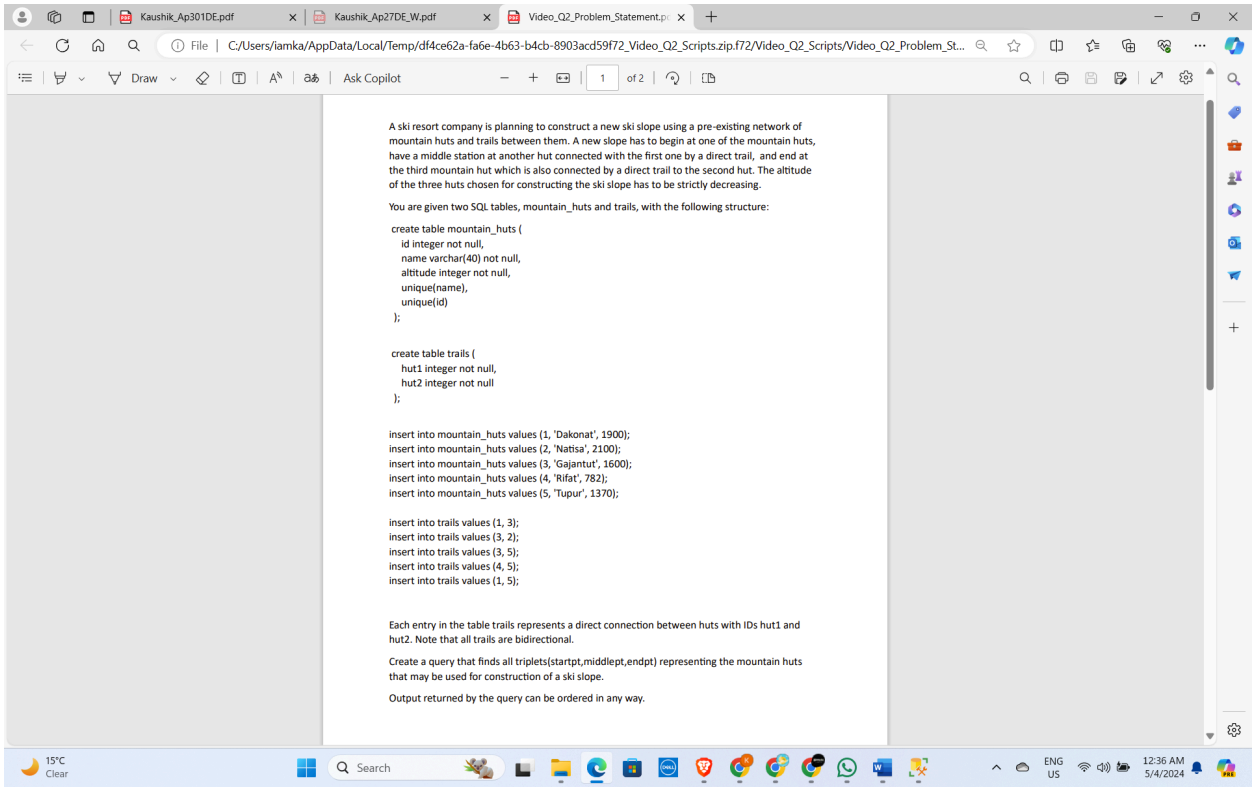
Day88 - May 3rd 2024

1. Started my day as usual
2. learning sliding window pattern
   Find the doc here :
   https://docs.google.com/document/d/1VxVQIBiGOiTrNB0Iwx2jt1Fz5Vz0WvPnPzNOMLZ2Tak/edit?usp=sharing

3. Solved a sliding window pattern problem on leetcode 3. Longest Substring Without Repeating Characters
   Pls find my sol doc here :
   https://docs.google.com/document/d/1gR9z5YZv3-B9GYBBTKYM7CItzLKou-6ePSLDpFapxOs/edit?usp=sharing

4. ENded my by solving a complex SQL from youtube



A ski resort company is planning to construct a new ski slope using a pre-existing network of mountain huts and trails between them. A new slope has to begin at one of the mountain huts, have a middle station at another hut connected with the first one by a direct trail, and end at the third mountain hut which is also connected by a direct trail to the second hut. The altitude of the three huts chosen for constructing the ski slope has to be strictly decreasing.

