CS 275 – Introduction to Databases

Week 1 - Overview of Databases

An Overview of Databases

- A database is a structured collection of related meaningful data
- Structured
  - There is some relationship and organization within the pieces of data
  - It is not simply a collection of related data
- Related
  - A database does not just collect data about everything
  - Data models some real world thing (your book calls it a *miniworld* or *universe of discourse.*
- Meaningful data
  - The data represents some aspect of the miniworld

Examples of Databases

- A grocery store
  - Data
    - Possible inventory
    - Current stock
    - Employees
    - Departments
  - Structure
    - Current stock must come from possible inventory
    - Employees maintain department
    - …
  - Miniworld
    - The stuff in a grocery store

Another example

- Bank
  - Data
    - Customers
    - Accounts
    - Employees
    - Rates
  - Structure
    - Customers have accounts
    - Accounts have interest rates
    - Employees contact customers
  - Miniworld
    - The bank and related outside information

What may not be a database

- A pile of hard drives
  - They have data, but may not have meaning
  - There is no structure to the data that we know
  - The data is not confined to describing a specific miniworld

What do we do with a database?

- There are many kinds of databases
  - Static geographic data in a hiking GPS
  - Dynamic data in an Excel spreadsheet that I use for grades
  - Dynamic data in a website like Amazon.com
- These are three examples, this class focuses on the latter example

Introducing the DBMS

- A DataBase Management System (DBMS) is a tool for accessing a database
- Provides
  - Redundancy control
  - Access control
  - Persistent storage
  - Querying system
  - Backup
  - Concurrent access
  - Enforcing constraints

The downsides to the DBMS

- DBMS gives us a lot but there are costs
  - Overhead
  - Complexity
- When not to use a DBMS?
  - Embedded system with little memory
  - Static data
  - Single user

Who works with the database?

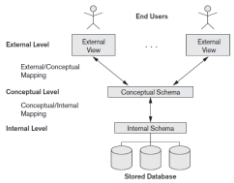
- Admins – Control user access, manage backups, work with physical hardware
- Designers – Create schemas
- Application developers – Write queries that work directly with the data
- End users – Generally work though an application to view and modify data

Who to think about

- Focus on the application developer
  - Design your data so it makes sense to them
  - Consider the types of queries they will want to run
- Then focus on the end user
  - Make sure that all the data they want actually exists
  - Make sure the database is the best tool to meet their needs

Layers of Abstraction

- People like to talk about layers
- External layer
  - For end user
  - Data must make sense
- Conceptual level
  - How the data is organized for the database designer
  - Diagrams and schemas are used here
- Internal level
  - Mappings to the actual data on the physical storage media



Data Independence

- Logical data independence
  - The idea that changes at the conceptual level should not break the external level
  - Adding data types should not break external level
  - Removing or changing usually will break the external level
- Physical data independence
  - Falling back to a backup hard drive should not break the conceptual level
  - Usually handled well by the DBMS