**The Concept of Normalization**

**What is Normalization in Database?**

Normalization in a database is a process used to organize the structure of data to reduce redundancy and improve data integrity. The main goals of normalization are to eliminate duplicate data and ensure data dependencies are logical and efficient.

**When to normalize?**

* **To Eliminate Redundancy**: When you notice that the same piece of data is being repeated multiple times across the database.
* **To Ensure Data Integrity**: When it's crucial that updates to data are propagated correctly and consistently throughout the database.
* **To Simplify Data Maintenance**: When you want to make your database easier to maintain and avoid complications during insert, update, or delete operations.
* **When Designing a New Database**: It’s a best practice to start with a normalized design to ensure a solid foundation.
* **When Restructuring or Migrating Data**: If you’re working on a project to clean up or migrate data, normalization helps to set a strong, clean structure.

**Types of Normalization forms.**

1. First Normal Form (1NF)
2. Second Normal Form (2NF)
3. Third Normal Form (3NF)
4. Boyce-Codd Normal Form (BCNF)
5. Fourth Normal Form (4NF)
6. Fifth Normal Form (5NF).

Normalization is a database design process aimed at organizing data to reduce redundancy and improve data integrity. It involves breaking a database into smaller, related tables while following specific rules called normal forms (NF). Below are the normal forms with an example of unnormalized data and the normalization process.

**Unnormalized Data (UNF)**

**Example Table:** A table that contains redundant and poorly organized data.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **OrderID** | **CustomerName** | **ItemsOrdered** | **TotalAmount** | **DeliveryAddress** |
| 1 | John Smith | Pen, Notebook | 15.00 | 123 Elm St, NY |
| 2 | Jane Doe | Marker, Eraser, Pen | 20.00 | 456 Oak St, LA |

* **Issues:**
  + Multiple values in the ItemsOrdered column (violates atomicity).
  + Repeating groups (different items in a single cell).

**First Normal Form (1NF)**

**Rules:**

* Ensure each column contains atomic (indivisible) values.
* Remove repeating groups.

**Normalized Table (1NF):** Split multiple items into individual rows.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **OrderID** | **CustomerName** | **Item** | **TotalAmount** | **DeliveryAddress** |
| 1 | John Smith | Pen | 15.00 | 123 Elm St, NY |
| 1 | John Smith | Notebook | 15.00 | 123 Elm St, NY |
| 2 | Jane Doe | Marker | 20.00 | 456 Oak St, LA |
| 2 | Jane Doe | Eraser | 20.00 | 456 Oak St, LA |
| 2 | Jane Doe | Pen | 20.00 | 456 Oak St, LA |

**Second Normal Form (2NF)**

**Rules:**

* Achieve 1NF.
* Remove partial dependencies (columns depend only on part of the primary key).

**Issues in 1NF:**

* CustomerName and DeliveryAddress depend on OrderID, not the combination of OrderID and Item.

**Normalized Tables (2NF):**

1. **Orders Table:**

|  |  |  |  |
| --- | --- | --- | --- |
| **OrderID** | **CustomerName** | **TotalAmount** | **DeliveryAddress** |
| 1 | John Smith | 15.00 | 123 Elm St, NY |
| 2 | Jane Doe | 20.00 | 456 Oak St, LA |

1. **OrderDetails Table:**

|  |  |
| --- | --- |
| **OrderID** | **Item** |
| 1 | Pen |
| 1 | Notebook |
| 2 | Marker |
| 2 | Eraser |
| 2 | Pen |

**Third Normal Form (3NF)**

**Rules:**

* Achieve 2NF.
* Remove transitive dependencies (non-key columns depend on other non-key columns).

**Issues in 2NF:**

* CustomerName and DeliveryAddress are related to each other, forming a transitive dependency.

**Normalized Tables (3NF):**

1. **Orders Table:**

|  |  |  |
| --- | --- | --- |
| **OrderID** | **CustomerID** | **TotalAmount** |
| 1 | 101 | 15.00 |
| 2 | 102 | 20.00 |

1. **Customers Table:**

|  |  |  |
| --- | --- | --- |
| **CustomerID** | **CustomerName** | **DeliveryAddress** |
| **101** | **John Smith** | **123 Elm St, NY** |
| **102** | **Jane Doe** | **456 Oak St, LA** |

1. **OrderDetails Table:**

|  |  |
| --- | --- |
| **OrderID** | **Item** |
| **1** | **Pen** |
| **1** | **Notebook** |
| **2** | **Marker** |
| **2** | **Eraser** |
| **2** | **Pen** |

**Boyce-Codd Normal Form (BCNF)**

**Rules:**

* Every determinant must be a candidate key (special case of 3NF).

If the 3NF tables have no anomalies, they already satisfy BCNF. In this case, there are no violations, so the structure remains unchanged.

**Higher Normal Forms (4NF and 5NF)**

**Fourth Normal Form (4NF)**

**Rule:**

* Remove **multivalued dependencies**, where two attributes are independent but tied to the same key.

**Example of 3NF/BCNF Table with Multivalued Dependency:**

| **OrderID** | **Item** | **DeliveryMode** |
| --- | --- | --- |
| 1 | Pen | Pickup |
| 1 | Pen | Courier |
| 1 | Notebook | Pickup |
| 1 | Notebook | Courier |

**Issue:**

* Item and DeliveryMode are independent but stored together, causing redundancy.

**Normalized Tables (4NF):**

1. **OrderItems Table:**

| **OrderID** | **Item** |
| --- | --- |
| 1 | Pen |
| 1 | Notebook |

1. **OrderDeliveryModes Table:**

| **OrderID** | **DeliveryMode** |
| --- | --- |
| 1 | Pickup |
| 1 | Courier |

**Fifth Normal Form (5NF)**

**Rule:**

* Eliminate **join dependencies**, ensuring no data redundancy occurs when tables are split and rejoined.

**Example of 4NF Table with Join Dependency:**

| **ProjectID** | **EmployeeID** | **Skill** |
| --- | --- | --- |
| 1 | E101 | Design |
| 1 | E102 | Design |
| 1 | E101 | Testing |
| 1 | E102 | Testing |

**Issue:**

* ProjectID, EmployeeID, and Skill form overlapping relationships that could be split.

**Normalized Tables (5NF):**

1. **Projects Table:**

| **ProjectID** | **EmployeeID** |
| --- | --- |
| 1 | E101 |
| 1 | E102 |

1. **EmployeeSkills Table:**

| **EmployeeID** | **Skill** |
| --- | --- |
| E101 | Design |
| E101 | Testing |
| E102 | Design |
| E102 | Testing |

1. **ProjectSkills Table:**

| **ProjectID** | **Skill** |
| --- | --- |
| 1 | Design |
| 1 | Testing |