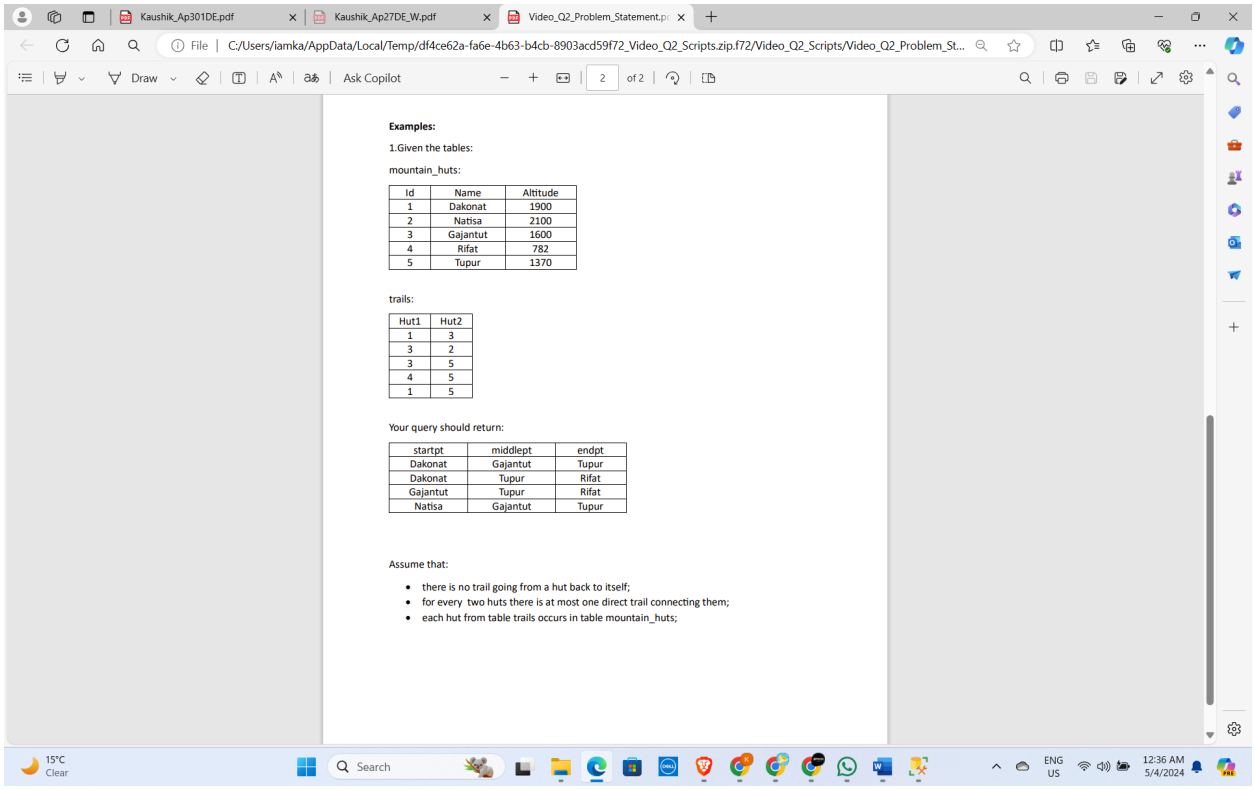
You are given two SQL tables, mountain_huts and trails, with the following structure:

```
create table mountain_huts (
  id integer not null,
  name varchar(40) not null,
  altitude integer not null,
  unique(name),
  unique(id)
);

create table trails (
  hut1 integer not null,
  hut2 integer not null
);

insert into mountain_huts values (1, 'Dakonat', 1900);
insert into mountain_huts values (2, 'Natisa', 2100);
insert into mountain_huts values (3, 'Gajantut', 1600);
insert into mountain_huts values (4, 'Rifat', 782);
insert into mountain_huts values (5, 'Tupur', 1370);

insert into trails values (1, 3);
insert into trails values (3, 2);
insert into trails values (3, 5);
insert into trails values (4, 5);
insert into trails values (1, 5);
```

Each entry in the table trails represents a direct connection between huts with IDs hut1 and hut2. Note that all trails are bidirectional.

Create a query that finds all triplets(startpt,middlept,endpt) representing the mountain huts that may be used for construction of a ski slope.

Output returned by the query can be ordered in any way.



Examples:

1.Given the tables:

mountain_huts:

Id	Name	Altitude
1	Dakonat	1900
2	Natisa	2100
3	Gajantut	1600
4	Rifat	782
5	Tupur	1370

trails:

