CMPT 354: Database Systems 1 – Unit 01 - Introduction

Dr. Jack Thomas
Simon Fraser University
Summer 2021

Welcome to Database Systems 1!

- Simon Fraser University's introduction to databases for third years.
- Not a core requirement, but always a perennial candidate.
- Also a prerequisite to CMPT 454, Database Systems 2, if you want more!

Our Goals for Today

- Handle the up-front **course administration** business.
  - Work together to **schedule office hours**.
- Introduce what we mean by a **database**.
- Give a **broad overview** of the concepts to be covered
- Time allowing, jump into our **first proper unit** (no spoilers here!)

Your Teaching Team

- I am **Dr. Jack Thomas** (jackt@sfu.ca), Sessional Instructor.

- Your TAs include:
  - **Emma Hughson** (emma_hughson@sfu.ca)
  - **Amirhossein Mozafari Khameneh** (amirhossein_mozafari_khameneh@sfu.ca)
  - **Peshotan Irani** (peshotan_irani@sfu.ca)

Course Website(s)

- **Canvas**
  - The **main course website**, hosting these virtual lectures, quizzes, assignments, midterms, etc.
- **CourSys**
  - Where **assignments** will be **uploaded** and all **final grades** will be tracked and released.
- **Discord**
  - The service we'll use for **office hours**, voice chats, and other forms of chatting.

Assessment

- 10 **Weekly Quizzes**: 20% (2% each)
  - Uploaded weekly on Fridays.
- 5 **Assignments**: 40% (8% each)
  - Two weeks apart, posted to Canvas, uploaded to CourSys.
- 2 **Midterms**: 20% (10% each)
  - Hosted on Canvas, the first in mid-June, the second in mid-July, both during class time.
- **Final Exam**: 20%
  - Also hosted on Canvas, schedule TBD but during the exam period in August.

Weekly Schedule

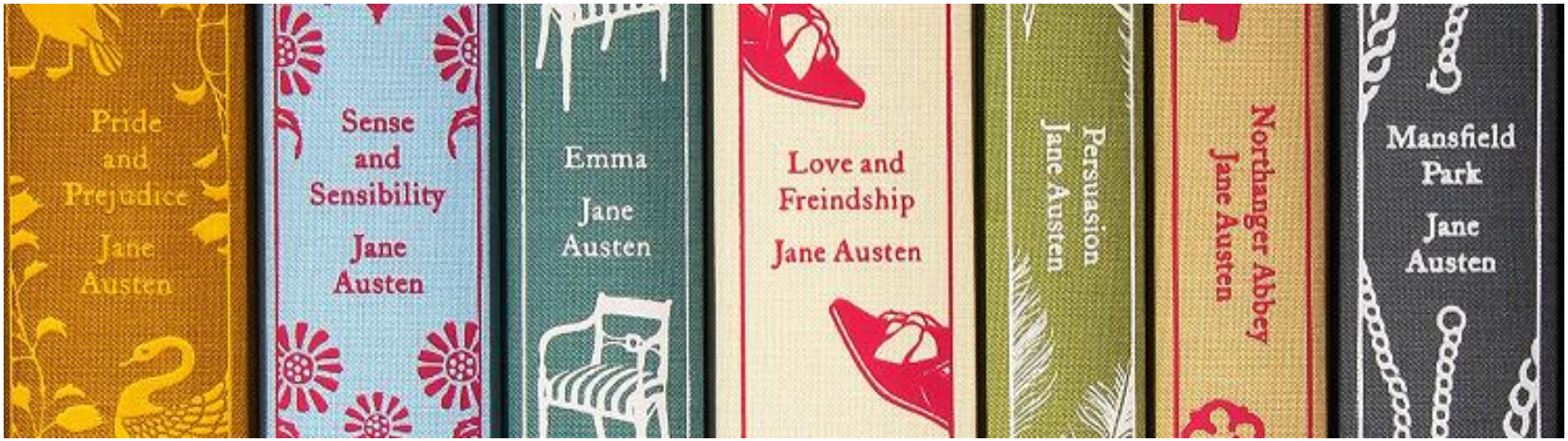
- **Lectures**
  - Monday from 8:30am to 9:30am.
  - Thursdays from 8:30am to 10:30am.
- **Quiz**
  - Goes up on Friday on Canvas.
  - You'll always have 48 hours to complete it to account for timezone and schdule issues.
- **Office Hours**
  - Offered through Discord, whose invite link can be found on the Canvas home page.
  - Let's talk about scheduling those now!

Special Thanks

- I'd like to extend a special thank-you to **Dr. John Edgar** here at SFU for sharing his previous course material during the development of this course.
- Did I include a picture of Dr. Edgar and not myself? Yes.

Data and Databases

- The least popular Jane Austen book.



- Also: the subject of this course

Image credit: https://janeaustenlf.org/pride-and-possibilities-more-articles/2019/01/26/issue-50-the-jane-austen-200th-commemoration-book-club

So What is a Database, Anyway?

- A database is a **collection of information**.
  - Databases of one sort or another have existed since the dawn of civilization.

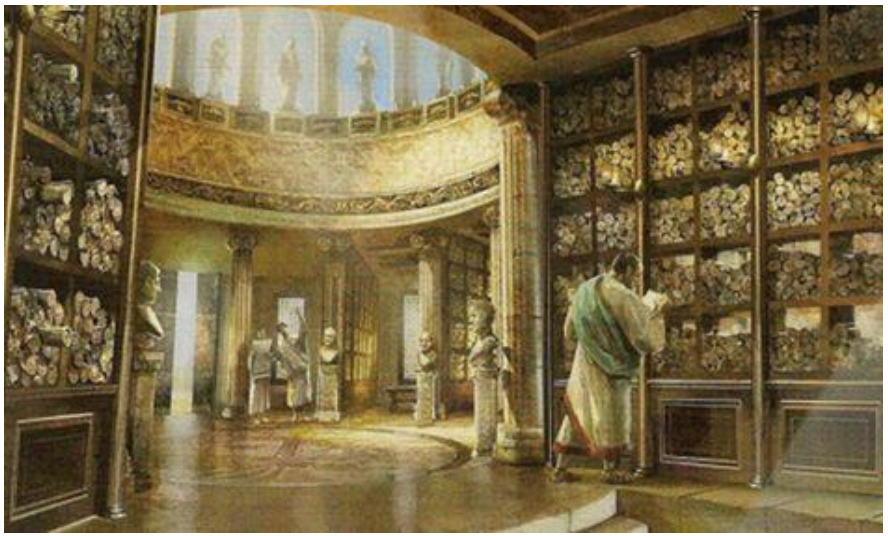


Image credit: https://outschool.com/classes/ancient-archives-book-club-lIzV5ENp
https://thedissolve.com/features/movie-of-the-week/68-brazil-forum-style-gallows-humor-the-past-as-futur/

The Modern Database

- In **Computer Science**, a database is a data collection managed by a *Database Management System*, or **DBMS**.
  - There are many different DBMS's out there.
- These databases are often represented by the *relational model*, though many recent NoSQL DBMS's don't use it.
  - What's NoSQL? Or SQL? We'll get there.

A Brief History of Time
(Just The Database Parts)

- https://www.computerhistory.org/revolution/memory-storage/8/265/2207

Database Applications

- Almost any application that handles a large amount of data will need a database.
- Databases can be found in:
  - The financial industry
  - Government agencies
  - Airlines
  - Universities (hello!)
  - Utility companies
  - Retailers
  - Manufacturing
  - Social Media
  - Games
  - And so much more!

Data, Data, Everywhere

- Early computer databases were primarily **used by large organizations to store textual data**.
  - In **1975** there were some **301** databases containing about 52 million records.
  - By **1998**, there were **11,339** databases holding 12.05 billion records.
- Databases are now used to store all kinds of different information – images, sounds, etc.

Citation: Martha E. Williams (1998), "State of Databases Today: 1999," in Gale Directory of Databases , L. Kumar,

Data in the Current Millennium

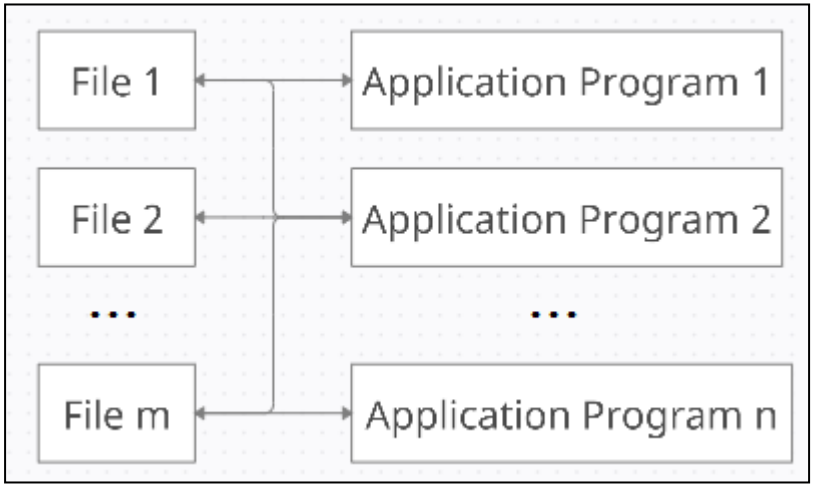
- How much data in the world?
  - **2010**: 1.2 zettabytes
  - **2012**: 2.8 zettabytes
  - **2020**: 40 zettabytes
- Growth of data is very recent
  - In **2017**, IBM estimated that **90%** of data had been created in the last **2 years**.
  - Much of this data is **unstructured** and **unanalyzed**.

What's a Zettabyte?

- A **zettabyte** is:
  - Often misspelled zetabyte
  - 2^{70} bytes, or 1,180,591,620,717,411,303,424 bytes
- That's a **big number**
  - There are estimated to be in the order of 100 billion **stars in the Milky Way Galaxy**.
    - 100,000,000,000 = 0.0000000000847 zettabytes
  - Estimates of the number of **stars in the observable universe** vary wildly, but here's one:
    - 10,000,000,000,000,000,000,000,000 = 847 zettabytes

How Data Storage Works Without a Database Management System

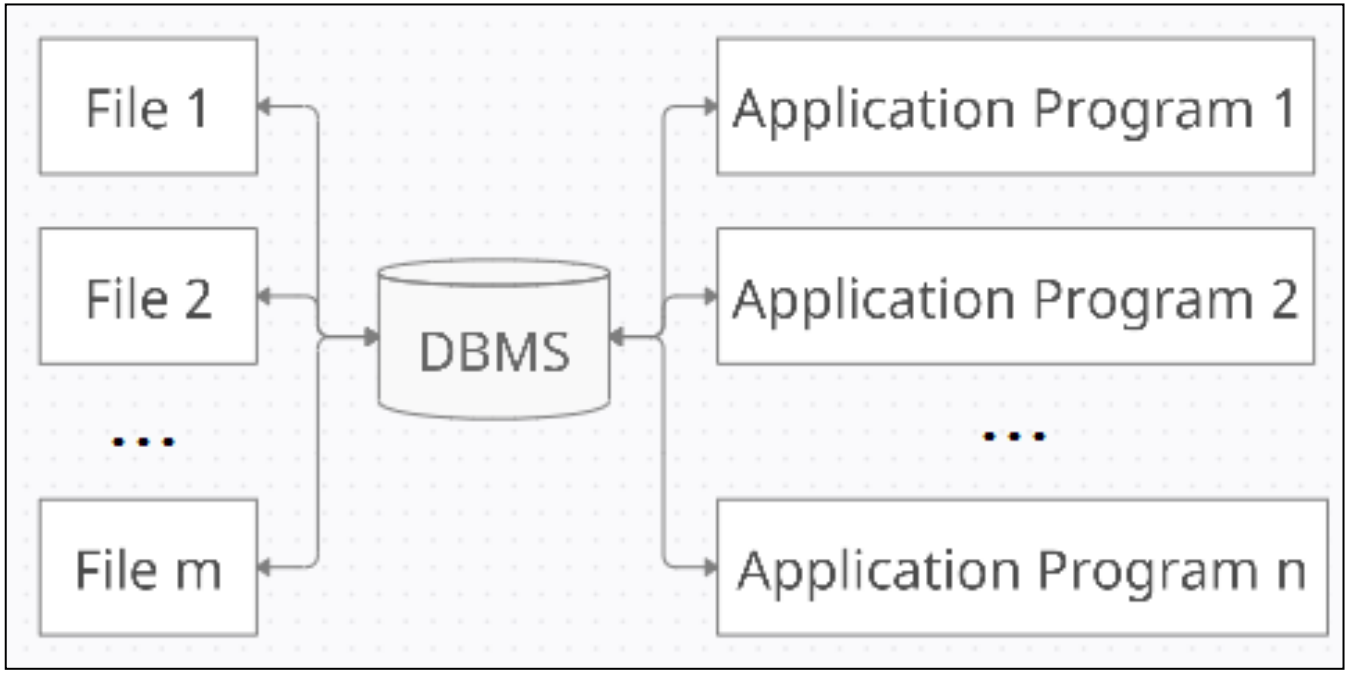
- Data is collected in different **files**.
- These files are used by many **application programs**, often shared between them.



What Happens If…

- An **attribute** is added to one of these files?
- Information that **is in more than one file** is **changed** by a program that interacts with **only one file**?
- We need to **repeatedly access a single record** out of millions of records?
- We need to retrieve data **stored in multiple files**?
- Several programs need to access and modify **the same records at the same time**?
- **The system crashes** while one of the application programs is running?

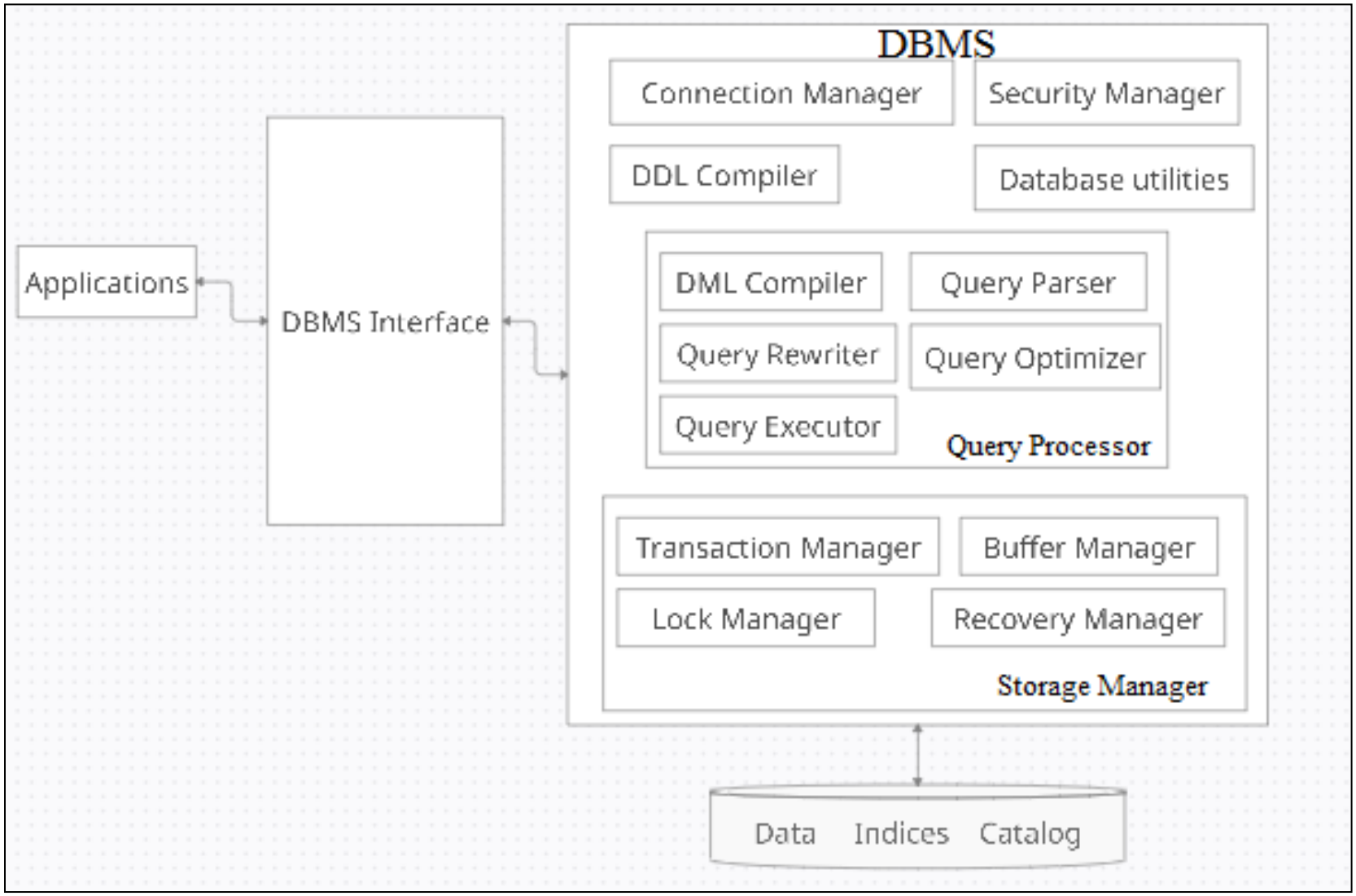
Data Storage With a Database Management System



DBMS Functions

- All **access** to data is **centralized** and **managed** by the **DBMS**.
- **Design and implementation advantages**
  - Logical data independence
  - Physical data independence
  - Reduced application development time
- **Use advantages**.
  - Efficient access.
  - Data integrity and security
  - Concurrent access and concurrency control
  - Crash recovery

DBMS Components



Which DBMS To Use?

