UNIVERSITÀ DEGLI STUDI DI UDINE

Data Management for Big Data
Introduction to databases

Andrea Brunello
andrea.brunello@uniud.it

Outline

- ➔ Different kinds of data
  - ◆ *There's a lot of different stuff out there*
- ➔ The need for databases
  - ◆ *Why can't we simply store files on the disk?*
- ➔ Abstraction levels and data models
  - ◆ *Things that allow us to define and organize data*
- ➔ History of information systems
  - ◆ *When did it all start?*

Data, information, knowledge

- Regardless of the domain considered, data are the lifeblood of corporate information systems

- **Data** can be defined as a collection of raw, unorganised and unanalysed material relating to a phenomenon (e.g., a sensor connected to an engine outputting decimal numbers)

- Through a processing and interpretation phase, the data become meaningful **information** for the recipient (e.g., the sensor is a thermometer, thus, numbers are interpreted as a temperature in degrees °C)

- Combining information, intuition and experience yields **knowledge**, which can be compared with knowledge already acquired and which allows us to interpret and guide our actions (e.g., based on previous knowledge, the person schedules a maintenance task on the engine)

Different kinds of data

➔ Traditionally, data considered by information systems used to be very simple

➔ Things started to change in the **2000s**, following the rise of the Internet and its applications, such as social networks

➔ We can broadly distinguish between the following kinds of data:

- **Structured** data
- **Semi-structured** data
- **Unstructured** data

Structured data can be seen as tabular data, represented by **rows** and **columns**

- ➔ Each row belonging to a same table has a fixed format, think about an Excel file
- ➔ For instance, personal data regarding customers
- ➔ Easy to store and process, but they convey a limited amount of information

	A	B	C	D	E	F
1	Name	Surname	Birth date	Gender	Address	Phone number
2	John	Doe	12/07/74	M	660 North Gonzales Ave. Canyon City, CA 91387	202-555-0123
3	Mary	Jane	15/08/90	F	7772 Shore St. Zion, IL 60099	202-555-0176
4	Darell	Smart	07/03/83	M	2 Pine Ave. Royal Oak, MI 48067	202-555-0197
5	Jessica	Miller	03/11/95	F	7015 Wilson St. South Portland, ME 04106	202-555-0197
6	Belinda	Barton	05/12/67	F	22 Wakehurst Street Jackson, NJ 08527	202-555-0197
7	Ned	Fitzgerald	26/10/77	M	12 Sage Rd. Hope Mills, NC 28348	202-555-0197

Semi-structured data do not have a fixed format; still, there is some kind of structuring within them

- ➔ They make use of **tags** or other markers that allow to organize the content and establish hierarchies within the data
- ➔ For example: XML files

```
{
    "EMPLOYEES": {
        "SALES": {
            "648229": {
              "NAME" : "Olivia Johnson"
              "DOB" : "1989-08-08"
            },
            "648666": {
              "NAME" : "Frank Mueller"
              "DOB" : "1985-05-11"
              "MISC" : "On paternal leave from 2019-01-01 until 2020-01-01"
            }
        }
    }
}
```

Unstructured data is **not organized** in any predefined manner

- ➔ Free text
- ➔ Audio and visual materials
- ➔ Difficult to store, index, and analyse

Maura

★★★★☆ **Soddisfatta**

Recensito in Italia il 30 ottobre 2022

Colore: Bianco | Nome stile: Cassettiera per l'armadio | **Acquisto verificato**

Buon designe, facile da montare, o cassetti sono capienti e robusti hanno il fondo rinforzato con un pannello estraibile di cartone rivestito. Il montaggio lo si può seguire dalle istruzioni illustrate nel libretto, il piano d'appoggio robusto, andrebbe migliorata la rifinitura. Leggero da spostare misure dopo il montaggio altezza 72,5 cm Larghezza 46,80 cm profondità 30 cm

Database management system (DBMS)

A DBMS contains information on a particular domain

- ➔ Collection of **interrelated data** (database)
- ➔ Set of **programs** to manage the data

For example, in the university domain:

- ➔ **Data**: students, professors, courses, …
- ➔ **Programs**: add a new student, modify the salary of a professor, enrol a student to a course, …

DBMSs can be very large, and they typically touch all aspects of our life!

