Databases 1: Introduction to Database Management Systems

Ioan Dragan

Sept. 2020

FMI

Course expectations

- Labs and courses will both introduce new concepts $\rightarrow$ it is recommended not to miss them

Course expectations

- Labs and courses will both introduce new concepts $\rightarrow$ it is recommended not to miss them
- Final grade computation:

```
int getGrade(int LectureGrade, int LabGrade){
if(LectureGrade ≥ 5 & LabGrade ≥ 5)
return (LectureGrade*0.5+LabGrade*0.5);
else return 4;}
```

Course expectations

- Labs and courses will both introduce new concepts $\rightarrow$ it is recommended not to miss them
- Final grade computation:
- Feedback is always welcome; there are no stupid questions!

Course expectations

- Labs and courses will both introduce new concepts $\rightarrow$ it is recommended not to miss them
- Final grade computation:
- Feedback is always welcome; there are no stupid questions!
- **DO INTERRUPT!**

FMI

Course expectations

- Labs and courses will both introduce new concepts $\rightarrow$ it is recommended not to miss them
- Final grade computation:
- Feedback is always welcome; there are no stupid questions!
- **DO INTERRUPT!**
- Attendance to course and labs: according to faculty's policy

FMI

Course expectations

- Labs and courses will both introduce new concepts $\rightarrow$ it is recommended not to miss them
- Final grade computation:
- Feedback is always welcome; there are no stupid questions!
- **DO INTERRUPT!**
- Attendance to course and labs: according to faculty's policy
- Contact:
  - Course and Lab: **ioan.dragan@e-uvt.ro**
  - Lab: **raul.horhat@e-uvt.ro** OR **felix.iacob@e-uvt.ro**

  Google Classroom: weekly update $\rightarrow$ **keep an eye on it**;

FMI

Course outline

1. Introduction to database approach.
2. The database environment.
3. Introduction to The Relational Model.
4. Views.
5. Transactions.
6. SQL Constraints.
7. Relational Database Design. Theory and practice.
8. An intro to Database Performance. Indexing.
9. JSON Support in Relational Database Management Systems
10. NoSQL Databases.

FMI

Agenda

1. Why do we need databases?

Agenda

1. Why do we need databases?
2. Short history of DBMSs

Agenda

1. Why do we need databases?
2. Short history of DBMSs
3. The database approach. Key concepts.

Agenda

1. Why do we need databases?
2. Short history of DBMSs
3. The database approach. Key concepts.
   - The database

FMI

Agenda

1. Why do we need databases?
2. Short history of DBMSs
3. The database approach. Key concepts.
   - The database
   - The Database Management System (DBMS)

FMI

Agenda

1. Why do we need databases?
2. Short history of DBMSs
3. The database approach. Key concepts.
    - The database
    - The Database Management System (DBMS)
    - The application programs

Agenda

1. Why do we need databases?
2. Short history of DBMSs
3. The database approach. Key concepts.
   - The database
   - The Database Management System (DBMS)
   - The application programs
4. Database aapproach environment

FMI

Agenda

1. Why do we need databases?
2. Short history of DBMSs
3. The database approach. Key concepts.
   - The database
   - The Database Management System (DBMS)
   - The application programs
4. Database aapproach environment
5. Data models

FMI

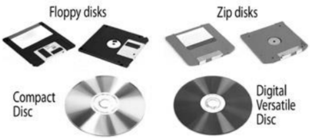
Agenda

1. Why do we need databases?
2. Short history of DBMSs
3. The database approach. Key concepts.
   - The database
   - The Database Management System (DBMS)
   - The application programs
4. Database aapproach environment
5. Data models
6. Advantages and shortcomings of DBMSs.

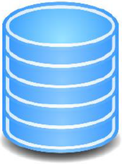
FMI

Why do we need databases?

