**Normalization In DBMS**

**What is Normalization**

**In Database Management Systems (DBMS)** Normalization is the process of organizing data in a database. It includes creating tables and establishing relationships between those tables according to rules designed both to protect the data and to make the database. Link: <https://learn.microsoft.com/en-us/office/troubleshoot/access/database-normalization-description>

Normalization is a database design technique that organizes tables in a manner that reduces redundancy (unless) and dependency of data and to avoid **insertion, update & deletion anomaly**. Normalization is multi-step process that puts the data in tubular form & remove duplicate data from relation table.

**Why we need normalization in DBMS**

* **Eliminate redundant data**: Remove duplicate data, Redundancy can cause anomalies that make it difficult for database administrators to maintain the database.
* **Improve data integrity**: Resolve data conflicts and avoid unwanted data connections and dependencies
* **Reduce errors:** Normalization can minimize data modification errors.
* **Optimize storage space**: Reduce the delay when new data is introduced
* **Improve workflow**: Normalization can increase the efficiency of database operations.
* **Increase security:** Normalization can help improve data integrity and consistency

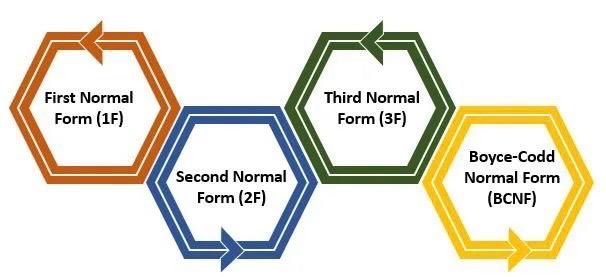
**There are three types of anomalies:**

* **Insertion Anomaly**
* **Deletion Anomaly**
* **Updatation Anomaly**
* **Insertion Anomaly:** If there is a new row inserted in the table and it creates the inconsistency in the table then it is called the insertion anomaly. Link: https://www.javatpoint.com/anomalies-in-dbms
* **Deletion Anomaly:** If we delete some rows from the table and if any other information or data which is required is also deleted from the database Link: <https://www.javatpoint.com/anomalies-in-dbms>
* **Updatation Anomaly:** The update anomaly is when an update of a single data value requires multiple rows of data to be updated. Link: <https://www.javatpoint.com/dbms-normalizatio>

**Normalization Types**

**The most commonly used normal forms:**

* First normal forms (1NF)
* Second normal forms (2NF)
* Third normal forms (3NF)
* Boyce & Codd normal forms (BCNF)



* **First Normal Form (1NF):**

This is the most basic level of normalization. In 1NF, each table cell should contain only a single value, and each column should have a unique name. The first normal form helps to eliminate duplicate data and simplify queries. Link: [*https://www.geeksforgeeks.org/normal-forms-in-dbms/*](https://www.geeksforgeeks.org/normal-forms-in-dbms/)

**Example:**

|  |  |  |
| --- | --- | --- |
| **Customer\_ID** | **Name** | **Mobile\_Number** |
| 1 | Sima | 915101/821020 |
| 2 | Harry | 8546501 |

**1NF Table:**

|  |  |  |
| --- | --- | --- |
| **Customer\_ID** | **Name** | **Mobile\_Number** |
| 1 | Sima | 915101 |
| 1 | Sima | 821020 |
| 2 | Harry | 8546501 |

* **Second Normal Form (2NF):**

2NF eliminates redundant data by requiring that each non-key attribute be dependent on the primary key. This means that each column should be directly related to the primary key, and not to other columns. Link: https://www.geeksforgeeks.org/normal-forms-in-dbms/

**Example:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Employee\_ID** | **Department\_ID** | **Employee\_Name** | **Department** |
| 1 | 101 | Smriti | Human Resource |
| 2 | 102 | Jock | Marketing |

**2NF Table:**

|  |  |
| --- | --- |
| **Employee\_ID** | **Employee\_Name** |
| 1 | Smriti |
| 2 | Jock |

|  |  |
| --- | --- |
| **Department\_ID** | **Department** |
| 101 | Human Resource |
| 102 | Marketing |

* **Third Normal Form (3NF):**

3NF builds on 2NF by requiring that all non-key attributes are independent of each other. This means that each column should be directly related to the primary key, and not to any other columns in the same table. Link: [*https://www.geeksforgeeks.org/normal-forms-in-dbms/*](https://www.geeksforgeeks.org/normal-forms-in-dbms/)

**Example:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Student\_ID** | **Course\_ID** | **Student\_Name** | **Age** |
| 1 | 1001 | Nancy | 35 |
| 2 | 1002 | Jerry | 40 |

**3NF Table:**

|  |  |  |
| --- | --- | --- |
| **Student\_ID** | **Student\_Name** | **Age** |
| 1 | Nancy | 35 |
| 2 | Jerry | 40 |

|  |  |  |
| --- | --- | --- |
| **Course\_ID** | **Student\_Name** | **Age** |
| 1001 | Nancy | 35 |
| 1002 | Jerry | 40 |

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

BCNF is a stricter form of 3NF that ensures that each determinant in a table is a candidate key. In other words, BCNF ensures that each non-key attribute is dependent only on the candidate key.

Link: https://www.javatpoint.com/dbms-third-normal-form

**Example:**

|  |  |  |
| --- | --- | --- |
| **Product\_ID** | **Product\_Name** | **Price** |
| 1 | laptop | 75000 |
| 2 | Mobile | 50000 |

**BCNF Table:**

|  |  |
| --- | --- |
| **Product\_Name** | **Price** |
| Laptop | 75000 |
| Mobile | 50,0000 |

|  |  |
| --- | --- |
| **Product\_Name** | **Price** |
| laptop | 75000 |
| Mobile | 50000 |

* **Fifth Normal Form (5NF):**
* The Fifth Normal Form(5NF) of database normalization, also know as the project- join normal from (PJNF)
* 5NF is the highest level of normalization
* It is involves decomposing a table into smaller tables to remove data redundancy
* Improve data integrity.

**Advantages of Normalization**

* Normalization reduces data redundancy.
* Improved database organization in general
* Improved data inconsistency
* Greater flexibility in database design
* Maintains relational integrity Link: https://er.yuvayana.org/normalization-in-dbms/

**Disadvantages of Normalization**

* You cannot start building the database before knowing what the user needs.
* The performance degrades when normalizing the relations to higher normal forms.
* It is very time-consuming and difficult to normalize relations of a higher degree.
* Careless decomposition may lead to a bad database design, leading to serious problems.