System Design

(Basics)

continued...

Video - 15

Facebook ⎤
Instagram ⎦ → codestorywithMIK

Twitter → cswithMIK

✆ → codestorywithMIK

Normalization & Denormalization in RDBMS

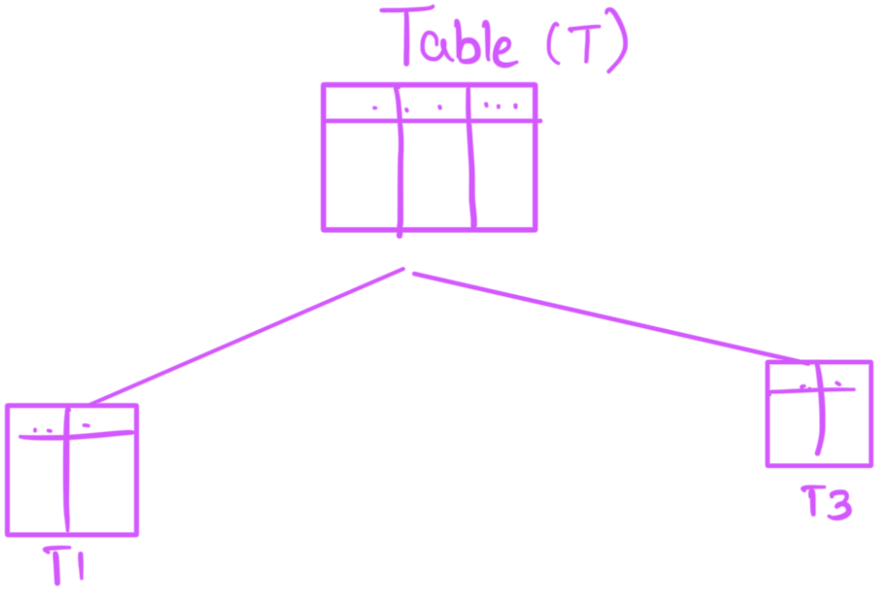
Normalization :-

- Breaking down large tables into smaller, related tables and establishing relationships between them.

- Minimizes redundancy , simplify .

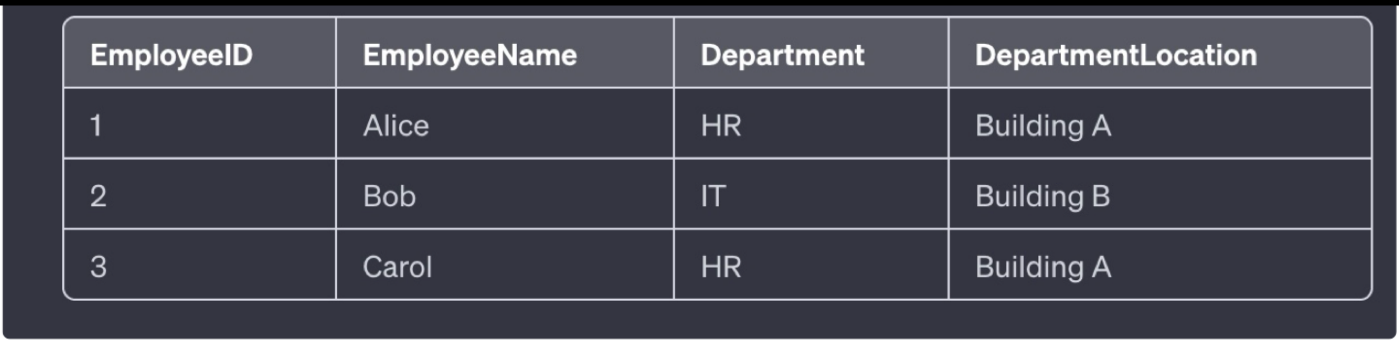
- Don't do over normalization.

" In simple language. "



