## Normalization

* Normalization is the process of organizing the data in the database.
* Normalization is used to minimize the redundancy from a relation or set of relations. It is also used to eliminate undesirable characteristics like Insertion, Update, and Deletion Anomalies.
* Normalization divides the larger table into smaller and links them using relationships.
* The normal form is used to reduce redundancy from the database table.
* Used to **eliminate** Insertion, update and. deletion **anomalies.**

**Data modification anomalies can be categorized into three types:**

* **Insertion Anomaly:** Insertion Anomaly refers to when one cannot insert a new tuple into a relationship due to lack of data.
* **Deletion Anomaly:** The delete anomaly refers to the situation where the deletion of data results in the unintended loss of some other important data.
* **Updatation Anomaly:** The update anomaly is when an update of a single data value requires multiple rows of data to be updated.

Above definition is get from : https://www.javatpoint.com/dbms-normalization

**Types of Normalisation in DBMS**

Normalization and its types in DBMS - DBMS normalization is achieved through a series of stages known as normal forms. Each normal form has a set of rules and requirements that a database table must meet to be considered normalized. Let's take a closer look at each of these normalization types in DBMS:

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

* The primary objective of achieving the first normal form is to eradicate duplicate data entries and streamline query processes.
* This means that each attribute in a table should hold a single value and cannot contain multiple values or sets.
* Each cell in the table should contain only a single value. Columns should have unique names.

Above definition is get from: https://www.almabetter.com/bytes/articles/normalization-in-dbms

**Example:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Student\_ID | Name | ContactNo | Course | Marks | Grade |
| 4103 | Biplab Parida | 9568292999,9876543212 | SQL | 85 | A |
| 4143 | Debasish Sahoo | 8835624682,9876578765 | UNIX | 72 | B |
| 4007 | Rakesh Kumar | 8763456476 | DBMS | 68 | C |
| 4103 | Biplab Parida | 9568292999,9876543112 | C++ | 92 | A+ |
| 4143 | Debasish Sahoo | 8835624682,9876578765 | SQL | 95 | A+ |

**1NF:** Each attribute should contain only 1 value and atomic values which cannot be split further.

In above table contactNo attribute contain more than 1 value and Name attribute can be split.

## 

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Student\_ID | First Name | Last Name | Mobile | Home | Course | Marks | Grade |
| 4103 | Biplab | Parida | 9568292999 | 9876543212 | SQL | 85 | A |
| 4143 | Debasish | Sahoo | 8835624682 | 9876578765 | UNIX | 72 | B |
| 4007 | Rakesh | Kumar | 8763456476 | 8763456476 | DBMS | 68 | C |
| 4103 | Biplab | Parida | 9568292999 | 9876543112 | C++ | 92 | A+ |
| 4143 | Debashis | Sahoo | 8835624682 | 9876578765 | SQL | 95 | A+ |

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

* In the 2NF, relational must be in 1NF.
* In the second normal form, all non-key attributes are fully functional dependent on the primary key.
* No Partialy Dependency

Example:

By making above table in Seond Normal Form where as it is in 1NF but Partial Dependency.

Student table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Student\_ID | First Name | Last Name | Mobile | Home |
| 4103 | Biplab | Parida | 9568292999 | 9876543212 |
| 4143 | Debasish | Sahoo | 8835624682 | 9876578765 |
| 4007 | Rakesh | Kumar | 8763456476 | 8763456476 |
| 4103 | Biplab | Parida | 9568292999 | 9876543112 |
| 4143 | Debashis | Sahoo | 8835624682 | 9876578765 |

Course:

|  |  |  |  |
| --- | --- | --- | --- |
| Student\_ID | Course | Marks | Grade |
| 4103 | SQL | 85 | A |
| 4143 | UNIX | 72 | B |
| 4007 | DBMS | 68 | C |
| 4103 | C++ | 92 | A+ |
| 4143 | SQL | 95 | A+ |

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

* **A relation will be in 3NF if it is in 2NF** and not contain any transitive partial dependency.
* 3NF is used to reduce the data duplication. It is also used to achieve the data integrity.
* If there is no **transitive dependency** for non-prime attributes, then the relation must be in third normal form.

IN the give table grade is indireactly dependent on student\_ID where as marks is direactly dependent on student\_id so the relation between student\_id and grade is transitive, so we have to remove this transitive dependency.

|  |  |  |  |
| --- | --- | --- | --- |
| Student\_ID | Course | Marks | Grade |
| 4103 | SQL | 85 | A |
| 4143 | UNIX | 72 | B |
| 4007 | DBMS | 68 | C |
| 4103 | C++ | 92 | A+ |
| 4143 | SQL | 95 | A+ |

|  |  |  |
| --- | --- | --- |
| Student\_ID | Course | Marks |
| 4103 | SQL | 85 |
| 4143 | UNIX | 72 |
| 4007 | DBMS | 68 |
| 4103 | C++ | 92 |

## 

|  |  |
| --- | --- |
| Marks | Grade |
| 85 | A |
| 72 | B |
| 68 | C |
| 95 | A+ |

# **Boyce Codd normal form (BCNF):**

* BCNF is the advance version of 3NF. It is stricter than 3NF.
* Table must be in 3NF.
* every determinant (attribute that determines another attribute) should be a candidate key.

Example:

|  |  |  |
| --- | --- | --- |
| Course | Marks | Grade |
| SQL | 85 | A |
| UNIX | 72 | B |
| DBMS | 68 | C |
| C++ | 92 | A+ |
| SQL | 95 | A+ |

## Here, "Grade" depends only on "Marks," which is a candidate key. This satisfies BCNF.