Normalization:

->It is the process of organizing the fields and tables of a database to minimize redundancy

And dependency

-.>It is dividing larger tables into smaller tables and defining a relationship between them

->It is a bottom-up approach

->Types of normalization:

1.1 NF

2.2 NF

3.3 NF

4.Bucee-codd normal form

Example:

|  |  |  |  |
| --- | --- | --- | --- |
| **SALUTATION** | **CUSTOMER NAME** | **CITY** | **BOOK ISSUED** |
| MR. | RAJ | BANGALORE | LET US C,ORACLE DATABSE, |
| MISS. | PRIYA | CHENNAI | PROGRAMMING WITH JAVA,C++ PROGRAMMING |
| MR. | RAJ | DELHI | DBA FUNDAMENTALS, ORACLE |

Consider the above table where the column Book issued has more than value

->1 NF

->Rule 1: It indicates each column in the table should contain only a single value

->Rule 2: Each record(row) needs to be unique or it should contain atomic values only

->Atomic values:

->Atomic values are those which cannot be divided further

|  |  |  |  |
| --- | --- | --- | --- |
| **SALUTATION** | **CUSTOMER NAME** | **CITY** | **Book Issued** |
| MR. | Raj | BANGALORE | LET US C |
| MR. | Raj | BANGALORE | ORACLE DATABSE |
| Miss | Priya | CHENNAI | PROGRAMMING WITH JAVA |
| Miss | Priya | CHENNAI | C++ PROGRAMMING |
| MR. | Raj | DELHI | DBA FUNDAMENTALS |
| MR. | Raj | DELHI | ORACLE PROGRAMMING |

->So in First normal form, the previous table is normalized as the above table such that each row or record is unique

->2NF:

->Rule 1: Be on 1 NF

->Rule 2: Each table should have a single column primary key

->Rule 3: Remove subsets of data that apply to multiple rows and make them in single rows

For example: In the above table , take city column where the city name is present in more than one row, so it should be avoided by dividing the tables and giving a primary key to each table

Divide the tables in such a way that one table contains member information and the other table contains customer information and the other table contains the Book Information

So primary columns named MEMBERSHIP ID and BOOK ID are added to the first and the second table respectively

So after doing 1 NF, the column which we reduce, try dividing the tables with respect to that column

Finally the tables becomes as follows:

Table1:

|  |  |  |  |
| --- | --- | --- | --- |
| **MEMBERSHIP ID** | **SALUTATION** | **CUSTOMER NAME** | **CITY** |
| 1 | MR. | RAJ | BANGALORE |
| 2 | MISS. | PRIYA | CHENNAI |
| 3 | MR. | RAJ | DELHI |
|  |  |  |  |

Table 2:

|  |  |  |
| --- | --- | --- |
| **OOK ID** | **MEMBERSHIP ID** | **BOOK ISSUED** |
| 1 | 1 | LET US C |
| 2 | 1 | ORACLE DATABSE |
| 3 | 2 | PROGRAMMING WITH JAVA |
| 4 | 2 | C++ PROGRAMMING |
| 5 | 3 | ORACLE PROGRAMMING |
| 6 | 3 | DBA FUNDAMENTALS |

3 NF:

->Rule 1: Be in 2 NF

->Rule 2: Divide the tables further in such a way that there are no functional dependencies

->Rule 3: For example now we can divide salutation but not name

Because person with the same name can exist but the salutation and entities like age cannot be changed, so only these entities like age and salutations should be further divided into separate tables.

Functional Dependency:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | | Employee Number | EmployeeName | **Salary** | **CITY** | | 1 | HAri | 470000 | Coimbatore | | 2 | Krishnan | 375000 | Chennai | | 3 | Hari Jayram | 2500000 | Coimbatore | |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

In the above table , if we know the employee number we can get the details of EmployeeName ,Salary and city details

So Employee Name and Salary are functionally dependent on Employee Number

Functional Dependency is denoted by ->

The functional dependency of X on Y is represented by X →Y

Transitive functional dependency:

|  |  |  |  |
| --- | --- | --- | --- |
| **Company** | **CEO** | **AGE** | **CITY** |
| Microsoft | SatyaNadella | 51 | BANGALORE |
| Google | Sundar Pichai | 35 | CHENNAI |
| Alibaba | JackMa | 55 | China |

In the above table

->if we know the company name , we can find the ceo name and if we know the ceo name we can find their age and city

->so according to the rule of transitive dependency :

If Company->CEO->AGE exists

Then Company->Age should also exist