# Normalization

**What is Normalization?**

* Normalization is used to Organize the data in the database.
* Normalization divides the larger table into smaller and links them using relationships.
* The normal form is used to reduce the repetition of the same data in multiple places within a database.

**Types of Normal Forms:**

First Normal Form (1NF):

* 1NF contains only Individual value
* Each cell in a table should hold a single piece of data, not a list or set of values.
* First normal form disallows the multi-valued attribute, composite attribute, and their combinations.

Example:

|  |  |  |
| --- | --- | --- |
| Student\_Id | Name | subject |
| 1 | Rupesh | Maths,chemistry |
| 2 | Vedant | History , English |
| 3 | Tolesh | Science , Marathi,Hindi |
| 4 | Vishant | Physics |

Convert To 1NF

|  |  |  |
| --- | --- | --- |
| Student\_id | Name | Subject |
| 1 | Rupesh | Maths |
| 1 | Rupesh | chemistry |
| 2 | Vedant | History |
| 2 | Vedant | English |
| 3 | Tolesh | Science |
| 3 | Tolesh | Marathi |
| 3 | Tolesh | Hindi |
| 4 | Vishant | Physics |

Second Normal Form(2NF):

* IN 2NF , The table must already be in 1NF.
* In a table with a composite primary key, each non-key column must depend on the entire composite key.

Example:

|  |  |  |
| --- | --- | --- |
| Student\_id | Student\_name | Book\_id |
| 1 | John | 101 |
| 1 | John | 102 |
| 2 | Lauren | 101 |
| 2 | Lauren | 103 |

Convert To 2NF:

|  |  |
| --- | --- |
| Student\_id | Student\_name |
| 1 | John |
| 1 | Lauren |

|  |  |
| --- | --- |
| Student\_id | Book\_id |
| 1 | 101 |
| 1 | 102 |
| 1 | 101 |
| 1 | 103 |

Third Normal Form(3NF):

* The table must already be in 2NF
* 3NF is used to reduce the data duplication.
* 3NF is used to achieve the data accuracy,completeness,consistency of an organization’s data.

Example:

|  |  |  |  |
| --- | --- | --- | --- |
| Employee ID | Employee Name | Department ID | Department Name |
| 1 | JOHN | D01 | HR |
| 2 | LAUREN | D02 | FINANCE |
| 3 | SRI | D03 | HR |

Convert to 3NF:

Employees:

|  |  |  |
| --- | --- | --- |
| Employee ID | Employee Name | Department ID |
| 1 | JOHN | D01 |
| 2 | LAUREN | D02 |
| 3 | SRI | D03 |

Department:

|  |  |
| --- | --- |
| Department ID | Department Name |
| D01 | HR |
| D02 | FINANCE |

Forth Normal Form(4NF):

* There should be no multi-valued dependency.
* For a dependency A → B, if for a single value of A, multiple values of B exists, then the relation will be a multi-valued dependency.

Example:

|  |  |  |
| --- | --- | --- |
| Student\_id | Course | Activity |
| 1 | Math | Basketball |
| 1 | Science | cricket |
| 2 | physice | volleyball |
| 3 | Chemistry | cricket |

Convert To 4NF:

Student\_course:

|  |  |
| --- | --- |
| Student\_id | Course |
| 1 | Math |
| 1 | Science |
| 2 | physice |
| 3 | Chemistry |

Student\_Activity:

|  |  |
| --- | --- |
| Student\_id | Activity |
| 1 | Basketball |
| 1 | cricket |
| 2 | volleyball |
| 3 | cricket |

Fifth Normal Form(5NF):

* In 5NF we can able to break the table as many Tables as possible to avoid the repetition of the same data in multiple places within a database.
* 5NF not contains any join dependency.
* 5NF is also known as Project-join normal form.

Example:

|  |  |  |
| --- | --- | --- |
| Employee\_id | Project\_id | location |
| 1 | A | New York |
| 1 | A | San Francisco |
| 1 | B | New York |
| 2 | A | New York |
| 2 | B | San Francisco |

Convert to 5NF:

**Employee Projects**

|  |  |
| --- | --- |
| Employee\_id | Project\_id |
| 1 | A |
| 1 | B |
| 2 | A |
| 2 | B |

Project Locations

|  |  |
| --- | --- |
| Project\_id | location |
| A | New York |
| A | San Francisco |
| B | New York |
| B | San Francisco |

Employee Locations

|  |  |
| --- | --- |
| Employee\_id | location |
| 1 | New York |
| 1 | San Francisco |
| 2 | New York |
| 2 | San Francisco |

Reference Link: <https://www.javatpoint.com/dbms-fifth-normal-form>