4.1 Data Base Design

2nd Edition

Ian F. C. Smith
EPFL, Switzerland

Module Information

- **Intended audience**
  - Beginners
  - Intermediate

- **Key words**
  - Data storage, data manipulation
  - Relational data base
  - Functional dependencies
  - Normal forms
  - Update anomalies

- **Author**
  - Ian Smith, EPFL, Switzerland

- **Reviewers (1st Edition)**
  - Carlos Caldas, University of Texas Austin, USA
  - Guillermo Salazar, Worcester Polytechnic Institute, USA

Why are data bases important?

Almost every company uses them to store data.

Some companies can trace their success to efficient uses of data bases.

Data bases may contain the core knowledge for the majority of business operations.

Much information on the Web is stored in a data base.

Bad implementations may have serious consequences.

Introduction

The amount of digital information related to civil engineering is **increasing exponentially**. For example, it is now standard practice to have the following information in digital form:

- Results of design calculations
- Simulation data (structural analysis, traffic model simulations, energy use, landslides, snow accumulation, etc.)
- Drawings

Introduction (cont’d.)

- Measurement data (energy use, traffic, loading, deformations, corrosion, humidity, hydraulic information, etc.)
- Experimental results
- Geographical information
- Geological data
- Weather data
- Cost data
- Product models

Why Use a Data Base?

In a well designed data base, data is organized so that information retrieval is easy, reliable and **robust**.

A constant in civil engineering is that information change is inevitable.

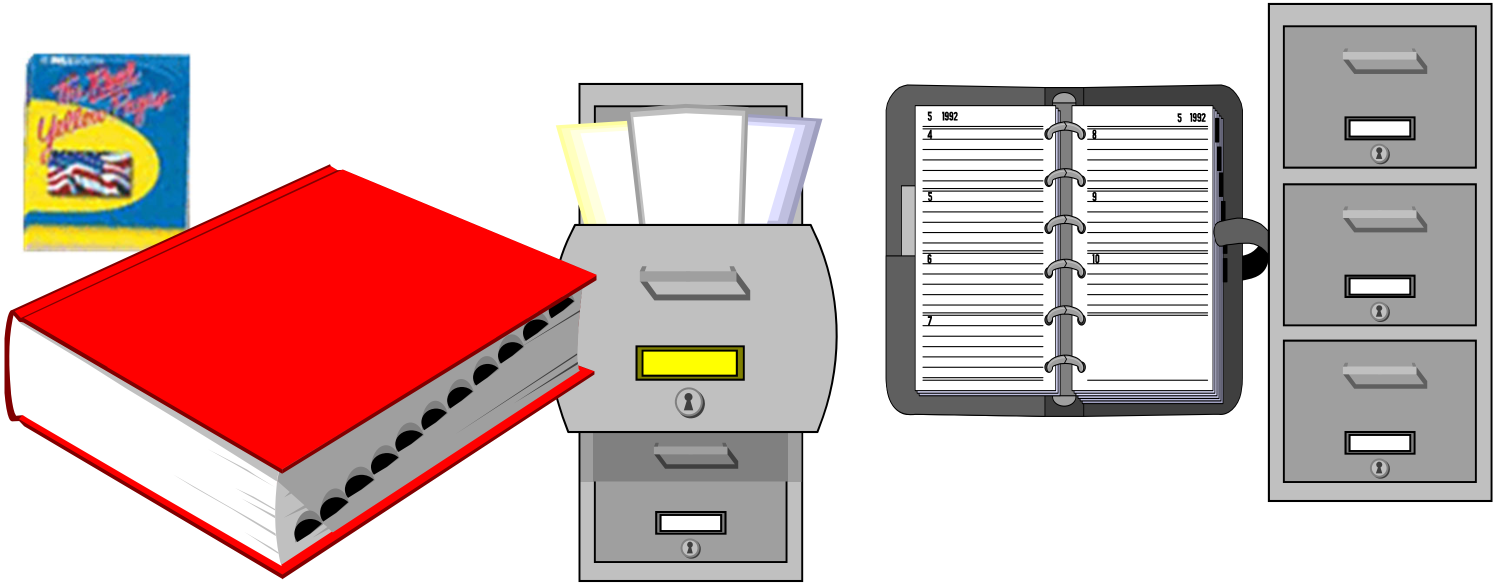
It is important to ensure that **data bases are robust so that data can be modified easily**. This module introduces techniques for good data-base design.

What is there to learn …

- Introduction to data bases
- Data base designs reflect **company practice**
- Engineers **cannot delegate** design to computer specialists
- How to **create** well designed data bases
- Good design **saves money** during updating
- Good design **avoids information loss**

What is a Data Base?