Hard-copies on paper

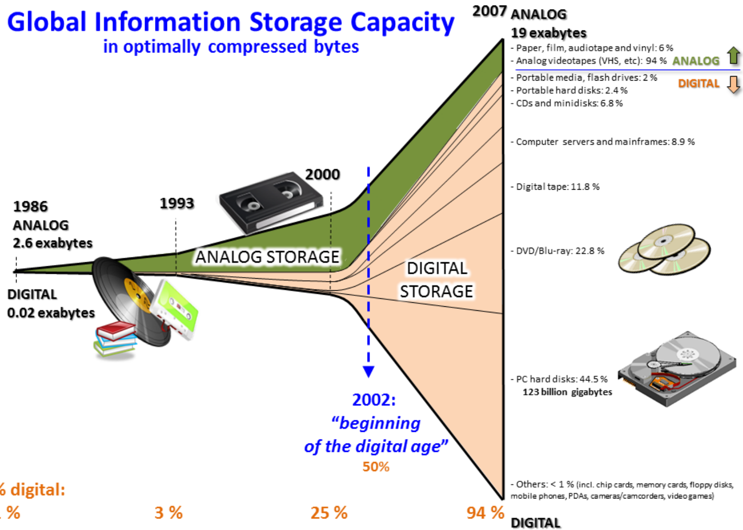


Electronically on files



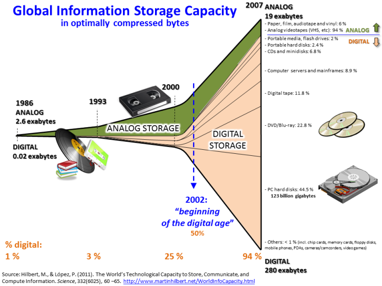
Database

Massive data



Source: Hilbert, M., & López, P. (2011). The World's Technological Capacity to Store, Communicate, and Compute Information. *Science*, 332(6025), 60 –65. http://www.martinhilbert.net/WorldInfoCapacity.html

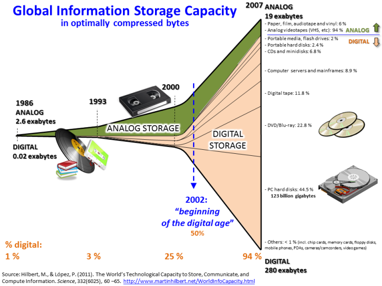
Massive data



- 90% of the data in the world today was created in the last two years alone

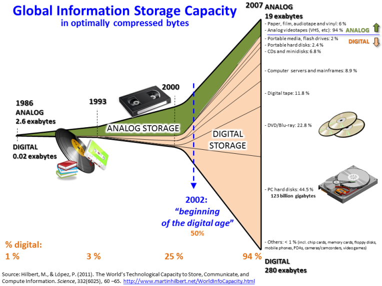
FMI

Massive data



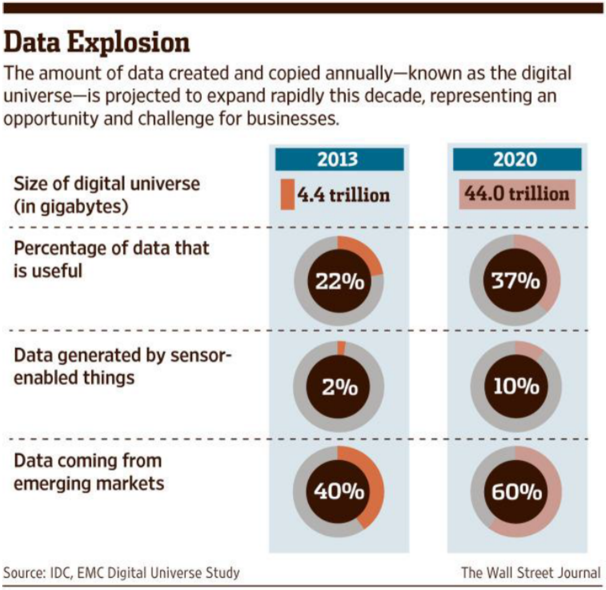
- 90% of the data in the world today was created in the last two years alone
- more than 12TB of tweets/day

Massive data



- 90% of the data in the world today was created in the last two years alone
- more than 12TB of tweets/day
- working with less than 0.001% if the sensor stream data, the data flow from all four LHC experiments represent 25PB annual rate before replication.

