SNS COLLEGE OF ENGINEERING

Kurumbapalayam (Po), Coimbatore – 641 107

An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING(IoT and Cybersecurity Including BCT)

COURSE NAME : 19SB504 DATABASE MANAGEMENT SYSTEMS

III YEAR / V SEMESTER

Unit III-**E-R Diagram models and NORMAL FORMS**

Topic :Normalizations: 1NF,2NF,3NF,BCNF,4NF,5NF

Normal Forms

- Normal forms are a **set of guidelines and rules** for designing and structuring a relational database to **minimize data redundancy and improve data integrity.**

- There are different normal forms, each with specific criteria that a database table should meet to be considered in that form.

1. **First Normal Form (1NF)**
2. **Second Normal Form (2NF)**
3. **Third Normal Form (3NF)**
4. **Boyce-Codd Normal Form (BCNF)**
5. **Fourth Normal Form (4NF)**
6. **Fifth Normal Form (5NF)**

1. First Normal Form (1NF)

First Normal Form (1NF) is a fundamental concept in database design that addresses the structure of tables in a relational database.

A table is said to be in 1NF if it meets the following criteria:

1.Each column in the table contains only atomic (indivisible) values.

2.Each cell in the table holds a single value, not a list or set of values.

3.Each row in the table is unique, meaning there are no duplicate rows.

Let's illustrate 1NF with a simple example. Consider a table for storing information about students and their courses. Here's an example of a table that is not in 1NF:

NORMALIZATION

StudentID	StudentName	Courses
101	Alice	Math, Physics, Chemistry
102	Bob	History, English
103	Carol	Physics, Chemistry, English

In this example, the "Courses" column violates 1NF because it contains a list of courses for each student, which is not atomic. To bring this table into 1NF, you would need to restructure it to separate the courses into individual rows. Here's the same data in 1NF:

NORMALIZATION

Students Table:

StudentID	StudentName
101	Alice
102	Bob
103	Carol

Courses Table:

StudentID	Course
101	Math
101	Physics
101	Chemistry
102	History
102	English
103	Physics
103	Chemistry
103	English

2. Second Normal Form (2NF)

- The Second Normal Form (2NF) is a level of normalization in relational database design that builds upon the First Normal Form (1NF). A table is in 2NF if it meets the following conditions:

1. It is in 1NF, which means that all attributes contain only atomic (indivisible) values, and each row is unique.

2. **It does not contain partial dependencies**, which means that non-key attributes (attributes that are not part of the primary key) are fully functionally dependent on the entire primary key.

- To illustrate 2NF, let's consider a simple example of a table representing information about students, their courses, and the course instructors. We'll show the process of bringing the table to 2NF.

Original Table:

StudentID	Course	Instructor
101	Math	Mr. A
101	History	Mr. B
102	Math	Mr. A
103	Physics	Mr. C

- In this table:
- "StudentID" is the primary key.
- "Course" and "Instructor" are non-key attributes.
- To bring this table to 2NF, we need to remove partial dependencies, which occur when non-key attributes depend on a part of the primary key. In this case, "Instructor" depends on "Course," which is part of the primary key.

- Step 1: Create Separate Tables for Courses and Instructors

Courses Table:

Course

Math

History

Physics

Instructors Table:

Course	Instructor
Math	Mr. A
History	Mr. B
Physics	Mr. C

Student Courses Table:

StudentID	Course
101	Math
101	History
102	Math
103	Physics

Third Normal Form (3NF)

- The Third Normal Form (3NF) is another level of normalization in relational database design, building upon the First Normal Form (1NF) and Second Normal Form (2NF).
- A table is considered to be in 3NF if it meets the following conditions:

1. It is in 2NF, which means it does not contain partial dependencies, where non-key attributes depend on a part of the primary key.

2. It **does not contain transitive dependencies**, which means that non-key attributes depend only on the primary key, not on other non-key attributes.

To illustrate 3NF, let's consider a simple example of a table representing information about employees and their projects. We'll show the process of bringing the table to 3NF.

