


Views in database

↳ Views are like virtual tables

↳ they don't occupy any spaces in the memory

employee → original (Admin access)

	emp-id	emp-name	emp-mobile	Dept	Salary
HR					

HR
Finance
Employee Services

- ↳ Create a virtual table
- ↳ and include logic in the view definition
- ↳ (emp-id, emp-name, department) $\rightarrow V1 \rightarrow HR$
- ↳ (emp-id, emp-name, salary) $\rightarrow V2 \rightarrow Finance$

↳ Hiding the logic from end user

* Union & UNION ALL in SQL

$$A = \{1, 2, 3, 5, 8, 9\}$$

$$B = \{2, 8, 10, 4, 6\}$$

→ no duplicates

$$A \cup B = \{1, 2, 3, 4, 5, 6, 8, 9, 10\}$$

$$A \underline{\cup} B = \{1, 2, 2, 3, 4, 5, 6, 8, 8, 9, 10\}$$

→ included duplicates

CTE (Common Table Expressions)

↳ Known as With Clause

↳ Two types of With Clause

↳ Iterative

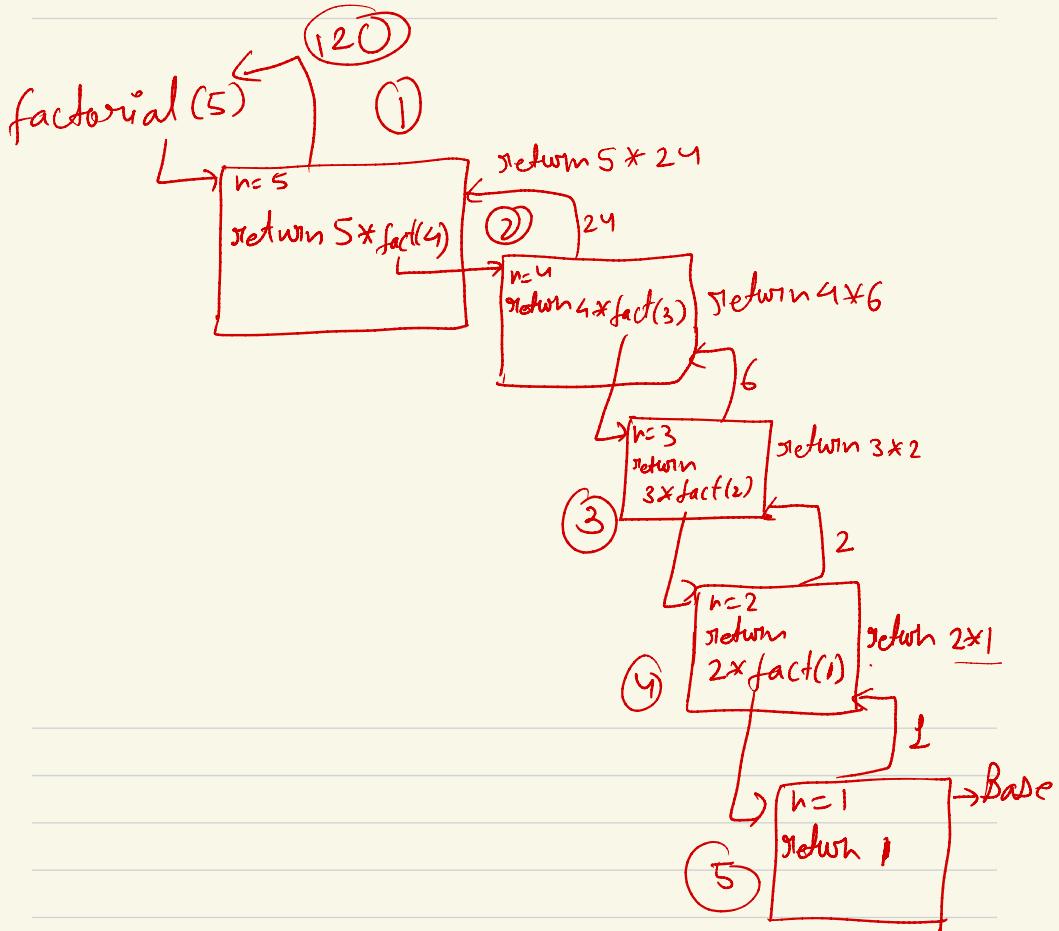
↳ Recursive

* Write a simple program to calculate factorial of a number?

