Database Technology

Topic 1: Introduction

Olaf Hartig
olaf.hartig@liu.se

LINKÖPING UNIVERSITY

Outline

1. Basic Terminology
2. The Database Approach
3. Using a Database System

4. Logistics of the Course

Basic Terminology

Most Basic Terminology

- **Data**: known facts that can be recorded
  and that have implicit meaning

- **Database**: collection of related data (logically coherent)
  - Represents some aspects of the real world (**miniworld**)
  - Built for a specific purpose

- Examples of large databases
  - Amazon.com's product data
  - Data collection underlying Webreg

Example of a Database

COURSE

Course_name	Course_number	Credit_hours	Department
Intro to Computer Science	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Database	CS3380	3	CS

SECTION

Section_identifier	Course_number	Semester	Year	Instructor
85	MATH2410	Fall	04	King
92	CS1310	Fall	04	Anderson
102	CS3320	Spring	05	Knuth
112	MATH2410	Fall	05	Chang
119	CS1310	Fall	05	Anderson
135	CS3380	Fall	05	Stone

GRADE_REPORT

Student_number	Section_identifier	Grade
17	112	B
17	119	C
8	85	A
8	92	A
8	102	B
8	135	A

PREREQUISITE

Course_number	Prerequisite_number
CS3380	CS3320
CS3380	MATH2410
CS3320	CS1310

Example from “Fundamentals of Database Systems” by Elmasri and Navathe, Addison Wesley.

Terminology (cont'd)

- **Database management system** (**DBMS**)
  - Collection of computer programs
  - Enables users to create and maintain a database (DB)
  - Supports concurrent access to a database by multiple users and programs
  - Protects the DB against unauthorized access and manipulation
  - Provides means to evolve the DB as requirements change

- Examples of database management systems
  - IBM's DB2, Microsoft's Access, Microsoft's SQL Server, Oracle, SAP's SQL Anywhere, MySQL, PostgreSQL