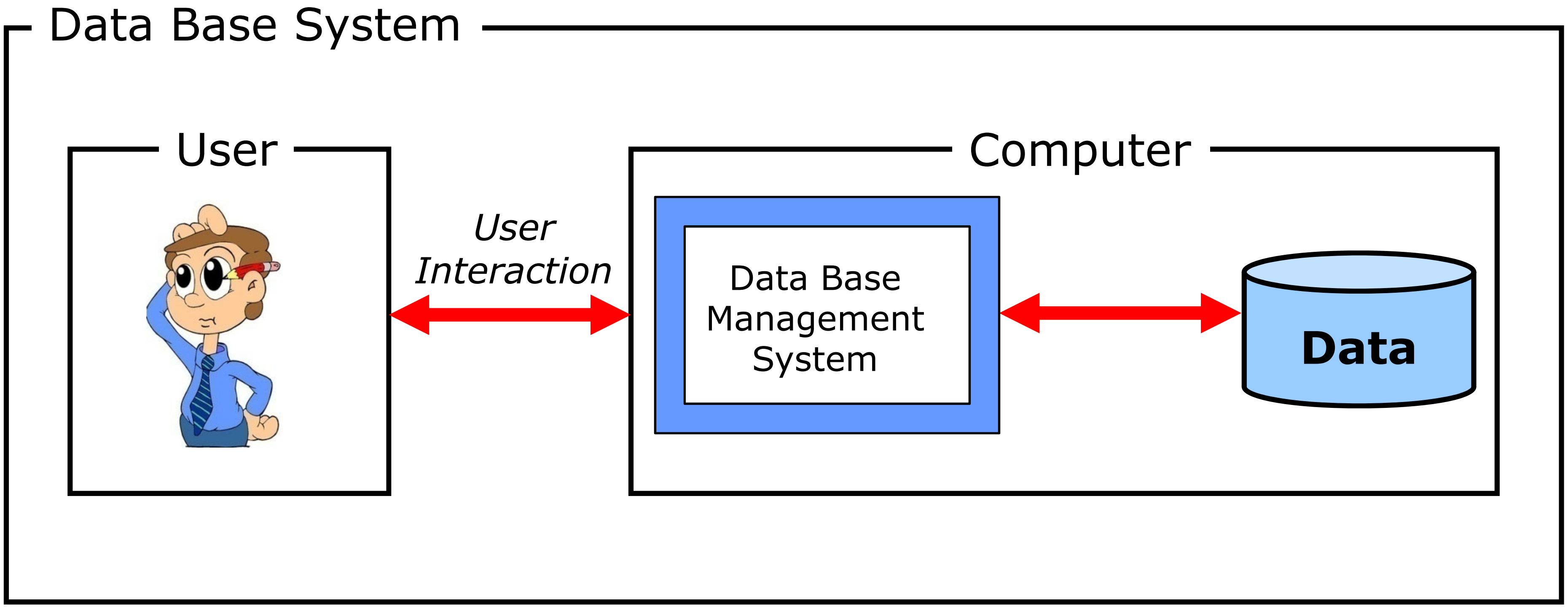
A **structured** collection of data



Why Have Structure in Data?

- Ease of **maintenance**
- Avoid **redundancy**
- Improve **efficiency when searching** for information

Data Base Systems



Types of Data Bases

- Hierarchical
- **Relational** (most widely used today)
- Object-oriented
- Logic-based
- Distributed
- Multimedia

This module focuses on **relational data bases**.

Relational: Mathematical term (not important for this introductory course)

Relational Data Base Systems

"*A model of data for large shared data banks.*"
Codd E. F. (1970)

Favorable conditions for applications

- Data can be placed in **tables**
- Large amounts of **structured data**
- Common operations: **finding the relevant entry**, finding all entries for one value of an attribute, ordering according to an attribute, etc.

Civil Engineering Examples

- Data from **past projects** in company records (used in this module)
- Drawing management
- Measurement data
- Load cases
- Topological data
- Material costs
- Product properties and dimensions

Data Base Design

Goals

- Provide representations that are **useful and intuitive** to those who will use them
- Avoid **redundancies**
- *Extensibility* and *robustness*: add, modify and delete data with as few side effects as possible. In other words, **minimize update anomalies.**

Data Base Design (cont'd.)

A **methodology** – steps to attain these goals

1. Understand how an organization works
2. Bring out **functional dependencies**
3. Aim for **highest normal form**
4. Make prototypes, test with users and iterate

Data Base Design

The first step is very important. Without a clear understanding of the data, its use and how it may change, data base design activities **may fail to meet the needs of its users.**

This course is aimed at improving understanding of the **Steps 2 and 3** of this methodology. This understanding will then be used to establish the strategic importance of **Step 1.**

Review Quiz - I

- Why are data bases important?
- What are three advantages of having structured data?
- What are good conditions for using a relational model?

Answers to Review Quiz - I

- Why are data bases important?

  In data bases, engineering data can be organized so that access and retrieval are easy, reliable and robust.

- What are three advantages of having structured data?
  - Ease of maintenance
  - No redundancy
  - Efficient search
- What are good conditions for using a relational model?

  Data can be organized in a structured form of two-dimensional tables (with columns of attributes and rows of records) and there is a need to retrieve information.

Outline

Introduction

Example

Functional Dependencies

Normal Forms

Why use Normal Forms?

A Civil Engineering Data Base

A consulting firm does design work on bridges and buildings. Often **past project information is hard to find and reuse** for new projects.

In order to reuse information from on-going and past projects, an engineer would like to create a data base that has **file locations** (server names) for design calculations and drawings for each part of each structure.

This example will illustrate **important** aspects of data base design.

Data Base Design (review)

A **methodology** – steps to attain these goals

1. Understand how an organization works
2. Bring out **functional dependencies**
3. Aim for **highest normal form**
4. Make prototypes, test with users and iterate

A Civil Engineering Data Base (cont'd.)

In this firm:

- Parts of a structure (foundations, piers, abutments, decks) can be designed in **different offices**
- For a given design part, design and drawing files are prepared in the same office
- All offices keep design files on one server and drawing files on another

A Civil Engineering Data Base (cont’d.)

Important characteristics of working procedures are

hidden

in data base designs.

A Civil Engineering Data Base (cont'd.)

Structure	Owner	Design Part	Office	Design File Location	Drawing File Location
School A	X	Foundation	Detroit	Server 1	Server 2
Bank A	Y	Steel Structure	Atlanta	Server 2	Server 3
Bridge D	Z	Piers	Houston	Server 3	Server 2
Bridge D	Z	Deck	Portland	Server 1	Server 3
Office C	P	Top floors	Atlanta	Server 2	Server 3
Bridge F	Q	Abutments	Boston	Server 2	Server 1

The real data base may have thousands of entries.

In the table above, the first row contains the **names of attributes**. Other rows contain **values** for these attributes.

Outline

Introduction

Example

Functional Dependencies

Normal Forms

Why use Normal Forms?

Functional Dependencies

When the value for attribute ***y*** *always* defines the value for attribute ***z***, we say that ***z* is functionally dependent on *y***.

Algebraically:

$$\boldsymbol{y \rightarrow z}$$

In other words, ***y*** functionally determines ***z***.

Note: **y** and **z** may represent multiple attributes.