In the early days, data-centric applications were built by means of:

- ➔ **Files**, e.g., in “csv” format
- ➔ **Programs** specifically designed to access such files

First relational database in the early 80s, after 10 years of work

- ➔ So, the need for a DBMS was seen from the beginning

Vehicles.txt

```
Plate,Year,Manufacturer,Model
AX252KF,2015,Ford,Mustang
BK356PQ,2020,Dodge,Charger
FH665HG,2019,Chevrolet,Corvette
CD124LP,2021,Chevrolet,Camaro
```

Belongs_to.txt

```
Plate,Customer_name,Customer_surname
AX252KF,John,Doe
BK356PQ,Mary,Jane
FH665HG,Jessica,Miller
CD124LP,Ned,Fitzgerald
```

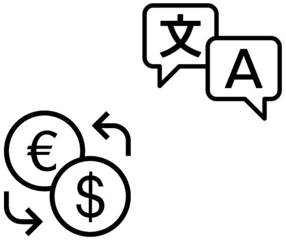
Customers.txt

```
Name,Surname,Birth_date,Gender
John,Doe,1974-07-12,M
Mary,Jane,1990-08-15,F
Darell,Smart,1983-03-07,M
Jessica,Miller,1995-11-03,F
Belinda,Barton,1967-12-05,F
Ned,Fitzgerald,1977-10-26,M
```

Drawbacks of relying on files and programs

Data **redundancy and heterogeneity**

- ➔ Multiple file formats and programming languages
- ➔ Duplication of information in different files
  - ◆ Possible data inconsistency!



Difficulty in **accessing data**

- ➔ Need to write a new program to carry out each new task
- ➔ Low interactivity

Drawbacks of relying on files and programs

Constraints management

- ➔ Domain constraints (e.g., car value >= 0) become «buried» in program code rather than being stated explicitly
- ➔ Hard to add new constraints and manage existing ones

Atomicity of updates

- ➔ Failures may leave the system in an inconsistent state
- ➔ E.g., a transfer of funds from one account to another should be either carried out in its entirety or not happen at all

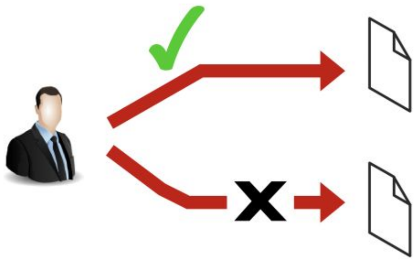
Drawbacks of relying on files and programs

Concurrent access by multiple users

- ➔ What happens if two users try to access and modify the same data?
- ➔ E.g., two users booking the last airline ticket

Security problems

- ➔ Grant users access to some, but not all, data



(Relational) DBMSs provide solutions to all these problems!

Characteristics justifying the use of a DBMS

- **Large** amounts of data

- **Global** data
  - They are relevant for a vast array of users and applications
  - The database is typically not tied to a particular user or application, as standard programs are

- **Persistent** data
  - The database is there independently from the interacting users or applications
  - Data lives on its own unlike, for instance, program variables

DBMS and stratification

- To be capable of dealing with all the previous stuff, a DBMS must be a very complex piece of software
- Thus, to allow for its development and management, it is typically organized following a **stratified approach**
- This means that the DBMS functionalities are arranged into **different hierarchical levels**
  - each level encapsulates/hides the details it deals with,
  - it offers its functionalities to the higher levels,
  - and it can rely on the functionalities made available by the lower levels, without caring about their implementation (**abstraction**)

From lowest to highest:

- ➔ **Physical level**: *how* data is stored on the disk (data structures, file arrangement, …)
- ➔ **Logical level**: *what* data is stored and represented in the database and their relationships
  - ◆ describes an entire database in terms of a small number of relatively simple structures
  - ◆ e.g., data is stored within tables, or graphs
  - ◆ they are *tables/graphs*, we do not care about how they are stored on the disk (the physical level deals with it)
- ➔ **View/External level**: it manages the presentation of information to users and programs. Views can also hide information to some kinds of users (e.g., show only some parts of a table)

