**SQL Section- – Solution’s**

**Name: Akilesh Vishnu Mohan Raj**

**Email:** [**akileshvishnu10@gmail.com**](mailto:akileshvishnu10@gmail.com)

**Batch 2 : Full Stack Big Data Analytics**

1. `

select username,activity,startDate,endDate from (

select \* , rank()over(partition by username order by startDate desc) as rnk,

count(username) over( partition by username order by startDate desc) as cnt

from UserActivity)tmp where rnk=2 or cnt=1;

1. `

select username,activity,startDate,endDate from (

select \* , rank()over(partition by username order by startDate desc) as rnk,

count(username) over( partition by username order by startDate desc) as cnt

from UserActivity)tmp where rnk=2 or cnt=1;

1. `

SELECT name from STUDENTS where MARKS >75 order by right(name,3);

1. `

select \* from EMPLOYEE order by name;

1. `

select \* from EMPLOYEE where salary > 2000 and months < 10 order by employee\_id ;

1. `
2. `

select \* from EMPLOYEE order by months\*salary;

1. `
2. `

(SELECT CONCAT(Name,'(',SUBSTRING(OCCUPATION,1,1),')') from OCCUPATIONS order

by Name asc ) UNION

(SELECT CONCAT('There are a total of ',count(OCCUPATION),'

',lower(OCCUPATION),'s','.') from OCCUPATIONS group by OCCUPATION ORDER BY

COUNT(occupation),occupation asc);

1. `
2. `
3. `

select N,

CASE

WHEN P is NULL then 'Root'

WHEN N in (select P from BST) then 'Inner'

else 'Leaf'

end

from BST order by N;

1. `
2. `

select N,

CASE

WHEN P is NULL then 'Root'

WHEN N in (select P from BST) then 'Inner'

else 'Leaf'

end

from BST order by N;

1. `

select N,

CASE

WHEN P is NULL then 'Root'

WHEN N in (select P from BST) then 'Inner'

else 'Leaf'

end

from BST order by N;

1. `

select N,

CASE

WHEN P is NULL then 'Root'

WHEN N in (select P from BST) then 'Inner'

else 'Leaf'

end

from BST order by N;

1. `

SELECT name from STUDENTS where MARKS >75 order by right(name,3);

1. `

select \* from EMPLOYEE order by name;

1. `

select \* from EMPLOYEE where salary > 2000 and months < 10 order by employee\_id ;

1. `

SELECT CASE

WHEN A + B <= C OR A + C <= B OR B + C <= A THEN 'Not A Triangle'

WHEN A = B AND B = C THEN 'Equilateral'

WHEN A = B OR B = C OR A = C THEN 'Isosceles'

ELSE 'Scalene'

END as result

FROM TRIANGLES;

1. `
2. `

SELECT item\_type,

case

when item\_type = 'prime\_eligible'

then Floor(500000/sum(square\_footage))\*count(item\_type)

else floor((500000 -

(select(floor(500000/sum(square\_footage)))\*sum(square\_footage) from inventory

where item\_type = 'prime\_eligible'))/sum(square\_footage))\*Count(item\_type)

end

from

inventory

group by item\_type

order by item\_type desc;

1. `

SELECT EXTRACT(MONTH FROM a1.event\_date) as month, COUNT(DISTINCT a1.users\_id)

as monthly\_active\_users

from user\_actions as a1,user\_actions as a2

where a1.users\_id = a2.users\_id AND

EXTRACT(MONTH FROM a1.event\_date) = 7

AND EXTRACT(MONTH FROM a2.event\_date) =6

AND EXTRACT(YEAR FROM a1.event\_date) = 2022

AND EXTRACT(YEAR FROM a2.event\_date) =2022

AND a1.event\_type in ( 'sign-in', 'like', 'comment') AND

a2.event\_type in ('sign-in', 'like', 'comment')

GROUP BY month;

1. `

WITH expanded AS(

SELECT searches

FROM search\_frequency

GROUP BY searches, GENERATE\_SERIES(1,num\_users)

)

SELECT PERCENTILE\_CONT(0.5) WITHIN GROUP (ORDER BY searches) AS median

FROM expanded

1. `
2. `
3. `
4. `
5. `

select gender,day, sum(score\_points) over(partition by gender order by

gender,day rows BETWEEN unbounded preceding and current row) as total

from Scores;

1. `

SELECT cc.name from person p inner join calls c on p.id=c.caller\_id or

p.id=c.callee\_id

inner join country cc on cc.country\_code=left(p.phone\_number,3) group by

cc.name having avg(c.duration) >

(select avg(duration) from calls);

1. `

select

avg(number) median

from

Numbers n

where

n.frequency >= abs(

(select sum(Frequency) from Numbers where

Number<=n.number)

-

(select sum(Frequency) from Numbers where

Number>=n.number));

1. `

select

pay\_month,

department\_id,

case when dept\_avg > comp\_avg then 'higher' when dept\_avg < comp\_avg then

'lower' else 'same' end comparison

from (

select date\_format(b.pay\_date, '%Y-%m') pay\_month, a.department\_id,

avg(b.amount) dept\_avg, d.comp\_avg

from employee a

inner join salary b

on (a.employee\_id = b.employee\_id)

inner join (select date\_format(c.pay\_date, '%Y-%m') pay\_month,

avg(c.amount) comp\_avg

from salary c

group by date\_format(c.pay\_date, '%Y-%m')) d

on ( date\_format(b.pay\_date, '%Y-%m') = d.pay\_month)

group by date\_format(b.pay\_date, '%Y-%m'), department\_id, d.comp\_avg) final

1. `
2. `
3. `
4. `
5. `
6. `
7. `
8. `
9. `
10. `
11. `
12. `
13. `
14. `
15. `