Database Systems

Yann Thierry-Mieg

Outline

1. Introduction
2. SQL : a Simple Query Language
3. Data integrity
4. PL/SQL : Programming Language / SQL

5. Database Design
6. Organization and Administration of a RDBMS (Oracle)

Bibliography

- Books
  - Bases de données Objet et relationnel de Georges Gardarin (Edition Eyrolles)
- Web:
  - http://www-inf.int-evry evry.fr/COURS/BD/accueil-ext.html .(modèle relationnel, SQL, Conception Entité-Association)
  - http://esesa1.supelec.fr/www/yb/poly_bd/sql/tdm_sql.html (language SQL)
  - http://cui.unige.ch/~nerima/index_bd.html (modèle relationnel, SQL, Conception Entité-Association)

Bibliography

- Eric Cestari's course at ECE 05-06 : http://www.ece.fr/~cestari/
- Jortiz course : cs.utsa.edu/~jortiz

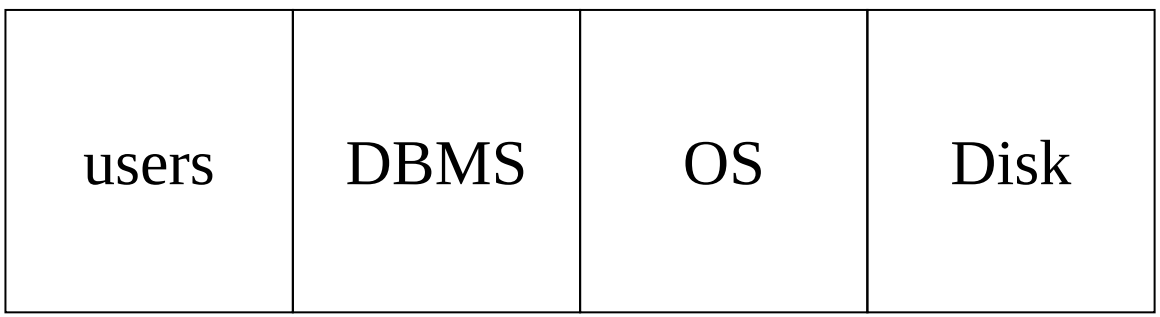
1. Introduction

Definition

- Database : structured data
  - recorded with minimal redundancy
  - to simultaneously satisfy multiple users
  - in a selective manner
  - with fast response time
- Data storage
  - using a permanent media
- Data adminisitration
  - Data Base Management System (DBMS)

Data Base Management System

- A DBMS is a software set that allows to search, update and save data on secondary storage
- Serves as additional layer between OS and user



DBMS

- A DBMS ensures
  - data **description**
  - **search** and **update** of data
  - **safety** : check user access rights, limit unauthorized access, crypt of sensitive information
  - **security :** saving and restoring data, limit impact of manipulation errors
  - **integrity :** define rules to maintain data integrity (depends on the nature of data stored)
  - **concurrency :** detect and offer resolution mechanisms for simultaneous/concurrent data access (transactions...)

Why Use a DBMS?

Suppose we need to build a university information system. How do we

- store the data? (use file structures…)
- query the data? (write programs…)
- Update data safely? (more programs…)
- provide different views on the same data? (registrar versus students) (more prog…)
- deal with crashes? (more prog…)

Way too complicated! Go buy a DBMS!

