1)/\*Number

-----------

1,2,3,5,6,8,9,15,19,21,22,24,29

Find the count of numbers missing in the given list.\*/

create table assign5\_1(Number int)

insert into assign5\_1 values(1),(2),(3),(5),(6),(8),(9),(15),(19),(21),(22),(24),(29)

select max(temp.count) from (select Number- row\_number() over(order by Number) as count from assign5\_1)as temp

--OR

select max(Number)-count(\*) as missing\_numbers from assign5\_1

2)/\*ID Val

----------- -----------

1 2

2 3

3 4

4 5

5 6

6 4

7 6

8 9

9 6

10 8

Find the value which is most repeated in the table.

Expected output

Val count

6 3 \*/

with cte(val,cnt,rank) as(select val,count(val) as cnt,rank() over( order by val) as rank from assign5\_2 group by val)

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

--OR

select distinct top(1) with ties val,count(val) as count from assign5\_2 group by val order by count desc

--<https://www.codeproject.com/Tips/480781/SQL-Server-TOP-WITH-TIES-A-Beauty-of-TSQL> Check this cout. Ties usage works but not recommended.

3)/\*id col1

----------- -----------

1 1

2 1

3 2

4 2

5 3

6 4

7 5

8 5

Delete records with duplicte col1 values. Keep the latest one and delete the earlier occurences \*/

create table assign5\_3(id smallint,val smallint)

insert into assign5\_3 values(1,1),(2,1),(3,2),(4,2),(5,3),(6,4),(7,5),(8,5)

delete a1 from assign5\_3 a1 join assign5\_3 a2 on a1.val=a2.val and a1.id<a2.id

4)/\*Sname SMarks SSubject

A 10 X

B 20 X

C 30 Y

D 40 Y

write a query to return the students who have scored more than avg mark in that subject.

Expected output

B 20 X

D 40 Y \*/

create table assign5\_4(name varchar(10),marks smallint,subject varchar(10))

insert into assign5\_4 values('A',10,'X'),('B',20,'X'),('C',30,'Y'),('D',40,'Y')

with cte1 as (select name,marks,subject,avg(marks) over(partition by subject) as average from assign5\_4 )

select name,marks,subject from cte1 where marks>average

5)/\*To find out the distance between two cities we have to call an API. We should call only once if the two cities were already considered irrespective of their order.

I.e. for below data for HYD-DEL it should call only once and we should ignore DEL-HYD record. Write a query to find out such unique cities.

Trip Table

TripID Source Target

1 HYD DEL

2 DEL HYD

3 HYD VIZ

4 VIZ HYD

5 HYD TPT

6 BEN KOC

7 KOC PUN

8 PUN KOC \*/

create table assign5\_5(TripId int,Source varchar(5),Target varchar(5))

insert into assign5\_5 values(1,'HYD','DEL'),(2,'DEL','HYD'),(3,'HYD','VIZ'),(4,'VIZ','HYD'),(5,'HYD','TPT'),(6,'BEN','KOC'),(7,'KOC','PUN'),(8,'PUN','KOC')

select a1.\* from assign5\_5 a1 left join assign5\_5 a2 on a1.source=a2.target and a1.target=a2.source

where a1.TripId < a2.TripId or a2.TripId is null

6)

/\*Below truck destination reached times. If any of the truck reached less than 15 mins of previous then those all belongs to same trip. Consider trip interval is starting point of a day.

\*/

create table assign5\_6(truckname char(1),reachedtime datetime)

insert into assign5\_6 values

('A','2017-06-07 14:39:09.000'),

('B','2017-06-07 14:43:11.000'),

('C','2017-06-07 14:39:04.000'),

('D','2017-06-07 13:11:09.000'),

('E','2017-06-08 12:45:44.000'),

('F','2017-06-08 12:51:09.000'),

('G','2017-06-07 04:11:04.000'),

('H','2017-06-07 13:13:09.000'),

('I','2017-06-07 03:41:59.000'),

('J','2017-06-07 18:55:09.000'),

('K','2017-06-07 19:02:09.000'),

('L','2017-06-07 21:30:09.000'),

('M','2017-06-07 12:52:09.000')

with cte as(select \*,row\_number() over(order by reachedtime) as id from assign5\_6 ),

rcte as(

select truckname,reachedtime,id,1 as tripnumber from cte where id=1

union all

select a.\*,case when abs(datediff(minute,a.reachedtime,r.reachedtime)) between 0 and 15 then tripnumber else tripnumber+1 end

from cte a join rcte r on a.id=r.id+1

)

select truckname,tripnumber from rcte group by truckname,tripnumber order by tripnumber

/\*why group by truckname,tripnumber????

Since we are selecting truckname,tripnumber ,group by should contain both of them

\*/

7)/\* Find 2nd highest of the employee. (Using CTEs)

Sample Input

Name Salary

e5 45000

e3 30000

e2 49000

e4 36600

e1 58000

Expected Output

Name Salary

e2 49000 \*/

with cte2(name,salary,rank) as(select name,salary,dense\_rank() over(order by salary desc) from assign5\_7 as rank)

select name,salary from cte2 where rank=2