Today’s Agenda

• Course Overview & Logistics
• Introduction of Databases

Course Objective

• Students will learn the fundamentals of database management systems.
  – When to use them
  – How to model data with them
  – How to store and retrieve information
  – How to search quickly for information
  – System internals & key algorithms

15-615 Wait List

• We are capped at 90 students.
• We do not follow CMU S3’s position list.
• You are not allowed to swap with somebody already enrolled in the course.
Course Logistics

- **Course Policies + Schedule:**
  - Refer to course web page.
- **Academic Honesty:**
  - Refer to CMU policy page.
  - If you’re not sure, ask the professors.
  - Don’t be stupid.

Homework Assignments

- All assignments are due at the beginning of the lecture (3:00pm), on the due date.
- All assignments are to be done individually.
- Late policy: Four “Slip” Days
- **HW3 & HW7** require more programming than the other assignments.

Exams

- **Midterm:** Wednesday October 19th
  - In-class (this room)
  - Materials up to October 12th & HW4 (inclusive)
- **Final:** TBA
  - Comprehensive

Course Rubric

- **Homework Assignments**
- **Midterm Exam**
- **Final Exam**
**Grading**

- **Assignments**: 30%
  - See course web page for HW weights.
- **Midterm**: 30%
- **Final**: 40%
- **No extra credit is offered.**

**Special Accommodations**

- Please contact the professors if you need special accommodations for the homework or exams **before** the due dates.
- Refer to [CMU Accommodations Page](#)

**Office Hours**

- **Christos (GHC 8019)**
  - Tuesdays @ 2pm-3pm
- **Andy (GHC 9019)**
  - Mondays @ 12pm-1pm
- Also available by appointment.
- See course website for TA hours.

**Course Message Board**

- On-line discussion through Blackboard:
  - [http://cmudb.io/15415-f16-blackboard](http://cmudb.io/15415-f16-blackboard)
- If you have a technical question about homework, please use Blackboard.
  - Don’t email profs or TAs directly.
- All non-project questions should be sent to the professors.
Lecture Questions

• Ask questions during the lecture.
• If you are unsure about something, then somebody else might have the same question.
• Don’t run up to talk to the professors immediately after the lecture.

Course Topics

• Introduction to Databases
• Data Models
• Query Language (SQL)
• Database Design
• Query Optimization & Indexing
• Transaction Management
• Advanced Topics

Spring 2017

• 15-721 – Database Systems
  – High-performance in-memory system internals
  – Programming intensive
• 15-826 – Multimedia DBs & Data Mining
  – Graph mining, time-series analysis, databases for machine learning.
  – Non-relational data

Database Talks (Optional)

• The CMU DB group hosts research/industry talks throughout the semester.
• More information:
  – http://db.cs.cmu.edu/events/
Research Positions (Optional)

• We are looking for students to help build CMU’s new flagship DBMS (Peloton)
  – Database Internals (C++11)
  – Autonomous Operation (TensorFlow)
• Come to the project info meeting:
  – Friday Sept 9th @ 12:30 (GHC 9115)
  – http://cmudb.io/fall2016-positions

What is a Database?

Database Example

• Or why should you take this course?
• Let’s build a simple application…

Database Example

• Create a database to keep track of the music that is available in our application.
Flat File Strawman

- Store the data in comma-separated value (CSV) files.
  - Use a separate file per entity.
  - The application has to parse the files each time they want to read/update records.

```python
for line in file:
    record = parse(line)
    if searchKey in record:
        // Do something!
```

Database Example

- Create a database to keep track of the music that is available in our application.

```
ARTIST  ALBUM  TRACK
name    name   name
description artist
year     album
country  year
number
```

Flat Files: Data Integrity

- How do we ensure that the artist is the same for each album entry?
- What if somebody overwrites the album year with an invalid string?
- What if there are multiple artists on an album?

Flat Files: Implementation

- How do you find a particular record?
- What if we now want to create a new application that uses the same database?
- What if two threads try to write to the same file at the same time?
Flat Files: Security

- What if only want some people to see some of the records data in a file?
- What about all of the records but only some of their attributes?

Flat Files: Durability

- What if the machine crashes while we’re updating a record?
- What if we want to replicate the database on multiple machines for high availability?

Database Management System

- A **DBMS** is software that allows applications to store and analyze information in a database.
- A general-purpose DBMS is designed to allow the definition, creation, querying, update, and administration of databases.
DBMS Types: Data Models

- Relational
- Key/Value
- Graph
- Document
- Column-family
- Array/Matrix
- Obsolete: Hierarchical, Network

DBMS Types: Target Workload

- On-line Transaction Processing (OLTP)
  - Fast operations that only read/update a small amount of data each time.
- On-line Analytical Processing (OLAP)
  - More complex read-only queries that read a lot of data all at once to compute aggregate data.

Relational Database Example

- Declare the attributes of each table.
  - Name / Value Types / Constraints

```
CREATE TABLE artist (name VARCHAR(32) NOT NULL, 
genre VARCHAR(32), 
country CHAR(3), 
PRIMARY KEY (name));
CREATE TABLE album (name VARCHAR(64) NOT NULL, 
artist VARCHAR(32) NOT NULL, 
year INT CHECK(year > 0), 
PRIMARY KEY (name));
CREATE TABLE track (name VARCHAR(64) NOT NULL, 
album VARCHAR(64) NOT NULL, 
tracknum SMALLINT CHECK(tracknum > 0), 
PRIMARY KEY (name, tracknum));
```

DBMS: Fundamental Concepts

- Three-level Architecture
- Logical Data Independence
- Physical Data Independence
Three-level Architecture

**View Level**
What information is exposed to users, what are they allowed to see...

**Logical Level**
What tables are there, what attributes do they have, what constraints should the DBMS enforce...

**Physical Level**
How data is stored, where it is located, how many bytes, what type of indexes...

Logical Data Independence

- We can modify our table definitions without having change our application’s views.
- Example:
  - Add/drop/rename attributes for a table.
  - Rename a table.

Physical Data Independence

- We can change how/where database objects are represented in the physical storage.
- Examples:
  - Use 32-bits instead of 64-bits for integers.
  - Convert an index from a B+Tree to a SkipList.
  - Compress a table when it is stored on disk.
  - Move a table to another disk/machine.

Course Topics

- Introduction to Databases
  - Data Models
  - Query Language (SQL)
  - Database Design
- Query Optimization & Indexing
- Transaction Management
- Advanced Topics
Next Class

• Application Modeling
  – Entities
  – Relationships
  – Attributes