

# DATA BASES INTRODUCTION

## DataBase Foundations

Author: Eng. Carlos Andrés Sierra, M.Sc.  
cavirguezs@udistrital.edu.co

Lecturer  
Computer Engineer  
School of Engineering  
Universidad Distrital Francisco José de Caldas

2024-III



- 1 Software Components and Applications
- 2 Glossary
- 3 DataBase Classification
- 4 MER Diagrams
  - Study Case: Spotify



# Outline

- 1 Software Components and Applications
- 2 Glossary
- 3 DataBase Classification
- 4 MER Diagrams
  - Study Case: Spotify



# Modular Software Components

- **Software Components** are the building **blocks** of software systems.
- **Modular Software** is a software design technique that emphasizes **separating** the **functionality** of a program into independent, interchangeable **modules**.
- **Software Applications** are the **final product** of software development.
- **Software Development** is the process of **creating** software applications.



# Modular Software Components

- **Software Components** are the building **blocks** of software systems.
- **Modular Software** is a software design technique that emphasizes **separating** the **functionality** of a program into independent, interchangeable **modules**.
- **Software Applications** are the **final product** of software development.
- **Software Development** is the process of **creating** software applications.



# Modular Software Components

- **Software Components** are the building **blocks** of software systems.
- **Modular Software** is a software design technique that emphasizes **separating** the **functionality** of a program into independent, interchangeable **modules**.
- **Software Applications** are the **final product** of **software development**.
- **Software Development** is the process of **creating** software applications.



# Modular Software Components

- **Software Components** are the building **blocks** of software systems.
- **Modular Software** is a software design technique that emphasizes **separating** the **functionality** of a program into independent, interchangeable **modules**.
- **Software Applications** are the **final product** of **software development**.
- **Software Development** is the process of **creating** software **applications**.



# History of Data

## HISTORY OF DATA

19,000 BC



The **Ishango bone** holds the first evidence of data collection and storage.

1600s



John Graunt introduces the **concept of data analysis** in 1663.

1800s



Herman Hollerith designs a **machine that helped complete the US census** in 1890.

1900s



Fritz Pfleumer invents the **magnetic tape** which later inspired the invention of floppy disks and hard disk drives.

1990s



Sir Tim Berners Lee invents the **World Wide Web**.





# Applications

- Software based on **layers** of **abstraction** and **modularity** lets implement different **database strategies**.
- Database Systems are fundamental for **data management**.
- Data analysis, data mining, data visualization, and data interpretation are **applications** of database systems.



# Applications

- Software based on **layers** of **abstraction** and **modularity** lets implement different **database strategies**.
- **Database Systems** are fundamental for **data management**.
- Data analysis, data mining, data visualization, and data interpretation are **applications** of **database systems**.



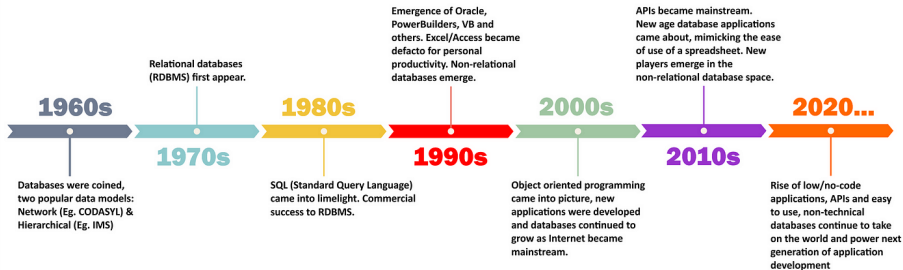
# Applications

- Software based on **layers** of **abstraction** and **modularity** lets implement different **database strategies**.
- **Database Systems** are fundamental for **data management**.
- **Data analysis**, **data mining**, **data visualization**, and **data interpretation** are **applications** of **database systems**.