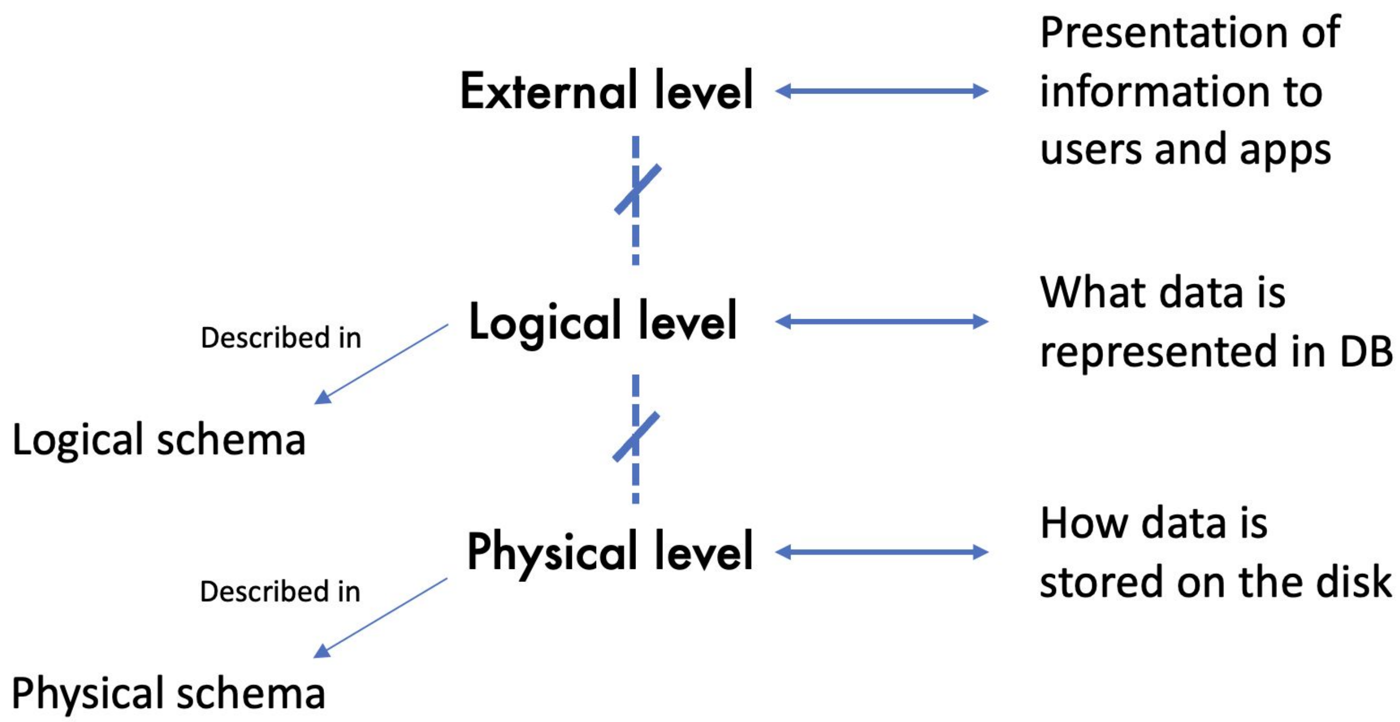
Database schemas

The levels of abstraction of a specific database are described by means of *schemas*:

- ➔ **Physical schema**: it lays out how data are stored physically on a storage system in terms of files and data structures
- ➔ **Logical schema**: it encodes the overall logical structure of the database
  - ◆ E.g., the database consists of a set of tables that are designed to store information about a set of customers (each characterized by name, address, and phone number) and their accounts in a bank; the relationships among customers and accounts; and, some constraints over them

Physical data independence: ability to make changes to the physical level without having to modify the logical one

Recap: levels and schemas



Data models

- To the describe the logical schema, a small number of relatively simple structures is used (data model)

- A **data model** is a collection of tools (like an alphabet) for describing
  - Data (and its structure)
  - Data relationships
  - Data constraints

- There exist several logical data models, including
  - The **relational data model**, which considers rows and tables (records and relations)
  - Graph data model, that deals with nodes and vertices
  - Object-based data models
  - Hierarchical model

Tables to represent data and relationship among the data

Students

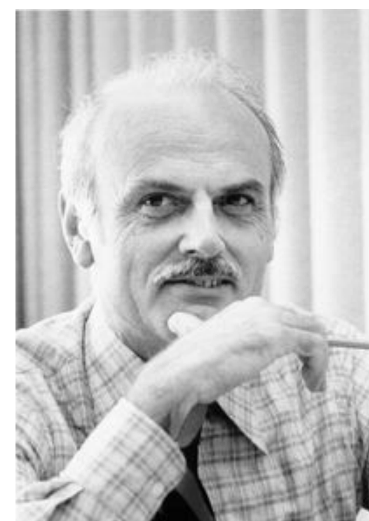
ID#	Name	Phone	DOB
500	Matt	555-4141	06/03/70
501	Jenny	867-5309	3/15/81
502	Sean	876-9123	10/31/82

