# 8. Common Table Expressions (CTEs) and Recursive Querie

## Objective

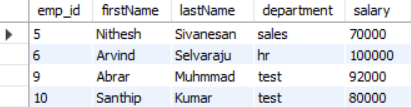
To observe the concepts of CTEs.

## Understanding

* CTEs are the temporary result set that exist in the scope of statement.
* Helps to breakdown complex queries and easy to understand by reading it.

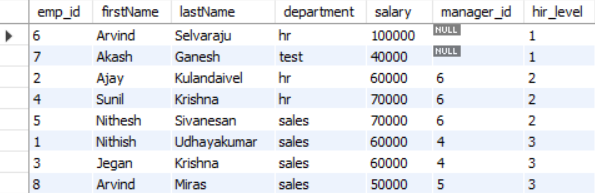
## Queries

1. ***WITH avgSalaries AS (***
2. ***SELECT department, AVG(salary) AS avg\_salary***
3. ***FROM employee***
4. ***GROUP BY department***
5. ***)***
6. ***SELECT e.\* FROM employee e***
7. ***JOIN avgSalaries a ON e.department = a.department***
8. ***WHERE e.salary > a.avg\_salary;***



* In CTE, I wrote the aggregation query to get the average salary of each department.
* And used it in the statement and joined the CTE and the employee table on department to get people who are getting more salary than the average salary of his/her department.

1. ***WITH recursive emp\_hir AS (***
2. ***select \*, 1 AS hir\_level***
3. ***from employee***
4. ***where manager\_id is null***
6. ***UNION ALL***
8. ***select e.\*,eh.hir\_level+1 AS hir\_level***
9. ***from employee e***
10. ***join emp\_hir eh on e.manager\_id = eh.emp\_id***
11. ***)***
12. ***select \* from emp\_hir order by hir\_level;***



* ***Arvind(6) – CEO*** 
  + ***Ajay(2) – Manager***
  + ***Sunil(4) – Manager***
    - ***Nithish(1) – Employee***
    - ***Jegan(3) - Employee***
  + ***Nithesh(5) – Manager***
    - ***Arvind(8) – Employee***
* This is the actual hierarchy of the company so wrote an **recursive CTE** to return the rows in the correct order.
* **Base query** - select \*, 1 AS hir\_level from employee where manager\_id is null;
  + Initial query, It will create column (**hir\_level**) and assign 1 value to the rows which having **null** in manager\_id.
* **Recursive query** - select e.\*,eh.hir\_level+1 AS hir\_level from employee e join emp\_hir eh on e.manager\_id = eh.emp\_id;
  + In this query , It will search for the employee whose manager\_id is already in the **CTE**(emp\_hir).
    - **employee e join emp\_hir eh on e.manager\_id = eh.emp\_id**;
  + If the employee id exists, It will create a column(**hir\_level**) and assign the value as of the manager\_id’s value by incrementing one.
* **Final query** - select \* from emp\_hir order by hir\_level;
  + It will select the details in emp\_hir CTE.