# History of DataBases

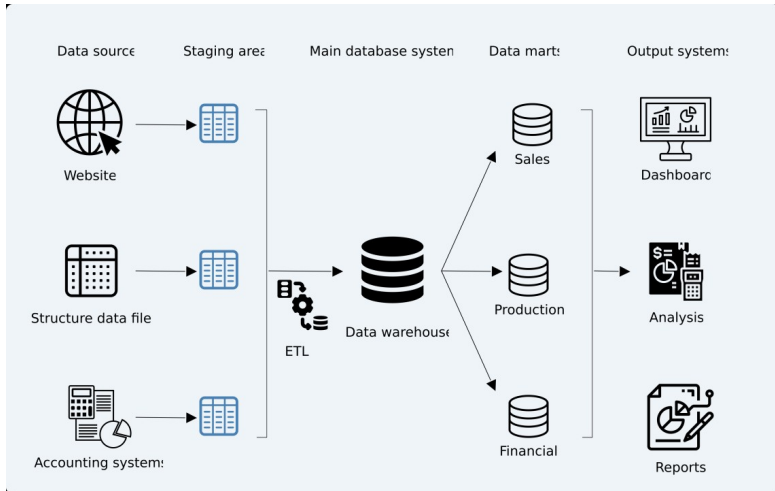
## History of Databases (1960-2020)



stackby.com



# Case of Study: DataBase System



# Outline

- 1 Software Components and Applications
- 2 Glosary**
- 3 DataBase Classification
- 4 MER Diagrams
  - Study Case: Spotify



# From Data to Information

- **Data**: is a set of **values** of **qualitative** or **quantitative** variables.
- **Data Management**: is the process of **collecting**, **storing**, **processing**, and **analyzing** data.
- **Data Analysis**: is a process of **inspecting**, **cleansing**, **transforming**, and **modeling** data with the goal of **discovering** useful **information**, informing **conclusions**, and supporting **decision-making**.



# From Data to Information

- **Data**: is a set of **values** of **qualitative** or **quantitative** variables.
- **Data Management**: is the process of **collecting**, **storing**, **processing**, and **analyzing** data.
- **Data Analysis**: is a process of **inspecting**, **cleansing**, **transforming**, and **modeling** data with the goal of **discovering** useful **information**, informing **conclusions**, and supporting **decision-making**.



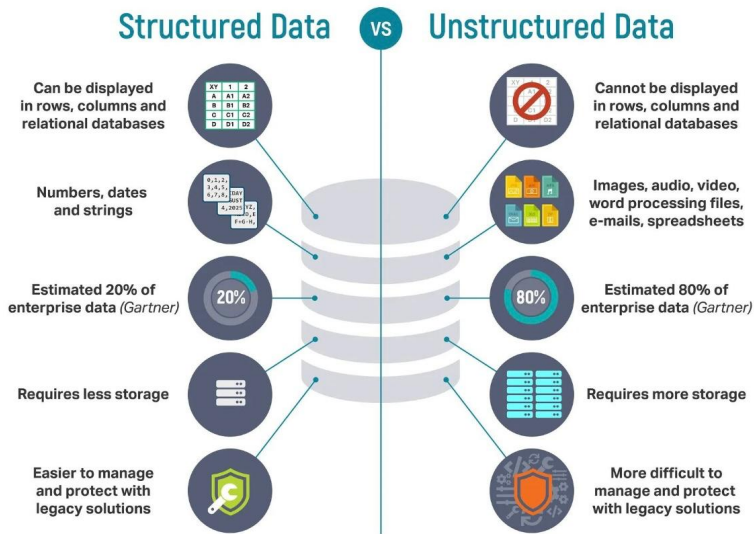


# From Data to Information

- **Data**: is a set of **values** of **qualitative** or **quantitative** variables.
- **Data Management**: is the process of **collecting**, **storing**, **processing**, and **analyzing** data.
- **Data Analysis**: is a process of **inspecting**, **cleansing**, **transforming**, and **modeling** data with the goal of **discovering** useful **information**, informing **conclusions**, and supporting **decision-making**.



# Structured and Unstructured Data



# Tables, Columns and Rows

- **Table** is a collection of **related** data held in a **structured** format within a **database**.
- **Column** is a set of **data values** of a particular **simple type**, one for each row of the table.
- **Row** is a set of **data values** of a particular **relationship**, one for each column of the table.



