**Q51**.

Ans:

Select name, population, area

From world where area >3000000 or population> 25000000

Q52.

Ans:

select name from customer

where referee\_id <>2

Q53.

Ans:

select name from customers

where id not in (select customerId from orders)

Q54.

Ans:

select employee\_id,

count(employee\_id) over (partition by team\_id) team\_size

from Employee

order by employee\_id;

Q55.

Ans:

with cte as(

select p.id ,p.name person\_name ,p.phone\_number, cntry.\*, c.\*

from

calls c

left join

person p

on c.caller\_id= p.id

left join

country cntry

on trim(leading '0' from substring\_index(p.phone\_number,'-',1))= cntry.country\_code

union all

select p.id ,p.name person\_name ,p.phone\_number, cntry.\*, c.\*

from

calls c

left join

person p

on c.callee\_id= p.id

left join

country cntry

on trim(leading '0' from substring\_index(p.phone\_number,'-',1))= cntry.country\_code

),

cte1 as (

select distinct name country\_name,

avg(duration) over (partition by name) avg\_cntry,

avg(duration) over () global\_avg

from cte

)

select country\_name from

cte1 where avg\_cntry>global\_avg

Q56.

Ans:

select

player\_id,

min(event\_date) first\_login

from activity

group by player\_id

Q57.

Ans:

With cte1 as (

select customer\_number ,

count(order\_number) num\_orders

from Orders

group by customer\_number

)

select customer\_number from cte1

where num\_orders = (select max(num\_orders) from cte1)

note: same query can work for follow-up case

Q58.

Ans:

select distinct a.seat\_id

from cinema a inner join cinema b

on abs(a.seat\_id - b.seat\_id) = 1 and a.free = 1 and b.free = 1

order by a.seat\_id

Q59.

Ans:

with cte as (

select distinct sp.sales\_id from

salesperson sp

left join

orders o

on sp.sales\_id = o.sales\_id

left join

company comp

on comp.com\_id = o.com\_id

where comp.name ='RED'

)

select name from salesperson

where sales\_id not in (select sales\_id from cte)

Q60.

Ans:

select x, y, z, if(x + y > z and y + z > x and z + x > y, 'yes', 'no') triangle

from triangle;

Q61.

Ans:

select min(abs(a.x-b.x)) shortest from point a

join

point b

on abs(a.x-b.x) <> 0

Q62.

Ans:

with cte as (

select actor\_id ,director\_id, count(\*) times from

ActorDirector

group by actor\_id ,director\_id

)

select actor\_id ,director\_id from cte

where times>=3

Q63.

Ans:

select p.product\_name, s.year, sum(s.price) from

sales s

left join

product p

on s.product\_Id = p.product\_Id

group by p.product\_name, s.year

Q64.

Ans:

select p.project\_id,avg(e.experience\_years) from

project p

left join

employee e

on p.employee\_id = e.employee\_id

group by p.project\_id;

Q65.

Ans:

with cte as (

select seller\_id, sum(quantity \* price) total\_sale from

sales

group by seller\_id

),

cte1 as (

select seller\_id,

dense\_rank() over (order by total\_sale desc) rnk

)

select seller\_id

from cte1

where rnk=1;

Q66.

Ans:

select a.buyer\_id from

(

select distinct s.buyer\_id from

sales s

left join

product p

on s.product\_id = p.product\_id

where p.product\_name='S8'

) a

left join

(

select distinct s.buyer\_id from sales s

left join

product p

on s.product\_id = p.product\_id

where p.product\_name in ('iPhone')

) b

on a.buyer\_id=b.buyer\_id

where b.buyer\_id is null

Q67.

Ans:

with cte as(

select visited\_on,

sum(amount) amount

from customer

group by visited\_on

),

cte1 as (

select visited\_on,

sum(amount) over (order by visited\_on range between interval 6 day preceding and current row) amount,

avg(amount) over (order by visited\_on range between interval 6 day preceding and current row) average\_amount

from

cte )

select visited\_on, amount, average\_amount

from cte1 where visited\_on>='2019-01-07';

Q68.

Ans:

select gender, day, sum(score) total from Scores

group by gender, day;

Q69.

Ans:

select log\_start.log\_id as start\_id, min(log\_end.log\_id) as end\_id from

(select log\_id from logs where log\_id - 1 not in (select \* from Logs)) log\_start,

(select log\_id from logs where log\_id + 1 not in (select \* from Logs)) log\_end

where log\_start.log\_id <= log\_end.log\_id

group by log\_start.log\_id;

Q70.