DBMS

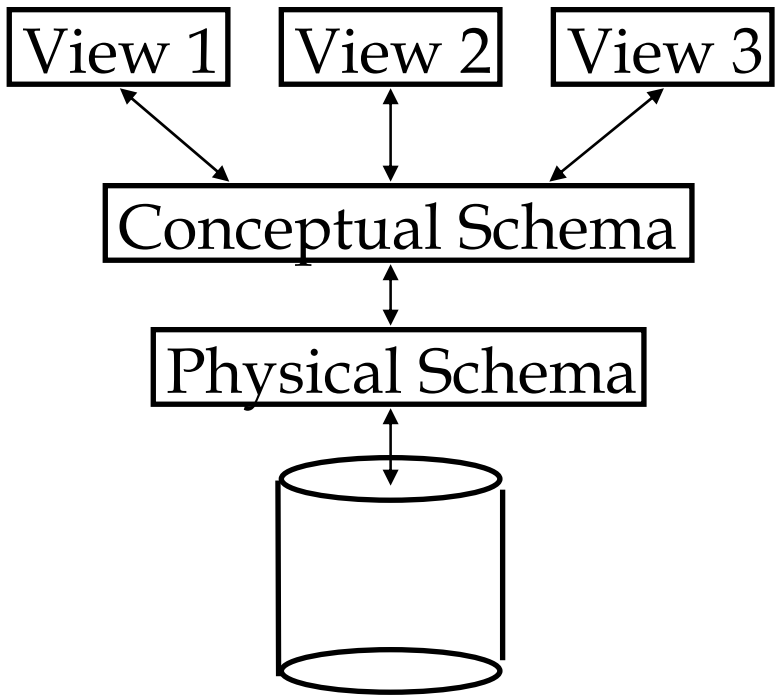
- Application Domain :
  - Knowledge base
  - Expert systems
  - Geographical Information System
  - Accounting and management
  - Digital content management
  - Genome projects
  - ...
- Any domain in which (large amounts) of structured data need to be manipulated
- The actual DBMS system used depends on the application and its needs

Data Description

- Three levels of description are distinguished:
  - External level (most abstract)
    - views on a database for instance
  - Conceptual schema
    - data structures of DBMS, basic types, access rights..
  - Physical schema
    - data storage media, file system, indexes...

Abstract levels of DB Schema

- Views describe how users see the data.
- Conceptual schema defines logical structure using a data model
- Physical schema describes the files and indices used.



Physical Schema

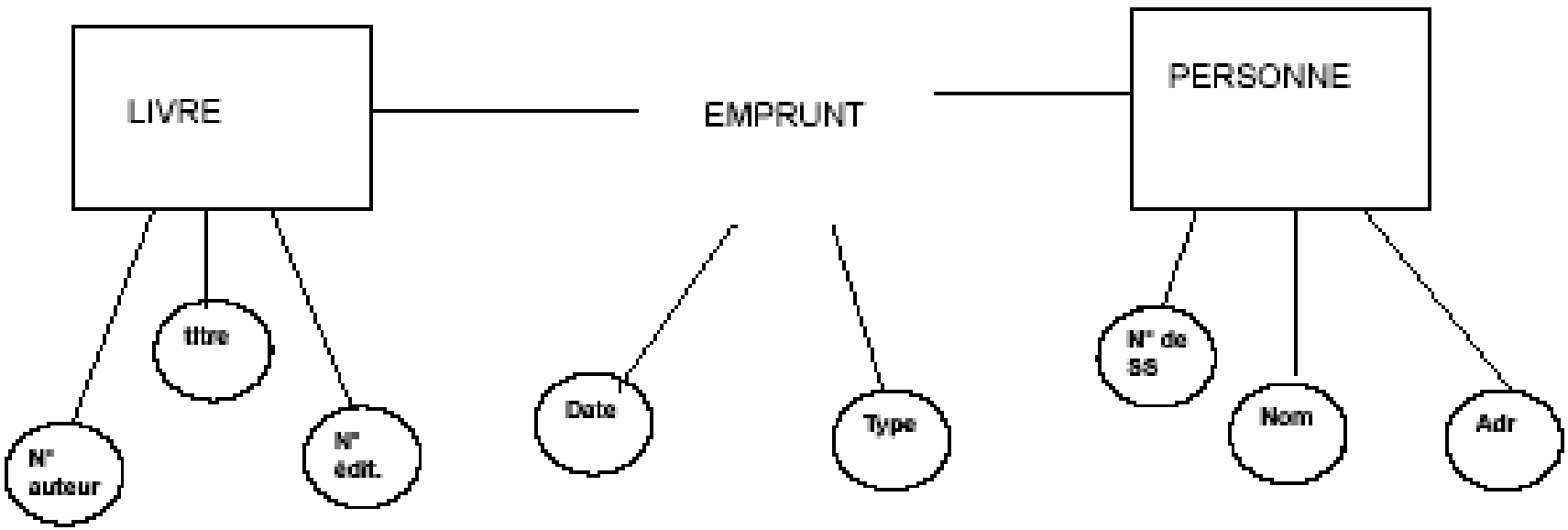
- Specifies how the data is physically stored (disk, tape storage...). The physical data model defines how the data is organized

