Politecnico
di Torino

1859

DBMG

Introduction to databases

Introduction to databases

- Information management
- Databases
- Data model
- Design of a database
- Data independence
- Data access
- Advantages and disadvantages of a DBMS
- Business intelligence

Information management

Introduction to databases

Information management

- Information is recorded and exchanged in different forms
- Forms of information organization and codification have been introduced over time

Computer systems

- In computer systems information is represented by means of data
  - the data are raw symbols which have to be interpreted and correlated to provide information
- Example
  - data: “Mario Rossi” and 424242
  - Information: result of looking up a telephone number in your personal telephone directory (e.g., list of contacts)

Data characteristics

Data are far more stable over time than the processes that manage them

Example:

- No variations in the structure of bank applications data for decades
- Procedures managing data may vary from year to year

Data are an important resource of the organization that manages them

Databases

Introduction to databases

Database

- General definition
  - a database is a collection of data that represents information interesting for a computer system
- “Technical” definition
  - A database is a collection of data managed by a DBMS

Types of databases - DBMS

Relational

MySQL

PostgreSQL

IBM DB2

Microsoft SQL Server

Object Oriented

Data warehouse

ORACLE

In-memory

SQLite

redis

NoSQL

mongoDB

cassandra

Document based

elasticsearch

Time series

influxdb

Real-time

Firebase

Graph

neo4j

… e altri…

Data Base Management System - DBMS

- A DBMS (Data Base Management System) is a software system able to manage collections of data that are
  - large
  - shared
  - Persistent

  ensuring their reliability and privacy

DBMS characteristics

- Far greater dimension than the central memory available
  - data management in secondary memory
- Data sharing between applications and user: a database is an *integrated* resource, shared by several company sectors
  - reduction of data redundancy
  - reduction of data inconsistency
  - competing access control mechanism

DBMS characteristics

Data persistence

Lifetime not limited to execution of programmes using them

Data reliability in the case of hardware and software malfunction/failure

Backup and recovery functionalities

Data privacy

Authorization mechanisms to enable users access

Efficiency

Capacity to carry out operations using a set of resources (time and space) acceptable for users

Efficacy

Capacity to make user activities productive

DBMS or file system?

- “Simplified” approach to data: data stored in a persistent mode in the mass/secondary memory inside a file
    - It is possible to memorize and look for data
    - Simple access mechanisms (sequential reading)
    - Simple sharing mechanisms (read only sharing, blocking write access rights)
- DBMS extends the functionalities of the file system, providing more integrated services

Data model

Introduction to databases

Data model

- A data model is a set of concepts utilized for organizing data of interest and describing its structure in a way that is understood by a computer
  - Elementary data types (integer, character, ...)
  - Structuring mechanism for defining more complex structures (record builder, array, …)

Types of data models

- Relational model
  - Most widespread data model
  - Data are organized into sets of homogeneous (fixed structure) records and represented as tables
- Before the relational model, other models closer to the physical structures of storing were used
  - Hierarchical model, network model
- Since the relational model
  - Object model, XML, database NOSQL, ...

Relational model

Teacher

Name	Department	Phone
Green	Computer Engineering	123456
White	Telecommunications	636363
Black	Computer Engineering	414243

Courses

Code	Name	Teacher
M2170	Information Systems	Green
M4880	Computer Networks	White
F0410	Databases	Black

Schema and instances

- In a database we define:
  - the *schema*, which describes the structure of the data
    - It is practically unvarying over time
    - It is represented by the header of each table (table name and column names)

Teacher

Name	Department	Phone
Green	Computer Engineering	123456
White	Telecommunications	636363
Black	Computer Engineering	414243

SCHEMA

Courses

Code	Name	Teacher
M2170	Information Systems	Green
M4880	Computer Networks	White
F0410	Databases	Black

SCHEMA

Schema and instances

- In a database we define:
  - the *istance*, composed of the content of each table, i.e. of the actual data values which are
    - Variable over time, also very rapidly
    - Represented by the rows in the tables