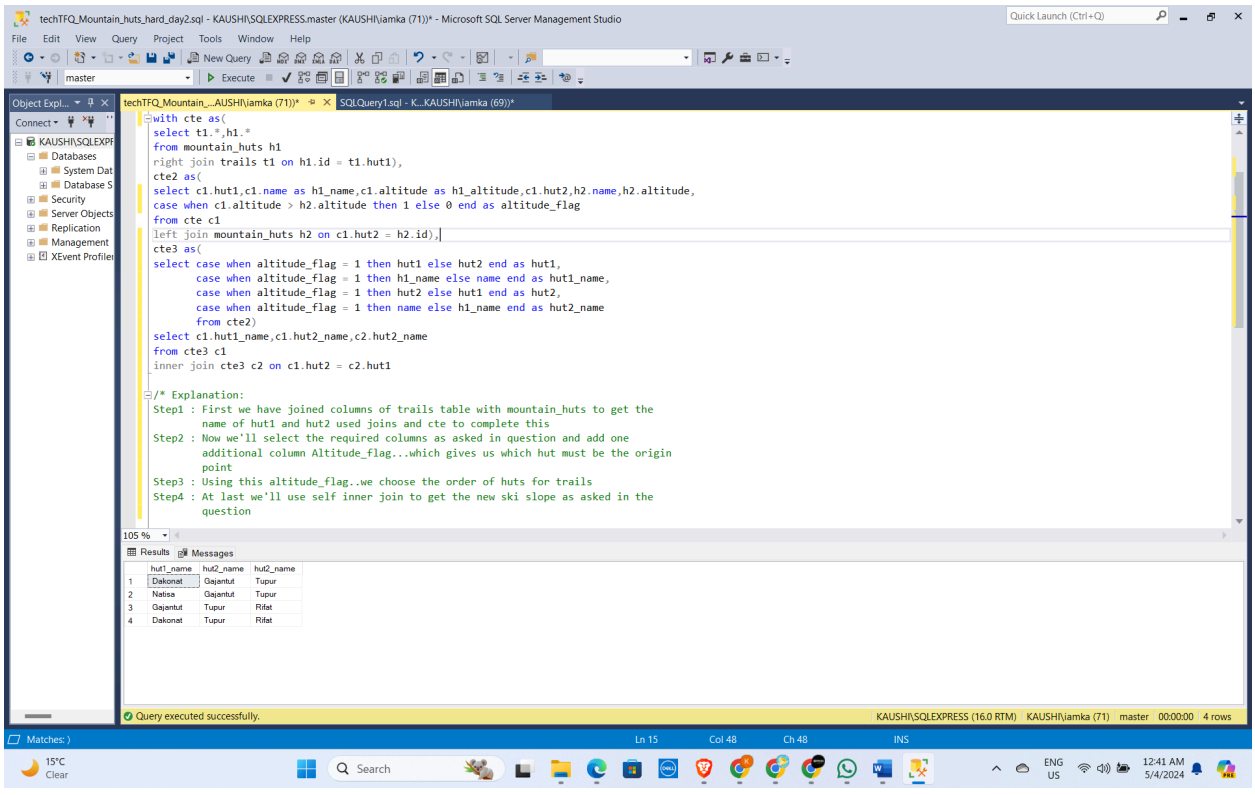
Hut1	Hut2
1	3
3	2
3	5
4	5
1	5

Your query should return:

startpt	middlepт	endpt
Dakonat	Gajantut	Tupur
Dakonat	Tupur	Rifat
Gajantut	Tupur	Rifat
Natisa	Gajantut	Tupur

Assume that:

- there is no trail going from a hut back to itself;
- for every two huts there is at most one direct trail connecting them;
- each hut from table trails occurs in table mountain_huts;



```
with cte as(
select t1.*,h1.*
from mountain_huts h1
right join trails t1 on h1.id = t1.hut1),
cte2 as(
select c1.hut1,c1.name as h1_name,c1.altitude as h1_altitude,c1.hut2,h2.name,h2.altitude,
case when c1.altitude > h2.altitude then 1 else 0 end as altitude_flag
from cte c1
left join mountain_huts h2 on c1.hut2 = h2.id),
cte3 as(
select case when altitude_flag = 1 then hut1 else hut2 end as hut1,
       case when altitude_flag = 1 then h1_name else name end as hut1_name,
       case when altitude_flag = 1 then hut2 else hut1 end as hut2,
       case when altitude_flag = 1 then name else h1_name end as hut2_name
       from cte2)
select c1.hut1_name,c1.hut2_name,c2.hut2_name
from cte3 c1
inner join cte3 c2 on c1.hut2 = c2.hut1

/* Explanation:
Step1 : First we have joined columns of trails table with mountain_huts to get the
        name of hut1 and hut2 used joins and cte to complete this
Step2 : Now we'll select the required columns as asked in question and add one
        additional column Altitude_flag...which gives us which hut must be the origin
        point
Step3 : Using this altitude_flag..we choose the order of huts for trails
Step4 : At last we'll use self inner join to get the new ski slope as asked in the
        question
```

	hut1_name	hut2_name	hut2_name
1	Dakonat	Gajantut	Tupur
2	Natisa	Gajantut	Tupur
3	Gajantut	Tupur	Rifat
4	Dakonat	Tupur	Rifat