Ans:

select stud.student\_id, stud.student\_name, exam.subject\_name, count(\*) attended\_exams from

Students stud

left join

examinations exam

on stud.student\_id =exam.student\_id

group by stud.student\_id, stud.student\_name, exam.subject\_name

Q71.

Ans:  
select a.employee\_id from employee a

left join

employee b

on a.manager\_id =b.employee\_id

left join employee c

on b.manager\_id =c.employee\_id

left join employee d

on c.manager\_id=d.employee\_id

where (b.employee\_name='Boss' or c.employee\_name='Boss' or d.employee\_name='Boss')

Q72.

Ans:

select distinct concat(year(trans\_date),'-',date\_format(trans\_date,'%m')) "month",

country,

count(\*) over (partition by country, concat(year(trans\_date),'-',month(trans\_date))) trans\_count,

sum(case when state='approved' then 1 else 0 end) over

(partition by country, concat(year(trans\_date),'-',month(trans\_date))) approved\_count,

sum(amount) over (partition by country, concat(year(trans\_date),'-',month(trans\_date))) trans\_total\_amount,

sum(case when state='approved' then amount else 0 end) over

(partition by country, concat(year(trans\_date),'-',month(trans\_date))) approved\_total\_amount

from Transactions

Q73.

Ans:

with cte as

(

select distinct action\_date,

(count(r.remove\_date) over (partition by action\_date) /count(action\_date) over (partition by action\_date) )\*100 avgg from

actions a

left join removals r

on a.post\_id=r.post\_id

where action='report'

)

select round(avg(avgg),2) from cte

Q74.

Ans:

with cte as

(

select player\_id, min(event\_date) first\_logg from activity

group by player\_id

),

cte1 as

(

select count(distinct player\_id) cnt from activity

),

cte2 as

(

select count(distinct player\_id) cnt from activity

where (player\_id, event\_date) in (select player\_id, date\_add(first\_logg, interval 1 day) from cte)

)

select round(cte2.cnt/cte1.cnt,2) fraction from

cte2, cte1

Q75.

Ans: repeated question

Q76.

Ans:

with cte as

(

select \*,

max(salary) over (partition by company\_id) max\_sal

from salaries

),

cte1

as

(

select \*,

case when max\_sal<1000 then 0

when max\_sal between 1000 and 10000 then 0.24

when max\_sal> 10000 then 0.49

end tax

from cte

)

select

company\_id, employee\_id, employee\_name,

round(salary \* (1-tax))

from

cte1

Q77.

Ans:

select sale\_date,

sum(case when fruit='apples' then sold\_num else 0 end) -

sum(case when fruit='oranges' then sold\_num else 0 end) diff

from sales

group by sale\_date

Q78.

Ans:

select e.left\_operand, e.operator, e.right\_operand,

case

when e.operator = '<' then if(l.value < r.value,'true','false')

when e.operator = '>' then if(l.value > r.value,'true','false')

else if(l.value = r.value,'true','false')

end as value

from expressions e

left join variables l on e.left\_operand = l.name

left join variables r on e.right\_operand = r.name

Q79.

Ans:

Part a)

with cte as (

select a.name, count(\*) cnt from

Users a

left join

MovieRating b

on a.user\_id=b.user\_id

group by a.name

)

select \* from cte

where cnt = (select max(cnt) from cte)

order by name;

part b)

with cte as (

select a.title, avg(b.rating) avg\_rat from

movies a

left join

MovieRating b

on a.movie\_id=b.movie\_id and b.created\_at between '2020-02-01' and '2020-02-29'

group by a.name

)

select \* from cte

where avg\_rat = (select max(avg\_rat) from cte)

order by title;

Q80.

Ans:

with cte as(

select p.id ,p.name person\_name ,p.phone\_number, cntry.\*, c.\*

from

calls c

left join

person p

on c.caller\_id= p.id

left join

country cntry

on trim(leading '0' from substring\_index(p.phone\_number,'-',1))= cntry.country\_code

union all

select p.id ,p.name person\_name ,p.phone\_number, cntry.\*, c.\*

from

calls c

left join

person p

on c.callee\_id= p.id

left join

country cntry

on trim(leading '0' from substring\_index(p.phone\_number,'-',1))= cntry.country\_code

),

cte1 as (

select distinct name country\_name,

avg(duration) over (partition by name) avg\_cntry,

avg(duration) over () global\_avg

from cte

)

select country\_name from

cte1 where avg\_cntry>global\_avg

Q81.

Ans:

select name from students where marks>75

order by substring(name,-3,3), id;

Q82.

Ans:

select name from employee

order by name

Q83.

Ans:

select name from employee

where salary>2000 and months<10

order by employee\_id

Q84.