... continuously growing



FMI

File Based systems

- It is defined as a collection of application programs that perform services for end users.

File Based systems

- It is defined as a collection of application programs that perform services for end users. E.g: production of reports
- Each program defines and manages its own data

File Based systems

- It is defined as a collection of application programs that perform services for end users.
- Each program defines and manages its own data
- File $\leftrightarrow$ collection of records

FMI

File Based systems

- It is defined as a collection of application programs that perform services for end users.
- Each program defines and manages its own data
- File $\leftrightarrow$ collection of records
    - each record contains related data

File Based systems

- It is defined as a collection of application programs that perform services for end users.
- Each program defines and manages its own data
- File $\leftrightarrow$ collection of records
    - each record contains related data
    - a record has one or more fields.

File Based systems

- It is defined as a collection of application programs that perform services for end users.
- Each program defines and manages its own data
- File $\leftrightarrow$ collection of records
  - each record contains related data
  - a record has one or more fields.
- Limitations:

FMI

File Based systems

- It is defined as a collection of application programs that perform services for end users.
- Each program defines and manages its own data
- File $\leftrightarrow$ collection of records
  - each record contains related data
  - a record has one or more fields.
- Limitations:
  - separation and isolation of data

FMI

File Based systems

- It is defined as a collection of application programs that perform services for end users.
- Each program defines and manages its own data
- File $\leftrightarrow$ collection of records
  - each record contains related data
  - a record has one or more fields.
- Limitations:
  - separation and isolation of data
  - duplication of data

File Based systems

- It is defined as a collection of application programs that perform services for end users.
- Each program defines and manages its own data
- File $\leftrightarrow$ collection of records
  - each record contains related data
  - a record has one or more fields.
- Limitations:
  - separation and isolation of data
  - duplication of data
  - data dependence

FMI

File Based systems

- It is defined as a collection of application programs that perform services for end users.
- Each program defines and manages its own data
- File $\leftrightarrow$ collection of records
  - each record contains related data
  - a record has one or more fields.
- Limitations:
  - separation and isolation of data
  - duplication of data
  - data dependence
  - incompatibility of files

FMI

File Based systems

- It is defined as a collection of application programs that perform services for end users.
- Each program defines and manages its own data
- File $\leftrightarrow$ collection of records
  - each record contains related data
  - a record has one or more fields.
- Limitations:
  - separation and isolation of data
  - duplication of data
  - data dependence
  - incompatibility of files
  - fixed queries

FMI

File Based systems

- It is defined as a collection of application programs that perform services for end users.
- Each program defines and manages its own data
- File $\leftrightarrow$ collection of records
  - each record contains related data
  - a record has one or more fields.
- Limitations:
  - separation and isolation of data
  - duplication of data
  - data dependence
  - incompatibility of files
  - fixed queries
  - inability to generate timely reports!

FMI

The origins of DBMSs

- 1960: Apollo Moon landing project

The origins of DBMSs

- 1960: Apollo Moon landing project $\rightarrow$ the GUAM (Generalized Update Access Method) - hierarchical approach

The origins of DBMSs

- 1960: Apollo Moon landing project $\rightarrow$ the GUAM (Generalized Update Access Method) - hierarchical approach $\rightarrow$ origin of the DBMS

FMI

The origins of DBMSs

- 1960: Apollo Moon landing project
- IBM joins and IMS (Information Management System) is born.
- Mid 60's GE creates IDS (Integrade Data Store) - network DBMS

The origins of DBMSs

- 1960: Apollo Moon landing project

- Mid 60's GE creates IDS (Integrade Data Store) - network DBMS
- 1960-1970 CODASYL (Conference on Data System Langauages), DNTG (Data Base Task Group) define DDL and DML