Teacher

Name	Department	Phone
Green	Computer Engineering	123456
White	Telecommunications	636363
Black	Computer Engineering	414243

ISTANCE

Courses

Code	Name	Teacher
M2170	Information systems	Green
M4880	Computer Networks	White
F0410	Databases	Black

ISTANCE

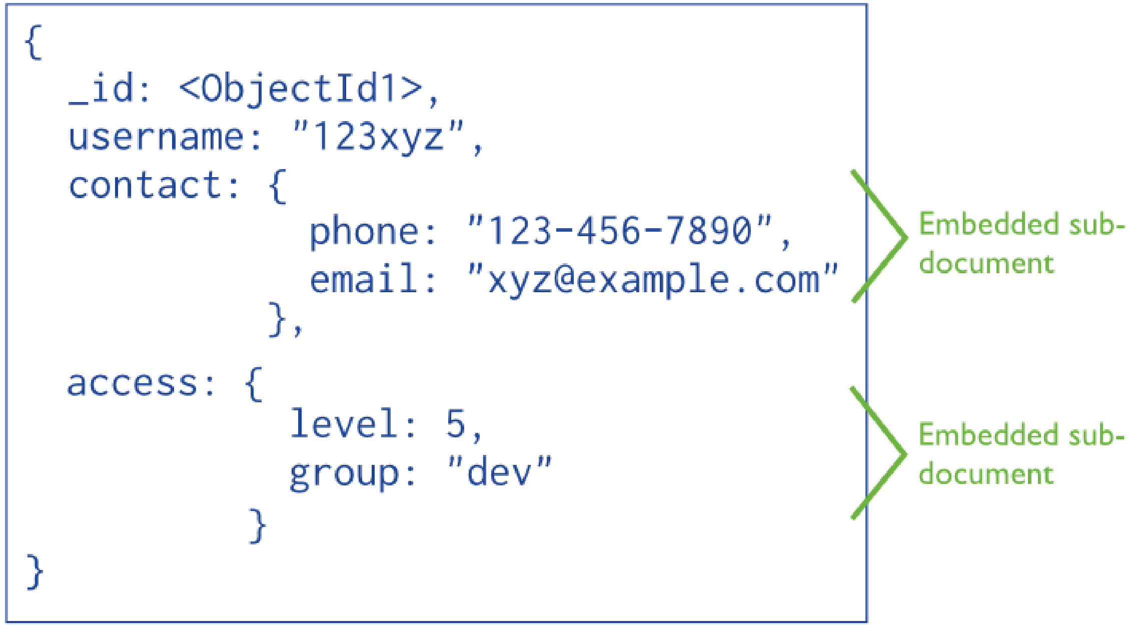
Example of other data models: Database NOSQL

- A database is a set of collections
- Each collection contains a set of documents
- Each document is described as a list of key-value fields and each field can contain any type of data
- Documents in the same collection can be heterogeneous
- Since the data representation is schema-less, it is not necessary to define in advance the document schema and documents in the same collection can be characterized by different fields

Relational database	NOSQL database
Tables	Collection
Row	Document
Column	Field

Example of document data

- Relationships between documents are inefficient and lead to denormalization
  - Document reference (ID), without native join



```
{
  _id: <ObjectId1>,
  username: "123xyz",
  contact: {
              phone: "123-456-7890",
              email: "xyz@example.com"
            },
  access: {
             level: 5,
             group: "dev"
           }
}
```

Source: https://www.mongodb.com/docs/manual/core/data-model-design/

Types of data models

Conceptual data model

- It is possible to represent data indipendently from the logical data model
  - describes real world concepts
  - used in the database design phase
- Example: Entity-Relationship model

