**Common Table Expression:**

**The query is designed to find the employee(s) in each department with the highest salary. It uses a Common Table Expression (CTE) defined by the WITH clause to simplify and structure the query. Here's a step-by-step explanation of how the query works, along with dummy tables and their execution process:**

**Concepts and Execution Order**

1. **WITH Clause (DeptMaxSal):**
   * **This defines a CTE named DeptMaxSal.**
   * **It calculates the maximum salary for each department (max(salary)), grouping by departmentId.**
   * **This part executes first and creates a temporary result set that can be referenced in the main query.**
2. **Main Query:**
   * **The main query uses the DeptMaxSal CTE to identify the employee(s) in each department whose salary matches the maximum salary.**
   * **It joins the DeptMaxSal CTE with the Employee table on departmentId and salary.**
   * **It further joins the Department table to include the department's name in the final output.**
3. **Execution Steps:**
   * **Step 1: Execute the CTE to create the temporary table DeptMaxSal.**
   * **Step 2: Use the temporary table in the main query.**
   * **Step 3: Perform the joins to filter and combine data as needed.**
   * **Step 4: Produce the final output.**

**Dummy Data**

**Employee Table**

| **id** | **name** | **departmentId** | **salary** |
| --- | --- | --- | --- |
| **1** | **Alice** | **101** | **5000** |
| **2** | **Bob** | **101** | **7000** |
| **3** | **Charlie** | **102** | **6000** |
| **4** | **David** | **102** | **6000** |
| **5** | **Eve** | **103** | **8000** |

**Department Table**

| **id** | **name** |
| --- | --- |
| **101** | **HR** |
| **102** | **Engineering** |
| **103** | **Marketing** |

**Execution of Query**

1. **WITH Clause Execution: The DeptMaxSal CTE calculates the maximum salary for each department:**
2. **SELECT departmentId, max(salary) as "salary"**
3. **FROM Employee**
4. **GROUP BY departmentId;**

**Result (DeptMaxSal):**

| **departmentId** | **salary** |
| --- | --- |
| **101** | **7000** |
| **102** | **6000** |
| **103** | **8000** |

1. **Main Query Execution:**
   * **First Join (Employee with DeptMaxSal):**
   * **SELECT e.\***
   * **FROM Employee e**
   * **JOIN DeptMaxSal dms**
   * **ON dms.departmentId = e.departmentId AND e.salary = dms.salary;**

**This matches employees in the Employee table whose departmentId and salary match the DeptMaxSal table.**

**Intermediate Result:**

| **id** | **name** | **departmentId** | **salary** |
| --- | --- | --- | --- |
| **2** | **Bob** | **101** | **7000** |
| **3** | **Charlie** | **102** | **6000** |
| **4** | **David** | **102** | **6000** |
| **5** | **Eve** | **103** | **8000** |

* + **Second Join (With Department Table):**
  + **SELECT d.name as "Department", e.name as "Employee", e.salary as "Salary"**
  + **FROM ...**

**This adds the department name from the Department table.**

**Final Result:**

| **Department** | **Employee** | **Salary** |
| --- | --- | --- |
| **HR** | **Bob** | **7000** |
| **Engineering** | **Charlie** | **6000** |
| **Engineering** | **David** | **6000** |
| **Marketing** | **Eve** | **8000** |

**Explanation of WITH and AS**

1. **WITH Clause:**
   * **Used to define a temporary result set that simplifies complex queries.**
   * **Improves readability and modularity.**
   * **Example:**
   * **WITH TempTable AS (**
   * **SELECT ...**
   * **)**
   * **SELECT ...**
   * **FROM TempTable;**
2. **AS Keyword:**
   * **Used to alias columns or tables for readability or convenience.**
   * **Example:**
   * **SELECT max(salary) AS "MaxSalary"**
   * **FROM Employee;**
   * **Here, "MaxSalary" is an alias for max(salary).**

**This breakdown shows how the query works step by step, demonstrating the concepts of CTEs, joins, and aliasing.**

**­­**