Ans:

select

case

when a+b <=c or b+c<=a or c+a <=b then 'Not a Triangle'

when a=b and b=c then 'Equilateral'

when a=b or b=c then 'Isosceles'

when a<>b and b<>c then 'Scalene'

end output

from triangles;

Q85.

Ans:

with cte as (

select year(transaction\_date) yr, product\_id,

sum(spend) spend from user\_transactions

group by year(transaction\_date) , product\_id

),

cte1 as (

select yr, product\_id, spend curr\_year\_spend,

lag(spend,1) over (partition by product\_id order by yr) prev\_year\_spend

from cte

)

select \*,

round((curr\_year\_spend- prev\_year\_spend)\*100/prev\_year\_spend,2) yoy\_rate

from cte1

;

Q86.

Ans:

with cte as (

SELECT item\_type, sum(square\_footage) summ , count(item\_type) cnt FROM inventory

group by item\_type

),

cte1 as (

select

500000- (floor(500000/summ)\*summ) sum\_left\_for\_nonprime,

(floor(500000/summ)\*summ) sum\_forprime,

floor(500000/summ)\* cnt prime\_cnt

from cte

where item\_type='prime\_eligible'

),

cte2 as(

select

floor(non\_prime.sum\_left/non\_prime\_summ.summ)\* non\_prime\_summ.cnt non\_prime\_cnt

from

(select sum\_left\_for\_nonprime sum\_left from cte1) non\_prime,

(select summ, cnt from cte where item\_type='not\_prime' )

non\_prime\_summ

)

select

cte1.prime\_cnt,

cte2.non\_prime\_cnt

from cte1,cte2

Q87

Ans:

with cte as (

SELECT user\_id, extract(MONTH from event\_date) mnth,

string\_agg(event\_type,',') actions

FROM user\_actions

group by user\_id, extract(MONTH from event\_date)

),

cte1 as

(select b.mnth "month", count(\*) monthly\_active\_users from cte a

inner join

cte b

on

a.user\_id=b.user\_id

and a.mnth = b.mnth-1

and (a.actions like '%sign-in%' or a.actions like '%like%' or a.actions like '%comment%')

and (b.actions like '%sign-in%' or b.actions like '%like%' or b.actions like '%comment%' )

group by b.mnth)

select cte1.month "month", monthly\_active\_users

from cte1

where

(cte1.month, monthly\_active\_users)

=(select max(cte1.month),max(monthly\_active\_users) from cte1)

Q88.

Ans:

with digit as (

select 0 as d union all

select 1 union all select 2 union all select 3 union all

select 4 union all select 5 union all select 6 union all

select 7 union all select 8 union all select 9

),

seq as (

select a.d + (10 \* b.d) + (100 \* c.d) + (1000 \* d.d) as num

from digit a

cross join

digit b

cross join

digit c

cross join

digit d

where a.d + (10 \* b.d) + (100 \* c.d) + (1000 \* d.d) <>0

),

cte as (

select searches,

row\_number() over (order by searches) ordr from search\_frequency sf

cross join

seq sq

where sq.num between 1 and sf.num\_users

order by sf.searches

),

cte1 as (

select searches, ordr,

ceil(avg(ordr) over ()) ceill,

floor(avg(ordr) over ()) flr

from cte

)

select round(avg(searches),1) median from cte1

where ceill=ordr or flr=ordr

Q89.

Ans:

WITH payment\_status

AS (

SELECT

advertiser.user\_id,

advertiser.status,

payment.paid

FROM advertiser

LEFT JOIN daily\_pay AS payment

ON advertiser.user\_id = payment.user\_id

UNION

SELECT

payment.user\_id,

advertiser.status,

payment.paid

FROM daily\_pay AS payment

LEFT JOIN advertiser

ON advertiser.user\_id = payment.user\_id

)

SELECT

user\_id,

CASE WHEN paid IS NULL THEN 'CHURN'

WHEN status != 'CHURN' AND paid IS NOT NULL THEN 'EXISTING'

WHEN status = 'CHURN' AND paid IS NOT NULL THEN 'RESURRECT'

WHEN status IS NULL THEN 'NEW'

END AS new\_status

FROM payment\_status

ORDER BY user\_id;

Q90.

Ans:

with cte as (

select \*,

count(case when session\_status='start' then 1 else null end) over (partition by server\_id order by status\_time) grp

from server\_utilization

),

cte1 as(

select

datediff(last\_value(status\_time) over (partition by server\_id,grp order by status\_time),

first\_value(status\_time) over (partition by server\_id,grp order by status\_time)) ttl

from cte

)

select sum(ttl) total\_uptime\_days from cte1;

Q91.

Ans: