**UNIT 4:**

**N ormalization:**

/\*var color = prompt('Enter your background color choice');

document.body.style.backgroundColor = color;

var textColor = prompt('Enter your text color choice');

document.body.style.color = textColor;

\*/

function changeStyle(color){

var div = document.getElementById('content');

div.style.backgroundColor = color;

div.style.color = 'white';

div.style.padding = '100px';

}

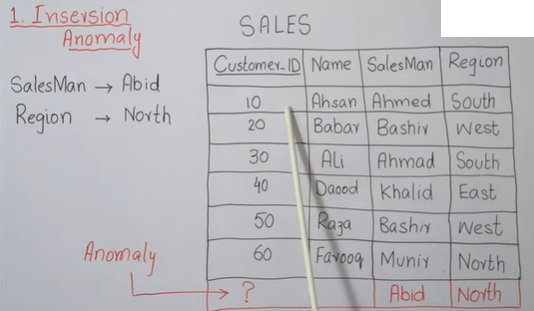
* Redundancy/duplication/repetition.
* Insertion, deletion, updating anomalies.

**Normal forms**

* First Normal Form (1NF)
* Second Normal Form (2NF)
* Third Normal Form (3NF)
* Fourth Normal Form (4NF)
* Boyce-Codd Normal Form (BCNF)

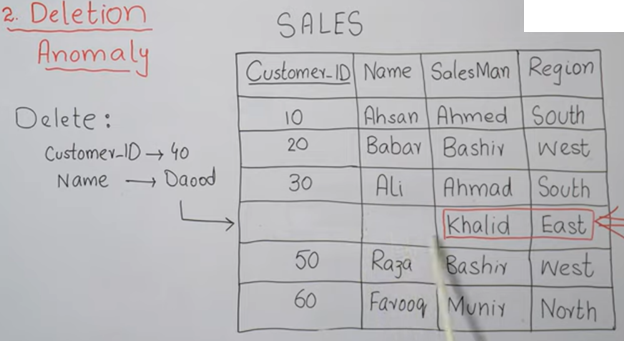
If a database design is not perfect, it may contain anomalies, which are like a bad dream for any database administrator. Managing a database with anomalies is next to impossible.

* **Insert anomalies** − We tried to insert data in a record that does not exist at all.



(Here, we tried to enter the new value Abid as a salesman and North as a Region. But without Primary Key (Customer\_ID) its not possible to do so. Or if we put any random customer id (like #,$) then it might create problem in future also we can’t leave it NULL.)

* **Deletion anomalies** − We tried to delete a record, but parts of it was left undeleted because of unawareness, the data is also saved somewhere else.

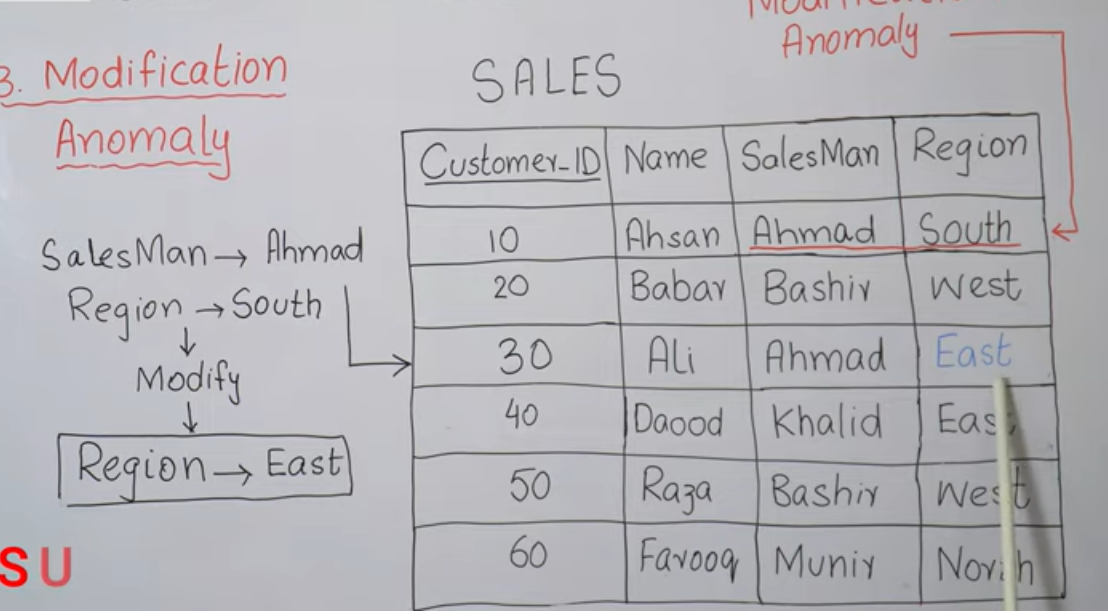


While deleting the customer id (Primary Key) the entry is not consistent and cannot be feasible to delete that single entry. Also, when we need to delete the customer id and name, we removed all the entry of that row from the table.

DELETE FROM SALES,

WHERE Customer\_ID=40; //it will remove the whole row including the salesman information too, which I cannot recover it later.

* **Update anomalies** − If data items are scattered and are not linked to each other properly, then it could lead to strange situations. For example, when we try to update one data item having its copies scattered over several places, a few instances get updated properly while a few others are left with old values. Such instances leave the database in an inconsistent state.



Normalization is a method to remove all these anomalies and bring the database to a consistent state. If we need to split the table, we can. And when we need to update, insert and update the table, then there won’t be any problem to do so.

/\*var color = prompt('Enter your background color choice');

document.body.style.backgroundColor = color;

var textColor = prompt('Enter your text color choice');

document.body.style.color = textColor;

\*/

function changeStyle(color){

var div = document.getElementById('content');

div.style.backgroundColor = color;

div.style.color = 'white';

div.style.padding = '100px';

}

**2NF:**

To be in second normal form, a relation must be in first normal form and relation must not contain any partial dependency. A relation is in 2NF if it has No Partial Dependency, i.e., no non-prime attribute (attributes which are not part of any candidate key) is dependent on any proper subset of any candidate key of the table.

**Partial Dependency** – If the proper subset of candidate key determines non-prime attribute, it is called partial dependency.

**Prime vs Non-Prime attributes**: **Prime Attribute** means which is part of Candidate Key. **Candidate Key** can be a person’s unique ID which can be made primary key like: Citizenship no., license no., voter id no., phone no, reg no, roll no, etc. **Non-Prime Attribute** an attribute that is not part of any candidate key.

**Partial vs Transitive Dependencies:**

As stated, the non-prime attributes i.e. StudentName and ProjectName should be functionally dependent on part of a candidate key, to be Partial Dependent. The StudentName can be determined by StudentID, which makes the relation Partial Dependent.

A transitive dependency occurs when one non-prime attribute is dependent on another non-prime attribute.

**Example:** Let's assume, a school can store the data of teachers and the subjects they teach. In a school, a teacher can teach more than one subject.

**TEACHER table**

|  |  |  |
| --- | --- | --- |
| **TEACHER\_ID** | **SUBJECT** | **TEACHER\_AGE** |
| 25 | Chemistry | 30 |
| 25 | Biology | 30 |
| 47 | English | 35 |
| 83 | Math | 38 |
| 83 | Computer | 38 |

In the given table, non-prime attribute TEACHER\_AGE is dependent on TEACHER\_ID which is a proper subset of a candidate key. That's why it violates the rule for 2NF. To convert the given table into 2NF, we decompose it into two tables:

**TEACHER\_DETAIL table:**

|  |  |
| --- | --- |
| **TEACHER\_ID** | **TEACHER\_AGE** |
| 25 | 30 |
| 47 | 35 |
| 83 | 38 |

**TEACHER\_SUBJECT table:**

|  |  |
| --- | --- |
| **TEACHER\_ID** | **SUBJECT** |
| 25 | Chemistry |
| 25 | Biology |
| 47 | English |
| 83 | Math |
| 83 | Computer |

**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.

A relation is in third normal form if it holds atleast one of the following conditions for every non-trivial function dependency X → Y.

1. X is a super key.
2. Y is a prime attribute, i.e., each element of Y is part of some candidate key.

**Example:**

**EMPLOYEE\_DETAIL table:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **EMP\_ID** | **EMP\_NAME** | **EMP\_ZIP** | **EMP\_STATE** | **EMP\_CITY** |
| 222 | Harry | 201010 | UP | Noida |
| 333 | Stephan | 02228 | US | Boston |
| 444 | Lan | 60007 | US | Chicago |
| 555 | Katharine | 06389 | UK | Norwich |
| 666 | John | 462007 | MP | Bhopal |

**Super key in the table above:**

The set of attributes that can uniquely identify a tuple is known as Super Key.

***i.e:{EMP\_ID}, {EMP\_ID, EMP\_NAME}, {EMP\_ID, EMP\_NAME, EMP\_ZIP}..so on***

**Candidate key:** {EMP\_ID}

**Non-prime attributes:** In the given table, all attributes except EMP\_ID are non-prime.

Here, EMP\_STATE & EMP\_CITY dependent on EMP\_ZIP and EMP\_ZIP dependent on EMP\_ID. The non-prime attributes (EMP\_STATE, EMP\_CITY) transitively dependent on super key(EMP\_ID). It violates the rule of third normal form.

That's why we need to move the EMP\_CITY and EMP\_STATE to the new <EMPLOYEE\_ZIP> table, with EMP\_ZIP as a Primary key.

**EMPLOYEE table:**

|  |  |  |  |
| --- | --- | --- | --- |
| **EMP\_ID** | **EMP\_NAME** | | **EMP\_ZIP** |
| 222 | | Harry | 201010 |
| 333 | | Stephan | 02228 |
| 444 | | Lan | 60007 |
| 555 | | Katharine | 06389 |
| 666 | | John | 462007 |

**EMPLOYEE\_ZIP table:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **EMP\_ZIP** | | **EMP\_STATE** | | **EMP\_CITY** |
| 201010 | UP | | Noida | |
| 02228 | US | | Boston | |
| 60007 | US | | Chicago | |
| 06389 | UK | | Norwich | |
| 462007 | MP | | Bhopal | |

**BCNF:**

* (**Left hand side of each Functional Dependency should be Candidate Key**)
* (Functional Dependency (FD) is a constraint that determines the relation of one attribute to another attribute)

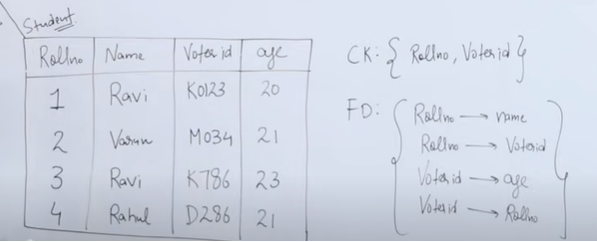
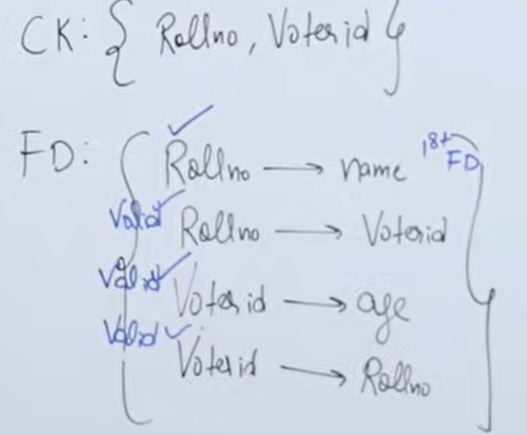
**CK** = Roll no, Voter Id

**FD** = Roll No --à Name

Roll No --à Voter ID

Voter ID -à Age

Voter ID -à Roll No

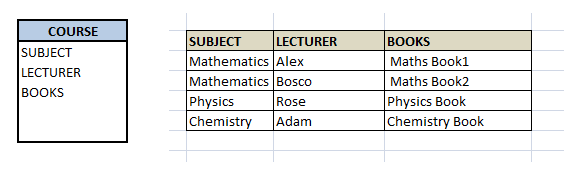


**4NF:**

In the fourth normal form,

* It should meet all the requirement of 3NF
* Attribute of one or more rows in the table should not result in more than one rows of the same table leading to multi-valued dependencies

To understand it clearly, consider a table with Subject, Lecturer who teaches each subject and recommended Books for each subject.



If we observe the data in the table above it satisfies 3NF. But LECTURER and BOOKS are two independent entities here. There is no relationship between Lecturer and Books. In the above example, either Alex or Bosco can teach Mathematics. For Mathematics subject, student can refer either ‘Maths Book1’ or ‘Maths Book2’. i.e.;

**SUBJECT –> LECTURER**

**SUBJECT–>BOOKS**

This is a multivalued dependency on SUBJECT. If we need to select both lecturer and books recommended for any of the subject, it will show up (lecturer, books) combination, which implies lecturer who recommends which book. This is not correct.

To eliminate this dependency, we divide the table into two as below:

