CSC343 Notes

Introduction to Databases

https://github.com/ICPRplshelp/

Last updated January 9, 2023

1 What is Data?

- Bits that represent values, such as:
  - Numbers
  - Strings
  - Images
- What do we want to do with it?
  - Manage it!!!
- We can manage a large collection of data with files
  - We used flat files, K/V pairs
  - If you combine K/V pairs with your favorite programming language, you get a database.
  - Of course, you'll have to implement all its methods yourself.
- The first commercial databased evolved using plain text. Now, how do we deal with:
  - Duplication
  - Organization (it is incredibly hard, especially when you have more than one person)
  - You'll have to reinvent various wheels:
    * Search
    * Sort
    * Modify, and so on
  - None of these would be optimized

What do we do instead

- We want an efficient, optimized system that specializes in handling **inconvincibly large amounts of data.**

In fact, some of the largest databases would already cram your hard drive in a second/

1.1 Databases and DBMS (Database Management System)

A **DBMS** is a powerful tool that can:

- Manage large amounts of data
- Allow it to persist for long amounts of time (protect it over time from being overwritten)

Every DBMS has a data model. It describes

- Structure (lowest level: rows and columns)
- Constraints (range, types)
- Operations (find all students with grades ≥ 85)

We'll be looking with the **relational data model.** This has been very widespread since it has been invented. It has been proposed as an efficient model.

1.2 The Relational Data Model

CSVs. Spreadsheets. They look a lot like them. The main concept is a relation. The concept of relation is borrowed from math.

A relation is a **SET** of **TUPLES.** Here's an example:

$$\{(x,y) \in \mathbb{R}^2 \ : x < y\}$$

That defines a relation. Every pair of values that obeys $x < y$ will be in it.

- A **column** (attribute, field) goes vertically.

- A **row** (tuple) goes horizontally).
- A table is a set of tuples (rows) (what about the header?)

A **schema** is a namespace where we collect some relations. When we do a lot of work with databases, we are likely to have some completely different areas of interest, and this helps us organize them at the top level.

1.3 What does a DBMS provide

- Can **explicitly** specify the logical structure of the data
  - And **enforce** it
- Can query (ask questions and it will respond) or modify data
  - Best to keep query and modify separate
- Good performance under heavy loads
  - We're talking about billions of data. We're getting into the range of 10^{15} but we're starting to push way more
- Durability of the data
  - Keep it safe and intact (data integrity)
- Concurrent access by multiple users
  - MERGE CONFLICTS!!! Us or us and our partners? We need to have ways of having concurrent access without causing a mess.
  - Suppose tables A and B are bank accounts. We have a couple of queries that remove $100 from A and deposit that amount in B. What happens if another user checks A before the $100 is removed and B after $100 is deposited?
  - If you're developing databases for multiple concurrent users, you'll have to worry about this exact issue, and it's important to pay attention to it.

The architecture of a DBMS relies **heavily** on the underlying operating system. You need access to direct pieces of the operating system. The DBMS sits between users and the data itself. Sometimes, it sits between an app you write and the database.

Rather than allowing someone who is pointing and clicking on the webpage, we have a DBMS that ensures that the queries sent by the person are well-formed and efficient.

Or maybe a Python program is forming proper DBMS queries first. We rarely have an end user directly interacting with the database.