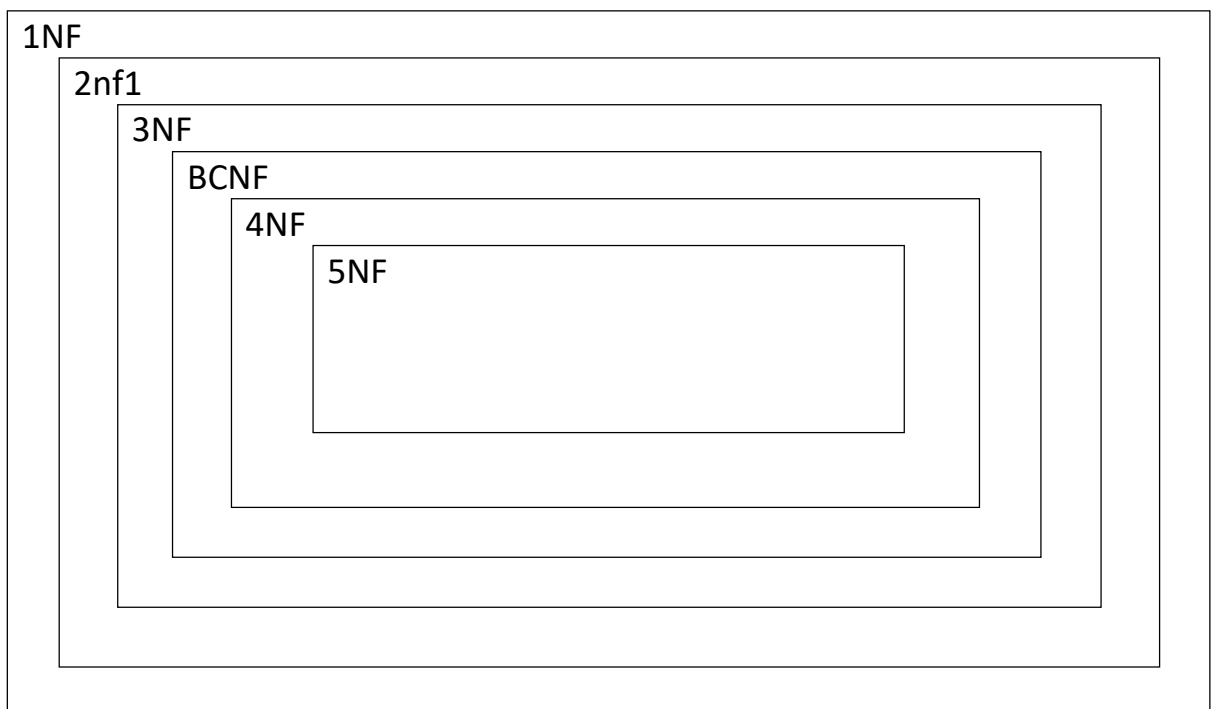


UNIT - III

Normalization

- ❖ Normalization is the process of reduce the redundancy.
- ❖ In another word remove the duplicate data from database.
- ❖ The basic objectives are three remove duplicated data from database without losing original information.
- ❖ After applying normalization we can't introduce inconsistency in database.
- ❖ There are three types of mapping of database.
 - Insert:- You can insert the row or column in a database with the help of insert we after the database.
 - Update:-We update row or column in a database with the help of update we after the database.
 - Delete:- We delete the row or column in the database with the help of delete we after the database
- ❖ Properties of normalization:-
 - Number of data should be duplicated.
 - The value of attribute in a relation must be self-contain.
 - A value must be specify in a row.
- ❖ Diagram:-



- ❖ BCNF (Boyce/Codd normal form)
- ❖ There are six types of normalization.