Logical data model

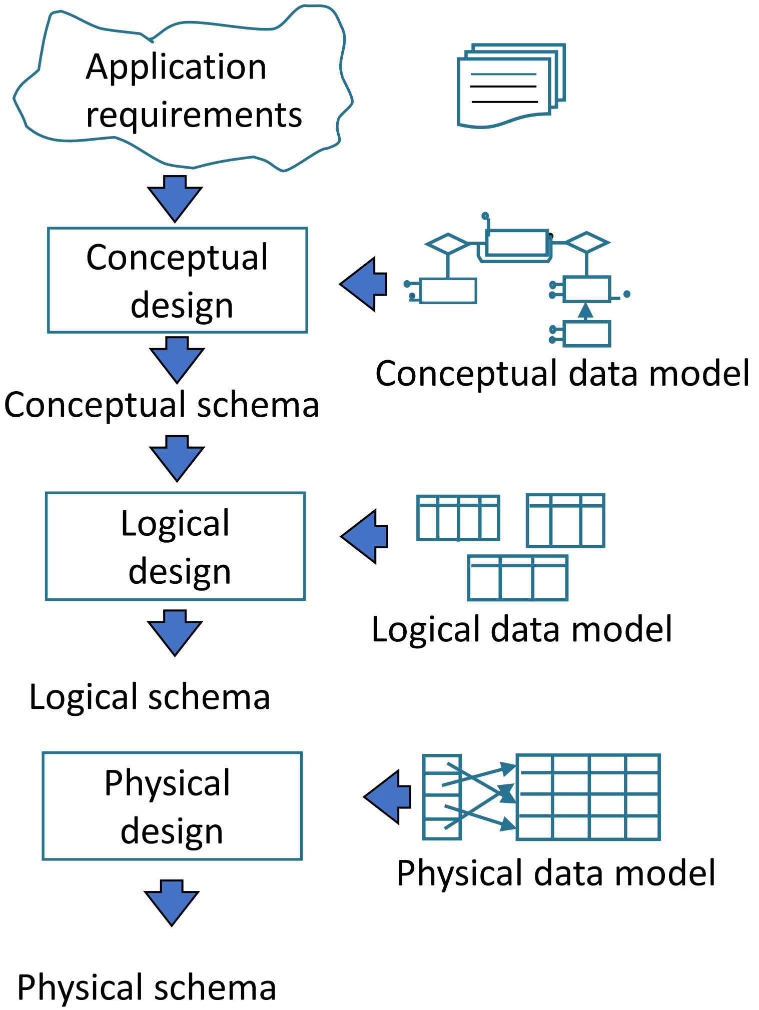
- Describes the data structure in the DBMS
  - used by the programmers accessing data
  - Indipendent from the physical data structures used in the DBMS
- Example: relational model

Design of a database

Represent the informal requirements of an application in terms of a conceptual schema that refers to a conceptual data model.

Translation of the conceptual schema previously defined into the database logical schema that refers to a logical data model

The logical schema is completed by the details of the physical implementation (file organization and indices) on a given DBMS. The result is called physical schema and refers to a physical data model



