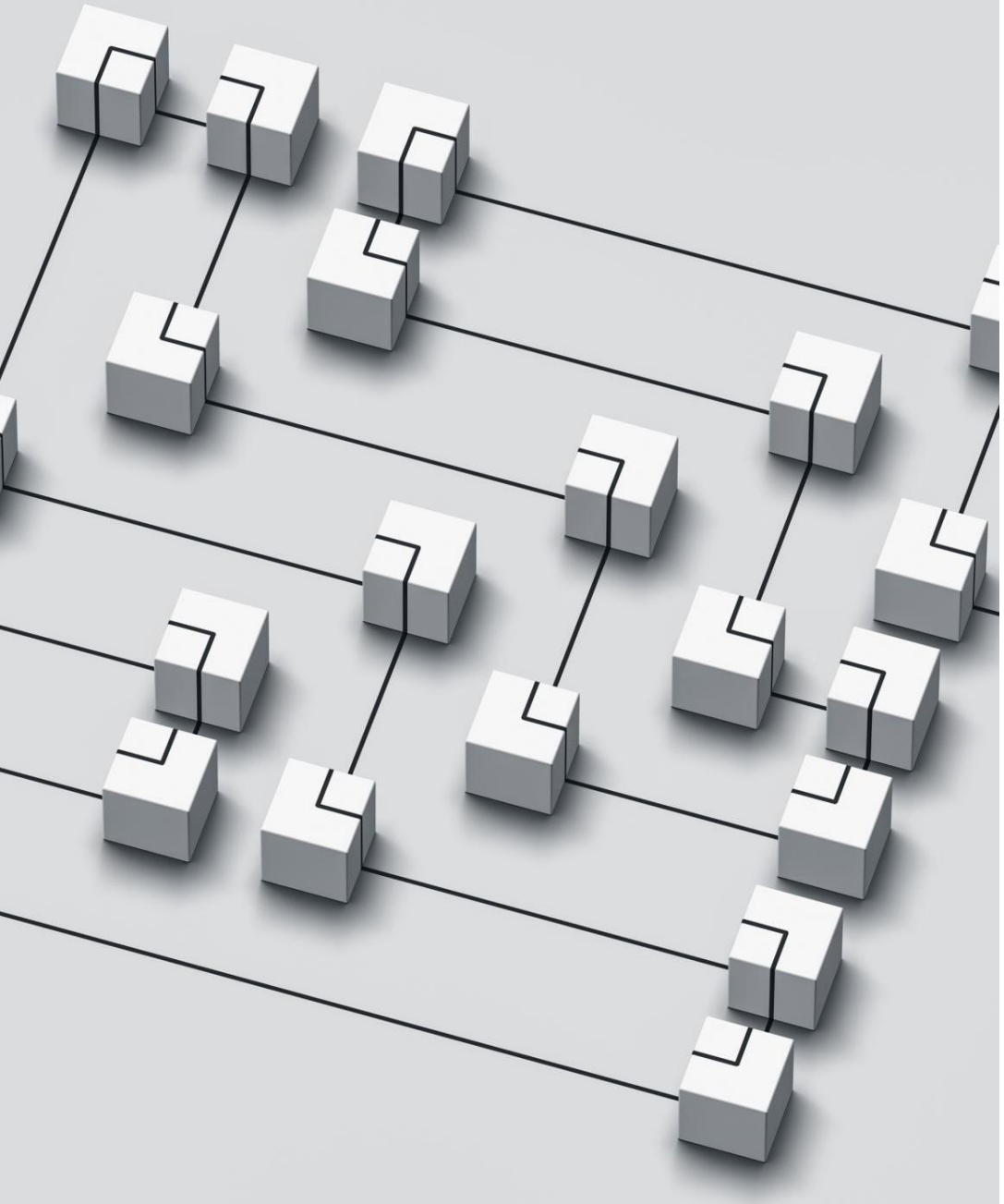
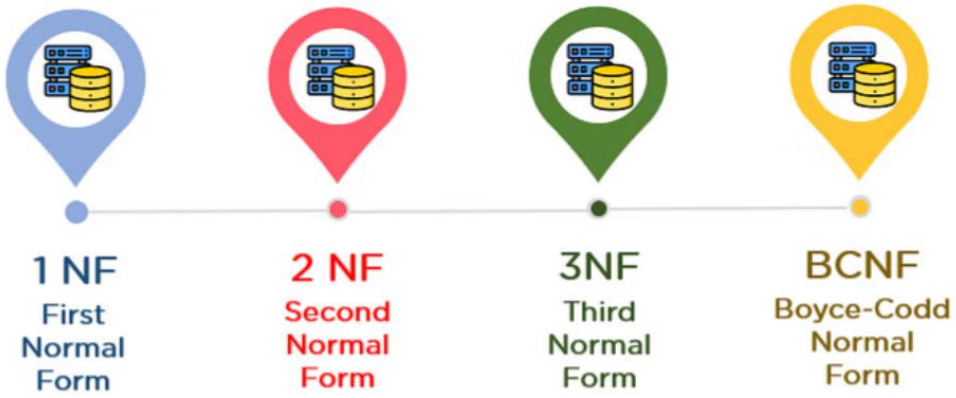
Simplifying Databasing ~Normalization