ID#	ClassID	Sem
500	1001	Fall02
501	1002	Fall02
501	1002	Spr03
502	1003	S203

Takes_Course

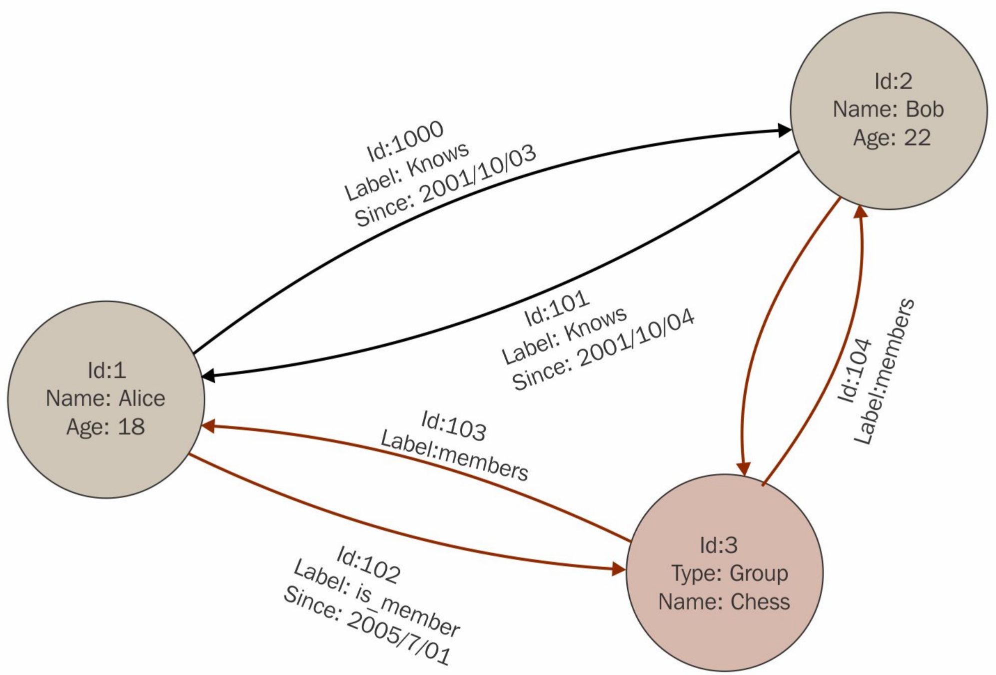
ClassID	Title	ClassNum
1001	Intro to Informatics	I101
1002	Data Mining	I400
1003	Internet and Society	I400