Database System

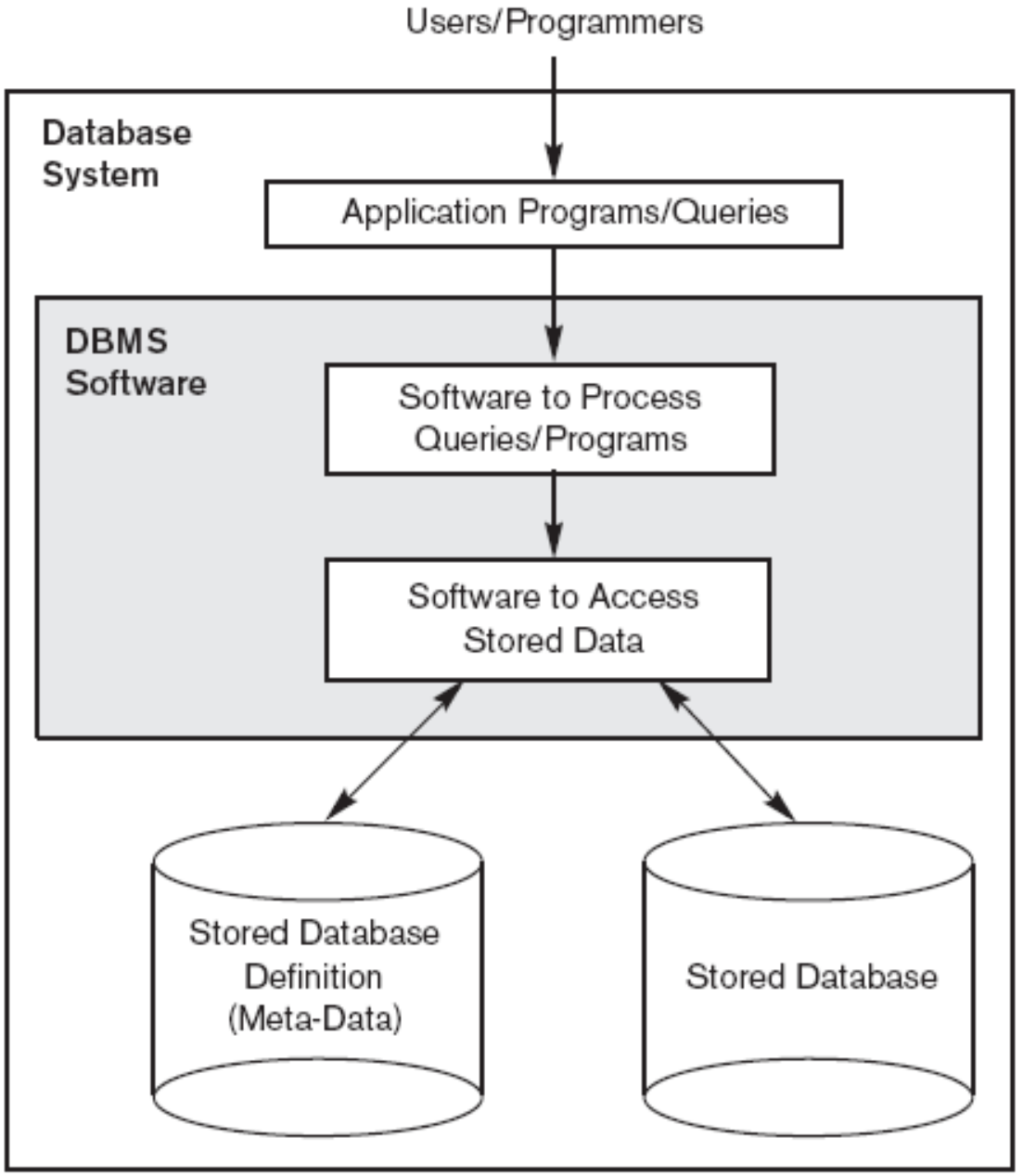


Figure from "Fundamentals of Database Systems" by Elmasri and Navathe, Addison Wesley.

The Database Approach

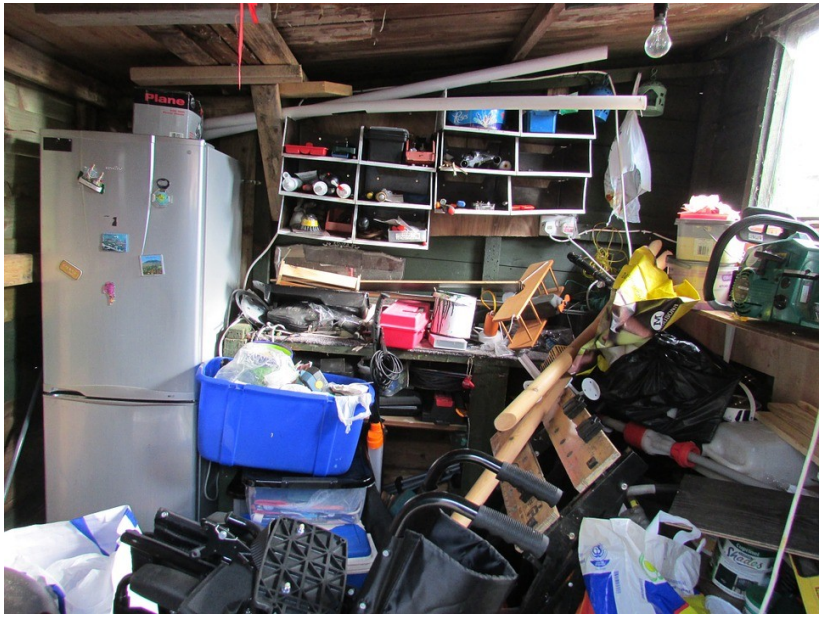
Pre-DBMS Data Management

- Used traditional **file processing**
  - Each user defines and implements the files needed for a specific software application

https://www.goodfreephotos.com/albums/other-photos/boxes-and-boxes-moving-storage.jpg

- As the application base grows
  - many shared files
  - a multitude of file structures
  - a need to exchange data among applications

Problems of Pre-DBMS Data Management



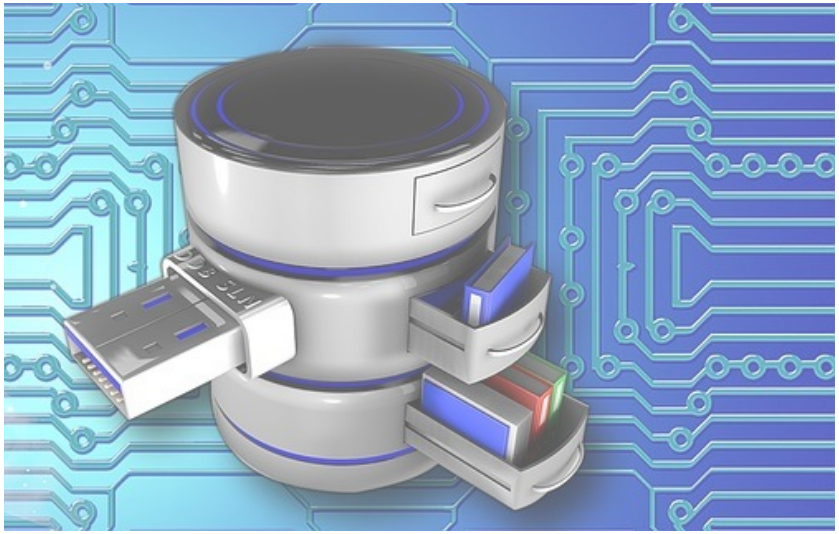
https://cdn.pixabay.com/photo/2014/06/01/22/26/clutter-360058_960_720.jpg

