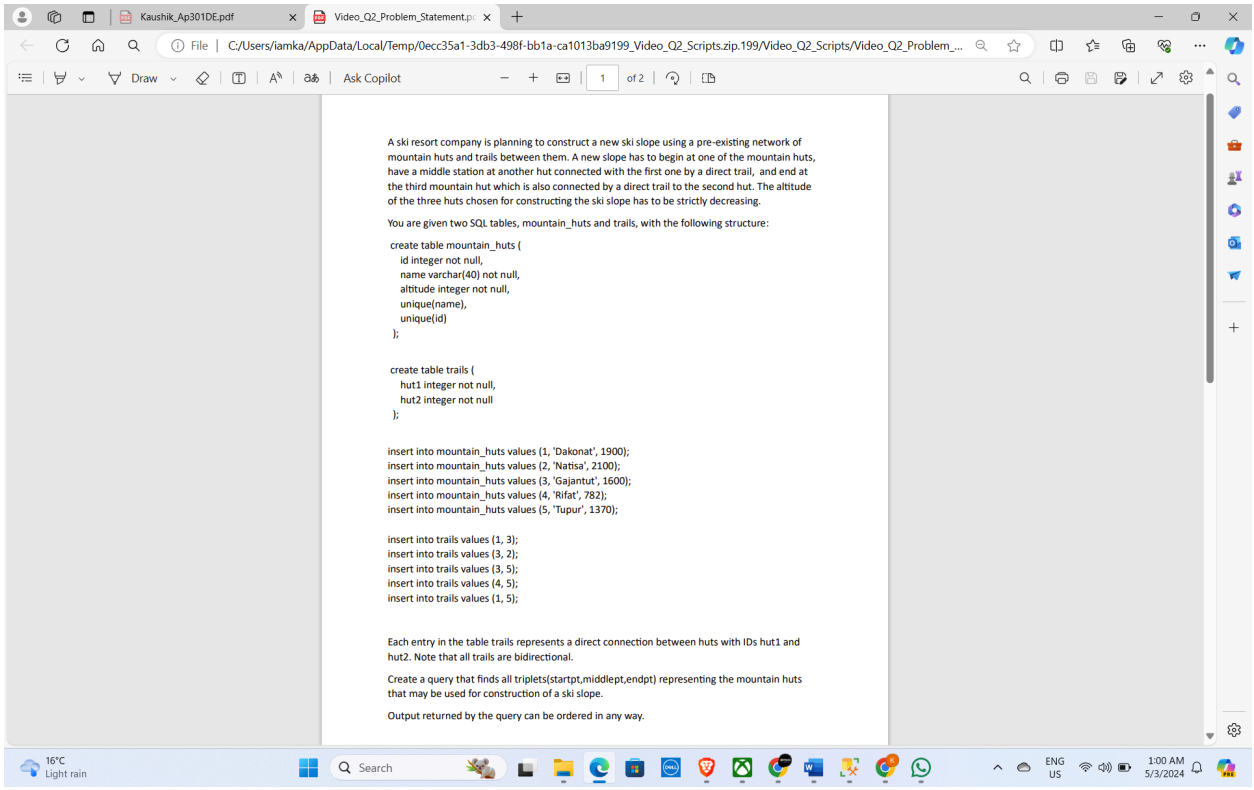
Day87 - May 2nd 2024

1. Started my day as usual
2. Solved a medium problem on leetcode..as it was easily solved did not prepared any solution doc
3. Ended my day by solving a complex SQL question online



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.