Courses



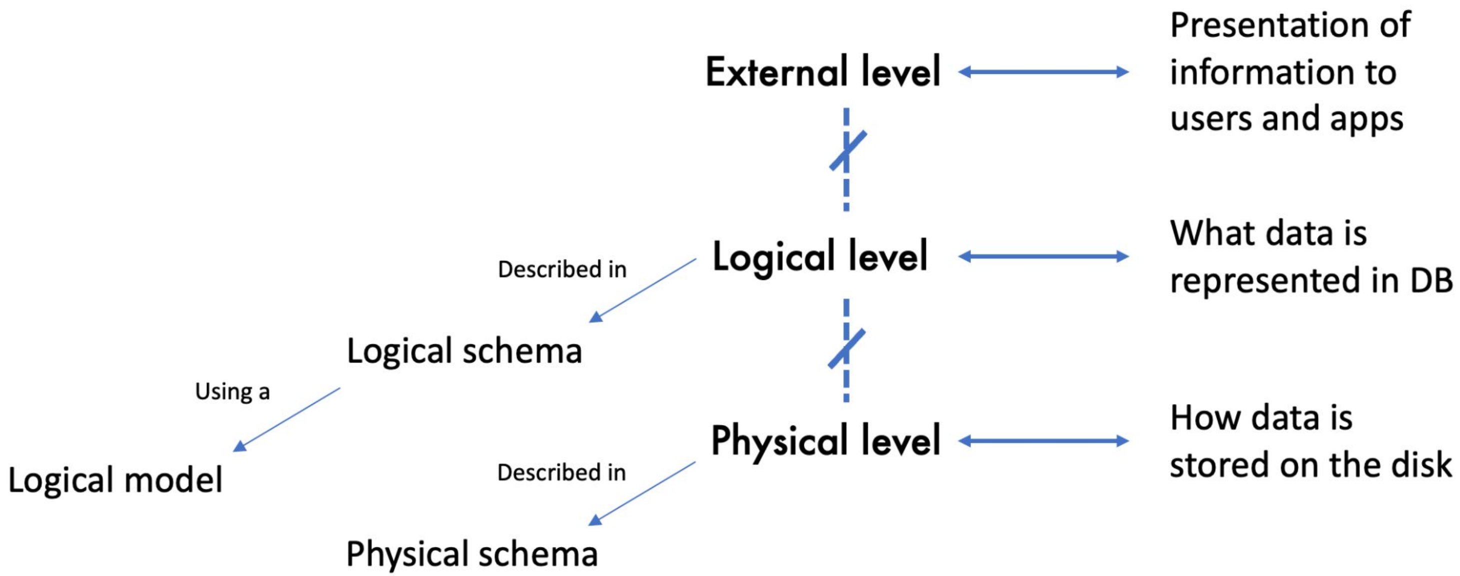
Graph data model

Nodes and arcs to represent data and relationship among the data



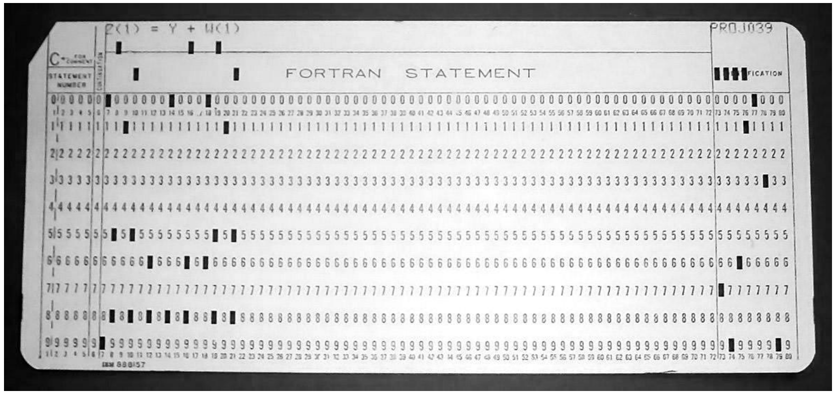
Records *physically* point to other records; more flexible structure for the nodes and the relationships

Recap: levels, schemas, and models



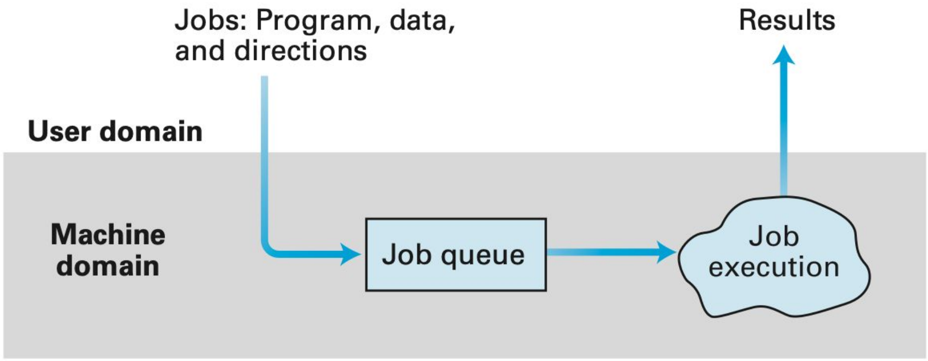
From bookkeeping to magnetic tapes

- Before electronic data processing, companies used to manage their customers, purchases and inventory using **traditional bookkeeping methods**
- Early electronic data (batch) processing came about in the 1950s; initial systems were based on **punch cards**, naturally exposed to damage and loss of data
- From 1960s magnetic tapes provided better data storage, but **sequential access** required full tape scan even for 5% of data. Many dedicated hardware as many formats available