FMI

The origins of DBMSs

- 1960: Apollo Moon landing project

- Mid 60's GE creates IDS (Integrade Data Store) - network DBMS
- 1960-1970 CODASYL (Conference on Data System Langauages), DNTG (Data Base Task Group) define DDL and DML
- 1970 Codd introduces the relational model

The origins of DBMSs

- 1960: Apollo Moon landing project

- Mid 60's GE creates IDS (Integrade Data Store) - network DBMS
- 1960-1970 CODASYL (Conference on Data System Langauages), DNTG (Data Base Task Group) define DDL and DML
- 1970 Codd introduces the relational model
- 1976 Chen introduces the ER (entity-relationship) model

FMI

The origins of DBMSs

- 1960: Apollo Moon landing project

- Mid 60's GE creates IDS (Integrade Data Store) - network DBMS
- 1960-1970 CODASYL (Conference on Data System Langauages), DNTG (Data Base Task Group) define DDL and DML
- 1970 Codd introduces the relational model
- 1976 Chen introduces the ER (entity-relationship) model
- Late 1970's SQL is developed

FMI

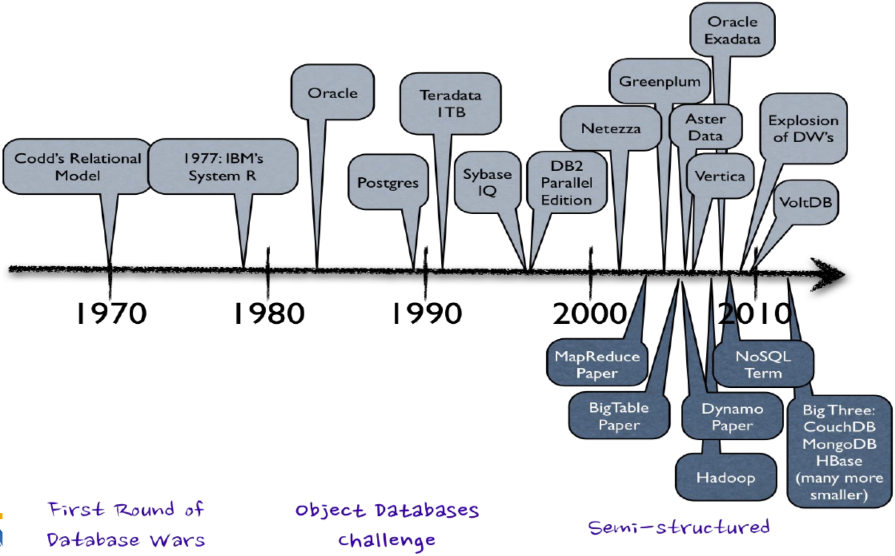
The origins of DBMSs

- 1960: Apollo Moon landing project

- Mid 60's GE creates IDS (Integrade Data Store) - network DBMS
- 1960-1970 CODASYL (Conference on Data System Langauages), DNTG (Data Base Task Group) define DDL and DML
- 1970 Codd introduces the relational model
- 1976 Chen introduces the ER (entity-relationship) model
- Late 1970's SQL is developed
- Today ... we have RDBMSs (DB2 from IBM, Oracle from Oracle, SQL Server from Microsoft and many more).