- Redundancy: multiple copies
- Inconsistency: independent updates
- Inaccuracy: concurrent updates
- Incompatibility: multiple formats
- Insecurity: proliferation
- Inauditability: poor chain of responsibility
- Inflexibility: changes are difficult to apply

Database Approach

- Eventually recognized that data is a critical corporate asset (along with capital and personnel)
  - Need to manage the data in a more systematic manner

- Database approach: Use a *single repository* to maintain data that is defined once and accessed by various users
  - Addresses the aforementioned problems



https://cdn.pixabay.com/photo/2017/06/12/04/21/database-2394312_960_720.jpg

Characteristics of the Database Approach

- Programs isolated from data through **abstraction**
  - DBMS does not expose details of how (or where) data is stored or how operations are implemented
  - Programs refer to an abstract model of the data, rather than data storage details
  - Data structures and storage organization can be changed without having to change the application programs
- Support of multiple **views** of the data
  - Different users may see different views of the database,which contain only the data of interest to these users
- Multi-user **transaction** processing
  - Encapsulates sequence of operations to behave atomically
    - e.g., transferring funds

Characteristics of the Database Approach

- Data is **self-describing**
  - Database system contains a *database catalog* with meta-data that describes structure and constraints of the database(s)
  - Database catalog used by DBMS, and by DB users who need information about DB structure
  - Example:

RELATIONS

Relation_name	No_of_columns
STUDENT	4
COURSE	4
SECTION	5
GRADE_REPORT	3
PREREQUISITE	2

COLUMNS

Column_name	Data_type	Belongs_to_relation
Name	Character (30)	STUDENT
Student_number	Character (4)	STUDENT
Class	Integer (1)	STUDENT
Major	Major_type	STUDENT
Course_name	Character (10)	COURSE
Course_number	XXXXNNNN	COURSE
….	….	…..
….	….	…..
….	….	…..
Prerequisite_number	XXXXNNNN	PREREQUISITE

Example from "Fundamentals of Database Systems" by Elmasri and Navathe, Addison Wesley.

Using a Database System

Defining a Database

- Specifying the data types, structures, and constraints of the data to be stored
- Uses a *Data Definition Language (DDL)*
- **Meta-data**: Database definition or descriptive information
  - Stored by the DBMS in a *database catalog* or *data dictionary*

- Phases for designing a database:
  - **Requirements specification and analysis**
  - **Conceptual design**
    - e.g., using the *Entity-Relationship model*
  - **Logical design**
    - e.g., using the *relational model*
  - **Physical design**

Database System Design Process

- Two main activities:
  - **Database design** focuses on defining the database
  - **Application design** focuses on the programs and interfaces that access the database (out of scope of this lecture)

