Advanced Machine Learning

Module 2: SQL and Advanced Data Manipulation

Fall '23 @ SBU

Professor Hadi Farahani

Module Overview

In this module, we'll be covering the following topics:

- What are databases, and what do they even look like
- Data structure, and why it's even important
- SQL, and the importance of similar tools
- SQL Basics: `SELECT`, `FROM`, `WHERE`, Group by, ...
- Joining data, unions
- EDA with SQL
- Writing efficient queries
- Analytics functions
- Creating ML datasets
- Storing and modifying data

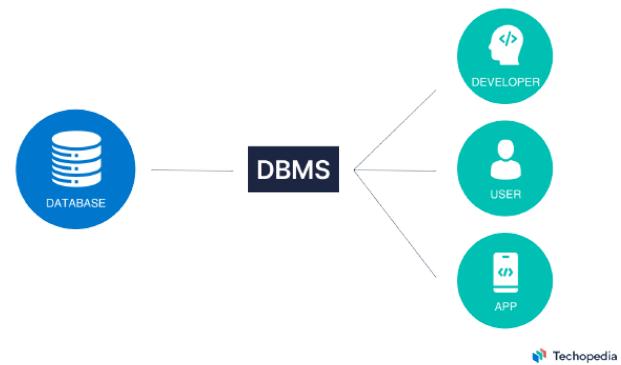
Introduction to Databases

So what is a database *exactly*?

A database is an organized collection of structured information, or data, typically stored electronically in a computer system.

- A database is usually controlled by a database management system (DBMS).

Together, the data and the DBMS, along with the applications that are associated with them, are referred to as a database system, often shortened to just database. (Source)



But what do we mean by “structured data”?

In general, when we’re talking about large, we have 3 types of data:

- **Structured data**: Structured data refers to information that is organized according to a specific format or schema, making it easy to store, search, and analyze.
  - It is typically found in databases and has a well-defined structure with fixed fields and data types. Examples include spreadsheets, relational databases, and tabular data.
- **Semi-structured data**: Semi-structured data is a type of data that does not have a rigid structure but contains some organizational elements. It may have tags, labels, or markers that provide a level of organization or hierarchy.
  - Semi-structured data is often represented in formats like CSV, XML or JSON and is commonly used in web-based applications, social media, and log files.

But what do we mean by “structured data”?

- **Unstructured data**: Unstructured data refers to data that does not have a predefined structure or organization.
  - It does not conform to any specific format and often includes text-heavy content such as emails, documents, audio, video, images, or free-form social media posts.

Unstructured data is challenging to analyze and store as it lacks a well-defined schema and requires advanced techniques like natural language processing and machine learning for processing and extracting valuable insights.

Data Structure

So for instance, CSV files (like we've previously worked with) and excel files are considered to be semi-structured data in the form of a text file, but it's not considered to be structured.

It's an incredibly simple task to convert a csv file into a relational database, which at that point we can use the CSV file for queries in SQL, a programming language designed for handling structured data.

(don't worry if these words are unfamiliar – we'll define them in a second)

Data strucure

So to recap, in this module we're looking to work with databases: structured data that uses certain tools to access and modify it.

A lot of the time, databases are incredibly large, and there are specific methods we need to use to be able to work with these vast amounts of data.

> So, are we talking about "big data"? No. Big data primarily refers to data sets that are too large or complex to be dealt with by traditional data-processing application software. (Source: Wikipedia).

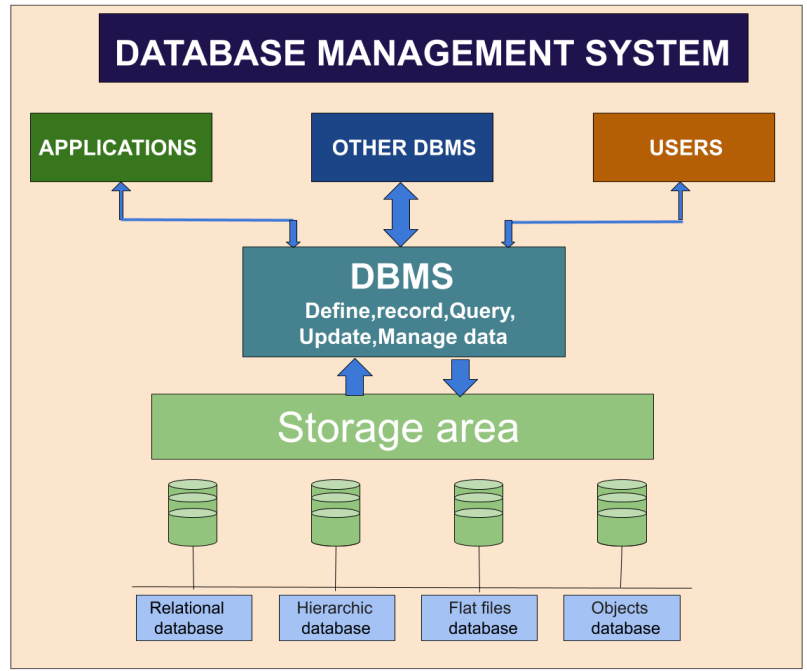
Here, we're talking about a middle ground of sorts; structured data that *might* be larger than your average csv file, but not so big that we need to use special tools to handle such data.