❖ 1NF (First Normal Form):-

- First normal form is an initialization step to convert your database into normalization.
- The first normal form is shortly called 1NF.
- First normal form is also called flat file.
- Rules :- in relation each and every attribute depends on atomic value.
- Example :-

Order_no	Order date	Item code	Item name	Quantity	Price
1456	25/07/2020	001	Pen	20	50
1456	25/07/2020	002	Pencil	20	40
1456	25/07/2020	003	Book	10	150
1460	08/09/2020	002	Pencil	20	40
1460	15/09/2020	002	Pencil	20	40

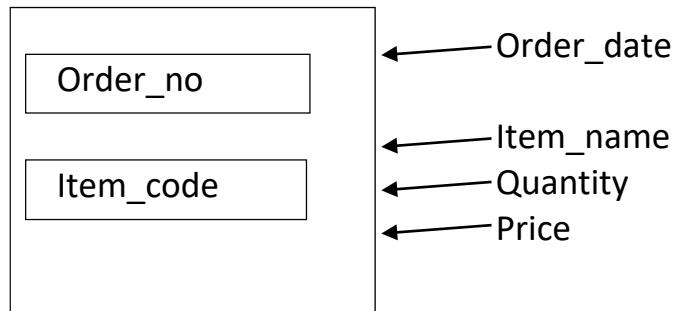
- The above table is already in first normal form.
- In a table order_no or item_code are atomic values, each and every attribute which are stored in relation that are dependent on atomic value.

❖ 2NF (Second Normal Form):-

- Second normal form is the improvement of first normal form.
- In another word second normal form derived from after first normal form.
- Rules:- in a relation all non key attribute depends on primary key.

Order_no	Order date	Item code	Item name	Quantity	Price
1456	25/07/2020	001	Pen	20	50
1456	25/07/2020	002	Pencil	20	40
1456	25/07/2020	003	Book	10	150
1460	08/09/2020	002	Pencil	20	40
1460	15/09/2020	002	Pencil	20	40

- Draw the dependency on table1.



- Order_no 1456 or Item code 001 is cancel then we loss, item name, price, quantity related to those item so we need to normalization table1.

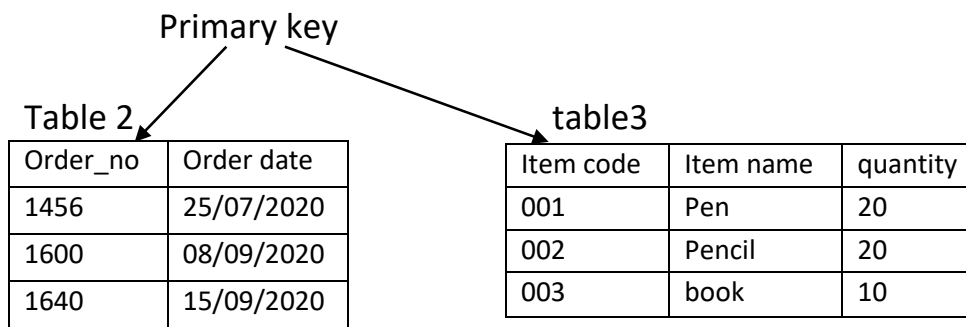


Table4

Price	Order_no	Item code
50	1456	001
40	1456	002
150	1456	003
40	1600	004
40	1640	005

- Above table are in second normal form.
- In table2 order no is a primary key and all other attribute depends on primary key.
- In table3 item code is a primary key and all other attributes depends on primary key.
- In table4 there is no primary key because it's a relational table.
- The above relationship satisfied all the condition of second normal form.