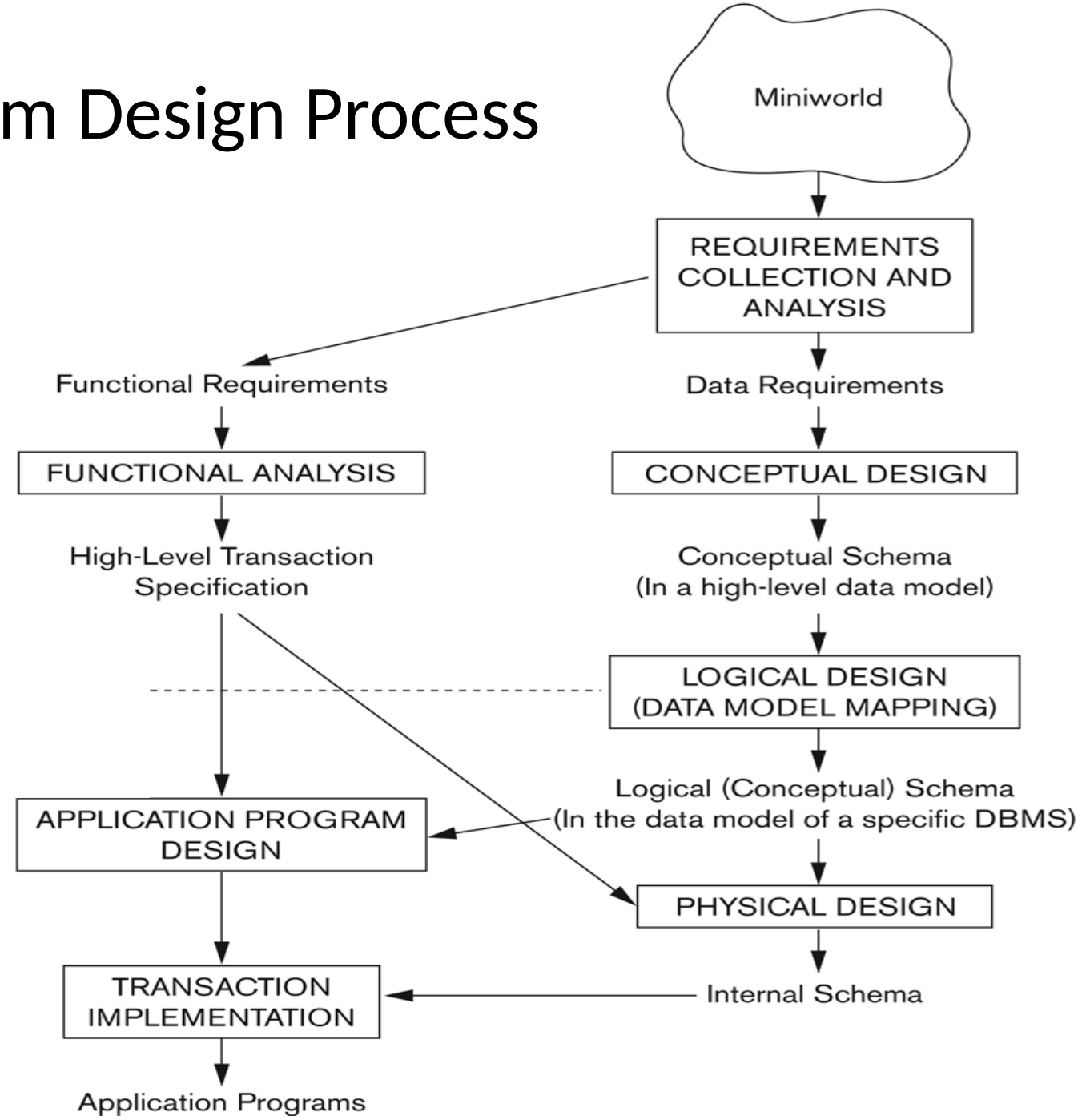


Figure from "Fundamentals of Database Systems" by Elmasri and Navathe, Addison Wesley.

Example of Data Requirements

A taxi company needs to model their activities.

There are two types of employees in the company: drivers and operators. For drivers it is interesting to know the date of issue and type of the driving license, and the date of issue of the taxi driver's certificate. For all employees it is interesting to know their personal number, address and the available phone numbers.

The company owns a number of cars. For each car there is a need to know its type, year of manufacturing, number of places in the car and date of the last service.

The company wants to have a record of car trips. A taxi may be picked on a street or ordered through an operator who assigns the order to a certain driver and a car. Departure and destination addresses together with times should also be recorded.

Another Example

- Movie database: information concerning movies, actors, awards
- **Data records**
  - Film
  - Person
  - Role
  - Honors
- Define structure of each type of data record by specifying **data elements** to include and **data type** for each element
  - String (sequence of alphabetic characters)
  - Numeric (integer or real)
  - Date (year or year-month-day)
  - Monetary amount
  - etc.