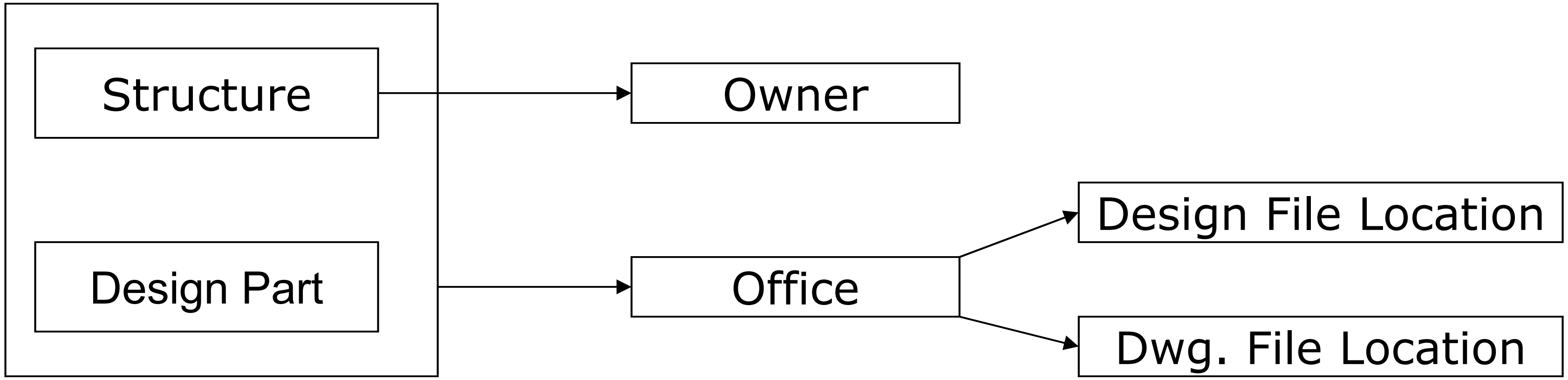
Terminology

Primary key: An attribute in a table is called the primary key if its values *uniquely* identify the other values in the table.

Composite primary key : The combination of two or more values in attributes uniquely identifies the other values in the table.

Functional Dependencies (cont’d.)

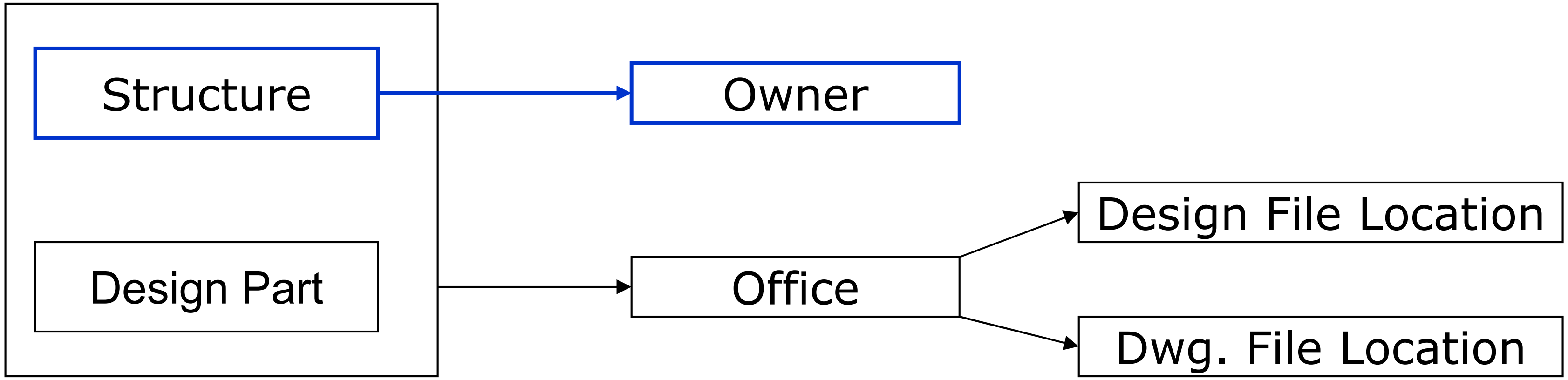
For the example shown earlier, the complete **functional dependency graph** is shown as follows:



The attributes, Structure and Design Part, form a possible **composite primary key**. The next slides describe important aspects of this graph.

Functional Dependencies (cont’d.)

Structure alone uniquely determines ***Owner***



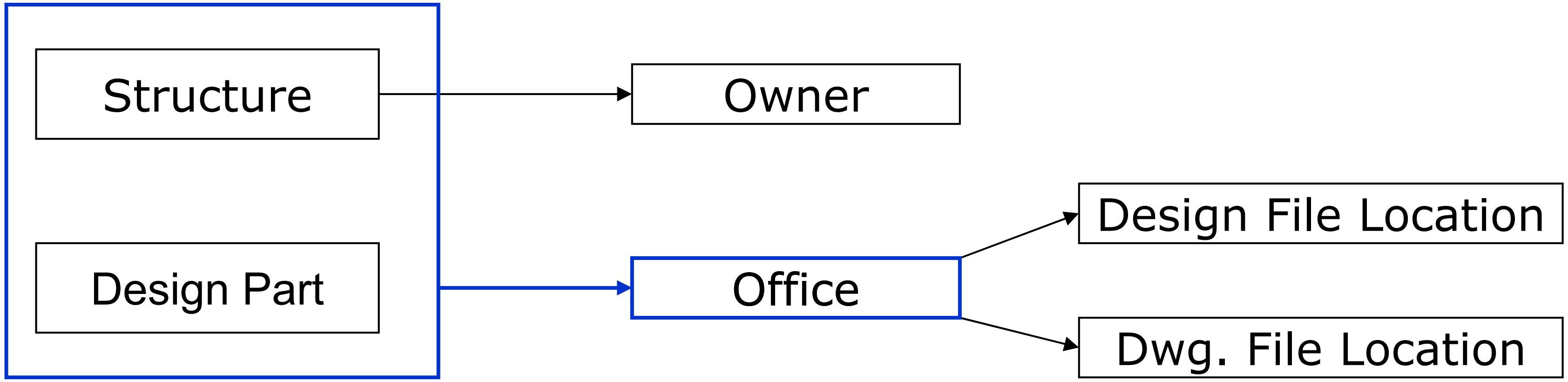
Functional Dependencies (cont’d.)