Application requirements

Conceptual design

Conceptual schema

Conceptual data model

Logical design

Logical schema

Logical data model

Physical design

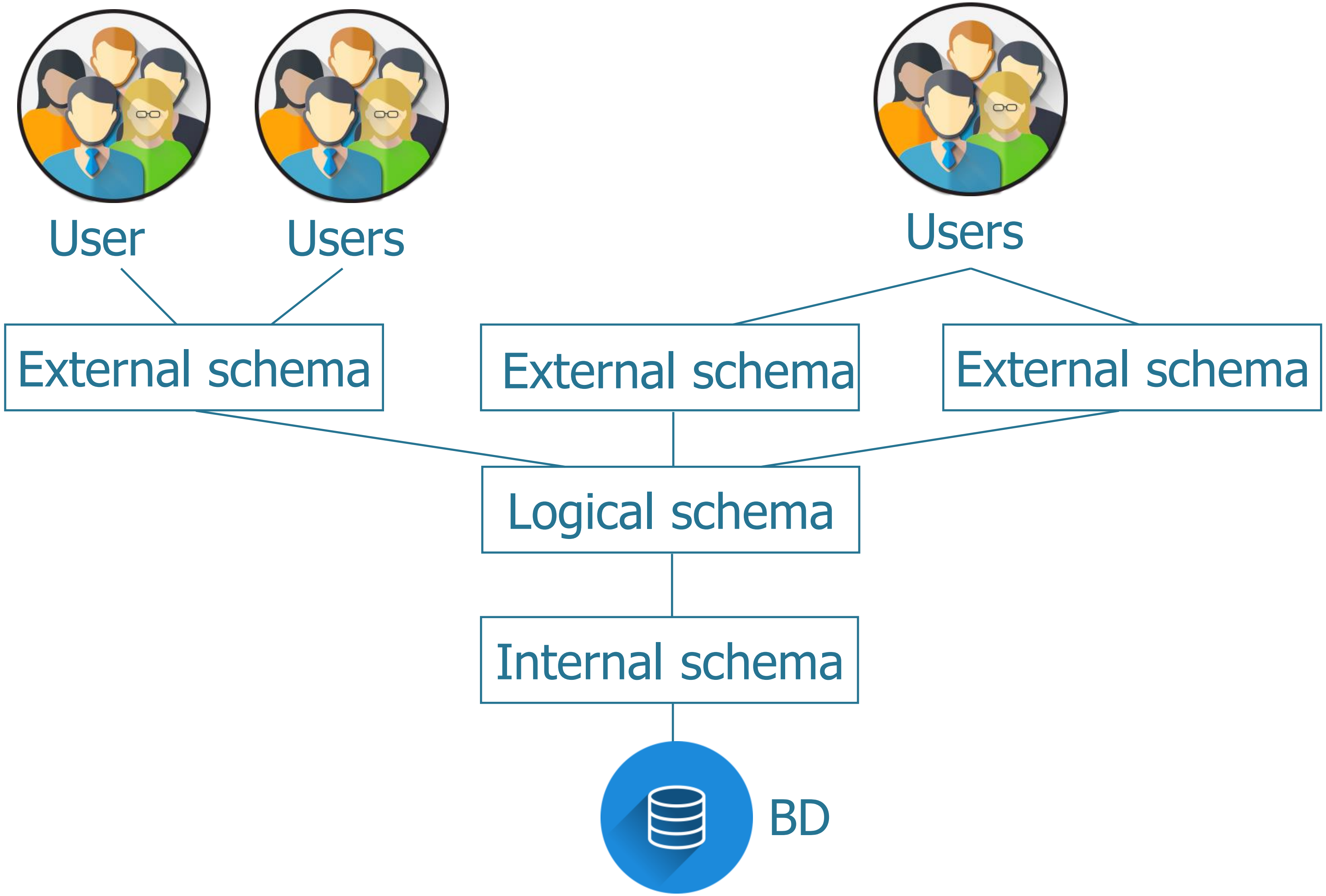
Physical schema

Physical data model

Data independence

Introduction to databases

Level of abstraction of a DBMS



Level of abstraction of a DBMS

- Logical schema
  - Description of the database using the logical model of the DBMS
- Internal schema
  - Representation of the logical schema using physical storing structures
- External schema
  - description of parts of the database, called "views", which reflect the point of view of particular users
  - Defined on the logical model

Data independence

- Data independence guarantees that users and application programmes which utilize a database can ignore the designing details used in the construction of the database
- It is a consequence of the subdivision into levels of abstraction
- Levels of data independence:
  - Physical independence
  - Logical independence

Data independence

- Physical independence
  - Enables interaction with the DBMS independently from the physical structure of the data
  - Access to a (logical or external level) relationship always takes place in the same way, independently from the means of memorization
  - it is possible to change the way the data is physically memorized without affecting the programmes utilizin the data