- e.g. data files (name, size, data organization)

Conceptual schema

- Describes data **independently of physical storage** constraints
  - Elementary data types
    - e.g. book title, author name, ...
  - Composite data types
    - e.g. book, person
  - Data associations between composite types
    - e.g. a book is borowed by a person
  - Some data integrity rules
    - e.g. a person may not borrow more than 3 documents ...

An example conceptual description



External Level

- Most abstract description level
- Contains **views**, which may be specific to a single user or user group, and only presents a partial view of the conceptual level
- Many views of a single database schema may coexist

History of DBMS

- Three main evolutions, each has its own data model
  - Hierarchical and network systems
  - Relational DBMS (90 % of DBMS today)
    - Oracle, DB2, MySQL, ...
  - Object-Oriented DBMS

Hierarchical and Networked Systems

- Hierarchical system (1960)
  - First generation DBMS
  - extension of file system with inter-file links (pointers)
  - Data stored in a tree
  - Access to a data is determined by traversing the tree

Hierarchical and Networked Systems

- Networked system (1970)
  - Same prinicples as hierarchical DBM
  - Data stored in a more general **graph** instead of a tree
  - Access to a data is determined by traversing the graph and following pointers

Relational DBMS

- Second generation DBM
- Based on relations <=> set theory
- Vast majority of DBM today (Oracle, DB2,...)
  - '70 : relational model is defined
  - '80 : first commercial RDBMS
  - '90 : Over 50% of DBMs based on relational
- Supports very large data sets, very efficiently

Relational DBMS

- RDBMS are based on **relational theory**
- Data represented as **TABLES** composed of **ROWS** and **COLUMNS**