Structure alone uniquely determines ***Owner***

Structure	**Owner**	**Design Part**	**Office**	**Design File Location**	**Drawing File Location**
School A	**X**	Foundation	Detroit	Server 1	Server 2
Bank A	**Y**	Steel Structure	Atlanta	Server 2	Server 3
Bridge D	**Z**	Piers	Houston	Server 3	Server 2
Bridge D	**Z**	Deck	Portland	Server 1	Server 3
Office C	**P**	Top floors	Atlanta	Server 2	Server 3
Bridge F	**Q**	Abutments	Boston	Server 2	Server 1

Functional Dependencies (cont'd.)

Structure & Design Part uniquely determine ***Office***



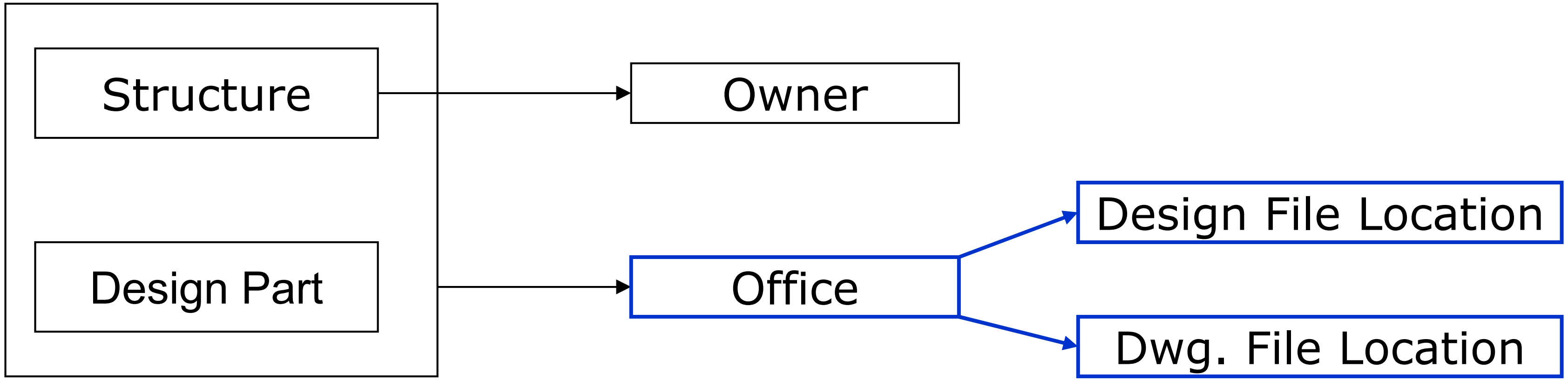
Functional Dependencies (cont'd.)

Structure & Design Part uniquely determine ***Office***

Structure	**Owner**	**Design Part**	**Office**	**Design File Location**	**Drawing File Location**
School A	X	**Foundation**	**Detroit**	Server 1	Server 2
Bank A	Y	**Steel Structure**	**Atlanta**	Server 2	Server 3
Bridge D	Z	**Piers**	**Houston**	Server 3	Server 2
Bridge D	Z	**Deck**	**Portland**	Server 1	Server 3
Office C	P	**Top floors**	**Atlanta**	Server 2	Server 3
Bridge F	Q	**Abutments**	**Boston**	Server 2	Server 1

Functional Dependencies (cont'd.)

Office uniquely determines ***Design File Location*** as well as ***Drawing File Location***



Functional Dependencies (cont'd.)

Office alone uniquely determines ***Design File Location*** and ***Drawing File Location***

Structure	**Owner**	**Design Part**	**Office**	**Design File Location**	**Drawing File Location**
School A	X	Foundation	**Detroit**	**Server 1**	**Server 2**
Bank A	Y	Steel Structure	**Atlanta**	**Server 2**	**Server 3**
Bridge D	Z	Piers	**Houston**	**Server 3**	**Server 2**
Bridge D	Z	Deck	**Portland**	**Server 1**	**Server 3**
Office C	P	Top floors	**Atlanta**	**Server 2**	**Server 3**
Bridge F	Q	Abutments	**Boston**	**Server 2**	**Server 1**

Outline

Introduction

Example

Functional Dependencies

Normal Forms

Why use Normal Forms?

Normal Forms

This course describes three normal forms

- **First Normal Form (1NF)**
- **Second Normal Form (2NF)**
- **Third Normal Form (3NF)**

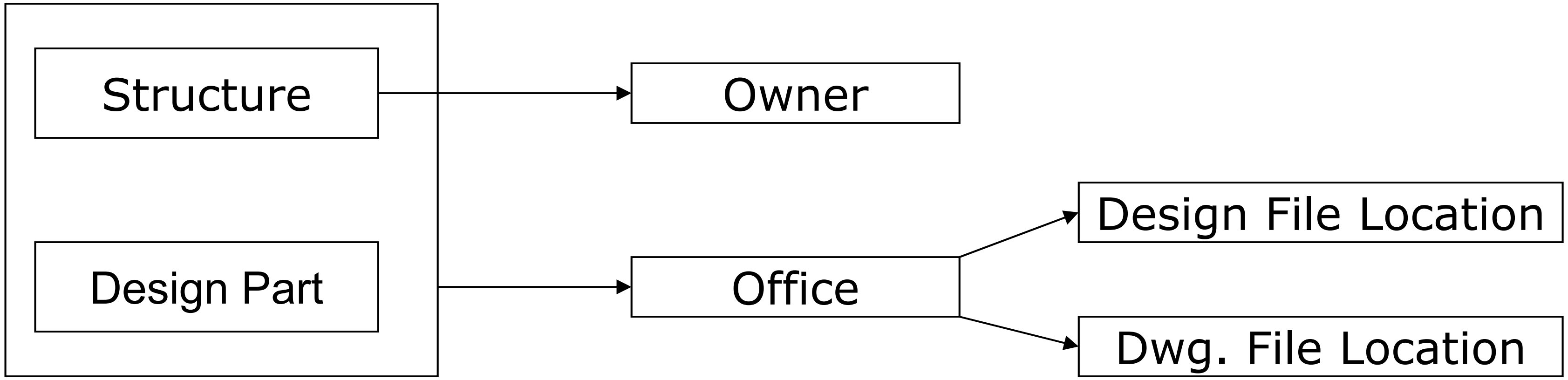
Each increment in form includes requirements of the previous form.

