UNIVERSITY of RWANDA

College of Science and Technology

School of ICT

Computer Science Department

Year 2

Lecturer: Muhayimana Odette
E-mail: muhayiodette06@gmail.com
Phone: 0788888571

CHAPV:Database normalization

Introduction

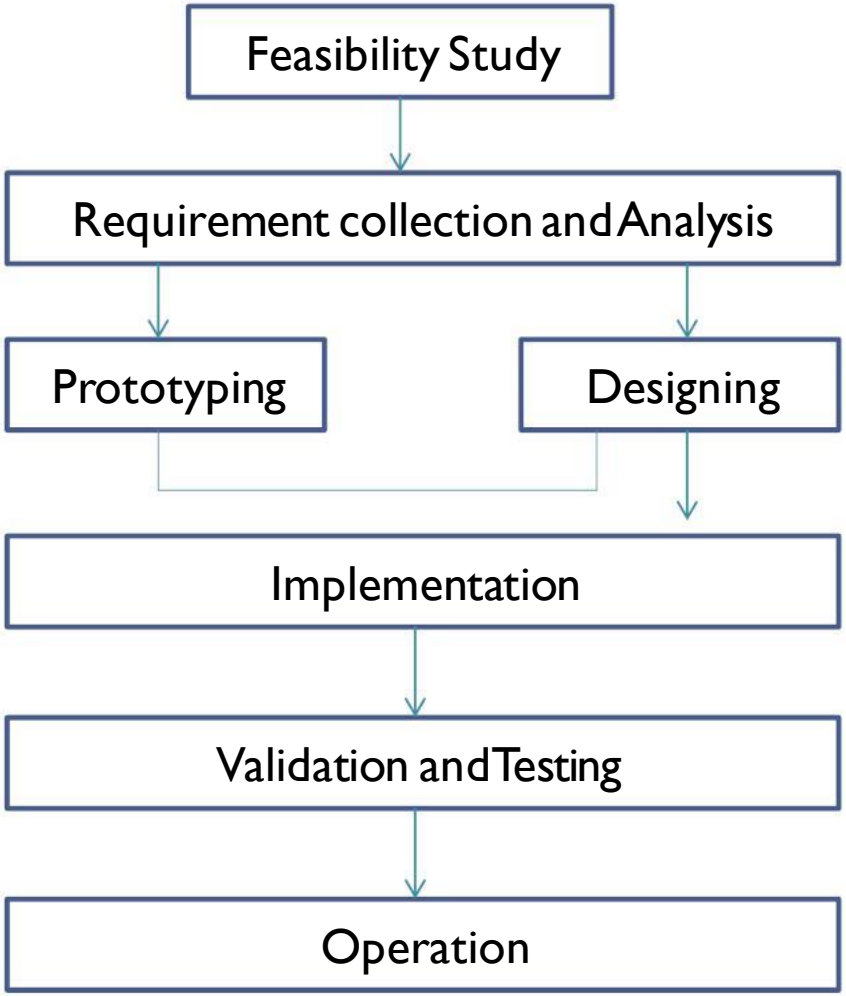
- Database design process integrates relevant data in such a manner that it can be processed through a mechanism for recording the facts.

- The database design is a complex process.

- The degree of complexity increases if there are many-to-many relationships among individual components

Introduction

- Steps in database design



Introduction

- Steps in database design

Feasibility Study

The objective of creating the database must be clear

Requirement collection and Analysis

Decide what data to be stored,

How data will be used

People who are going to use the database must be interviewed repeatedly

Assumptions about the stated relationships between various parts of the data must be questioned again and again.

Introduction

- Steps in database design

Prototyping and Design

It implies a procedure for analyzing and organizing data into a form suitable to support business requirements and make use of strategic technology

It includes conceptual, logical and physical design

Implementation

It involves development of code for database processing

It also involves the installation of new database contents.