FMI

Evolution of DBMS

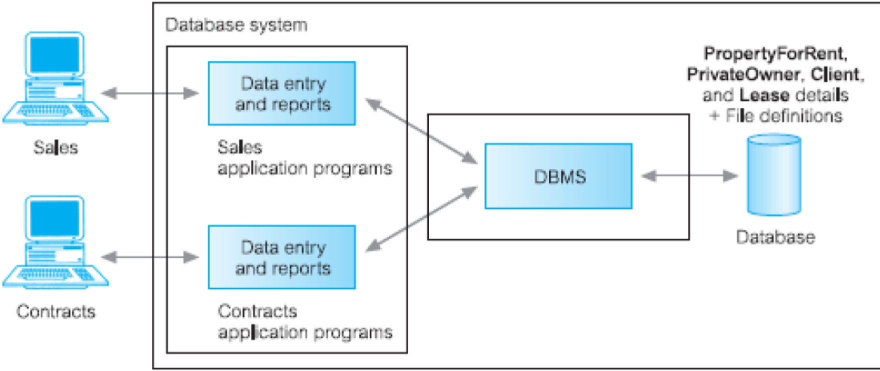


Source: http://www.benstopford.com/2012/07/28/thoughts-on-big-data-technologies-part-1/

The Database Approach

- The Database
- The Data Base Management System
- (Database) Application Program

The Database Approach



PropertyForRent (propertyNo, street, city, postcode, type, rooms, rent, ownerNo)
PrivateOwner (ownerNo, fName, lName, address, telNo)
Client (clientNo, fName, lName, address, telNo, prefType, maxRent)
Lease (leaseNo, propertyNo, clientNo, paymentMethod, deposit, paid, rentStart, rentFinish)

The Database

- Database = a shared collection of logically related data (and description of this data), designed to meet the information needs of an organization.
- Database = self-describing collection of integrated records
  - Schema : description of data = system catalog, metadata
  - Data Model: set of records, XML documents, graphs, collection of objects etc.

FMI

Database Management System

DBMS is a software system that enables users to define, create, and maintain the database by providing controlled access to this database. It provides efficient, reliable,

convenient, and safe multi-user storage of and access to massive amounts of persistent data.

FMI

Database Management System

- Persistent : outlive the programs that create/access he data
- Safe: hardware/software failures, malicious users
- Multi-user: concurrently access to data (concurrency control)
- Convenient:
  - Physical Data Independence: huge difference between physical representation of data on disk and the logical way of seeing and working with;
  - High level, declarative (what, not how) query languages (e.g. SQL)
- Efficient: thousands of operations (query/update) per second
- Reliable: 99.999% uptime

FMI

Database Management System

- Allow users to define the database - Data Definition Language (DDL)
- Allows users to insert, update, delete and retrieve data from the database - Data Manipulation Language (DML), e.g: SQL
- Controlled access to the database:
  - Security system
  - Integrity system
  - Concurrency control system
  - Recovery control system
  - User-accessible catalog

FMI

Database Languages

- Data Definition Language: used to specify the database schema
- Data Manipulation Language: used to update the database (insert, update, delete)
- Data Query Language: used to extract (read) data needed at a moment.

FMI

Data Definition Language

- A descriptive language that allows the DBA or user to describe and name entities required for the application, together with any associated integrity and security constraints
- The result of compiling a DDL statement $\rightarrow$ the system catalog (data dictionary) which integrates metadata
- System catalog (=data dictionary = data directory) may, or may not, be accessible by database users.
- DDL used to specify external & conceptual schema

FMI

Data Manipulation Language (DML)

- A language that provides a set of operations to support the basic data manipulation operations on data held in the database
- Operations include insertion, modification or deletion
- Procedural DML: a language that allows users to tell the system what data is needed and exactly HOW to retrieve the data (network and hierarchical DMLs are typically procedural)
- Declarative DML: a language that allows the user to state WHAT data is needed rather than how it is to be retrieved.