Original Table:

EmployeeID	EmployeeName	Department	DepartmentLocation
101	Alice	HR	New York
102	Bob	IT	India
103	Carol	HR	New York
104	David	Sales	Londen

In this table:

"EmployeeID" is the primary key.

"Department" and "DepartmentLocation" are non-key attributes.

Step 1: Create a "Departments" Table

Departments Table:

Department	**Department Location**
HR	New York
IT	**India**
Sales	Londen

Step 2: Create a Linking Table

Now, to establish the relationship between employees and departments, you can create a linking table that stores the associations between employees and departments.

EmployeeDepartments Table:

EmployeeID	Department
101	HR
102	IT
103	HR
104	Sales

Boyce-Codd Normal Form (BCNF)

- Boyce-Codd Normal Form (BCNF) is a higher level of database normalization that aims to eliminate certain types of anomalies in a relational database.
- BCNF is a stricter form of normalization than the Third Normal Form (3NF). To be in BCNF, a relation (table) must meet two main criteria:
- **No partial dependencies**
- **No transitive dependencies**:

NORMALIZATION

- simple example
- Suppose we have a relation (table) called Student_Course with the following attributes:

Student_ID (Primary Key)

Course_ID (Primary Key)

Student_Name

Course_Name

Instructor

Student_ID	Course_ID	Student_Name	Course_Name	Instructor
1	101	John	Math 101	Mr. Smith
2	102	Alice	Science 201	Mrs. Johnson
3	101	Bob	Math 101	Mr. Smith

NORMALIZATION

- In this example, we have information about students, courses, and the instructors of those courses.

- Both Student_ID and Course_ID together serve as the primary key, ensuring each combination of student and course is unique.

Fourth Normal Form (4NF)

The Fourth Normal Form (4NF) is a level of database normalization that goes beyond the Third Normal Form (3NF).

4NF is concerned with eliminating multi-valued dependencies in a relational database.

To be in 4NF, a relation (table) must meet the following criteria:

Example
STUDENT

STU_ID	COURSE	HOBBY
21	Computer	Dancing
21	Math	Singing
34	Chemistry	Dancing
74	Biology	Cricket
59	Physics	Hockey

- The given STUDENT table is in 3NF, but the COURSE and HOBBY are two independent entity. Hence, there is no relationship between COURSE and HOBBY.

- In the STUDENT relation, a student with STU_ID, **21** contains two courses, **Computer** and **Math** and two hobbies, **Dancing** and **Singing**. So there is a **Multi-valued dependency on STU_ID**, which leads to unnecessary repetition of data.

- So to make the above table into 4NF, we can decompose it into two tables:

STUDENT_COURSE

STU_ID	COURSE
21	Computer
21	Math
34	Chemistry
74	Biology
59	Physics

STUDENT_HOBBY

STU_ID	HOBBY
21	Dancing
21	Singing
34	Dancing
74	Cricket
59	Hockey

- A relation is in 5NF if it is in 4NF and not contains any join dependency and joining should be lossless.

- 5NF is satisfied when all the tables are broken into as many tables as possible in order to avoid redundancy.

- 5NF is also known as Project-join normal form (PJ/NF).

Example

SUBJECT	LECTURER	SEMESTER
Computer	Anshika	Semester 1
Computer	John	Semester 1
Math	John	Semester 1
Math	Akash	Semester 2
Chemistry	Praveen	Semester 1

In the above table, John takes both Computer and Math class for Semester 1 but he doesn't take Math class for Semester 2. In this case, combination of all these fields required to identify a valid data.

- So to make the above table into 5NF, we can decompose it into three relations P1, P2 & P3:

P1

SEMESTER	SUBJECT
Semester 1	Computer
Semester 1	Math
Semester 1	Chemistry
Semester 2	Math

P2

SUBJECT	LECTURER
Computer	Anshika
Computer	John
Math	John
Math	Akash
Chemistry	Praveen

P3

SEMSTER	LECTURER
Semester 1	Anshika
Semester 1	John
Semester 1	John
Semester 2	Akash
Semester 1	Praveen

Thank You .....