Objectives of Databases Design

1. Efficiency

- The design should make full and efficient use of the facilities provided.
- With online database, the users should interact without any time delay

2. Integrity

- The database should be as accurate as possible,
- Ensure that the database is not corrupted by hardware or software errors.

3. Privacy

- Database should not allow unauthorized access to files.

Objectives of Databases Design

4. Security

- Database once loaded, it should be safe from physical corruption whether from hardware or software failure or from unauthorized access
- This is a general requirement of most databases

5. Implementation

- Conceptual model should be simple and effective so that mapping from conceptual model to logical model is easy.

6. Flexibility

- The database should not be implemented in a rigid way that assumes the business will remain constant forever
- Database must be capable of responding readily to changes

Database Design Tools

- Advantages and disadvantages of database design tools

✓Advantages	Disadvantages
✓The amount of code to be written is reduced as a result the database design time is reduced	✓More expenses involved for the tool itself.
✓Chances of errors because of manual work are reduced	✓Developers might require special training to use the tool
✓Easy to convert the business model to working database model	
✓Easy to ensure that all business requirements are met with	
✓A higher quality, more accurate product is produced	

Database Design Tools

- Commercial Database Design tools

1. CASE Studio2

   Powerful database modeling, management, and reporting tool
   http://www.casestudio.com/enu/default.aspx

2. Design for Databases V3

   Database development tool using an entity relationship diagram
   http://www.datanamic.com/dezign

3. DBDesignr4

   Visual database design system that integrates database design, modeling

Database Design Tools

- Commercial Database Design tools

8. Qdesigner

   It is an enterprise modeling and design solution that empowers architects, DBAs, developers, and business analysts to produce IT solutions
   http://www.quest.com/QDesigner

9. Power designer

   It offers a modeling solution that analysts, DBAs, can tailor
   http://www.sybase.com/products/powerdesigner/

10. xCase

    Database design tools which provides data modeling environment.
    www.xcase.com

Redundancy and Data Anomaly

- Redundant data means storing the same information more than once.
- Redundancy can lead to anomalies.

Note:The different anomalies are:

- Insertion anomalies
- Updating anomalies
- Deletion anomalies

Redundancy and Data Anomaly

- Given a table

Staff no.	Job	Dept. no.	Dept.name	City
100	Sales man	10	Sales	Trichy
101	Manager	20	Accounts	Coimbatore
102	Clerk	30	Accounts	Chennai
103	Clerk	30	Operations	Chennai

Insertion Anomaly

We cannot insert a department without inserting a member of staff that works in that department.

Update Anomaly

Change the name of the department that "101" works in without simultaneously changing the department that "102" works

Deletion Anomaly

By removing employee 100, we have removed all information pertaining to the sales department

Redundancy and Data Anomaly

- Repeating Group

It is an attribute (or set of attributes) that can have more than one value for a primary key value.

Staff no.	Job	Dept. no.	Dept.name	City	Contact number
100	Sales man	10	Sales	Trichy	43545, 45385, 98766
101	Manager	20	Accounts	Coimbatore	67545, -------, --------
102	Clerk	30	Accounts	Chennai	-------, --------, ---------
103	Clerk	30	Operations	Chennai	-------, --------, ---------

For each contact number, we have to store the data of the STAFF which leads to more storage space (more memory)

Repeating groups are not allowed in a relational design, since all attributes have to be atomic.

Database Normalization

Edgar F. Codd (1970), the inventor of the **relational model**, introduced the concept of normalization and what we now know as the **First Normal Form (1NF).** Codd (1971) went on to define the **Second Normal Form (2NF) and Third Normal Form (3NF)** in 1971, and **Codd and Raymond F. Boyce (1974)** defined the **Boyce-Codd Normal Form (BCNF).** Higher normal forms were defined by other theorists in subsequent years, the most recent being the **Sixth Normal Form (6NF)** introduced by **Chris Date, Hugh Darwen, and Nikos Lorentzos** (2002).