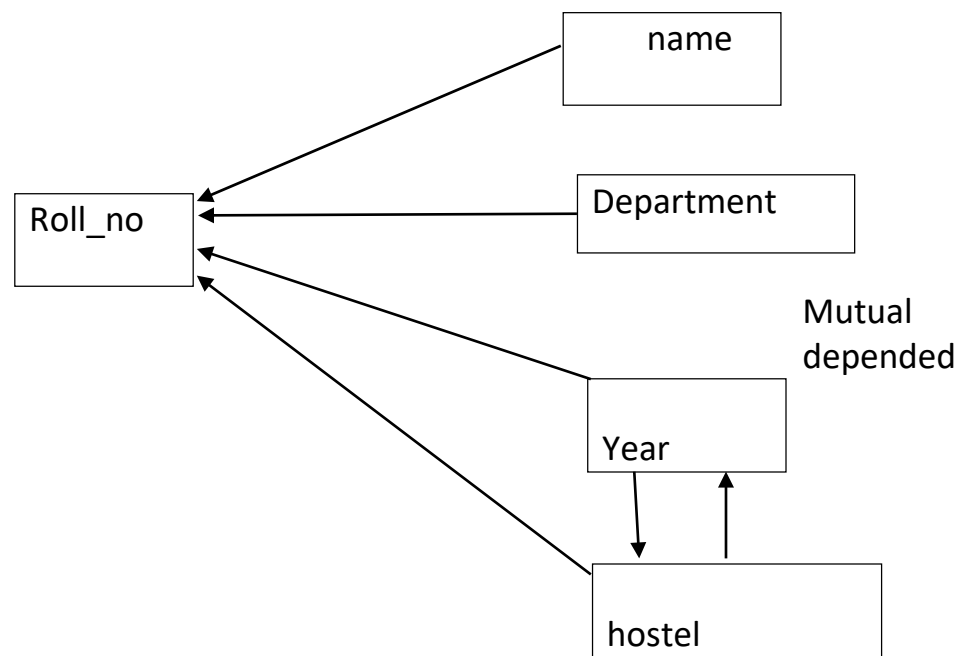
❖ 3NF (Third Normal form):-

- The third normal form improvement of second normal form.
- In another word third normal form is derived after second normal form.
- Rules:- in a relation all non key attribute are functionally depended on 1. Primary key 2. Mutual independent
- Here mutual independent means key attribute are functionally depend on each other.
- The third normal form will be needed where all attributes in a relation are not functionally depended on primary key and mutual depended.
- Example :-

Table1

Roll no	Name	Department	Year	hostel
1	Ankit	Maths	1	Ganga
2	Rahul	Science	1	Ganga
3	Pinal	Gujarati	2	Kaveri
4	Jenul	Hindi	2	Kaveri
5	jitendra	english	3	godavari

- In above table hostel name and year mutual depended on each other.
- Dependency diagram for this table.



- In a above relationship roll no is a primary key all other attributes are depended on roll no.
- From table 1 you can see first year student stay in ganga hostel.
- Same as second year and third year student stay in kaveri and Godavari hostel.
- This is mutual depended between year and hostel for the converting database into third NF. We have remove this mutual depended.
- For converting this table 1 into 3NF we devide table into two part.
- 1. Student master table.
- 2. Hostel master table.

Student master table

Roll no	Name	Department	Year
1	Ankit	Maths	1
2	Rahul	Science	1
3	Pinal	Gujarati	2
4	Jenul	Hindi	2
5	Jitendra	english	3

Hostel master table

Year	Hostel
1	Ganga
1	Ganga
2	Kaveri
2	Kaveri
3	Godavari

- In above table 2 and 3 we remove mutual depended between hostel and year and all known key attribute are functionally depended on primary key.

What is Denormalization?

Traditionally data is stored in normalized databases, in which multiple separate tables are maintained in a relational database to minimize the redundant data. Therefore, whenever we have to access data from multiple tables, we need to perform complex and costly join operations on the required tables.

Denormalization is a database optimization technique where we add redundant data in the database to get rid of the complex join operations. This is done to speed up database access speed. Denormalization is done after normalization for improving the performance of the database. The data from one table is included in another table to **reduce the number of joins** in the query and hence helps in speeding up the performance.

Example: Suppose after normalization we have two tables first, Student table and second, Branch table. The student has the attributes as *Roll_no* , *Student-name* , *Age* , and *Branch_id* .

Student table

Roll_no	Student_name	Age	Branch_id
1	Andrew	18	10
2	Angel	19	10
3	Priya	20	10
4	Analisa	21	11
5	Anna	21	12

Advantages of Denormalization

1. Minimizing the need for joins.
2. Reducing the number of tables.
3. Queries to be retrieved can be simpler.
4. Less likely to have bugs.
5. Precomputing derived values.
6. Reducing the number of relations.
7. Reducing the number of foreign keys in relation.

