Unit 1: Introduction

1 - Data processing

2 - Files. Pros and cons. Operations.

3 - Files management systems.

4 - Databases systems.

5 - Components and functions in database systems.

6 - Advantages and drawbacks.

Data processing

What is data?

We will consider data the representation and identified record of any value, attribute, fact or magnitude, either in a quantitative way or in a qualitative way.

Examples:

- red
- 10 meters
- 185 km/h

Data and information

Are they the same thing? Do we mean it?

Information = data processed and transmitted

Information must add to the data some sort of signification, this way it will get utility and value.

Examples:

- Red → colour of a house
- 10 meters → the width of a certain road
- 185 km/h → maximum speed of a given vehicle

Data processing

In order to process data in any organisation or field, the existence a system that allows their conservation, tracking or preservation is required.

Data processing has undergone an evolution throughout time.

Origin: files and paper based documents.

Manual handling

Data processing

When computer technology came up, then spread and became popular, the data went on to be stored in electronic format.

Then the manipulation of data became

AUTOMATIC

Files

Electronic computer files are the first form of data electronic treatment.

Definition:

A set of data (information) with some kind of relationship between them, and that are stored as a unit in a hardware element (disk, tape, memory card, …) in a permanent, non-volatile way.

The cause for the files to come up and usage is precisely the volatility of the computer memory (RAM, random access memory) because they make up for that non-persistence of the data; thus, allowing the massive storage of all of them without anything getting lost when the systems are switched off.

Files

A first initial classification, according to their use or function, can be established:

- Permanents (master, constants, logs, history)
- Temporary (auxiliary, intermediate, of movements or data transfer, results)

Files are internally structured in records, and in turn, these are divided into fields.

We consider the latter ones as the minimum unity of information with proper signification that can be stored and we can operate with as well.

File structure

registro 0	C-102	Navacerrada	400
registro 1	C-305	Collado Mediano	350
registro 2	C-215	Becerril	700
registro 3	C-101	Centro	500
registro 4	C-222	Moralzarzal	700
registro 5	C-201	Navacerrada	900
registro 6	C-217	Galapagar	750
registro 7	C-110	Centro	600
registro 8	C-218	Navacerrada	700

Records and fields

The previous records are considered logical records. Not necessarily they are meant to match the physical records → in one of these, either part of one or several logical records might be contained.

The physical records are in turn divisions of the non-volatile memory, and they are to be read or written in any access operation.

Once again, they may have a fixed or variable length with a certain number of bytes.

Record access

- Sequential (in given order)
- Direct (by key: each record has a single unique, key that is used by an algorithm to retrieve the address in the physical memory)
- Indexed (consistent on a table with keys and relative directions)
- Dynamic (a combination of the direct or indexed access with the sequential access; that is, the first redirection will lead to a group or set of records)

Files: sequential organisation

Pros:

- Quick way to access adjacent records, placed next to each other
- Compact (good use of space with no empty places)

Cons:

- Sequential access to n-th placed record
- Inefficient when it comes to queries due to comparisons
- No physical removal of records (solution: chained sequential organisation)
- Sorting

Files: direct organisation

Pros:

- Immediate access to every record → speed
- Read and write operations can be performed simultaneously

Cons:

- Sequential accesses (must go through empty records)
- Inefficiency (due to empty spaces)
- Collisions

Files: indexed / dynamic organisation

Pros:

- Sorting the records is not required
- Immediate access to the record
- Sequential accesses

Cons:

- When a full file content must be retrieved, the query is slow
- Empty spaces, use of storage
- Complex algoryihms

Operations with files

On the file:

- Open
- Close
- Sort
- Compact spaces
- Copy
- Remove
- Concat / chain / merge

On the records:

- Add / insert
- Update
- Delete
- Check / access
- Copy
- Move

Files management systems

The story, so far → all data are contained within a single file

However, as the data volume processing goes up, so do the requirements.

Solution: use multiple files

Files management system:

Dedicated software to handle and manipulate sets of files, typically presented in a variety of different formats and for multiple applications.

Files management systems: drawbacks

Let’s imagine an organisation with a number of departments:

- We’ll need to create or embrace new programs to suite the ones already in use → important effort of understanding the way the data are structured and stored
- Difficult changes (multiple and diverse formats, different programming languages, etc.)
- Data not integrated → they are isolated!
- The data may be repeated in an unnecessary way (redundancy) → this lead in turn to problems of integrity and consistency (wrong or incomplete information, not updated everywhere at the same time)

Files management systems: drawbacks

All of the previously exposed reasons mean:

- inconvenience (when updating)
- Storage issued (due to the occupied volume of data)
- Efficiency (both in space storage and in access time)

Therefore, in order to remove the redundancies it is required a process of integration, both for the data and for the software.

Files management systems: drawbacks

To sum up:

- Problems regarding the programs and software:
  - Semantic integrity control, authorisations, data access concurrency
  - Restrictions of consistency
  - Security controls and complex simultaneous accesses
- Problems regarding the data:
  - Redundancy
  - Inconsistency
  - Fragmentation
  - Access

Database systems

They come up in the 60s, trying to be the response to the problems of the file management systems.

To do so, they will provide tools and resources for a more efficient and effective data management, and easy ways to their update and usage.

Main advantage:

- Logical independency of the data (update some data does not imply other changes)
- Physical independency of the data (the used device has no influence in the software and its operation)

→ Lower costs of storage and maintenance

Concepts

Database: a set of inter-related data with an optimal internal organisation for their manipulation.

Database Management System (DBMS): dedicated software (a set or group of programs) that allow to define and create databases as well as inserting, retrieving, updating or deleting data from those.

Examples: Oracle, MySQL, Postgres, MariaSQL, MongoDB, Neo4j, Redis

Database system = database + DBMS

Functions of a database system

- Data definition
- Data manipulation
- Grant the integrity
- Provide security and recovering tools and mechanisms
- Foresee and allow the access concurrency
- Optimal and efficient physical data organisation

Components of a database system

- Hardware
- Dedicated software:
  - DDL module
  - DML module
  - User interface
  - File management module
  - Concurrency access and control module
- Data (including the data dictionary)
- Users (administrator, end users, operators, …)

Types of databases

- Centralized: when the database is located in a unique physical place (a single machine).

- Distributed: in this case, data and software are scattered throughout several machines that are connected in the same computer network.

Database systems: advantages

* *Integrity*: set of restrictions that allow or deny to store certain values in a database

- Centralised control:
  - data
  - users
- Integration: the implemented elements neither give rise to inconsistencies nor redundancies, and mechanisms for a better error control are provided
- Efficient and self-managed physical storage (therefore it is feasible to delegate tasks on it)
- Database integrity is granted even though is data are to be shared between users or locations, or when their objects or components are updated
- **Data dictionary**: a very powerful tool that stores metadata, i.e. specifications about the self database → this makes it easier aspects such as the understanding, potential changes, efficient re-designs, and so on

Database systems: drawbacks

- Initial (high) cost → departmental transformations, software purchasing, initial work to embrace adaptation, creation or configuration, equipment and hardware up-to-date, ...
- Amortization of investment → the system will start to be profitable only after some time has gone by
- Database designs → they are not unique for every problem or situation, nor they are obtained in an immediate or mechanical way
- If the decision to make the database distributed is taken, checking potential redundancies or inconsistencies issues will be a requirement
- Even if there are standards, it is difficult or unlikely that the DBMS (and commercial solutions and products in particular) fulfil all the specifications. Moreover, sometimes they deliver features that are not compatible at all between database systems!