Database Management Systems

Previously, we briefly mentioned DBMS in relation to databases.

A database management system (DBMS) is a software application that allows users to efficiently store, organize, manage, and retrieve data within a database.

- It serves as an interface between the users and the database, ensuring data integrity, security, and efficient data access.



The Role of DBMS

By now you might be wondering what the need for a DBMS might be when we can just interface directly with a .db file?

Using DBMS as an intermediary layer between applications and databases provides numerous benefits in terms of data management, security, performance, scalability, and maintainability, making it a preferred approach in most software systems.

- **Data abstraction**: DBMS provide a layer of abstraction that hides the complexities of interacting with the database at a low level
- **Integrity and security**: DBMS enforce constraints, integrity, encryption, authentication and much more, which helps protect the database from unauthorized access and potentially (unintentionally) destructive actions.
- **Concurrent access and performance optimization**: Using a DBMS, multiple users can access a database simultaneously
- **Data independence**: DBMS provides a layer of separation between the application and the physical database. This separation allows changes to the database structure, storage, or organization to be made without impacting the application code.

SQL

We can interact with the database using the DBMS. But how do we work with the DBMS?

SQL, or **Structured Query Language**, is a programming language that is used for database management and manipulation.

In data science, SQL is used for several purposes:

1. **Data Extraction**: SQL is used to query and extract data from databases. Data scientists can write SQL queries to retrieve specific data points or subsets of data that are relevant to their analysis.
2. **Data Cleaning and Preparation**: SQL is used to clean and transform raw data stored in databases. It can handle tasks such as removing duplicates, filtering, and aggregating data, as well as formatting and structuring data in a way that is suitable for analysis.
3. **Data Exploration and Analysis**: SQL allows data scientists to perform descriptive analysis by writing SQL queries to summarize and aggregate data, calculate statistics, identify patterns, and generate reports. SQL can also be used for exploratory data analysis to gain insights into the data before further analysis.

SQL

4. **Data Integration**: SQL is used to join and merge data from multiple tables or databases. It enables data scientists to combine data from different sources and create a unified view for analysis.
5. **Data Manipulation**: SQL allows for updating, inserting, and deleting data in databases. Data scientists can use SQL to modify and manipulate data as needed for their analysis or to perform data wrangling tasks. Overall, SQL is a powerful tool for data scientists to access, manipulate, and analyze data stored in databases, providing a foundation for various data science tasks.

Why SQL?

> But why don't we just load these databases in Pandas and treat them like any other dataframe?

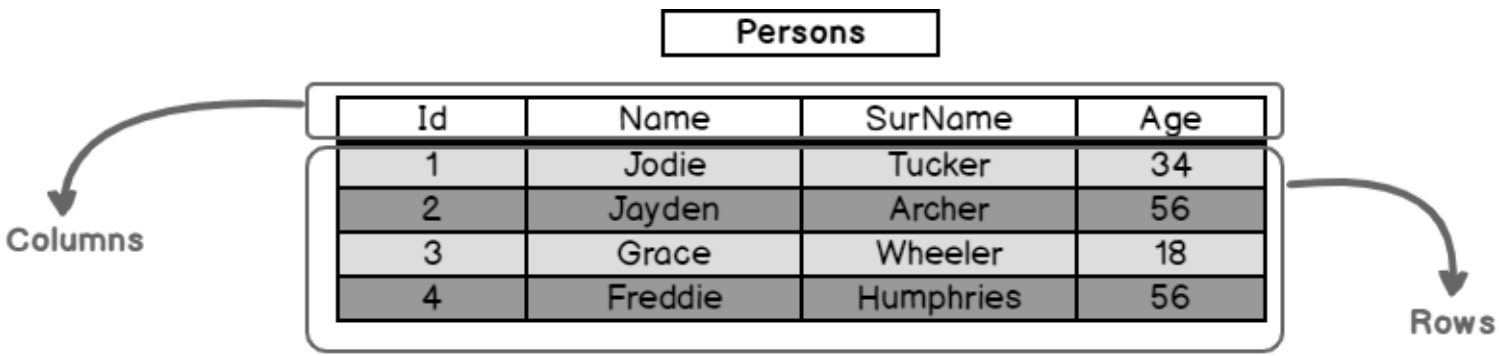
Working with SQL has it's advantages for several reasons:

- Using SQL means speaking the "native language" designed for these databases. Using SQL will help us extract data from databases efficiently, write complex queries, join multiple tables, and aggregate data.
- When dealing with large datasets, SQL queries can significantly outperform equivalent operations in Pandas, as SQL allows us to leverage performance optimization techniques like database indexing and caching.
- Many organizations still have legacy systems and data stored in relational databases. To extract insights from such systems or integrate them with modern data analysis workflows, SQL is necessary. In summary, while Pandas is powerful for working with data in memory, SQL is crucial for accessing databases, optimizing performance, collaborating with data engineers, handling legacy systems, and leveraging existing data assets.

What our data looks like

So now that we know about what we're working with, and the tools we'll be using, let's finally get into *finally* applying our knowledge.

The databases we will be working with consist of tables, which consist of rows and columns much like dataframes in pandas;

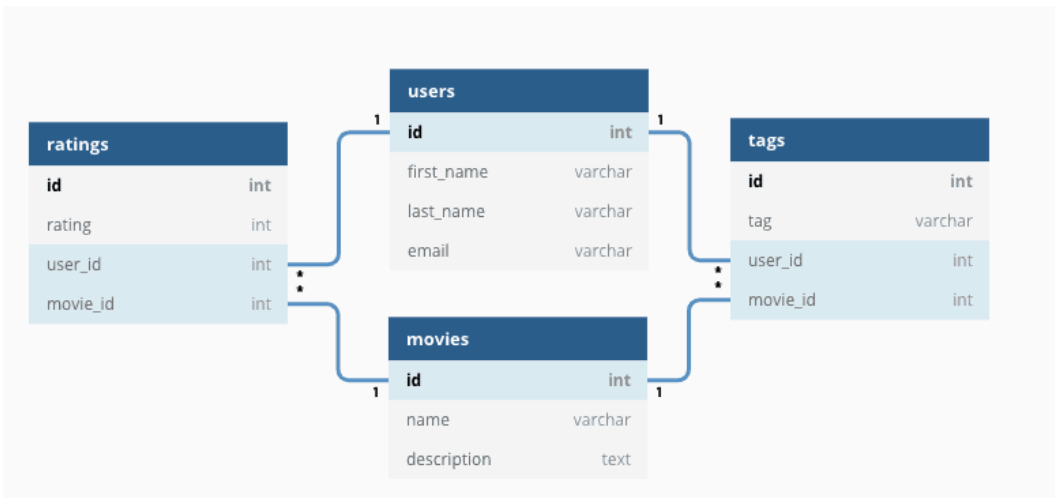


Relational Databases

A database that has one or more of these tables, in which these tables can have relations to one another are called relational databases.

Each table has a number of columns, and each column must have a name and data type, making our data completely structured.

> A DBMS meant for dealing with relational databases, like the ones we'll be working with, are called RDBMSes.



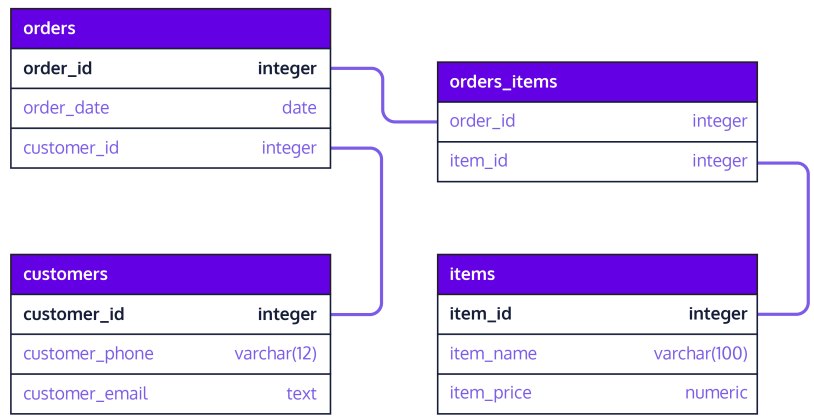
Database Schemas

In the context of relational databases, a schema refers to the logical structure or blueprint that describes the organization of a database.

It defines the tables, their fields (columns), data types, relationships, constraints, and other attributes.

A schema provides a framework for designing and managing the database objects and ensures data integrity and consistency.

- It acts as a roadmap for how the data is organized, how the tables are related to each other, and how information can be retrieved and manipulated within the database.



> TLDR; it's the structure in our structured data.

Onto code!