Informally, a table/relation is often described as "normalized" if it is in the Third Normal Form *(Date, C, 2003)*. Most 3NF tables are free of insertion, update, and deletion anomalies, i.e. in most cases 3NF tables adhere to BCNF, 4NF, and 5NF (**but typically not 6NF).**

Purpose of Normalization

1. To avoid redundancy by storing each fact within the database only once
2. To put data into the form that is more able to accurately accommodate change
3. To avoid certain updating “anomalies”
4. To facilitate the enforcement of data constraint
5. To avoid unnecessary coding.

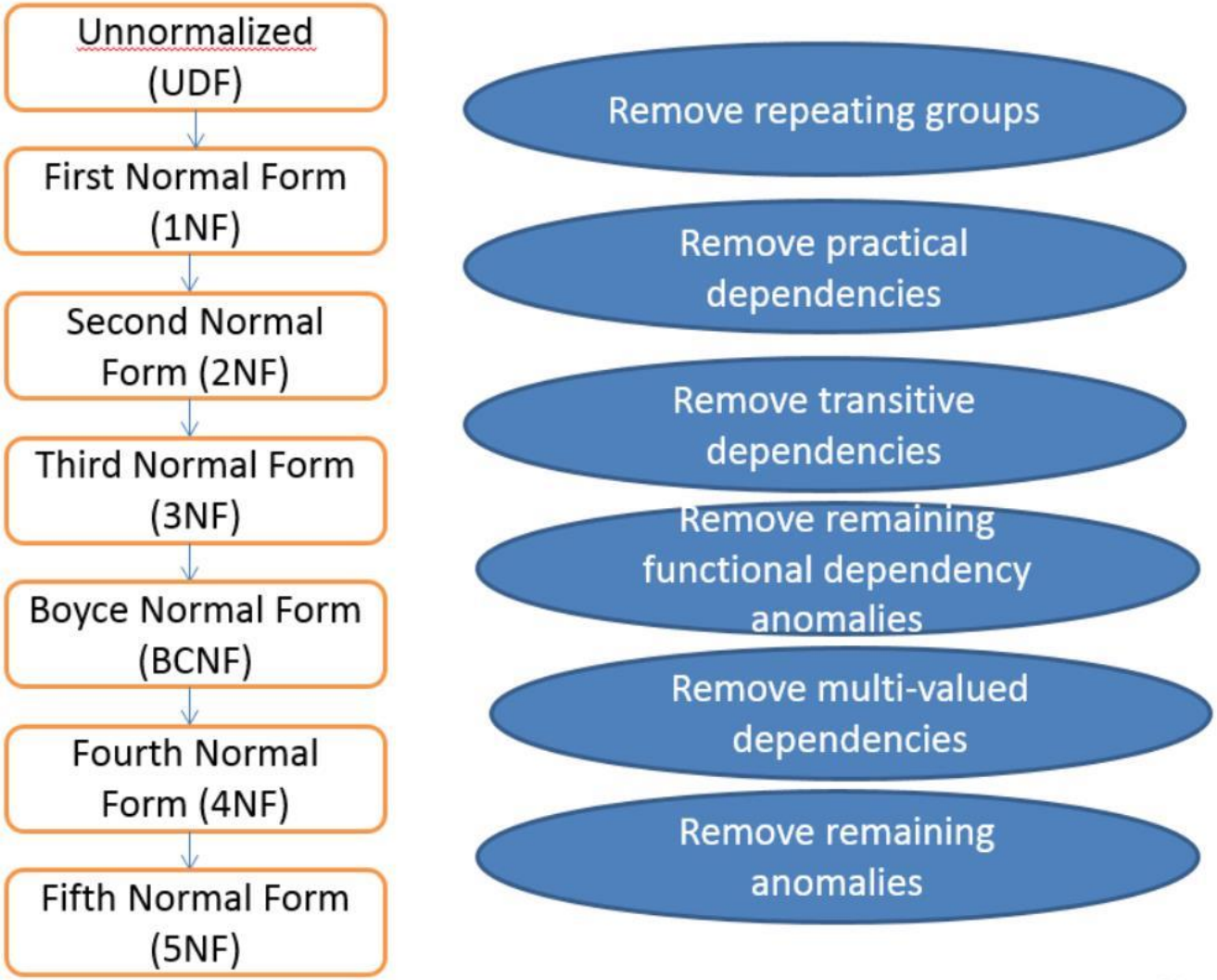
Normalization Process

Normalization is a formal technique used to evaluate the quality of a relational database schema. It determines whether a database schema contains any of the "wrong" kinds of redundancy and defines specific methods to eliminate them. Normalization is based on a concept called ***functional dependency*** *and on a series of* ***normal forms***

First normal form (1NF). A table is in first normal form if it contains no repeating fields or groups of fields.

- **Second normal form (2NF).** A table is in second normal form if it is in first normal form and if each non-key element is functionally dependent on the entire primary key.
- **Third normal form (3NF).** A table is in third normal form if it is in second normal form and if no non-key element is functionally dependent on any other non-key element.

steps of normalization



Functional Dependency

A *Functional Dependency* describes a relationship between *attributes* within a single relation. An attribute is *functionally dependent* on another if we can use the value of one attribute to determine the value of another.

Example:

Employee_Name is functionally dependent on Social_Security_Number because Social_Security_Number can be used to uniquely determine the value of Employee_Name.

We use the arrow symbol → to indicate a functional dependency. X →Y is **read *X functionally determines Y or Y is functionally dependent on X***

Functional Dependency

