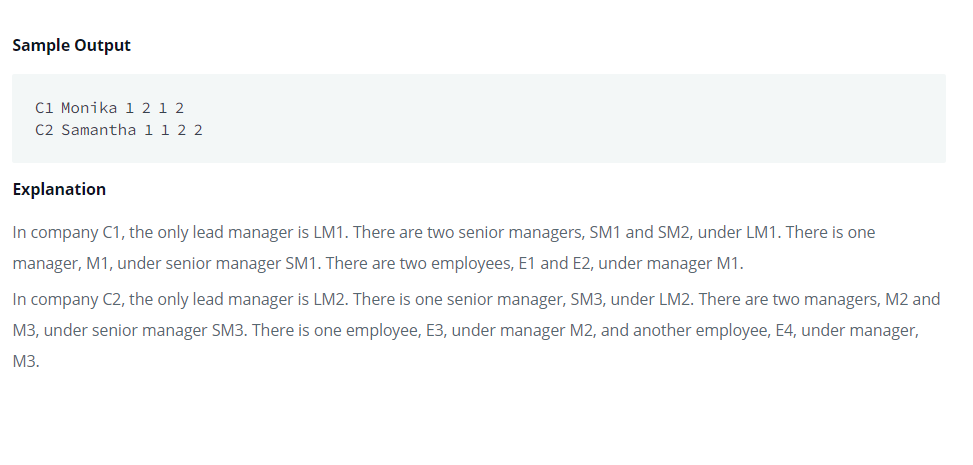
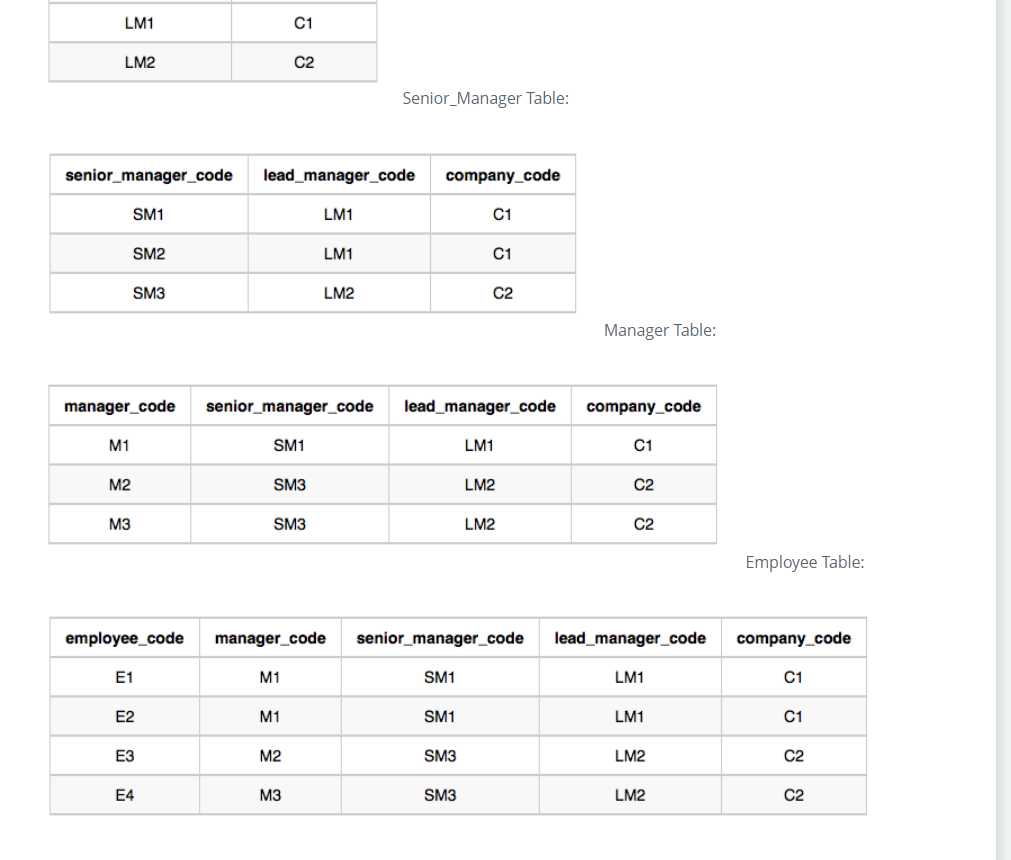
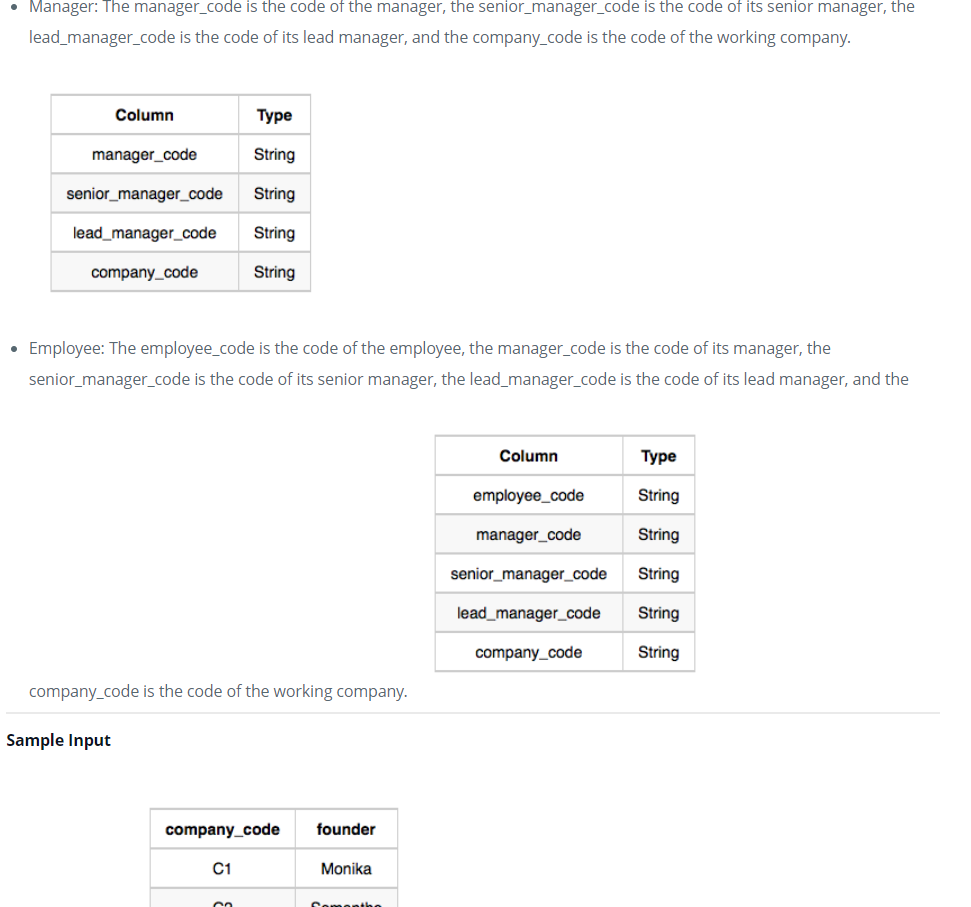
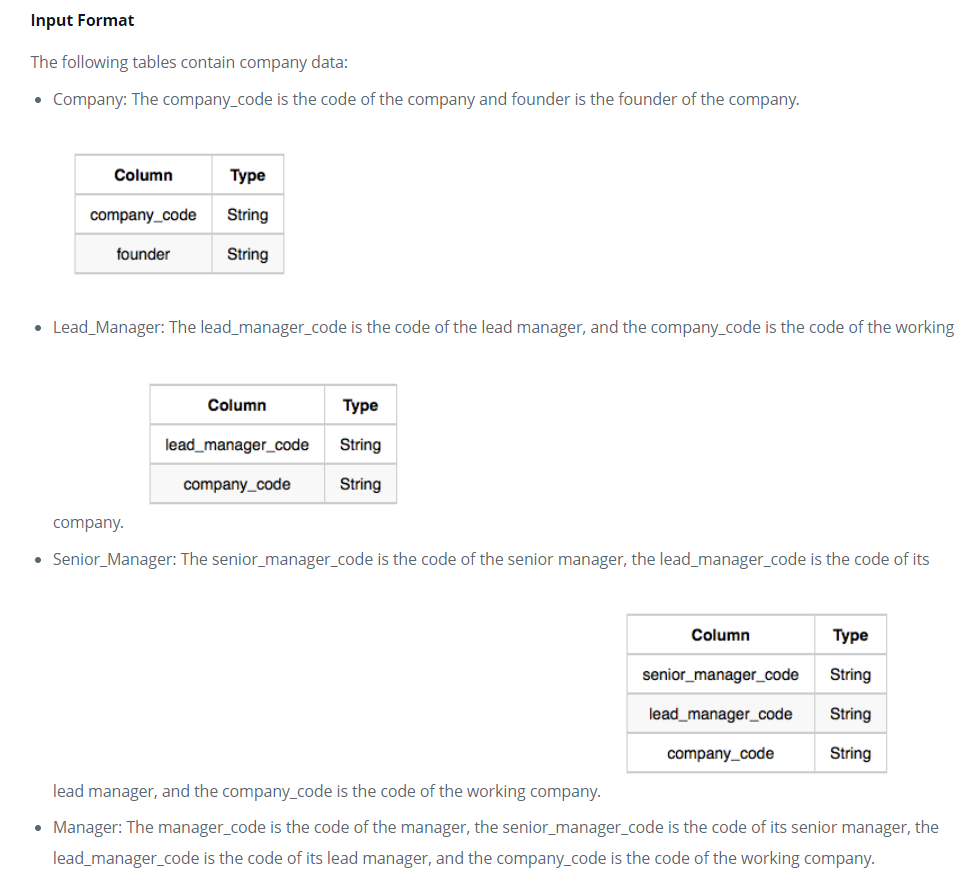
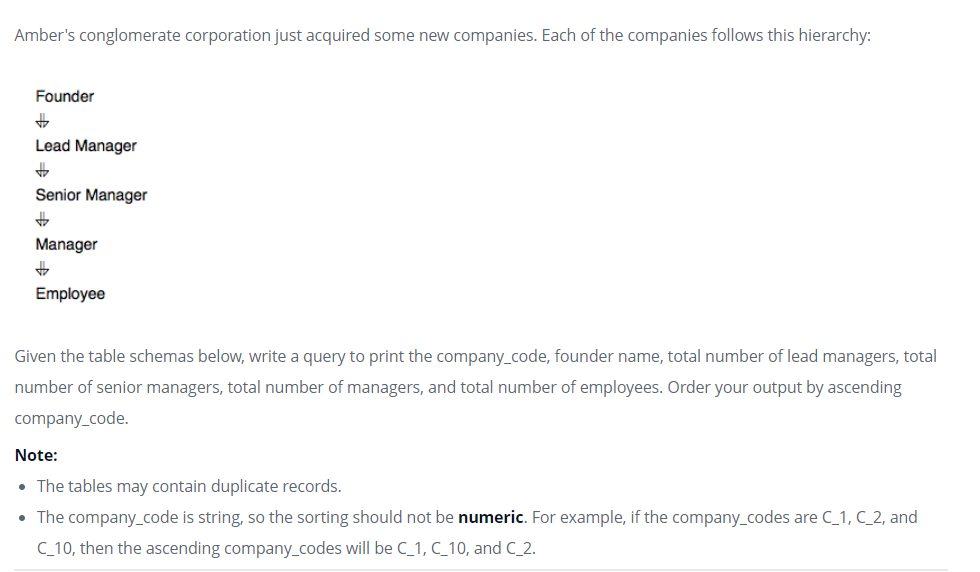
Top of Form

Bottom of Form

Top of Form

Bottom of Form

|  |  |
| --- | --- |
|  | 1. 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. |
|  | 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 |
|  |  |
|  |  |
|  | SELECT CASE |
|  | WHEN P IS NULL THEN CONCAT(N, 'Root') |
|  | WHEN N IN(SELECT DISTINCT P FROM BST) THEN CONCAT(N, 'inner') |
|  | ELSE CONCAT(N, 'Leaf') |
|  | END |
|  | FROM BST |
|  | ORDER BY N ASC; |

2.

SELECT C.company\_code,

C.founder,

COUNT(distinct L.lead\_manager\_code),

COUNT(distinct S.senior\_manager\_code),

COUNT(distinct M.manager\_code),

COUNT(distinct E.employee\_code)

FROM Company as C,

Lead\_Manager as L,

Senior\_Manager as S,

Manager as M,

Employee as E

WHERE E.manager\_code = M.manager\_code

AND M.senior\_manager\_code = S.senior\_manager\_code

AND L.lead\_manager\_code = S.lead\_manager\_code

AND C.company\_code = L.company\_code

GROUP BY C.company\_code, C.founder

ORDER BY C.company\_code