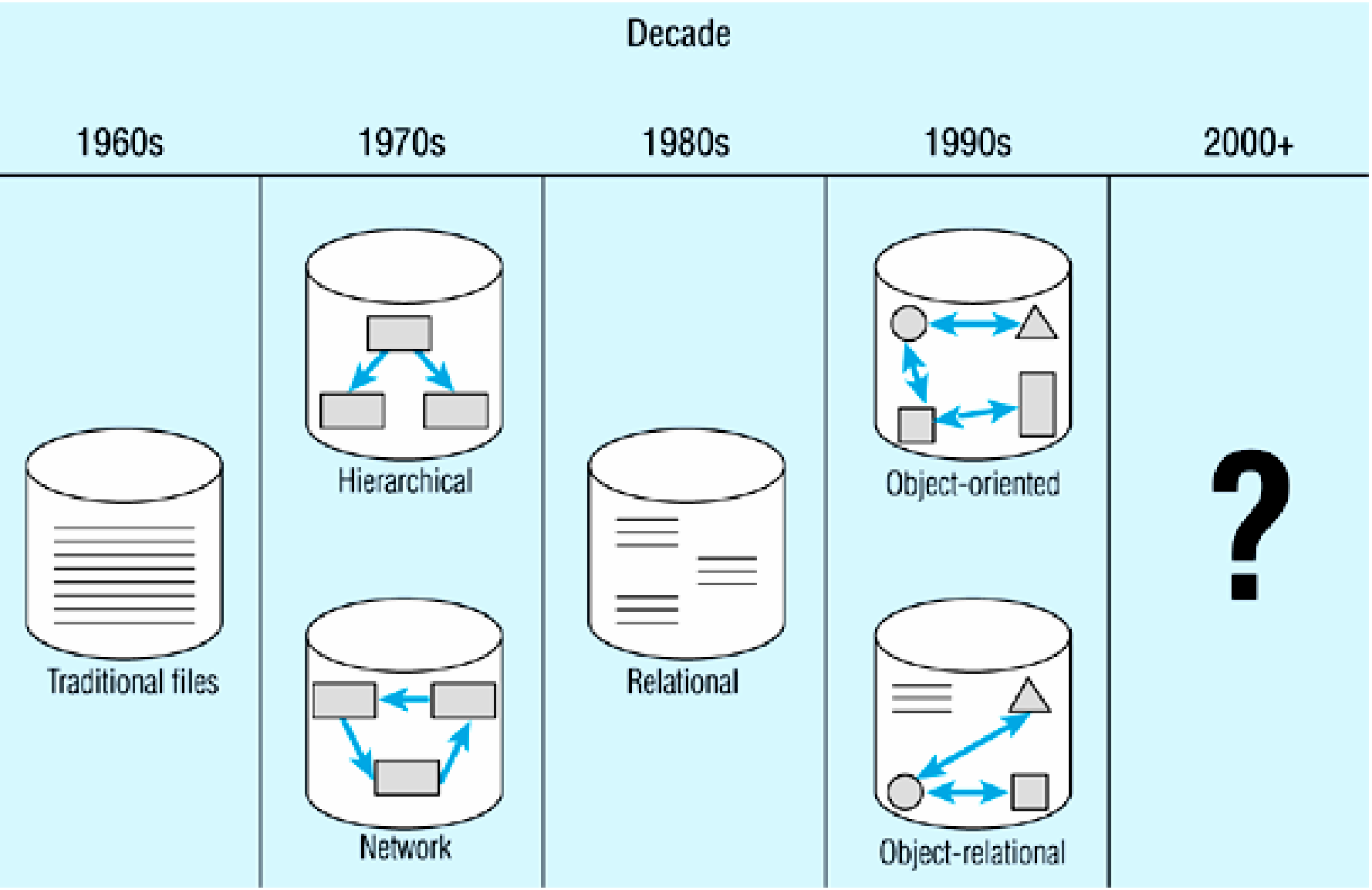
BOOK	
BOOK_ID	TITLE
8399	"Tom Sawyer"
1664	"Lord of the Rings"

PERSON	
PERS_ID	NAME
A1234	"John Smith"
E3456	"J. F. Kennedy"

Object-Oriented DBMS

- O-O DBMS are based on the object model ('90s)
- Pro :
  - flexibility of schema definition
  - efficiency of cross indexing
- Con :
  - Object loading less efficient
  - poor commercial support

Figure 1-12 Evolution of database technologies



RDBMS

- Basic Concepts
  - Relational model due to Codd 1970
- Definition :
  - A **domain** is a set of values
- **e.g:**
  - integer domain
  - string of length 25 domain

Relational basic concepts

- Definition
  - A **relation** is a subset of a Cartesian product of domains, characterized by it's **name**
- Example
  - let Country={China, USA, France} and Money={dollar, yuan, euro, florin} be two domains

CURRENCY	
COUNTRY_NAME	MONEY_NAME
China	Yuan
France	Euro
USA	dollar

Relational Schema

- Definition
  - an **attribute** of a relation is one of it's columns, characterized by a domain and a name
- e.g. COUNTRY_NAME:COUNTRY
- Definition:
  - A **relational schema** is noted by the relation name, followed by the definition of it's attributes in the form NAME:DOMAIN
- **R(A1:D1,A2:D2,....An:Dn)**

Key and Unicity

- Definition :
  - a subset of a relation's attributes (columns) is a **KEY** for a relation if it allows to uniquely identify a row of the relation
  - In other wortds, for any two rows of a relation, the columns forming the key have distinct values
- **Any relation has at least one KEY = PRIMARY KEY**
  - at worst, use all columns of the relation
  - usually, choose a **minimal** key, such that if you remove one element from the key, the unicity property is violated

Key Unicity

- Examples :
  - PERSON(**ss_num**, name, address, birth_date)
  - GRADE(**student_id, course_id**, grade)
    - note that this means a student may not have more than one grade for a given course !!