

Lecture Notes – IntroductionToCTE

Section 1: Lecture Summary

The lecture introduces **CTE** (Common Table Expression) as a named subquery that creates a temporary, lightweight virtual table on the fly, not stored in the database. It explains the syntax using the **WITH** clause to name a query, which can then be referenced in the main query's **FROM** clause like a table, improving readability over inline subqueries. CTEs are reusable within the main query and their columns can be used in **SELECT**, **WHERE**, **GROUP BY**, **HAVING**, and **ORDER BY** clauses. Examples demonstrate selecting employee details and filtering by salary using the **Employees** table.

Section 2: Key Concepts and Explanations

CTE is a named subquery defined with **WITH CTE_name AS (subquery)**, executed temporarily to produce results used as a table in the main query. Unlike unnamed subqueries, CTEs separate complex logic for better readability and allow reuse in the main query. The subquery can include **SELECT**, **FROM**, **JOIN**, **WHERE**, **GROUP BY**, and other clauses. In the main query, the CTE acts as a table in the **FROM** clause, with its columns available for filtering (e.g., **WHERE salary > 90000**) or other operations. CTEs are session-specific and discarded after execution.

Section 3: Example Code and Use Cases

```
WITH EMP AS (  
    SELECT EmpID, FirstName, Salary  
    FROM Employees  
)  
SELECT *  
FROM EMP;
```

This CTE selects **EmpID**, **FirstName**, and **Salary** from **Employees**, then the main query retrieves all columns from the **EMP** CTE.

```
WITH high_pay AS (  
    SELECT EmpID, FirstName, Salary  
    FROM Employees  
    WHERE Salary > 90000  
)
```

```
SELECT *  
FROM high_pay;
```

This CTE filters **Employees** for **Salary > 90000**, then the main query shows **EmpID**, **FirstName**, and **Salary** for high earners.

Section 4: Key Takeaways

- **CTE** uses **WITH** to name a subquery, creating a temporary table for the main query.
- Improves query readability and reusability compared to inline subqueries.
- CTE columns work in **SELECT**, **FROM**, **WHERE**, **GROUP BY**, **HAVING**, **ORDER BY**.
- Defined query results are lightweight and not persisted in the database.