---Normalization---

Normalization: Introduction

Normalization is a process of organizing the data in database to avoid **data redundancy**, **insertion anomaly, update anomaly & deletion anomaly**.

Prepared by: Sanchita Saha Ray

Anomalies in DBMS

There are **three types** of **modification anomalies** that occur when the database is **not Normalized**.

– **Insertion Anomaly**

– **Update Anomaly**

– **Deletion Anomaly**

Prepared by: Sanchita Saha Ray

Insertion Anomalies

Situation 1:

Suppose a new employee joins in the company-- who is under training and currently not assigned to any department.

Problem 1:

Would we be able to insert data of that employee into the table?

Situation 2:

Suppose a new department is introduced in the company-- no employee is assigned to that department.

Problem 2:

Would we be able to insert data of that department into the table?

Prepared by: Sanchita Saha Ray

Insertion Anomaly (Example)

EmployeeID	Ename	DeptID	Salary	Dname	Dlocation
1001	John	2	4000	IT	New Delhi
1002	Anna	1	3500	HR	Mumbai
1003	James	1	2500	HR	Mumbai
1004	David	2	5000	IT	New Delhi
1005	Mark	2	3000	IT	New Delhi
1006	Steve	3	4500	Finance	Mumbai
1007	Alice	3	3500	Finance	Mumbai
Null	**Null**	**4**	**Null**	**Marketing**	**Kolkata**

Prepared by: Sanchita Saha Ray

Deletion Anomalies

Situation:

Suppose, the company terminates an employee who alone used to represent a department in that company.

Problem:

Would we be able to delete the record of that employee and still be able to keep existence of that department in the table?

Prepared by: Sanchita Saha Ray

Deletion Anomaly (Example)

EmployeeID	Ename	DeptID	Salary	Dname	Dlocation
1001	John	2	4000	IT	New Delhi
1002	Anna	1	3500	HR	Mumbai
1003	James	1	2500	HR	Mumbai
1004	David	2	5000	IT	New Delhi
1005	Mark	2	3000	IT	New Delhi
1006	Steve	3	4500	Finance	Mumbai

Universal Relation

Update Anomalies

Situation:

Company wants to change the name of a department.

Problem:

Would we be able to update all rows where that department name has appeared in the table?

*An **update anomaly** is a **data inconsistency** that results from **data redundancy** and a **partial update**.*

Update Anomaly (Example)

EmployeeID	Ename	DeptID	Salary	Dname	Dlocation
1001	John	2	4000	IT	New Delhi
1002	Anna	1	3500	HR	Mumbai
1003	James	1	2500	HR	Mumbai
1004	David	2	5000	IT	New Delhi
1005	Mark	2	3000	IT	New Delhi
1006	Steve	3	4500	Finance	Mumbai
1007	Alice	3	3500	Finance	Mumbai

Prepared by: Sanchita Saha Ray

How To Avoid Anomalies ?

The best approach to create tables without anomalies is to ensure that the tables are normalized, and that's accomplished by understanding functional dependencies.

In other words, it will eliminate redundancies and anomalies.

Prepared by: Sanchita Saha Ray

What is Normalization ?