FMI

Data Query Language (DQL)

- A language that provides a set of operations to support the basic data extraction on the data hed in the database
- Operations include querying the database
- Procedural DQL vs Declarative DQL

FMI

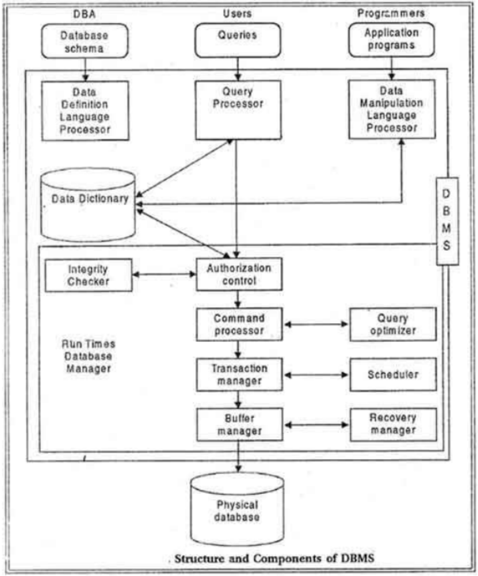
4GL

- Non procedural
- Presentation languages (query languages, report generators)
- Speciality languages (spreadsheets)
- Application/ forms/ graphics generators
- Examples: SQL, QBR (Query by Example)

FMI

Components of a DBMS

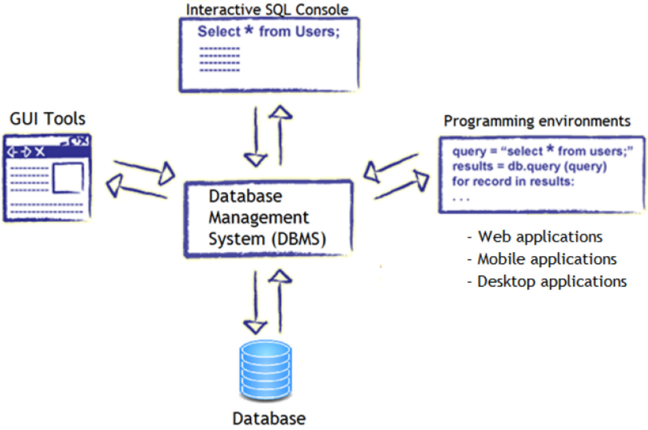
- Query processor
- DML processor
- DDL compiler
- Data dictionary
- Run-time database manager



Source: `https://ecomputernotes.com/fundamental/what-is-a-database/components-of-dbms`

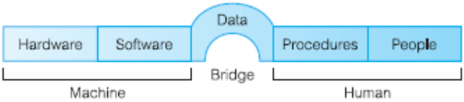
(Database) Application Programs

Application Program: a computer program that interacts with the database by issuing an appropriate request (typically an SQL statement) to the DBMS.



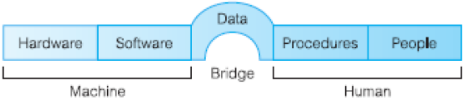
FMI

Database Approach Environment



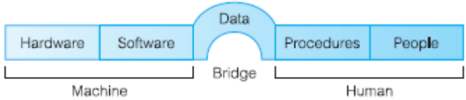
- Hardware

Database Approach Environment



- Hardware
  - Single server
  - Distributed architecture

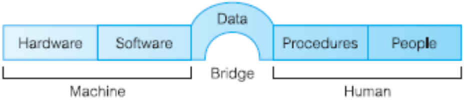
Database Approach Environment



- Hardware
- Software (OS, DBMS, application programs)
  - Apps are usually in 3GL (eg. C++, Java, Visual Basic, PHP)
  - The DBMS may have 4GL for query languages, report, form, graphics, or application generators (e.g MS Access, SQL).

FMI

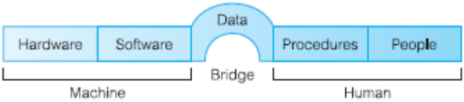
Database Approach Environment



- Hardware
- Software
- Data (schema : structure of the database, tables, attributes)
  - Data Models
  - Names, Types, and sizes of data items
  - Names of relationships
  - Integrity constraints
  - Names of authorized users
  - Indexes, storage structures

