Solution:

Ans 1:

SELECT \*

FROM

Logs l1,

Logs l2,

Logs l3

WHERE

l1.Id = l2.Id - 1

AND l2.Id = l3.Id - 1

AND l1.Num = l2.Num

AND l2.Num = l3.Num

;

Ans 2 :

SELECT

name AS 'Name'

FROM

Candidate

JOIN

(SELECT

Candidateid

FROM

Vote

GROUP BY Candidateid

ORDER BY COUNT(\*) DESC

LIMIT 1) AS winner

WHERE

Candidate.id = winner.Candidateid

Ans 3:

SELECT

dept\_name, COUNT(student\_id) AS student\_number

FROM

department

LEFT OUTER JOIN

student ON department.dept\_id = student.dept\_id

GROUP BY department.dept\_name

ORDER BY student\_number DESC , department.dept\_name

LIMIT 3

;

Ans 4:

WITH firstConditionCTE AS

(

SELECT tiv\_2015 FROM Insurance GROUP BY tiv\_2015 HAVING COUNT(\*) > 1

),

SecondConditionCTE AS

(

SELECT concat(lat, ' ', lon) as uniq\_lat\_lon FROM Insurance GROUP BY concat(lat, ' ', lon) HAVING COUNT(\*) = 1

)

SELECT

ROUND(SUM(tiv\_2016),2) AS tiv\_2016

FROM Insurance

WHERE tiv\_2015 IN (SELECT \* FROM firstConditionCTE)

AND

concat(lat, ' ', lon) IN (SELECT uniq\_lat\_lon FROM SecondConditionCTE)

Ans 5:

SELECT

id, 'Root' AS Type

FROM

tree

WHERE

p\_id IS NULL

UNION

SELECT

id, 'Leaf' AS Type

FROM

tree

WHERE

id NOT IN (SELECT DISTINCT

p\_id

FROM

tree

WHERE

p\_id IS NOT NULL)

AND p\_id IS NOT NULL

UNION

SELECT

id, 'Inner' AS Type

FROM

tree

WHERE

id IN (SELECT DISTINCT

p\_id

FROM

tree

WHERE

p\_id IS NOT NULL)

AND p\_id IS NOT NULL

ORDER BY id;

Ans 6:

select round(min(shortest\_distance),2) as shortest from (

select \*, sqrt(POW((x2 - x1),2) + POW((y2 - y1),2)) as shortest\_distance

from (

Select A.x as x1, A.y as y1, B.x as x2, B.y as y2

from Point2D A, Point2D B

) abc

where (x1,y1)<> (x2,y2)

)mno

Ans 7:

select

case

when id % 2 = 0 then id - 1

when id % 2 = 1 and id < (select count(\*) from seat)then id + 1

else id

end as id,

student from seat

order by id

Ans 8:

with debit as

(

select paid\_by as user\_id,

-sum(amount) as amount

from transactions group by paid\_by

),

credit as

(

select paid\_to as user\_id,

sum(amount) as amount

from transactions group by paid\_to

),

all\_transaction as

(

select \* from debit

union all

select \* from credit

),

amount\_with\_users as

(

select u.user\_id,

u.user\_name,

coalesce(u.credit + sum(amount), u.credit) as final\_amt

from all\_transaction t

right join users u on t.user\_id = u.user\_id

group by user\_id, user\_name, credit

),

final as

(

select user\_id,

user\_name,

final\_amt as credit,

if (final\_amt>0, 'No', 'Yes') as credit\_limit\_breached

from amount\_with\_users

)

select \* from final

Ans 9 :

select

if (from\_id < to\_id, from\_id, to\_id) as person1,

if (from\_id < to\_id, to\_id, from\_id) as person2,

count(\*) as call\_count,

sum(duration) as total\_duration

from calls

group by person1,person2;