Using a Database

- **Populating** a DB: Inserting data to reflect the miniworld
  - e.g., store data to represent each film, actor, role, director, etc

Film

title	genre	year	director	runtime	budget	gross
The Company Men	drama	2010	John Wells	104	15,000,000	4,439,063
Lincoln	biography	2012	Steven Spielberg	150	65,000,000	181,408,467
War Horse	drama	2011	Steven Spielberg	146	66,000,000	79,883,359
Argo	drama	2012	Ben Affleck	120	44,500,000	135,178,251
Fire Sale	comedy	1977	Alan Arkin	88	1,500,000	0

Person

name	birth	city
Ben Affleck	1972	Berkeley
Alan Arkin	1934	New York
Tommy Lee Jones	1946	San Saba
John Wells	1957	Alexandria
Steven Spielberg	1946	Cincinnati
Daniel Day-Lewis	1957	Greenwich

Honors

movie	award	category	winner
Lincoln	Critic's Choice	actor	Daniel Day-Lewis
Argo	Critic's Choice	director	Ben Affleck
Lincoln	Screen Actors Guild	supporting actor	Tommy Lee Jones
Lincoln	Screen Actors Guild	actor	Daniel Day-Lewis
Lincoln	Critic's Choice	screenplay	Tony Kushner
Argo	Screen Actors Guild	cast	Argo
War Horse	BMI Flim	music	John Williams

Role

actor	movie	persona
Ben Affleck	Argo	Tony Mendez
Alan Arkin	Argo	Lester Siegel
Ben Affleck	The Company Men	Bobby Walker
Tommy Lee Jones	The Company Men	Gene McClary
Tommy Lee Jones	Lincoln	Thaddeus Stevens
Alan Arkin	Fire Sale	Ezra Fikus
Daniel Day-Lewis	Lincoln	Abraham Lincoln

Using a Database (cont'd)

- **Populating** a DB: Inserting data to reflect the miniworld
- **Query**: Interaction causing some data to be retrieved
  - Uses a *Query Language*

- Examples of queries:
  - List the cast of characters for *Lincoln*.
  - Who directed a *drama* in *2012*?
  - Who directed a film in which he or she also played a role?
  - What awards were won by *War Horse*?

Using a Database (cont'd)

- **Populating** a DB: Inserting data to reflect the miniworld
- **Query**: Interaction causing some data to be retrieved
  - Uses a *Query Language*
- **Manipulating** a DB
  - Querying and updating the DB to understand/reflect miniworld
  - Generating reports
  - Uses a *Data Manipulation Language (DML)*

- Examples of updates:
  - Record that *Argo* won a Golden Globe award for best picture.
  - Add another $395,533 to the gross earnings for *Lincoln*.
  - Change the birthplace for *Daniel Day-Lewis* to *London*.
  - Delete all data about the movie *Fire Sale* from the database.

Using a Database (cont'd)

- **Populating** a DB: Inserting data to reflect the miniworld
- **Query**: Interaction causing some data to be retrieved
  - Uses a *Query Language*
- **Manipulating** a DB
  - Querying and updating the DB to understand/reflect miniworld
  - Generating reports
  - Uses a *Data Manipulation Language (DML)*
- **Application program**
  - Accesses DB by sending queries and updates to DBMS

Reorganizing a Database

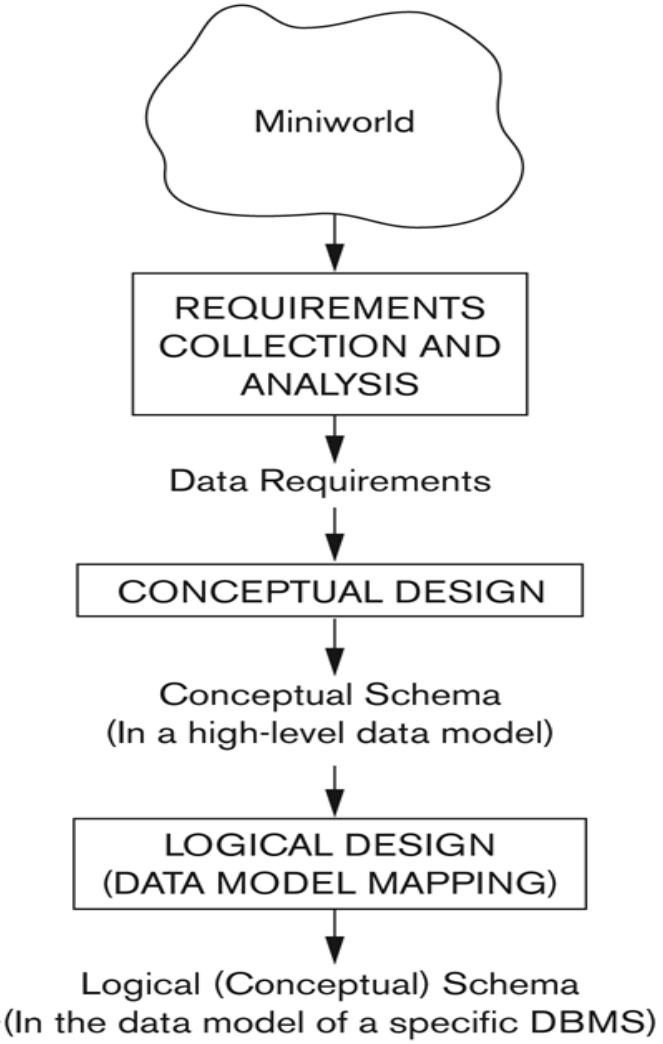
- Changes the metadata rather than the data
- More drastic than data updates
  - May require massive changes to the data
  - May require changes to some application programs
- Uses the *Data Definition Language (DDL)* again
- Examples:
  - Move *director* from FILM to a separate relation DIRECTOR with columns for *person* and *movie*
  - Change *birth* from *yyyy* to *yyyy/mm/dd*
  - Split name in PERSON to separate *surname* from *given names*.
  - Include data element *movieID* in FILM (to accommodate remakes and other duplications of film title); update other relations accordingly