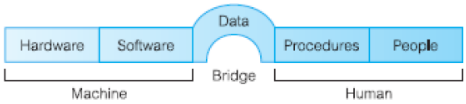
FMI

Database Approach Environment



- Hardware
- Software
- Data
- Procedures: instructions and rules that govern the design and use of the DB
  - logs to the DBMS
  - use of a particular DBMS facility or application
  - start and stop the DBMS
  - make backup copies of the database
  - handle hardware or software failures
  - coding standards, guidelines
  - monitoring and notifications
  - migration

Database Approach Environment



- Hardware
- Software
- Data
- Procedures
- People (Roles in the Database Environment)
  - Data administrators (DA): planning, development and maintenance of standards, policies and procedures + conceptual/logical design
  - Database Administrators (DBA): physical design and implementation, security, integrity, maintenance of the OS, ensuring satisfactory performance for apps and users
  - Database designer: logical/conceptual database designer - business rules, physical database designer
  - Application programmers
  - End users: naive users, sophisticated users.

FMI

Data models

- Data Model: integrated collection of concepts for describing data, relationship between data, and constraints on the data in an organization.
- The purpose of a data model is to represent data and to make the data understandable
- Types of data models:
  - External data models
  - Conceptual data models
  - Internal data models

FMI

Data models

- Maine roles: communicate the semantic of data & discover the semantic of data
- Characteristics of logical (external/conceptual) data models:
  - Graphical diagram
  - Explicit representation of semantic
  - Appropriate level of detail
  - DBMS independent
  - Tool support

FMI

Internal data models

- Relational model
- Object-oriented model
- Graph model
- Key-value model
- Columnar model
- Document model

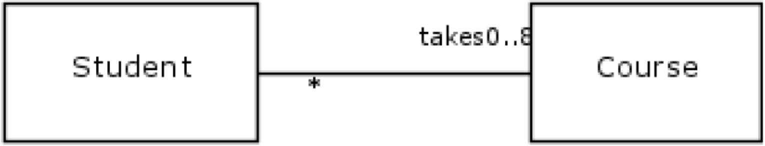
Running Example

Students should enroll in courses they want to attend. One student may enroll in up to 8 courses. In order for one course to run it requires at least 10 enrolled students. As places in courses are limited, for each enrollment request there will be a decision associated whether the student is accepted or not in the course.
Courses are offered by different departments of the university, each course is uniquely identified by their title and each course is credited a fixed number of credits. Students may enroll to courses offered by different departments.

FMI

Example

Conceptual model



Advantages of DBMSs

- Flexibility: more information from the same ammount of data
- Control of data redundancy
- Data consistency
- Sharing of data
- Improved data integrity
- Improved security
- Enforcement of standards
- Economy of scale
- Balance of conflicting requirements
- Improved data accessibility and responsiveness
- Increased productivity
- Improved maintenance
- Increased concurrency
- Improved backup and recovery services

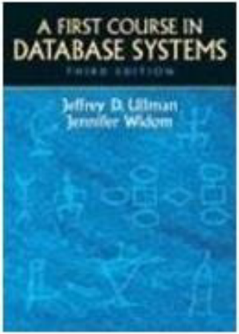
Shortcomings of DBMSs

- Complexity and cost (HW+DBMS costs+ cost of conversion to DB approach)
- Size
- Performance
- Higher impact of a failure

Common uses of DBMSs

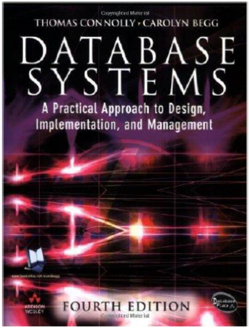
- Web applications
- Super/Hypermarket
- Credit card management
- Library
- Insurance
- Manufacturing
- Financial/accounting
- Social media

Bibliography (recommended)



A First Course in Database Systems (3rd edition) by Jeffrey Ullman and Jennifer Widom, Prentice Hall, 2007

<u>**Chapter 1**</u>



Database Systems - A Practical Approach to Design, Implementation, and Management (4th edition) by Thomas Connolly and Carolyn Begg, Addison-Wesley, 2004

<u>**Chapter 1**</u>

FMI