- There are different types of **DBMS products**:
  - Relational DBMS (or RDBMS)
  - Non-SQL (What's SQL? We're getting there…)
- Cost also varies:
  - Some are **free**, like MySQL or Microsoft SQL.
  - Some are **quite expensive**, like Oracle.
- It is important to select the product that is **right for the organization or application** at hand.

Data Models

- A database **models** a real-world enterprise.
- A *data model* is a **formal language** for describing data.
  - A *schema* is a description of a **particular collection of data** using a particular data model.
- One of the most **widely used data models** is the *relational data model*.
  - The main concept of this model is a *relation*, or set, which can be represented by a **table with rows and columns**.
  - Web databases and Big Data databases often do not use the relational model.

Relational Model

- This course covers the relational data model **used by most traditional commercial DBMS's**.
- The model can be used during the design process to **describe the enterprise** that requires a DB.
  - An example of **abstraction**, since it doesn't require the implementation details yet.
  - Data can be described at different levels, allowing the levels of a system to be **relatively independent from each other**.

Levels of Abstraction

- Data can be described at **three levels of abstraction**:
  1. **Physical Schema**
     - The lowest level schema, which describes how data is stored and indexed.
  2. **Conceptual (or Logical) Schema**
     - What (not how) data is stored, described in terms of the data model.
  3. **External (or View) Schema**
     - The highest level schema, describing how some users interact with the data. There can be multiple views.

Data Independence

- **Physical data independence**
  - Allows the physical schema to be modified without rewriting application programs.
  - Usually to improve performance, like adding or removing an index or moving a file to a new disk.
- **Logical data independence**
  - Shields users from changes in the logical schema – i.e. their views remain unchanged.
  - Allows the logical schema to be modified without rewriting application programs, like adding an attribute to a relation.

Database Languages

- A **database language** allows a database to be **created**, **modified**, or **queried**.
  - We will use *Structured Query Language* (**SQL**)
- SQL has **four components**:
  - *Data Definition Language* (**DDL**), used to create and modify database schemas.
  - *Data Manipulation Language* (**DML**), used to modify and query records.
  - *Transaction Control Language* (**TCL**) and *Data Control Language* (**DCL**), which we won't be covering.

Data Definition Language

- The DDL allows entire databases to be created, and allows **integrity constraints** to be specified:
  - Domain constraints
  - Referential integrity
  - Assertions
  - Authorization
- The DDL is also used to **modify** existing DB schema:
  - Addition of new tables
  - Deletion of tables
  - Addition of attributes

Data Manipulation Language

- The DML allows users to **access** or **change** data in a database.
  - Retrieve information stored in the database.
  - Insert new information into database.
  - Delete information from the database.
  - Modify information stored in the database.
- There are two basic types of DML:
  - **Procedural** – users specify what data is required and how it should be retrieved.
  - **Declarative** (nonprocedural) – users specify what data is required without specifying how it should be retrieved.

CMPT 354 and 454

- CMPT 354 covers database **specification** and **implementation**.
  - Database design – the relational model and the ER model.
  - Creating and accessing a database
    - Relational algebra
    - Creating and querying a DB using SQL
  - Database application development
- CMPT 454 – **DBMS Issues**
  - Disk and buffer management and storage
  - Query evaluation
  - Transactions and recovery
  - Advanced topics

CMPT 354 Topics

- Designing a database using the **Entity Relationship model**, and **Entity Relationship diagrams**.
- The **relational model**, converting an **ERD** into an **SQL database**.
- **Relational algebra**, the basis of SQL.
- **SQL**
- **Specifying constraints** on a database
- **Database applications**
- **Normalization**
- And more!

Recap – The Basics of Data

- In **Computer Science**, databases are collections of data organized with a **Database Management System**.
- Databases are based on a **data model**, which for us will usually mean the **relational model**.
- This allows us to describe data at **three levels of abstraction** (**physical**, **conceptual**, and **external** schema).
- **Database languages** like **SQL** are used to create, modify, and query these databases.