For this course, the highest possible normal form is given to the examples.

Normal Forms (cont'd.)

A data base is said to be in

- **First Normal Form (1NF)**: if it contains only scalar (simple) values and not, for example, nested tables.



Normal Forms (cont'd.)

Our original example is in **1NF**.

Structure	**Owner**	**Design Part**	**Office**	**Design File Location**	**Drawing File Location**
School A	X	Foundation	Detroit	Server 1	Server 2
Bank A	Y	Steel Structure	Atlanta	Server 2	Server 3
Bridge D	Z	Piers	Houston	Server 3	Server 2
Bridge D	Z	Deck	Portland	Server 1	Server 3
Office C	P	Top floors	Atlanta	Server 2	Server 3
Bridge F	Q	Abutments	Boston	Server 2	Server 1

In 1NF, three types of update anomalies may occur.

1NF Update Anomalies

Modification

If the owner of a structure changes, many places in the data base need to be modified.

For instance, if Bridge D changes hands from Company Z to Company XY^{new}, changes have to be made at each instance of Bridge D.

In a data base with thousands of entries distributed on many computers, modifications can be costly to ensure.

1NF Update Anomalies

Modification

Before

Structure	Owner	Design Part	Office	Design File Location	Drawing File Location
School A	X	Foundation	Detroit	Server 1	Server 2
Bank A	Y	Steel structure	Atlanta	Server 2	Server 3
Bridge D	**Z**	Piers	Houston	Server 3	Server 2
Bridge D	**Z**	Deck	Portland	Server 1	Server 3
Office C	P	Top floors	Atlanta	Server 2	Server 3
Bridge F	Q	Abutments	Boston	Server 2	Server 1

1NF Update Anomalies

Modification

After

Structure	**Owner**	**Design Part**	**Office**	**Design File Location**	**Drawing File Location**
School A	X	Foundation	Detroit	Server 1	Server 2
Bank A	Y	Steel structure	Atlanta	Server 2	Server 3
Bridge D	**XY$^{\text{new}}$**	Piers	Houston	Server 3	Server 2
Bridge D	**XY$^{\text{new}}$**	Deck	Portland	Server 1	Server 3
Office C	P	Top floors	Atlanta	Server 2	Server 3
Bridge F	Q	Abutments	Boston	Server 2	Server 1

1NF Update Anomalies

Deletion

If a design part is subcontracted, this deletion could lead to **loss of information.**

For example, if the foundation for School A is subcontracted, then we lose the information that there X owns school A.

1NF Update Anomalies

Deletion

Before

Structure	Owner	Design Part	Office	Design File Location	Drawing File Location
School A	**X**	**Foundation**	**Detroit**	**Server 1**	**Server 2**
Bank A	Y	Steel structure	Atlanta	Server 2	Server 3
Bridge D	Z	Piers	Houston	Server 3	Server 2
Bridge D	Z	Deck	Portland	Server 1	Server 3
Office C	P	Top floors	Atlanta	Server 2	Server 3
Bridge F	Q	Abutments	Boston	Server 2	Server 1

1NF Update Anomalies

Deletion

After

Structure	Owner	Design Part	Office	Design File Location	Drawing File Location
Bank A	Y	Steel structure	Atlanta	Server 2	Server 3
Bridge D	Z	Piers	Houston	Server 3	Server 2
Bridge D	Z	Deck	Portland	Server 1	Server 3
Office C	P	Top floors	Atlanta	Server 2	Server 3
Bridge F	Q	Abutments	Boston	Server 2	Server 1

1NF Update Anomalies

Insertion

If there is a new project and further details, such as which office is handling which design part, are not yet decided, it cannot be added to the data base.

If Building R owned by Owner S has been allotted, this information cannot be added to the data base in 1NF until all other information is known to complete the row.

This could mean that we would not know that a new project, Building R, is beginning.

1NF Update Anomalies

Insertion

Before

Structure	Owner	Design Part	Office	Design File Location	Drawing File Location
School A	X	Foundation	Detroit	Server 1	Server 2
Bank A	Y	Steel structure	Atlanta	Server 2	Server 3
Bridge D	Z	Piers	Houston	Server 3	Server 2
Bridge D	Z	Deck	Portland	Server 1	Server 3
Office C	P	Top floors	Atlanta	Server 2	Server 3
Bridge F	Q	Abutments	Boston	Server 2	Server 1

