**Normalization**

Normalization is a technique of minimizing redundancy and undesirable characteristics from a table or a set of tables.

It reduces dependency and redundancy of data, thus eliminating insert, update and delete anomalies.

It divides larger tables into smaller tables that are linked through relationship. Normal forms are used to perform normalization of tables.

The normal forms are described as follows:-

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

A relation is in first normal form (1NF) if it satisfies the following condition:-

* If every attribute of the relation contains only a single value i.e. all the attributes of the relation have atomic (or indivisible) domains.

This can be explained with the example as follows:-

The table **Student** contains phone number as multivalued attribute i.e. a student can have more than one phone number. So, it is not in First Normal Form.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STUDENT\_ID | STUDENT\_NAME | STUDENT\_PHONE | STUDENT\_STATE | STUDENT\_CONTRY |
| 1 | ABC | 9998887770, 7555455961 | RAJASTHAN | INDIA |
| 2 | XYZ | 2223331045 | GUJRAT | INDIA |
|  |  |  |  |  |

**Student table**

To convert this table into First Normal Form, the following steps are performed-

* The attributes containing multiple values in individual tables are eliminated by creating a separate table for each set of such attribute.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STUDENT\_ID | STUDENT\_NAME | STUDENT\_PHONE | STUDENT\_STATE | STUDENT\_CONTRY |
| 1 | ABC | 9998887770 | RAJASTHAN | INDIA |
| 1 | ABC | 7555455961 | RAJASTHAN | INDIA |
| 2 | XYZ | 2223331045 | GUJRAT | INDIA |

**Student table (1NF)**

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

A relation is in first normal form (2NF) if it satisfies the following conditions:-

* If it is in First Normal Form (1NF).
* It should not contain any partial dependency i.e. no non-prime attribute (attributes which are not part of any candidate key) is dependent on any proper subset of candidate key of the relation where,

Partial Dependency is where an attribute in a table depends on only a part of the primary key and not on the whole key.

. This can be explained with the example as follows:-

For example, if there is another table **Course** in the database, for storing name of courses opted by a student,

|  |  |  |
| --- | --- | --- |
| STUDENT\_ID | COURSE\_ID | COURSE\_NAME |
| 1 | C1 | C++ |
| 1 | C2 | DBMS |
| 2 | C2 | DBMS |

**Course table**

The primary key for this table is the combination of two attributes i.e. STUDENT\_ID and COURSE\_ID but COURSE\_NAME is only dependent on COURSE\_ID and not STUDENT\_

ID. So, COURSE\_NAME has nothing to do with STUDENT\_ID in this table.

So, here exists partial dependency and the table is not in second normal form. To convert this table into second normal form, the relation should be decomposed into 2 tables to be in 2NF as follows:-

|  |  |
| --- | --- |
| STUDENT\_ID | COURSE\_ID |
| 1 | C1 |
| 1 | C2 |
| 2 | C2 |

**Course\_Enrolled**

|  |  |
| --- | --- |
| COURSE\_ID | COURSE\_NAME |
| C1 | C++ |
| C2 | DBMS |

**Course\_Information**

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

A relation is in third normal form (3NF) if it satisfies the following condition:-

* If it is in 2nd Normal Form.
* If there is no transitive dependency for non-prime attributes where ,

Transitive Dependency is when a non-prime attribute depends on other non-prime attributes rather than depending upon attributes of primary key. For example, if attribute B is functionally dependent in attribute A, A->B and C is functionally dependent on B, B->C, then A->C is called transitive dependency.

This can be explained with the example as follows:-

In the student table, STUDENT\_COUNTRY is transitively dependent on STUDENT\_ID as STUDENT\_COUNTRY is functionally dependent on STUDENT\_STATE and STUDENT\_STATE is functionally dependent on STUDENT\_ID and is not in 3rd Normal form.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STUDENT\_ID | STUDENT\_NAME | STUDENT\_PHONE | STUDENT\_STATE | STUDENT\_CONTRY |
| 1 | ABC | 9998887770 | RAJASTHAN | INDIA |
| 1 | ABC | 7555455961 | RAJASTHAN | INDIA |
| 2 | XYZ | 2223331045 | GUJRAT | INDIA |

To convert this table in 3NF we decompose this table into two as follows;

|  |  |  |  |
| --- | --- | --- | --- |
| STUDENT\_ID | STUDENT\_NAME | STUDENT\_PHONE | STUDENT\_STATE |
| 1 | ABC | 9998887770 | RAJASTHAN |
| 1 | ABC | 7555455961 | RAJASTHAN |
| 2 | XYZ | 2223331045 | GUJRAT |

|  |  |
| --- | --- |
| STUDENT\_STATE | STUDENT\_CONTRY |
| RAJASTHAN | INDIA |
| GUJRAT | INDIA |

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

A relation is in Boyce-Codd Normal Form (BCNF) if it satisfies the following conditions:-

* If it is in Third Normal Form (3NF).
* For any functional dependency A->B, A should be a **super key** (A super key is a set of attributes within a table whose values can be used to uniquely identify a tuple)

This form deals with certain type of anomaly that is not handled by 3NF. A 3NF table which does not have multiple overlapping candidate keys is said to be in BCNF.

This can be explained with the example as follows:-

|  |  |
| --- | --- |
| STUDENT\_ID | COURSE\_ID |
| 1 | C1 |
| 1 | C2 |
| 2 | C2 |