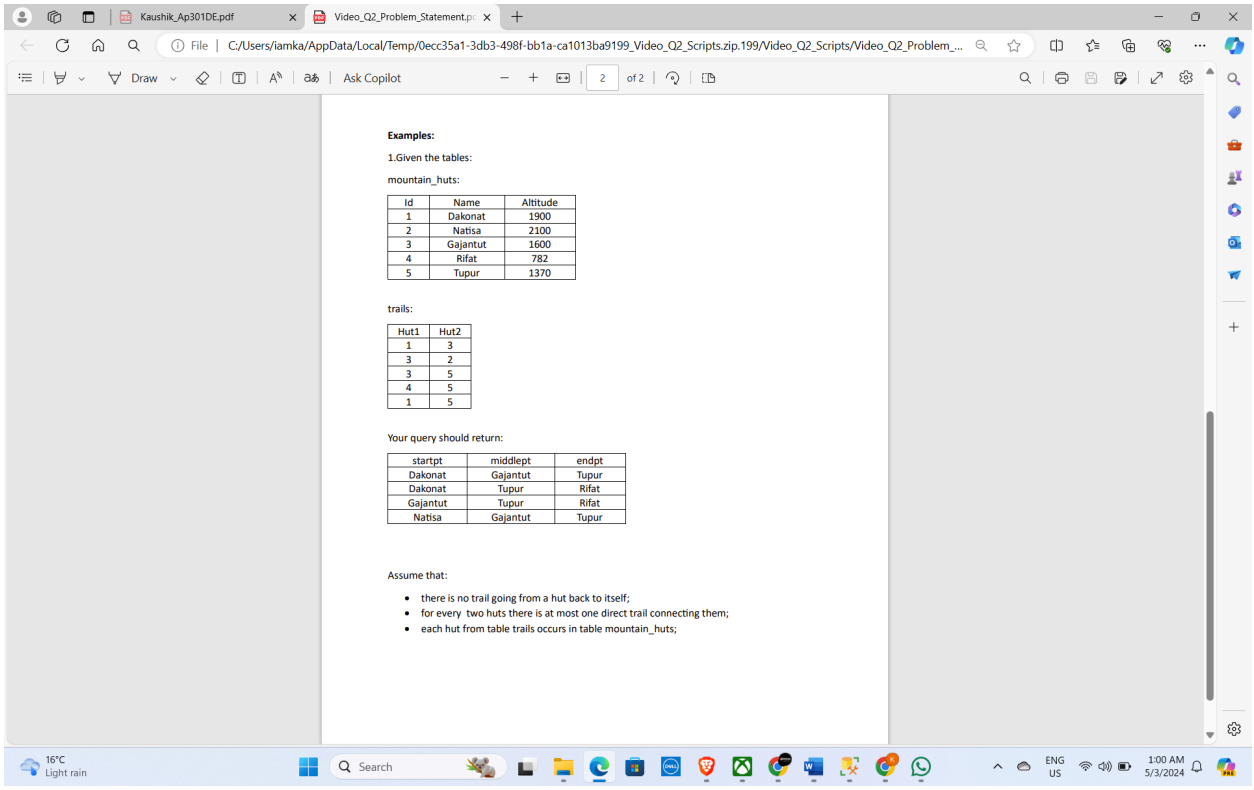
You are given two SQL tables, mountain_huts and trails, with the following structure:

```
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:

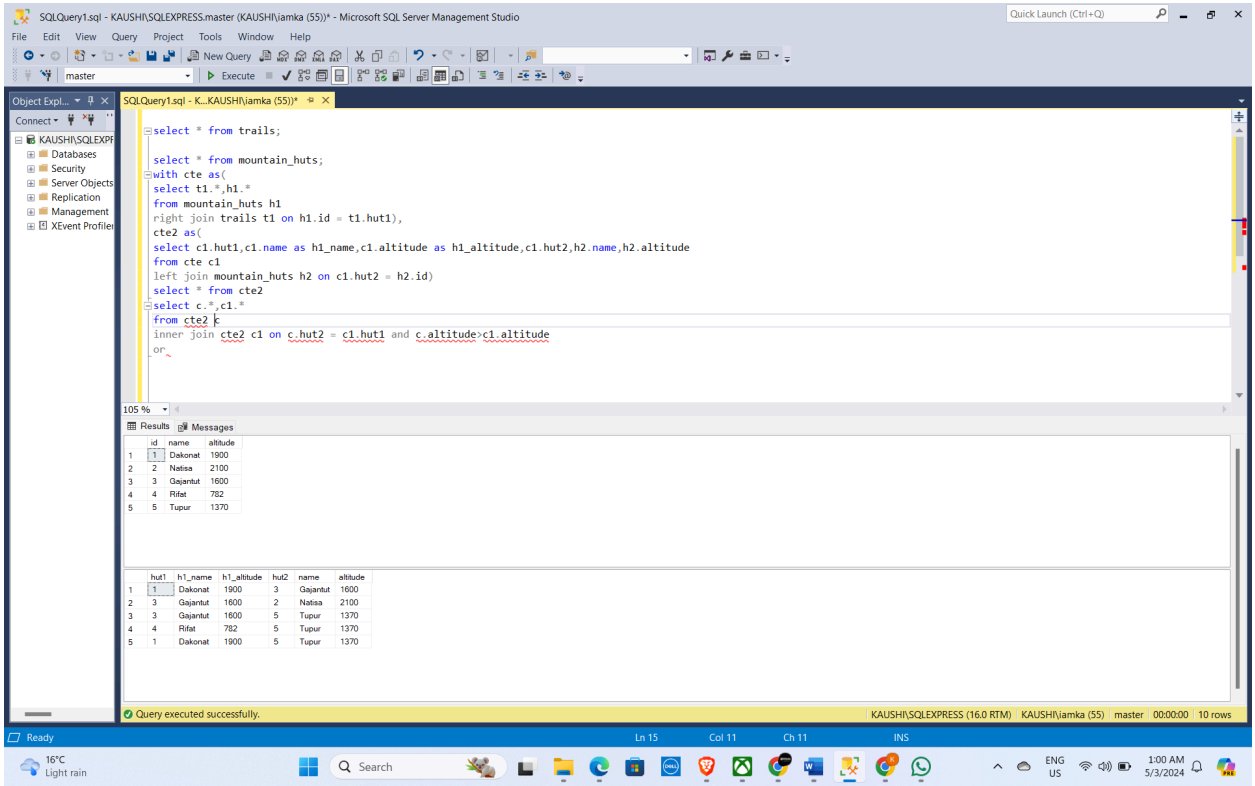
Hut1	Hut2
1	3
3	2
3	5
4	5
1	5

Your query should return:

startpt	middlept	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;



```
select * from trails;

select * from 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
from cte c1
left join mountain_huts h2 on c1.hut2 = h2.id)
select * from cte2
select c.*,c1.*
from cte2 c
inner join cte2 c1 on c.hut2 = c1.hut1 and c.altitude>c1.altitude
or
```

id	name	altitude
1	Dakonat	1900
2	Natisa	2100
3	Gajantut	1600
4	Rifat	782
5	Tupur	1370

hut1	h1_name	h1_altitude	hut2	name	altitude
1	Dakonat	1900	3	Gajantut	1600
3	Gajantut	1600	2	Natisa	2100
3	Gajantut	1600	5	Tupur	1370
4	Rifat	782	5	Tupur	1370
1	Dakonat	1900	5	Tupur	1370

completed it on May2nd mid night(may 3rd)