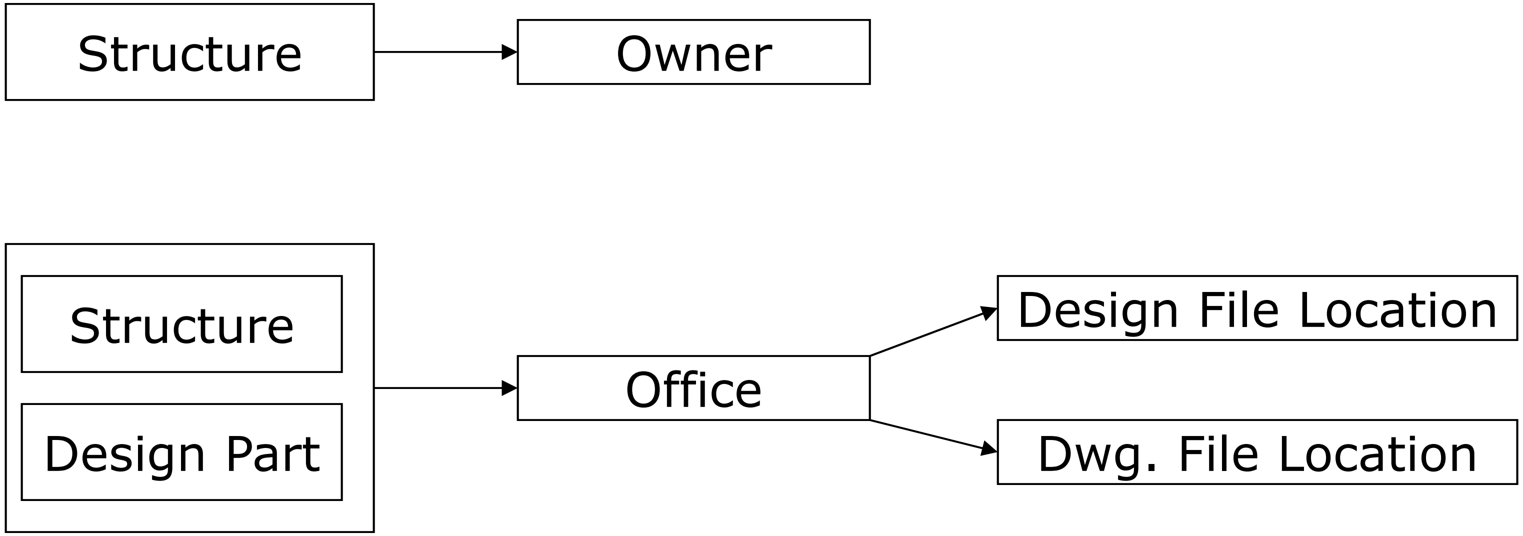
1NF Update Anomalies

Insertion After

Structure	**Owner**	**Design Part**	**Office**	**Design File Location**	**Drawing File Location**
School A	X	Foundation	Detroit	Server 1	Server 2
Bank A	Y	Steel structure	Atlanta	Server 2	Server 3
Bridge D	Z	Piers	Houston	Server 3	Server 2
Bridge D	Z	Deck	Portland	Server 1	Server 3
Office C	P	Top floors	Atlanta	Server 2	Server 3
Bridge F	Q	Abutments	Boston	Server 2	Server 1
Building R	**S**	*error*	*error*	*error*	*error*

Normal Forms (cont'd.)

- **Second Normal Form (2NF)**: if the data base is in 1NF and if each non-key attribute depends on a complete key attribute.



Our example, split into two tables, is now in **2NF**.

Normal Forms (cont'd.)

Our example is now in **2NF**.

Structure	**Owner**
School A	X
Bank A	Y
Bridge D	Z
Office C	P
Bridge F	Q

Structure	**Design Part**	**Office**	**Design File Location**	**Drawing File Location**
School A	Foundation	Detroit	Server 1	Server 2
Bank A	Steel Structure	Atlanta	Server 2	Server 3
Bridge D	Piers	Houston	Server 3	Server 2
Bridge D	Deck	Portland	Server 1	Server 3
Office C	Top floors	Atlanta	Server 2	Server 3
Bridge F	Abutments	Boston	Server 2	Server 1

Normal Forms (cont'd.)

A data base in 2NF overcomes the update anomalies that were identified in its 1NF form.

However …

the same three kinds of anomalies may occur in 2NF as well!!!

2NF Update Anomalies

Modification

If the server details for any office change, these changes have to be reflected at each location in the data base where that office shows up.

For instance, if in Atlanta office the Design File Location changes from Server 2 to Server 4, changes have to be made at **each instance** of Atlanta.

2NF Update Anomalies

Modification

Structure	Owner
School A	X
Bank A	Y
Bridge D	Z
Office C	P
Bridge F	Q

Before

Structure	Design Part	Office	Design File Location	Drawing File Location
School A	Foundation	Detroit	Server 1	Server 2
Bank A	Steel Structure	**Atlanta**	**Server 2**	Server 3
Bridge D	Piers	Houston	Server 3	Server 2
Bridge D	Deck	Portland	Server 1	Server 3
Office C	Top floors	**Atlanta**	**Server 2**	Server 3
Bridge F	Abutments	Boston	Server 2	Server 1

2NF Update Anomalies

Modification

Structure	Owner
School A	X
Bank A	Y
Bridge D	Z
Office C	P
Bridge F	Q

After

Structure	Design Part	Office	Design File Location	Drawing File Location
School A	Foundation	Detroit	Server 1	Server 2
Bank A	Steel Structure	**Atlanta**	**Server 4**	Server 3
Bridge D	Piers	Houston	Server 3	Server 2
Bridge D	Deck	Portland	Server 1	Server 3
Office C	Top floors	**Atlanta**	**Server 4**	Server 3
Bridge F	Abutments	Boston	Server 2	Server 1

2NF Update Anomalies

Deletion

If there is subcontracting of designs and drawings, this deletion may lead to loss of information.

For example, if the designs of the piers for Bridge D are subcontracted, we **lose the information** related to where the files at Houston are stored.

2NF Update Anomalies

Deletion

Structure	Owner
School A	X
Bank A	Y
Bridge D	Z
Office C	P
Bridge F	Q

Before

Structure	Design Part	Office	Design File Location	Drawing File Location
School A	Foundation	Detroit	Server 1	Server 2
Bank A	Steel Structure	Atlanta	Server 2	Server 3
Bridge D	**Piers**	**Houston**	**Server 3**	**Server 2**
Bridge D	Deck	Portland	Server 1	Server 3
Office C	Top floors	Atlanta	Server 2	Server 3
Bridge F	Abutments	Boston	Server 2	Server 1

2NF Update Anomalies

Deletion

Structure	Owner
School A	X
Bank A	Y
Bridge D	Z
Office C	P
Bridge F	Q

After

Structure	Design Part	Office	Design File Location	Drawing File Location
School A	Foundation	Detroit	Server 1	Server 2
Bank A	Steel Structure	Atlanta	Server 2	Server 3
Bridge D	Deck	Portland	Server 1	Server 3
Office C	Top floors	Atlanta	Server 2	Server 3
Bridge F	Abutments	Boston	Server 2	Server 1

2NF Update Anomalies

Insertion

If a new office is acquired, this additional information cannot be added to the data base until it is doing a project.

For example, if an office in New York is acquired, the file location information cannot be added until New York begins a project for the parent company.