Data independence

- Logical independence
  - Enables interaction with the external level independently from the logical level
  - it is possible to change the logical level maintaining the external structures unaltered (as long as the correspondences are unaltered)
  - it is possible to add new views or alter existing views without changing the logical schema

Data access

Introduction to databases

Data access languages

- User-friendly interfaces that enable specific queries without using a textual language
- Interactive languages (es., SQL, DSL)
- Command similar to interactive commands introduced into traditional programming (C, C++, COBOL, Java, Python, …), so-called host languages
- ORM (Object-Relational Mapping): mapping objects in programming languages with database tables/documents
- Commands similar to interactive commands introduced into ad hoc development languages, often with specific functionalities (generation of graphics, complex prints, screens)

Data access languages

- Languages are divided into two categories
  - *Data Definition Languages* (DDL) used to define the logical, external and physical schemas, and access authorizations
  - *Data Manipulation Languages* (DML) used for querying and updating database instances

Users

- **Database administrator**: in charge of (centralized) control and management of the databse
  - Guarantees sufficient performance
  - ensures system reliability
  - Manages authorizations and access to data
- **Designers and programmers**: they define and realize
  - the structure of the database
  - the programmes accesing the database
- **Users**: utilize the database for their activities
  - end users: they use transactions, i.e. programmes that carry out predefined activities
  - casual users: they formulate queries (or updates) which are not predefined by the interactive access languages of the database

Transactions

- Programmes that carry out frequent predefined activities
- Example
  - Flight bookings
  - Bank transfers
  - purchase on e-commerce
- Generally realized by introducing SQL into a host language

Operational database, data warehouse Business intelligence

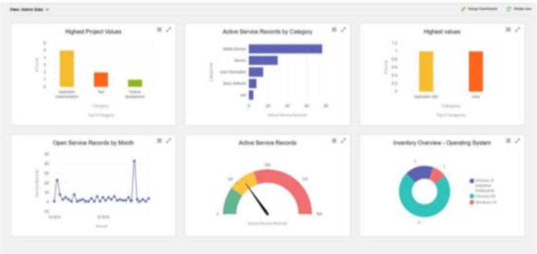
Introduction to databases

Operational database vs. data warehouse

- *Operational Database* includes detailed information used to run the day to day operations of the business.
  - Applications frequently access operational databases to both read and update the database content
  - Operational Database Management Systems also called as OLTP (Online Transactions Processing Databases), are used to manage dynamic data in real-time.
  - Operational databases are the source of information for the data warehouse.
- *Data Warehouse* is data repository devoted to decision support, which is kept separate from company operational databases, and integrates data from different data sources
  - Data warehourese is a data repository used to support data analysis and decision-making.
  - These systems are called as Online-Analytical Processing (OLAP) Systems.
- Data Warehouse and the OLTP database are both relational databases. However, the goals of both these databases are different.

Business Intelligence

- BI provides support to strategic decision support in companies
- Objective: transforming company data into actionable information
  - at different detail levels
  - for analysis applications
- BI requires an appropriate hardware and software infrastructure
- Data to be analysed are usually stored in a data warehouse
- Mined information can be visualized using informative dashboards


Highest Project Values
Active Service Records by Category
Highest values
Open Service Records by Month
Active Service Records
Inventory Overview - Operating System

Advantages and disadvantages of DBMS

Introduction to databases

DBMS advantages

- Data as a common resource of the whole organization
    - Reduction of redundancies and inconsistencies
- Unified and precise data model of facts of interest to the organization
- Possible centralized control of data
    - standardization, economies of scale
- Data independence

DBMS disadvantages

- These are expensive, complex products that require
  - direct investment
    - Purchase of the product
  - indirect investments
    - Purchase of the necessary hardware and software resources
    - Conversion of the applications
    - Training of personnel
- They provide a set of services in an integrated form
  - it is not possible to separate out unsued services that cause a reduction in perdormance