The total score of a hacker is the sum of their maximum scores for all of the challenges. Write a query to print the hacker\_id, name, and total score of the hackers ordered by the descending score. If more than one hacker achieved the same total score, then sort the result by ascending hacker\_id. Exclude all hackers with a total score of 0  from your result.

select temp.hacker\_id, h.name, sum(max\_score) as total\_score from

(select hacker\_id, challenge\_id, max(score) as max\_score

from submissions group by hacker\_id, challenge\_id) temp

inner join hackers h on

h.hacker\_id = temp.hacker\_id

group by temp.hacker\_id, h.name

having total\_score !=0

order by total\_score desc, temp.hacker\_id;

Given the table schemas below, write a query to print the company\_code, founder name, total number of lead managers, total number of senior managers, total number of managers, and total number of employees. Order your output by ascending company\_code

select c.company\_code, founder, count(distinct lead\_manager\_code),count(distinct senior\_manager\_code),

count(distinct manager\_code), count(distinct employee\_code)

from employee e inner join company c on c.company\_code = e.company\_code

group by c.company\_code, founder

order by c.company\_code;

You are given a table, BST, containing two columns: N and P, where N represents the value of a node in Binary Tree, and P is the parent of N.

select N, case when

P is null then 'Root'

when N in (select distinct P from BST) then "Inner"

else "Leaf" end as note\_type

from BST order by N;

Given the names and grades for each student in a class of  students, store them in a nested list and print the name(s) of any student(s) having the second lowest grade.

if \_\_name\_\_ == '\_\_main\_\_':

    ls = []

    score =[]

    for \_ in range(int(input())):

        name = str(input())

        marks = float(input())

        score.append(marks)

        ls.append(list((name, marks)))

    print(ls)

    sorted\_score = sorted(score)

    print(sorted\_score)

    min2 = sorted\_score[1]

    print([name for name, score in ls if score == float(min2)])

**Query to find the city with maximum and minimum number of charecters**

(Select city, length(city) from station order by length(city), city limit 1)

union

(Select city, length(city) from station order by length(city) desc, city limit 1);