2NF Update Anomalies

Insertion

Structure	Owner
School A	X
Bank A	Y
Bridge D	Z
Office C	P
Bridge F	Q

Before

Structure	Design Part	Office	Design File Location	Drawing File Location
School A	Foundation	Detroit	Server 1	Server 2
Bank A	Steel Structure	Atlanta	Server 2	Server 3
Bridge D	Piers	Houston	Server 3	Server 2
Bridge D	Deck	Portland	Server 1	Server 3
Office C	Top floors	Atlanta	Server 2	Server 3
Bridge F	Abutments	Boston	Server 2	Server 1

2NF Update Anomalies

Insertion

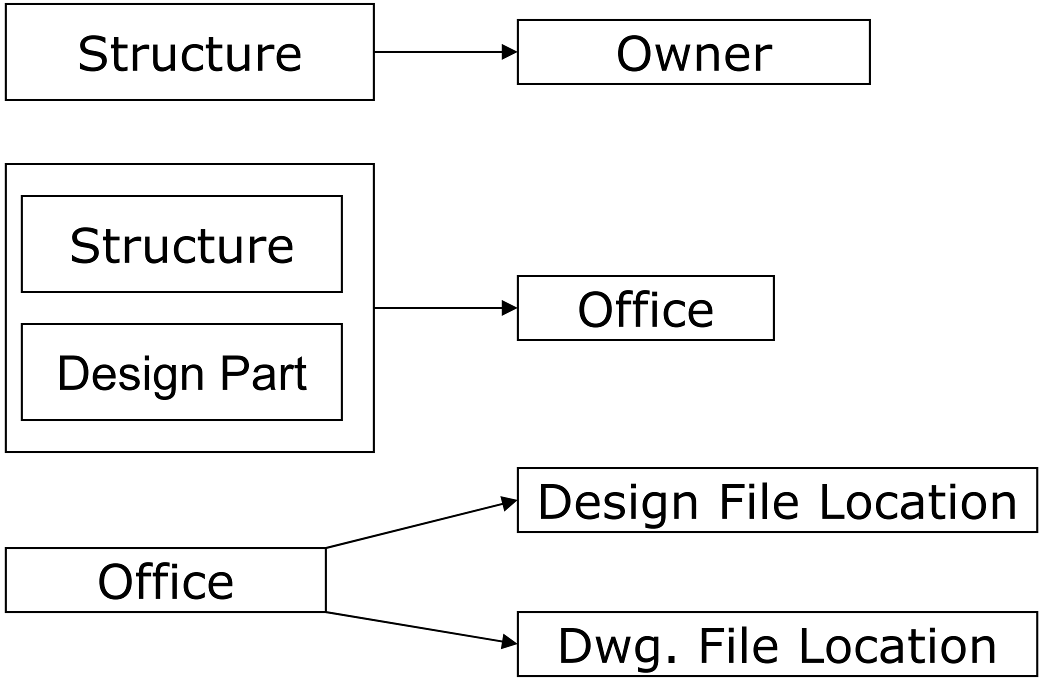
Structure	Owner
School A	X
Bank A	Y
Bridge D	Z
Office C	P
Bridge F	Q

After

Structure	Design Part	Office	Design File Location	Drawing File Location
School A	Foundation	Detroit	Server 1	Server 2
Bank A	Steel Structure	Atlanta	Server 2	Server 3
Bridge D	Piers	Houston	Server 3	Server 2
Bridge D	Deck	Portland	Server 1	Server 3
Office C	Top floors	Atlanta	Server 2	Server 3
Bridge F	Abutments	Boston	Server 2	Server 1
error	*error*	**New York**	**Server 5**	**Server 7**

Normal Forms (cont'd.)

- **Third Normal Form (3NF)**: if it is in 2NF and if each non-key column is **directly dependent** on the primary key column.



Normal Forms (cont'd.)

Through designing in the Third Normal Form we reduce many risks associated with changing information.

Higher forms exist when there are several candidate composite primary keys. This is not within the scope of this course.

Normal Forms (cont’d.)

Our example is now in **3NF**.

Structure	**Owner**
School A	X
Bank A	Y
Bridge D	Z
Office C	P
Bridge F	Q

Structure	**Design Part**	**Office**
School A	Foundation	D
Bank A	Steel Structure	At
Bridge D	Piers	H
Bridge D	Deck	P
Office C	Top floors	At
Bridge F	Abutments	B

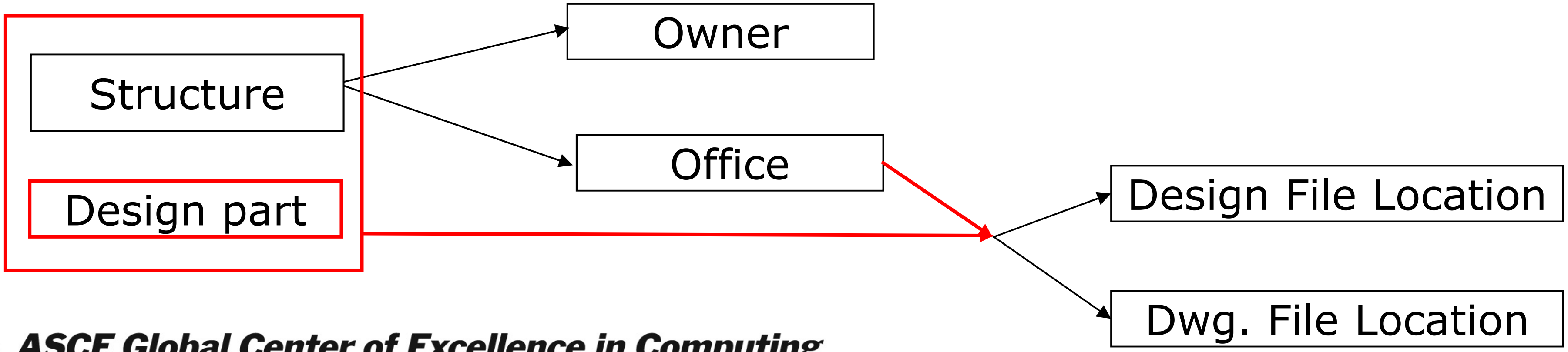
Office	**Design File Location**	**Drawing File Location**
Detroit	Server 1	Server 2
Atlanta	Server 4	Server 3
Houston	Server 3	Server 2
Portland	Server 1	Server 3
Boston	Server 2	Server 1
New York	Server 5	Server 7

Functional Dependencies (revisited)

In another firm, the dependencies are different.