Database normalization is a process of making the database consistent by-

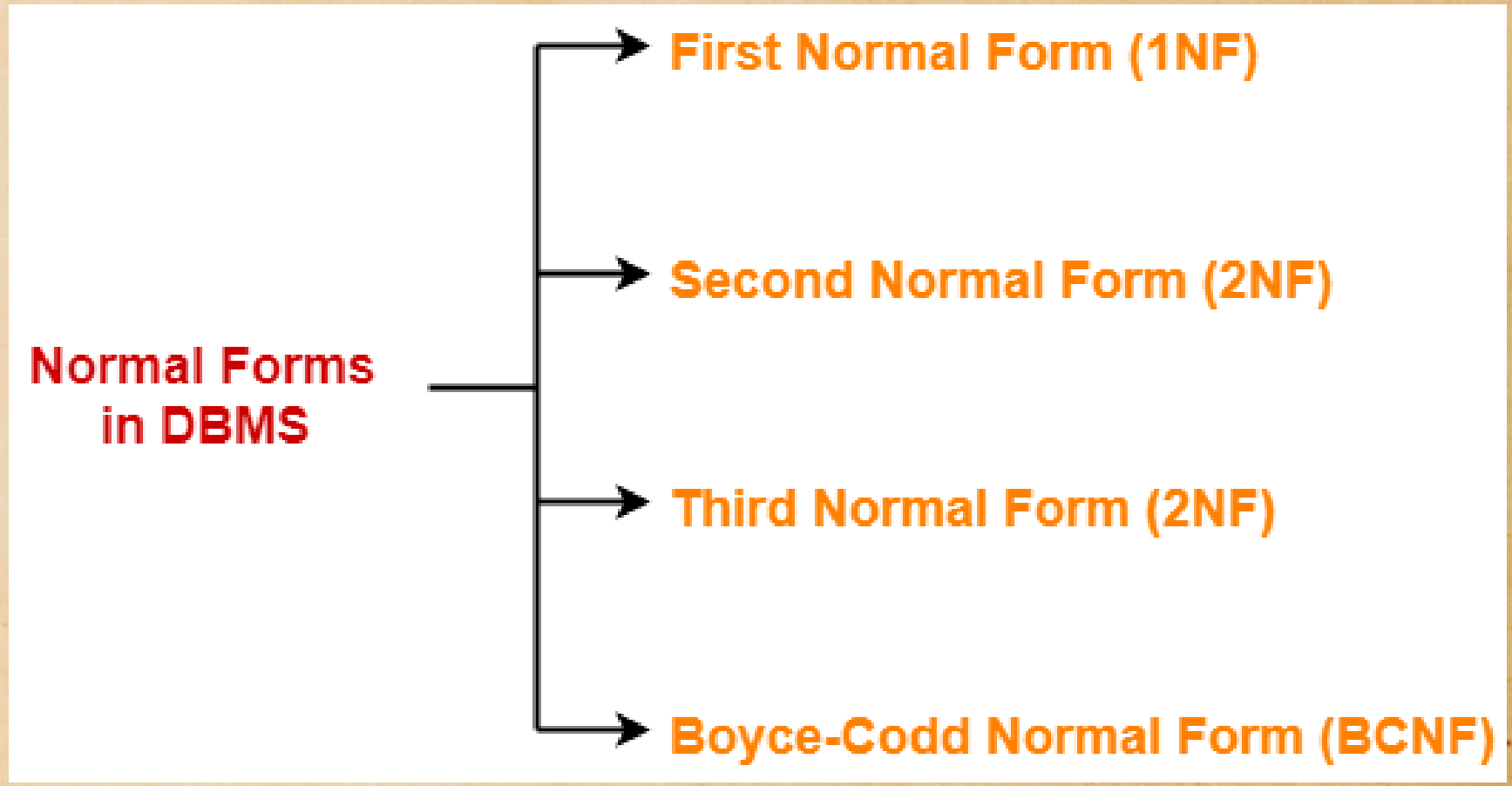
-- Reducing the *redundancies.*

-- Ensuring *integrity of data* through lossless decomposition.

-- Normalization is done through Normal forms (NF).

Prepared by: Sanchita Saha Ray

Normal Forms ?



Prepared by: Sanchita Saha Ray

First Normal Form (1NF)

A given **relation** is called in **First Normal Form (1NF)** if each **cell** of the table contains only an **atomic value**.

OR

A given **relation** is called in First Normal Form (1NF) if the **attribute** of every **tuple** is either **single valued** or a **NULL value**.

Prepared by: Sanchita Saha Ray

First Normal Form (1NF)

The following relation is not in 1NF-----

Student_id	Name	Subjects
100	Akshay	Computer Networks, OS
101	Aman	Database Management System
102	Anjali	Automata, Compiler Design

The following relation is in 1NF--------

Student_id	Name	Subjects
100	Akshay	Computer Networks
100	Akshay	Designing
101	Aman	Database Management System
102	Anjali	Automata
102	Anjali	Compiler Design

By default, every relation is in 1NF. Formal definition of a Relation states that value of all the attributes must be atomic.

Second Normal Form (2NF)

A given relation is called in **Second Normal Form (2NF)** iff -

-- Relation already exists in 1NF.

-- No partial dependency (w.r.t. non-prime attributes) exists in the relation.

Prepared by: Sanchita Saha Ray

Second Normal Form (2NF)

TABLE_PURCHASE_DETAIL

Customer ID	Store ID	Purchase Location
1	1	Los Angeles
1	3	San Francisco
2	1	Los Angeles
3	2	New York
4	3	San Francisco

Relation is not in --------2NF-----

Composite primary key: [Customer ID, Store ID].

Non-key attribute: [Purchase Location].

TABLE_PURCHASE

Customer ID	Store ID
1	1
1	3
2	1
3	2
4	3

TABLE_STORE

Store ID	Purchase Location
1	Los Angeles
2	New York
3	San Francisco

Relations are in ------2NF--------

Table [TABLE_STORE]:
***Primary Key:* [Store ID]**

Table [TABLE_PURCHASE]:
***Primary Key*:**
[Customer ID, Store ID]

Third Normal Form (3NF)

A given relation is called in **Third Normal Form (3NF)** iff -

-- Relation already exists in 2NF.
-- No transitive dependency (w.r.t. non-prime attributes) exists in the relation.

Any one condition should hold:
For every **non-trivial** functional dependency **A → B**

-- A is a *super key*
-- B is a *prime attribute*

Prepared by: Sanchita Saha Ray

Third Normal Form (3NF)

Customer

The table in this example is in 1NF and in 2NF.

CustID	CustName	AccNo	BankCode	Bank
1001	Rajesh	1099901	8921	HDFC
1002	Akash	1099902	8921	HDFC

3NF

Customer

CustID	CustName	AccNo	BankCode
1001	Rajesh	1099901	8921
1002	Akash	1099902	8921

Bank

Bank Code	Bank
8921	HDFC
8901	HDFC

FD1:
CustID --> CustName, AccNo, BankCode

FD2:
BankCode --> Bank

Prepared by: Sanchita Saha Ray

Boyce-Codd Normal Form (BCNF)

A relation is called in **Boyce-Codd Normal Form (3NF)** iff -

-- **Relation already exists in 3NF.**

-- For every **non-trivial** functional dependency **A → B,**
'A' is a *super key* of the relation

Prepared by: Sanchita Saha Ray

---Thank You---

Prepared by: Sanchita Saha Ray