Logistics of the Course

Intended Learning Outcomes

After the course you should be able to …

- *Design relational databases* for different types of example domains by first creating a conceptual schema using the Enhanced Entity-Relationship (EER) model and then translating this conceptual schema into a corresponding logical schema captured in the relational data model.
- Analyze and improve the quality of given relational database schemas based on the formal measure of *normal forms*.



After the course you should be able to …

- *Employ the SQL language* to query and to modify several example relational databases, as well as to create such a database with a given relational database schema.
- Compare the cost of finding and updating records in database storage files when using different approaches to organize and to index such files.
- *Apply basic techniques* that DBMSs may use to identify and to avoid problems that may occur when multiple users access a database concurrently.
- *Apply recovery algorithms* that DBMSs use to guarantee persistence of data even in the case of system failures.

Course Topics

1. Introduction
2. Relational databases
3. SQL
4. EER modeling
5. Mapping of EER diagrams to relations
6. Functional dependencies and normalization
7. Stored procedures and triggers
8. Data structures for DBs
9. Introduction to Transaction Processing
10. Concurrency Control
11. Database Recovery
12. Query Processing

- Attention: topics do not map 1-to-1 to lectures

Logistics of the Course

Examination

Written Exam

- After the course
- Dates: see pointer at the course Website
- Old example exam and a collection of old exam questions (with solutions) on the course Website
- Covers all topics of the course except for topic 7 (triggers & stored procedures)

Four Assignments

1. SQL
2. Database design and EER modeling
3. Functional dependencies and normalization
4. BrianAir project, *lab4a*: initial design, *lab4b*: improved design
   lab4c: implementation, *lab4d*: urkund analysis

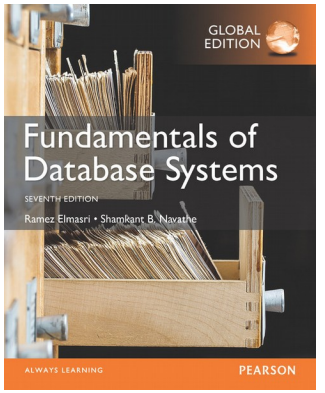
- Deadlines on the course Website
  - hard deadlines for lab4a and lab4b (before lab2 and lab3!)
- To be solved in pairs
  - register with a lab partner in Webreg before the end of this week
- Use MySQL server for lab1 and lab4c
  - need access to MySQL server provided by LiU IT
  - instructions on the course Website

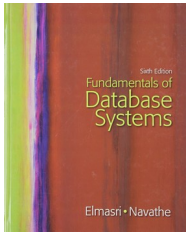
Logistics of the Course

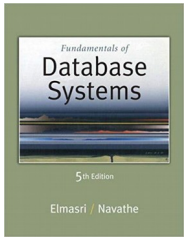
Organization

Structure of the Course

- Schedule see the course Website
- 12 lectures
  - Text book: Elmasri and Navathe. *Fundamentals of Database Systems*, Addison Wesley, 7th edition.
- 10 lab sessions
  - First two: focus on assignment lab1
  - Next two: focus on assignment lab2
  - Remaining six: focus on assignment lab4c
    (only three of these six lab sessions will be supervised)
- 2 teaching sessions
  - Discussion of lab4a hand-ins (mandatory!)
  - Practice with functional dependencies and normalization (related to assignment lab3)







www.liu.se

LINKÖPING UNIVERSITY