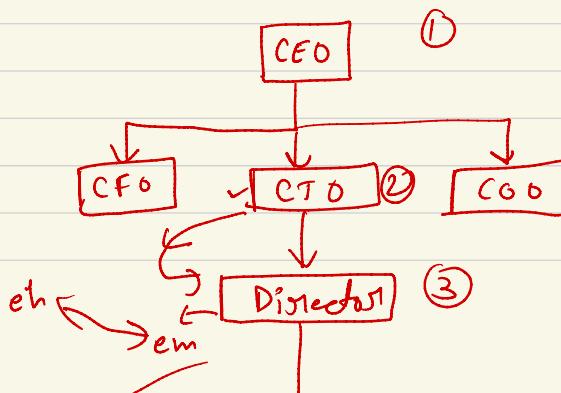
{ def factorial(n):

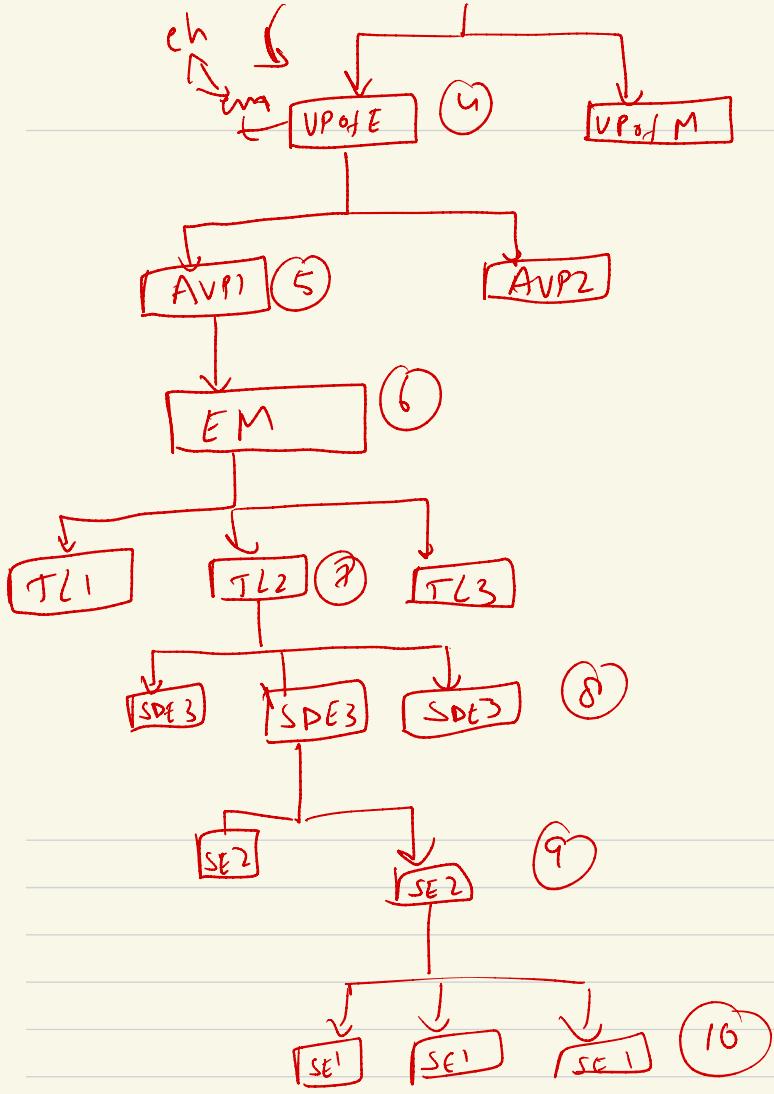
 if n == 0 or n == 1:] → Base Condition
 return 1]

return $n \times \text{factorial}(n-1)$ } \Rightarrow Recursive call



organization Chart





with recursive gen-numbers

{ select 1 as n } → Anchor Query Base Condition
 union recursive call
 select n+1 from gen-numbers where n < 10

select x from gen-numbers;

Select 1] \rightarrow gen-number

Union

Select 2 from gen-number where $1 < n = 1$

Union

Select 3 from gen-num where $2 < 10$

Union

Select 4 from gen-num where $3 < 10$

⋮

Select 10 from gen-num where $9 < 10$

⋮

Select 11 from gen-num here $10 < 10$ *