A *Functional Dependency* describes a relationship between *attributes* within a single relation. An attribute is *functionally dependent* on another if we can use the value of one attribute to determine the value of another.

Example:

Employee_Name is functionally dependent on Social_Security_Number because Social_Security_Number can be used to uniquely determine the value of Employee_Name.

We use the arrow symbol → to indicate a functional dependency. X →Y is **read *X functionally determines Y or Y is functionally dependent on X***

First Normal Form (1NF)

First normal form defines a structural constraint on table rows. Repeating fields such as **Dependent** in the Figure below are not allowed within any table in a relational database. Repeating groups of fields are also prohibited. In practice, this constraint is not difficult to enforce because relational DBMS do not allow a designer to define a table containing repeating fields.

EMPLOYEES						
SSN	Name	Department	Salary	Dependent 1	Dependent 2	Dependent 3
111-222	Uwera Caritas	IT	500,000	Alain	Claudine	
222-333	Kayiranga Alain	Accounting	460,000			
333-444	Kayitesi Daphrose	Administration	430,000	Peter	John	

If your ERD contains such a structure, consider creating **a new Dependent Table**

EMPLOYEES			
SSN	**Name**	**Department**	**Salary**
111-222	Uwera Caritas	IT	500,000
222-333	Kayiranga Alain	Accounting	460,000
333-444	Kayitesi Daphrose	Administration	430,000

DEPENDENTS		
Dependent_ID	**Name**	**SSN**
0001	Alain	111-222
0002	Claudine	111-222
0003	Peter	333-444
0004	John	333-444

Second Normal Form (2NF)

A table is in second normal form if it is in first normal form and if each non-key element is functionally dependent on the entire primary key.

It is generally more complex to verify the 2NF when the table has a composite primary key (two or more attributes as primary key). Let's consider the following table ORDERS:

ORDERS			
Product_ID	**Order_ID**	**Quantity**	**Order_Date**
P010	C012	12	05/06/2014
P012	C050	15	06/07/2014

Product_ID and Order_ID functionally determines the Quantity of the product, but not the Order_Date. The Order_Date is only functionally dependent on Order_ID.

We need to remove Order_Date from the table ORDERS and create another table.