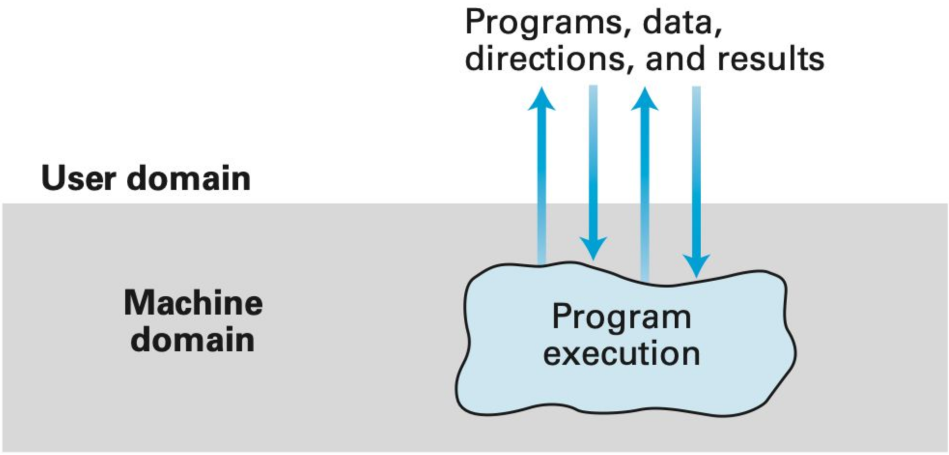


In the decades 1950s-1960s we have also the switch from batch to interactive processing

Batch processing



Interactive processing



Direct access storage device

- The 1970s saw the advent of **disk storage**, also referred to as direct access storage device (DASD)
- No need to go through records 1, 2, 3, …, n to get record $n + 1$ once the location address of $n + 1$ is known
- The time to locate a record is now measured in milliseconds
- New, more complex data structures were developed, such as lists and trees to be stored on disk
- Along with the DASD, it came a new type of system software known as a **database management system** (DBMS)

Online transaction processing

- ➔ With DBMS it was easier to store and access data on a DASD; moreover, the DBMS took care of tasks such as storing data on a DASD, indexing data, managing access rights, and so forth

- ➔ By the mid-1970s, **online transaction processing** (OLTP) made even faster access to data possible. Applications like bank teller systems and manufacturing control systems became possible

- ➔ At this point, network and hierarchical data models became of widespread use

Meanwhile, Ted Codd defines the **relational data model**:

- ➔ Would win the ACM Turing Award for this work
- ➔ Points of strength: simplicity and hiding of physical details
- ➔ IBM Research begins work on *System R* prototype
- ➔ UC Berkeley begins work on *Ingres* prototype
- ➔ They both still exhibit computationally worse performance with respect to previous network and hierarchical data models
- ➔ By the 1980s the prototypes evolve into commercial systems
- ➔ *SQL* becomes industrial standard

Wal-Mart and the birth of Data Warehousing

- Around 1990 Wal-Mart retail corporation began to achieve wide acclaim for its mastery of supply chain management
- Behind this success there was a **data warehouse**, and a new way of interacting with data, called **online analytical processing** (OLAP)
- Data is collected by its point-of-sales systems to achieve unprecedented insight into the purchasing habits of its 100 million customers and the logistics guiding its 25,000 suppliers
- Wal-Mart’s data warehouse was the first commercial Enterprise Data Warehouse to reach 1 terabyte of data in 1992

- ➔ Around 2000s, the types of data stored in database systems evolved rapidly, pushed by an ever increasing usage of the Internet and multimedia
- ➔ The variety of new data-intensive applications led to **NoSQL** systems, which gave programmers greater flexibility to work with new types of data, but lacked a high level query language
- ➔ **Distributed** storage and computing frameworks were developed, such as Hadoop
- ➔ To allow for the interchange of information between systems, formats such as XML and JSON became of widespread usage
- ➔ Data Mining applications started to emerge
- ➔ Most of large and medium sized organizations started to rely on **decision support systems**

- A decision is the selection of a course of action from a set of alternatives
- A decision support system recommends such an action by offering managers information upon which to build ideas so to come up with the final choice
- A **decision management system** is an *action-oriented* evolution of a decision support system
- It makes one step more and takes decisions without human intervention based on known information and a set of coded business rules or artificial intelligence models
- Of course, not all judgments may be automated (strategic vs operational decisions)

References

- A. Silberschatz, H.F. Korth, S. Sudarshan, *Database system concepts*, 7th Edition, 2020
- W.H. Inmon, *Building the Data Warehouse*, 4th Edition, 2005