Hacker rank SQL

**Q. Query the list of CITY names from STATION that do not start with vowels. Your result cannot contain duplicates.**

**Ans**. SELECT DISTINCT CITY FROM STATION WHERE CITY NOT LIKE 'A%' AND CITY NOT LIKE 'E%' AND CITY NOT LIKE 'I%' AND CITY NOT LIKE 'O%' AND CITY NOT LIKE 'U%'

//////

**Q. Query the list of CITY names from STATION that do not end with vowels. Your result cannot contain duplicates.**

**Ans**. SELECT DISTINCT CITY FROM STATION WHERE CITY NOT LIKE '%A' AND CITY NOT LIKE '%E' AND CITY NOT LIKE '%I' AND CITY NOT LIKE '%O' AND CITY NOT LIKE '%U'

/////

**Q. Query the list of CITY names from STATION that either do not start with vowels or do not end with vowels. Your result cannot contain duplicates.**

**Ans**. SELECT DISTINCT CITY FROM STATION WHERE (CITY NOT LIKE 'A%' AND CITY NOT LIKE 'E%' AND CITY NOT LIKE 'I%' AND CITY NOT LIKE 'O%' AND CITY NOT LIKE 'U%') OR (CITY NOT LIKE '%A' AND CITY NOT LIKE '%E' AND CITY NOT LIKE '%I' AND CITY NOT LIKE '%O' AND CITY NOT LIKE '%U')

Q. Amber's conglomerate corporation just acquired some new companies. Each of the companies follows this hierarchy: 

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*.

**Note:**

* The tables may contain duplicate records.
* The *company\_code* is string, so the sorting should not be **numeric**. For example, if the *company\_codes* are *C\_1*, *C\_2*, and *C\_10*, then the ascending *company\_codes* will be *C\_1*, *C\_10*, and *C\_2*.

**Ans**.. select company\_code, founder,

(select count(distinct lead\_manager\_code) from Employee where company\_code=c.company\_code)as a,

(select count(distinct senior\_manager\_code) from Employee where company\_code=c.company\_code)as b,

(select count(distinct manager\_code) from Employee where company\_code=c.company\_code)as c,

(select count(distinct employee\_code) from Employee where company\_code=c.company\_code)as d

from Company c

order by company\_code

Q. [Pivot](https://en.wikipedia.org/wiki/Pivot_table) the Occupation column in **OCCUPATIONS** so that each Name is sorted alphabetically and displayed underneath its corresponding Occupation. The output column headers should be Doctor, Professor, Singer, and Actor, respectively.

**Note:** Print **NULL** when there are no more names corresponding to an occupation.

**Input Format**

The **OCCUPATIONS** table is described as follows:

Table

Description automatically generated

Occupation will only contain one of the following values: **Doctor**, **Professor**, **Singer** or **Actor**.

**Ans**. select

min(case when occupation = 'Doctor' then Name end),

min(case when occupation = 'Professor' then Name end),

min(case when occupation = 'Singer' then Name end),

min(case when occupation = 'Actor' then Name end)

from

(select name, occupation, Row\_number() Over (PARTITION BY occupation ORDER BY name)as num from occupations)as d group

by num;

Q. 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*.

Table

Description automatically generated

Write a query to find the node type of *Binary Tree* ordered by the value of the node. Output one of the following for each node:

* *Root*: If node is root node.
* *Leaf*: If node is leaf node.
* *Inner*: If node is neither root nor leaf node.

**Ans**. select n,

case

when p is Null then 'Root'

when n in (select p from BST) then 'Inner'

else

'Leaf'

end

FROM BST order by n ;