Normalization

Normalization Is the process of organizing the data in the database.

Normalization is used to reduce the redundancy from the database.

Normalization divides the larger table into the smaller table and links them using relationship.

It is also used to eliminate the undesirable characteristics like insertion, update and deletion anomalies.

1st Normal Form (1NF)

**Rules**: In this normal form we tackle the problem of atomicity. It means a single cell cannot hold multiple values.

If a table contains a composite or multivalued attribute, it violates the First Normal Form.

|  |  |  |  |
| --- | --- | --- | --- |
| Employee ID | Employee Name | Phone Number | Salary |
| 1E8976 | Alex | 5642685465,  8646744666 | 80000 |
| 1E5678 | Mangraj | 2456675676 | 55000 |
| 1E5432 | David | 7777745646 | 70000 |
| 1E9864 | Roxx | 2344553567,  6756854467 | 45000 |

In the above table, we can clearly see that the phone number column has two values.

Thus, it violates the 1st NF. Now if we apply the 1st NF to the above table, we get the below table as the result.

|  |  |  |  |
| --- | --- | --- | --- |
| **Employee ID** | **Employee Name** | **Phone Number** | **Salary** |
| 1E8976 | Alex | 5642685465 | 80000 |
| 1E8976 | Alex | 8646744666 | 80000 |
| 1E5678 | Mangraj | 2456675676 | 55000 |
| 1E5432 | David | 7777745646 | 70000 |
| 1E9864 | Roxx | 2344553567 | 45000 |
| 1E9864 | Roxx | 6756854467 | 45000 |

By this we achieved atomicity and also each and every column have unique values.

2nd Normal Form (2NF)

**Rules**: The table must be 1NF.

The table also should not contain partial dependency. Here partial dependency means proper subset of candidate key determines non-prime attributes.

It means all non-key attributes are fully functional dependent on the primary key.

|  |  |  |
| --- | --- | --- |
| **Employee Id** | **Department Id** | **Office Location** |
| 1EDU001 | ED-T1 | Pune |
| 1EDU002 | ED-S3 | Bangalore |
| 1EDU001 | ED-Z2 | Mumbai |
| 1EDU004 | ED-T1 | Pune |

This table has a composite primary key Employee Id, Department Id.

Here the non-key attribute is Office Location.

In this case Office Location, only depends upon Department Id, which is only part of the primary key. So here partial dependency occurs. Which violates the 2NF.

To bring this table to second Normal Form, we need to break the table into two parts.

|  |  |
| --- | --- |
| **Employee Id** | **Department Id** |
| 1EUD001 | ED-T1 |
| 1EUD002 | ED-S3 |
| 1EUD001 | ED-Z2 |
| 1EUD004 | ED-T1 |

|  |  |
| --- | --- |
| **Department ID** | **Office location** |
| ED-T1 | Pune |
| ED-S3 | Bangalore |
| ED-Z2 | Mumbai |

Now Office Location is fully dependent on the primary key of that table, which is Department Id.

3rd Normal Form (3NF)

**Rules**: Must be 2NF. There should no transitive dependency for non-prime attributes (which doesn’t form a candidate key).

That means non-prime attributes shouldn’t dependent on other non-prime attributes.

X ----> Y

Y -----> Z

Then, X ---> Z

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Student Id** | **Student Name** | **Subject Id** | **Subject** | **Address** |
| 1DT00001 | Alex | 15CS11 | SQL | Odisha |
| 1DT00002 | David | 15CS12 | JAVA | Goa |
| 1DT00003 | Rahul | 15CS14 | JS | Jharkhand |
| 1DT00004 | Arjeet | 15CS13 | C++ | Rajasthan |

Here Student Id (Super Key) determines Subject Id, and Subject Id, determines Subject.

The Non-Prime attributes (Subject) transitively dependent on Super Key (Student Id) via Subject Id.

This implies that we have a transitive functional dependency, and this structure doesn't satisfy the third normal form.

|  |  |  |  |
| --- | --- | --- | --- |
| **Student Id** | **Student Name** | **Subject Id** | **Address** |
| 1DT00001 | Alex | 15CS11 | Odisha |
| 1DT00002 | David | 15CS12 | Goa |
| 1DT00003 | Rahul | 15CS14 | Jharkhand |
| 1DT00004 | Arjeet | 15CS13 | Rajasthan |

|  |  |
| --- | --- |
| **Subject Id** | **Subject** |
| 15CS11 | SQL |
| 15CS12 | JAVA |
| 15CS14 | JS |
| 15CS13 | C++ |