

Data Output

Messages

Notifications

≡

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈

	name character varying (20)	weight integer	turn integer	capacity bigint
1	Mike	300	4	932
2	Smith	232	3	632
3	Adam	200	2	400
4	John	200	1	200

-- 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

```

Output:

1	select	customer_id,	count(distinct	products)
2	from	customer_orders		
3	group by	customer_id		
4	having	count(distinct	products) =	(select count(distinct
5	order by	customer_id	asc	
6				

Data Output

Messages

Notifications

customer_id

integer

count

bigint

1	2	3
---	---	---

-- 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:

1	select	customer_id,	count(distinct	products)
2	from	customer_orders		
3	group by	customer_id		
4	having	count(distinct	products) !=	(select count(distinct
5	order by	customer_id	asc	
6				

Data Output

Messages

Notifications

customer_id

integer

count

bigint

1	1	2
2	3	1

-- 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
```

Output:

```

1 select customer_id, count(products)
2 from customer_orders
3 group by customer_id, products
4 having count(products)>1
5 order by customer_id asc
6

```

Data Output Messages Notifications



	customer_id integer	count bigint
1	3	2

-- Display the highest selling health drink

```

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

```

Output:

```

1 select products, count(*) from customer_orders
2 group by products
3 order by count(*) desc
4

```

Data Output Messages Notifications



	products character varying (20)	count bigint
1	Boost	4
2	Horlicks	2
3	Bournvita	1

*****-----End-----*****

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
```

Output:

```
1 select *, (sales-lag(sales) over(partition by company_name order by year)) as revenue_diff
2 from companies
3
```

	year	company_name	sales	revenue_diff
	integer	character varying (10)	integer	integer
1	2007	A	100	[null]
2	2008	A	120	20
3	2008	B	80	[null]
4	2008	B	100	20
5	2007	C	120	[null]
6	2008	C	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
```

Query		Query History
1	with a as (
2	select *, lag(sales) over(partition by company_name order by year) as revenue_diff	
3	from companies)	
4	select company_name,round(((sales::decimal-revenue_diff::decimal)/revenue_diff::decimal)*100,2) from a	
5	where revenue_diff is not null	
6		

Data Output		Messages	Notifications
<div> <div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> </div> </div>			
	company_name character varying (10)	round numeric	
1	A	20.00	
2	B	25.00	
3	C	33.33	

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

Note: Logic of drawing a triangle 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;
```

Output:

```

1 select x,y,z,
2 case
3     when x+y>z and y+z>x and z+x>y then 'Yes'
4     else 'No' end as status
5 from triangle;
6
7

```

Data Output Messages Notifications

	x integer	y integer	z integer	status text
1	13	14	24	Yes
2	15	16	17	Yes
3	18	19	20	Yes
4	13	14	29	No

*****--End--*****

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;
```

Output:

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

Data Output Messages Notifications

	hire_date date	employee_id [PK] integer	first_name character varying (20)	extract numeric
1	1989-09-21	101	Neena	21
2	1993-01-13	102	Lex	13
3	1990-01-03	103	Alexander	3
4	1991-05-21	104	Bruce	21

-- Display day the week from the Hire_date field

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

Output:

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

	hire_date date	employee_id [PK] integer	first_name character varying (20)	extract numeric
1	1989-09-21	101	Neena	4
2	1993-01-13	102	Lex	3
3	1990-01-03	103	Alexander	3

-- Display month from the Hire_date field

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

Output:

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

	hire_date date	employee_id [PK] integer	first_name character varying (20)	extract numeric
1	1989-09-21	101	Neena	9
2	1993-01-13	102	Lex	1
3	1990-01-03	103	Alexander	1
4	1991-05-21	104	Bruce	5

-- Display year from the Hire_date field

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

Output:

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

Data Output Messages Notifications



	hire_date date	employee_id [PK] integer	first_name character varying (20)	extract numeric
1	1989-09-21	101	Neena	1989
2	1993-01-13	102	Lex	1993
3	1990-01-03	103	Alexander	1990
4	1991-05-21	104	Bruce	1991

-- 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:

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

Data Output Messages Notifications



	hire_date date	employee_id [PK] integer	first_name character varying (20)	Day1-Training date
1	1989-09-21	101	Neena	1989-09-22
2	1993-01-13	102	Lex	1993-01-14
3	1990-01-03	103	Alexander	1990-01-04

-- 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;
```

Output:

```
2 select hire_date, employee_id, first_name, (hire_date+ interval '180 days') as Permanent_Employee
3 from employees;
4
```

Data Output Messages Notifications



	hire_date date	employee_id [PK] integer	first_name character varying (20)	permanent_employee timestamp without time zone
1	1989-09-21	101	Neena	1990-03-20 00:00:00
2	1993-01-13	102	Lex	1993-07-12 00:00:00
3	1990-01-03	103	Alexander	1990-07-02 00:00:00

-- Display the dayname from the Hire_date

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

Output:

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

	hire_date date	employee_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:

```
1 select hire_date, employee_id, first_name, extract('dow' from hire_date) as dayofweek,  
2 to_char(hire_date, 'dd') as dayofmonth from employees;  
3
```

	hire_date date	employee_id [PK] integer	first_name character varying (20)	dayofweek numeric	dayofmonth text
1	1989-09-21	101	Neena	4	21
2	1993-01-13	102	Lex	3	13
3	1990-01-03	103	Alexander	3	03

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

Output:

```

1 with a as(
2 select hire_date, employee_id, first_name, extract('dow' from hire_date)+1 as dayofweek,
3 to_char(hire_date, 'dd') as dayofmonth, to_char(hire_date, 'day') as daywise from employees)
4
5 select employee_id, daywise, dayofweek, hire_date, (dayofmonth::integer / 7)
6 from a where (dayofmonth::integer / 7) = 2 and daywise = 'saturday '
7

```

Data Output

Messages

Notifications

+

📄

▼

📋

▼

🗑️

🗑️

📦

⬇️

📈

	employee_id [PK] integer	daywise text	dayofweek numeric	hire_date date	?column? integer
1	201	saturday	7	1996-02-17	2

-- 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

```

Output:

Query		Query History	
1	with a as (SELECT word		
2	FROM (
3	VALUES		
4	('Madam'), ('Bob'), ('Wow'), ('deed'), ('rotator'), ('Tech')		
5) AS word_table(word))		
6			
7	select word, reverse(word),		
8	case when lower(word) = lower(reverse(word)) then 'Palindrome' else 'No' end as Palin from a		
9			

Data Output		Messages	Notifications
word	reverse	palin	
text	text	text	
1	Madam	madaM	Palindrome
2	Bob	boB	Palindrome
3	Wow	woW	Palindrome
4	deed	deed	Palindrome
5	rotator	rotator	Palindrome
6	Tech	hceT	No

-- 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 )

```

Output:

Query		Query History	
1	with a as (SELECT word		
2	FROM (
3	VALUES		
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)		
9			
Data Output		Messages	Notifications
			
			
word	reverse	palin	
text	text	text	
1	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 )
```

Output:

Query Query History

```
1 with a as (SELECT word
2 FROM (
3   VALUES
4     ('Madam'), ('Bob'), ('Wow'), ('deed'), ('rotator'), ('Tech')
5 ) AS word_table(word))
6
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
11 having length(word) = (select max(length(word)) from a )
12
```

Data Output Messages Notifications



	word text	reverse text	palin text	count bigint
1	rotator	rotator	Palindrome	1