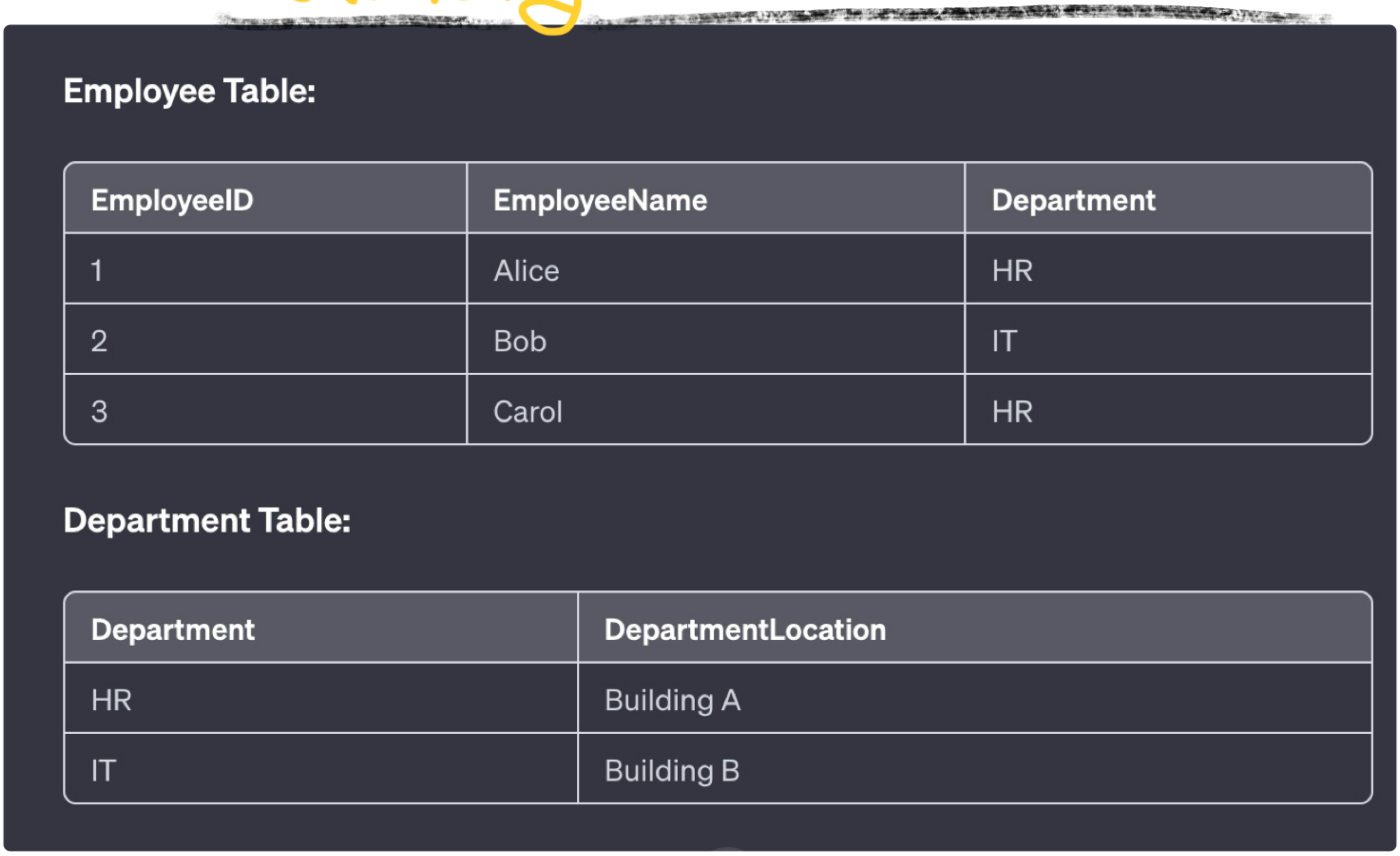
Example :-

Employee Table :-



EmployeeID	EmployeeName	Department	DepartmentLocation
1	Alice	HR	Building A
2	Bob	IT	Building B
3	Carol	HR	Building A

Normalize the table above:-



Employee Table:

EmployeeID	EmployeeName	Department
1	Alice	HR
2	Bob	IT
3	Carol	HR

Department Table:

Department	DepartmentLocation
HR	Building A
IT	Building B

Qn:- Find the Department location of Empl ID:2

Ans:- EID, EName, Department Location

Employee का DepartLocation = Department का DepartLocation

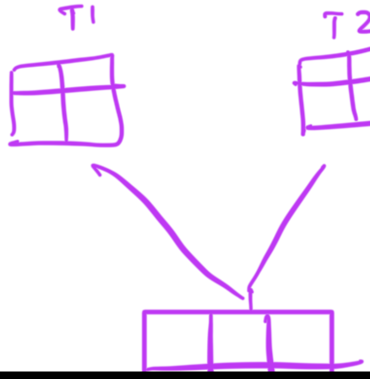
EmployeeID = 2

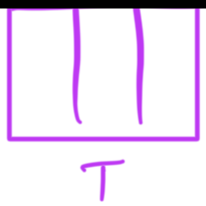
```
Select E.EmloyeeID, E.EmloyeeName,
D.DepartmentLocation
From Employee E
JOIN Department D on
E.Department = D.Department
where E.EmployeeID =2;
```

Denormalization :-

- Deliberately introducing redundancy into a database table.

- Opposite of Normalization.





But Why???

(अरे दिक्कत क्या थी ???)

☺ Pros of Denormalization :-

1. Improved query performance. (No joins)
2. Simplified query.
3. Reduce joins.
4. Read intensive scenarios → Best

☹ Cons of Denormalization :-

1. Redundancy → memory waste.
2. Inconsistent Data (Suppose you forgot to update

DepartmentLocation of HR in some rows)

③ Upadate/write operations → Extremely slow

Because you will have to update in multiple places (redundant)