

## Lecture Notes – QueriesOnRecursiveCTEs

### Section 1: Lecture Summary

The lecture demonstrates writing a **recursive CTE** to display employee hierarchy levels and paths in an organization, using a special table EMP2 with EmployeeID, FirstName, JobTitle, MgrID, and Dept. It shows top manager Rago at level 1, subordinates like Tan and Mira at level 2, and further employees like Niraj at level 3, building a tree structure with concatenated paths like "Rago -> Tan -> Niraj".

### Section 2: Key Concepts and Explanations

**Recursive CTE** starts with WITH RECURSIVE, defining a CTE name like EMP\_hierarchy. It has two parts unioned: anchor query (seed) selects top-level employees where **MgrID IS NULL**, providing initial EmployeeID, FirstName, level=1, and path as CAST(FirstName AS CHAR(200)). Recursive part self-joins the table (e) with the CTE (eh) on e.MgrID = eh.EmployeeID, selecting e.EmployeeID, e.FirstName, eh.level + 1 as level, and CONCAT(eh.path, ' -> ', e.FirstName) as path. Main query selects from the CTE to output EmployeeID, FirstName, level, path. Useful for hierarchical data like employee trees or product categories/subcategories.

### Section 3: Example Code and Use Cases

```
WITH RECURSIVE EMP_hierarchy AS (  
    SELECT EmployeeID, FirstName, 1 AS level,  
           CAST(FirstName AS CHAR(200)) AS path  
    FROM EMP2  
    WHERE MgrID IS NULL  
  
    UNION ALL  
  
    SELECT e.EmployeeID, e.FirstName, eh.level + 1 AS level,  
           CONCAT(eh.path, ' -> ', e.FirstName) AS path  
    FROM EMP2 e  
    JOIN EMP_hierarchy eh ON e.MgrID = eh.EmployeeID  
)  
SELECT EmployeeID, FirstName, level, path  
FROM EMP_hierarchy;
```

This query on EMP2 (temporarily added to companyDB) outputs hierarchy: level 1 Rago (path: Rago), level 2 Tan/Mira (paths: Rago -> Tan, Rago -> Mira), level 3 Niraj (path: Rago -

> Tan -> Niraj). Adaptable to companyDB Employees by adding MgrID column for self-joins on EmpID = MgrID.

#### Section 4: Key Takeaways

- Recursive CTE requires anchor (starting point, e.g., MgrID IS NULL) union all recursive part (self-join on MgrID = prior EmployeeID).
- Use level increment (eh.level + 1) and CONCAT for paths with separators like '-> '.
- Cast strings to CHAR(200) for path growth; ideal for hierarchies in employees or categories.
- Table must exist (e.g., create EMP2 in companyDB to test).