All design parts for a given structure are designed in the **same office**. Also within an office, design and drawing information are on many servers (not just two as before). Consequently, values for ***Structure*** *(Bridge D, Bank A, School A)* uniquely determine values for ***Office.***

New dependency graph



Functional Dependencies (revisited)

Second company in **3NF**

Structure	**Owner**	**Office**
School A	X	Detroit
Bank A	Y	Atlanta
Bridge D	Z	Housto
Office C	P	Atlanta
Bridge F	Q	Boston

Structure	**Office**	**Design Part**	**Design File Location**	**Drawing File Location**
School A	Detroit	Foundation	Server 1	Server 2
Bank A	Atlanta	Steel Structure	Server 2	Server 3
Bridge D	Houston	Piers	Server 3	Server 2
Bridge D	Houston	Deck	Server 1	Server 3
Office C	Atlanta	Top floors	Server 2	Server 1
Bridge F	Boston	Abutments	Server 2	Server 1

Functional Dependencies (revisited)

The 3NF tables are **different** in each new situation

Functional dependencies reflect the way a **company conducts business**.

Computer specialists may not be aware of important dependencies!

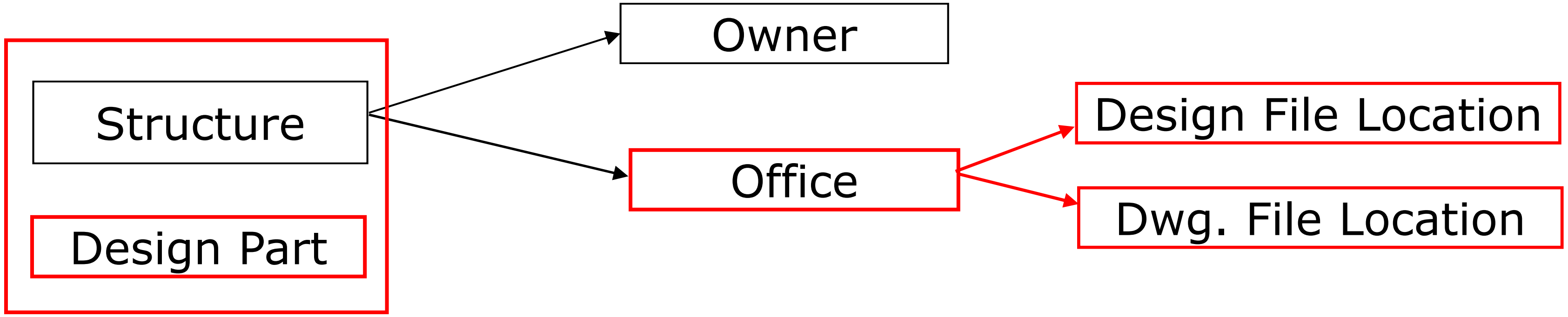
If functional dependencies are wrongly identified, the third (incorrect) normal form will not guard against update anomalies.

Functional Dependencies (revisited)

In a third firm, dependencies change again.

There is only one location for design files and one for drawing files in a given office. Only the values of ***Office*** determine the ***file locations***, independent of the values of *Design Part*.

New dependency graph



Functional Dependencies (revisited)

Third company in **3NF**

Structure	Owner	Office
School A	X	Detroit
Bank A	Y	Atlanta
Bridge D	Z	Houston
Office C	P	Atlanta
Bridge F	Q	Boston

Structure	Design Part
School A	Foundation
Bank A	Steel Structure
Bridge D	Piers
Bridge D	Deck
Office C	Top floors
Bridge F	Abutments

Office	Design File Location	Drawing File Location
Detroit	Server 1	Server 2
Atlanta	Server 4	Server 3
Houston	Server 3	Server 2
Portland	Server 1	Server 3
Boston	Server 2	Server 1

Functional Dependencies (revisited)

⚠ **It is essential that engineers are involved in designing data bases for their projects.**

Functional Dependencies (revisited)

The most reliable design strategy is to **develop dependency graphs within teams** of engineers and computer specialists.

Engineers need to appreciate the importance of functional dependencies in order to **help specialists do the best job.**

Review Quiz - II

- Define
  - Primary key
  - Composite key

- Name three update anomalies that can occur in the first and second normal forms?

- Which normal form is the best? Why?

- What inherent information do functional dependencies contain?

Answers to Review Quiz – II

- Define
  - Primary key

    An attribute is called a primary key if their values uniquely identifies the row.
  - Composite key

    Two or more attributes form a composite key if their values uniquely identifies the row.
- Name three update anomalies that can occur in the first and second normal forms?
  - Modification
  - Deletion
  - Insertion

Answers to Review Quiz – II

- Which normal form is the best? Why?

  The third normal form (3NF) is best because a data base in 3NF is less at risk of update anomalies.

- What inherent information do functional dependencies contain?

  Information related to business processes is inherently represented by functional dependencies.

Outline

Introduction

Example

Functional Dependencies

Normal Forms

Why use Normal Forms?

Why Use Normal Forms?

- Databases with higher normal forms are **easier to modify**. The main risks that arise during modification are information loss and data inconsistencies. These are called **update anomalies.**

- Lower normal forms have dependencies which contain **information that could be lost** upon deletion of records. For example, partial dependencies and transitive dependencies are lost when records containing unique attribute values are eliminated.

Why Use Normal Forms? (cont'd.)

- Consistency problems arise when new records introduce values that contradict existing values. For example, in the 1NF and 2NF (first example) it was possible to name **two different servers** for design information at an office.

- Finally, redundancy is possible. For example, in a 2NF relation, transitive dependencies may mean that the same information is present in several records. Therefore modifications require changes to every relevant record.

Why Use Normal Forms?

- Data management **requirements exist everywhere**.
- Data should be organized so that they are easily modifiable, **without update anomalies**
- The most widely used DB type is the relational DB.
- Good DB design requires a **sound knowledge of company behavior** in order to identify correct functional dependencies among data types.
- Data base designers should aim to create data bases in highest possible normal form.

Further Reading

- Date, C. J. *An Introduction to Database Management Systems*, Addison Wesley, 1995
- Bhavani M. T., *Data Management Systems*, CRC Press, 1997
- Raphael, B. and Smith, I.F.C. *Engineering Informatics - Fundamentals of Computer-Aided Engineering*, Wiley, 2013