# Key-Value Data Structures

- **Key-Value Data Structures** are a type of **data structure** that can map **keys** to **values**.
- **Key** is a **unique** identifier for a **record** in a **data fragment**. **Value** is the **data** that is **associated** with the **key**.



# Primary and Foreign Keys

- **Primary Key** is a **unique** identifier for a **record** in a **data set**.
- **Foreign Key** is a **column** or **group of columns** in a **table** that **links** to a **primary key** in another **table**.



# CRUD Operations

- **CRUD** is an acronym for **Create**, **Read**, **Update**, and **Delete**.
- **Create** is the process of **adding** new **records** to a **data set**.
- **Read** is the process of **retrieving records** from a **data set**.
- **Update** is the process of **modifying records** in a **data set**.
- **Delete** is the process of **removing records** from a **data set**.



# Outline

- 1 Software Components and Applications
- 2 Glossary
- 3 DataBase Classification**
- 4 MER Diagrams
  - Study Case: Spotify



# DataBase Classification

- **DataBase** is a collection of **data** that is **organized** so that it can be **easily accessed, managed, and updated**.
- **Relational DataBase** is a type of **database** that stores and provides access to **data points** that are **related** to one another.
- **NoSQL DataBase** is a type of **database** that provides a mechanism for **storage and retrieval** of **data** that is **modeled** in **means other** than the **tabular relations** used in **relational databases**.





# DataBase Classification

- **DataBase** is a collection of **data** that is **organized** so that it can be **easily accessed, managed, and updated**.
- **Relational DataBase** is a type of **database** that stores and provides access to **data points** that are **related** to one another.
- **NoSQL DataBase** is a type of **database** that provides a mechanism for **storage and retrieval** of **data** that is **modeled** in **means other** than the **tabular relations** used in **relational databases**.



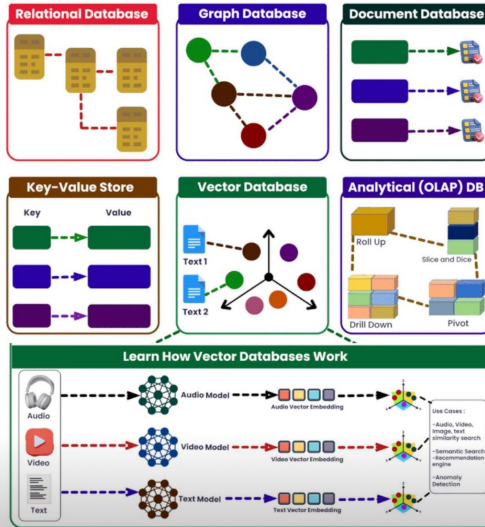
# DataBase Classification

- **DataBase** is a collection of **data** that is **organized** so that it can be **easily accessed, managed, and updated**.
- **Relational DataBase** is a type of **database** that stores and provides access to **data points** that are **related** to one another.
- **NoSQL DataBase** is a type of **database** that provides a mechanism for **storage** and **retrieval** of **data** that is **modeled** in **means** other than the **tabular relations** used in **relational databases**.



# Types of Database

## How Many Types of Database Do You Know?



# Outline

- 1 Software Components and Applications
- 2 Glossary
- 3 DataBase Classification
- 4 MER Diagrams**
  - Study Case: Spotify



# Entity-Relationship Model

**Entity-Relationship Model** is a **data model** for **describing** the **data** or **information** aspects of a **business domain** or its **processes**.



# Entity Definition

**Entity** is a **thing** or **object** in the **real world** that is **distinguishable** from other



# Relationship between Entities

**Relationship** is a **connection** between **entities**. This connection could be **one-to-one**, **one-to-many**, and **many-to-many**.



# Creating our own **Espotifai**





# Outline

- 1 Software Components and Applications
- 2 Glossary
- 3 DataBase Classification
- 4 MER Diagrams
  - Study Case: Spotify



# Thanks!

## Questions?



Repo: <https://github.com/EngAndres/ud-public/tree/main/courses/databases-foundations>