ORDERS		
Product_ID	**Order_ID**	**Quantity**
P010	C012	12
P012	C050	15

ORDERS_INFO	
Order_ID	**Order_Date**
C012	05/06/2014
C050	06/07/2014

Third Normal Form (3NF)

A table is in third normal form if it is in second normal form and if no non-key element is functionally dependent on any other non-key element.

Let's consider the following table:

STUDENTS			
StudentID	Name	Department	Faculty
001	Uwimana Odile	IT	ICT
002	Mwiza Emery	CS	ICT
003	Makuza Augustin	IS	ICT
004	Niyigena Sosthene	C.Eng	Engineering

Faculty is functionally dependent on Department (Department →Faculty) i.e. if we know the Department, we can determine the Faculty. So we need to create another table for this dependency.

STUDENTS		
StudentID	**Name**	**Department**
001	Uwimana Odile	IT
002	Mwiza Emery	CS
003	Makuza Augustin	IS
004	Niyigena Sosthene	C.Eng

DEPARTMENTS	
Department	**Faculty**
IT	ICT
CS	ICT
IS	ICT
C.Eng	Engineering

Database Denormalization

After the normalization process we normally have a normalized LDM. This latter represents the database architecture. The tables and relations of the LDM are then implemented in a DBMS to create physical tables that compose the Physical Data Model (PDM) is then created. The PDM is also normalized. ***The denormalization process consists of splitting the physical tables of the PMD into non-normalized data structures after a querying process***.

*** The results of queries (*they are written in SQL*) are also presented in the form of tables i.e. columns and rows. These new structures are not normalized.

The Figure below depicts an example of denormalization.

CLIENTS

ClientID	SurName	FirstName	Telephone
001	Uwera	Diane	08754172
002	Kayiranga	Alain	08965216
003	Kayitesi	Aimée	08952141

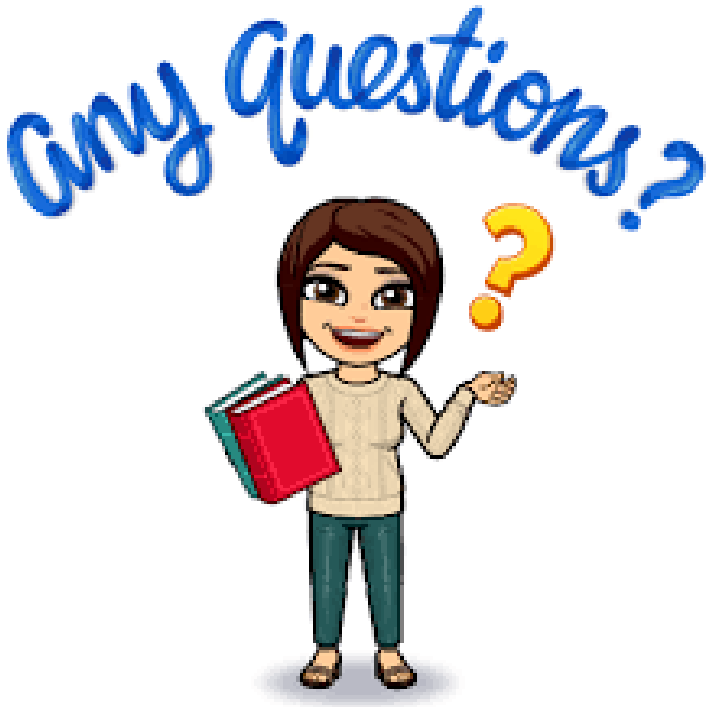
CARS

CarID	Manufacturer	Meter	ClientID
4698 SJ 45	Renault	123450	003
4568 HD 16	Toyota	56000	002
6576 VE 38	Benz	12000	001

DenormalizedTable:

SurName	FirstName	CarID	Manufacturer
Uwera	Diane	6576 VE 38	Benz
Kayiranga	Alain	8562 EV 23	Benz


THANK
YOU


Any questions?