Disadvantages of Denormalization

1. As data redundancy is there, update and insert operations are more expensive and take more time. Since we are not performing normalization, so this will result in redundant data.
2. Data Integrity is not maintained in denormalization. As there is redundancy so data can be inconsistent.

Functional Dependency

The functional dependency is a relationship that exists between two attributes. It typically exists between the primary key and non-key attribute within a table.

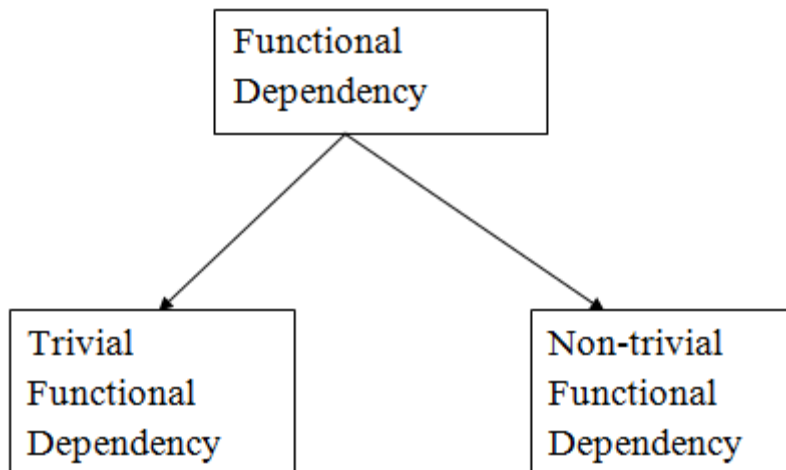
1. $X \rightarrow Y$

The left side of FD is known as a determinant, the right side of the production is known as a dependent.

For example:

Assume we have an employee table with attributes: Emp_Id, Emp_Name, Emp_Address.

Types of Functional dependency



1. Trivial functional dependency

- $A \rightarrow B$ has trivial functional dependency if B is a subset of A.
- The following dependencies are also trivial like: $A \rightarrow A$, $B \rightarrow B$

Example:

1. Consider a table with two columns Employee_Id and Employee_Name.
2. $\{Employee_id, Employee_Name\} \rightarrow Employee_Id$ is a trivial functional dependency as
3. Employee_Id is a subset of $\{Employee_Id, Employee_Name\}$.
4. Also, $Employee_Id \rightarrow Employee_Id$ and $Employee_Name \rightarrow Employee_Name$ are trivial dependencies too.

2. Non-trivial functional dependency

- $A \rightarrow B$ has a non-trivial functional dependency if B is not a subset of A.
- When $A \cap B$ is NULL, then $A \rightarrow B$ is called as complete non-trivial.

Example:

1. $ID \rightarrow Name$,
2. $Name \rightarrow DOB$
- 3.

What is Transitive Dependency?

Consider a relation $R(A\ B\ C)$. Here A, B, and C are known as the attribute of the relation R. When the following condition arises; then the transitive dependency occurs in the DBMS. The conditions are $A \rightarrow B$, $B \rightarrow C$. Therefore, the condition becomes $A \rightarrow C$. In transitive functional dependencies, a dependent is directly dependent on the determinant.

Let's understand the transitive dependencies with the help of the following example.

Author_ID	Author	Book	Author_Nationality
A1	Arundhati Roy	The God of Small Things	India
A1	Kiran Desai	The Inheritance Of Loss	India
A2	R. K. Narayan	The Man Eater of Malgudy	India

Practice Questions on Transitive Dependency in DBMS

1. Which of these anomalies is a result of transitive dependency in a database management system?

- A. Deletion
- B. Insertion
- C. Modification
- D. All of the mentioned

Answer: D. All of the mentioned

2. The 4th Normal Form is designed so as to cope up with:

- A. Transitive dependency
- B. Join dependency
- C. Multi-valued dependency
- D. None

Answer: A. Transitive dependency

3. For some of the relations, the changing of data can deliver some undesirable consequences known as:

- A. Modification anomalies
- B. Referential Integrity constraints
- C. Transitive dependencies
- D. Normal forms

Answer: C. Transitive dependencies