Problem Statement

- **Current Database Design:** The EMPLOYEE table has multiple fields related to salary, designation, department, and other attributes, all in one large table.
- **Issues Identified:**
- **Data Redundancy:** Repeating details for department and salary of each employee.
- **Update Anomalies:** Changes in department information or salary structure could require updates across multiple rows, leading to inconsistency.
- **Insert Anomalies:** Inserting new records for departments or salary details requires duplicating information, which could lead to errors.
- **Delete Anomalies:** Deleting an employee record might result in the loss of critical department information or salary records.

Solution ? Normalization



Why?

- **Reduces Data Redundancy:**

  We can decompose the entire EMPLOYEE table into 3 smaller tables:

  1. Employee, 2.Salary, 3.Department.

- **Prevents Update Anomalies:** Updating salary or department information in one place **ensures consistency**, without needing to change multiple records.
- **Simplifies Data Management:** Smaller tables allow **separation of concerns** to manage the huge data into smaller segments at a time.
- **Improves Query Efficiency:** Querying is more efficient due to the **reduced time complexity** of querying entire rows even when only some columns information is needed.
- **Minimizes Anomalies:** Now the client works on smaller tables rather than one single table, so lesser chances of failure.

Step 1: 1NF Form – First Normalization Form

- **Original Table:**

EMPLOYEE (EID, NAME, AGE, SALARY, BASIC, HRA, TA, DA, PF, DESIGNATION, DEPARTMENT, HOD)

1NF requires that:

- Each column must contain atomic (indivisible) values.
- Each column must contain values of a single type.
- Each record must be unique.

-> The Table Doesn't need any changes since there are no columns with multiple values.

Step 2: Second Normal Form (2NF)

- **Current Table After 1 NF Form:**
- **EMPLOYEE** (EID, NAME, AGE, SALARY, BASIC, HRA, TA, DA, PF, DESIGNATION, DEPARTMENT, HOD)

2NF requires:

- The table must be in 1NF.
- There should be no partial dependency

  We assume **EID** (Employee ID) is the primary key. Now let's check for partial dependencies:

- Attributes like **SALARY, BASIC, HRA, TA, DA, PF** depend on the **EID** and not on just part of the primary key. These are dependent on the employee, so they are fine.
- However, **DESIGNATION, DEPARTMENT, HOD** are related to the **employee's job and department**, which may be treated as a separate entity.

  So we need to split the table into two:

- EMPLOYEE **(EID, NAME, AGE, SALARY, BASIC, HRA, TA, DA, PF)**
- DEPARTMENT **(DEPARTMENT, DESIGNATION, HOD)**

  This removes the dependency of **DESIGNATION, DEPARTMENT, HOD** on the employee attributes and places them in a separate table.

Step 3: Third Normal (3NF)

- **Current Table after 2NF Form:**
- EMPLOYEE **(EID, NAME, AGE, SALARY, BASIC, HRA, TA, DA, PF)**
- DEPARTMENT **(DEPARTMENT, DESIGNATION, HOD)**
- **3NF** requires:

The table must be in 2NF.

There should be no transitive dependency, meaning that non-key attributes should not depend on other non-key attributes.

Issue in current table:-

HOD should be only depended on department, but it is dependent

on **EID as well ,** hence creating a transitive dependency .

Also Salary's components doesn't exclusively depend on EID rather , they depend on basic salary not EID logically so make a separate table for Salary as well

HOD- > EID -> Department

- EMPLOYEE **(EID, NAME, AGE, SALARY, BASIC, HRA, TA, DA, PF, DEPARTMENT_ID)**
- DEPARTMENT **(DEPARTMENT_ID, DEPARTMENT, DESIGNATION, HOD)**
- SALARY **EID, BASIC, HRA, TA, DA, PF, SALARY)**

Step 5: Boyce-Codd Normal Form (BCNF)

- Current Table after 3NF Form:
- **EMPLOYEE (EID, NAME, AGE, SALARY, DEPARTMENT)**
- **DEPARTMENT (DEPARTMENT_ID, DEPARTMENT, DESIGNATION, HOD)**
- **SALARY (EID, BASIC, HRA, TA, DA, PF, SALARY)**
- **BCNF** Requires:
- For every functional dependency, the left-hand side must be a superkey.

At this point, both tables (EMPLOYEE ,DEPARTMENT and SALARY) are already in BCNF since every non-trivial functional dependency involves a superkey. We don't need any further decomposition.