Today ...

- Survey
- Course Overview

Homework

- HW1 (out)
Course Overview

Course webpage

- www.cs.gonzaga.edu/bowers/courses/cpsc321
- Please check frequently (schedule, notes, assignments, etc.)

Course discussion forum

- Piazza

Please read the syllabus

- **Office hours**: Tues & Thurs 9:30-10:30, Wed 1:00-3:00 or by appointment
- **Grading**: 30% assignments, 10% project, 15% quizzes, 45% exams
- **Homework late policy**: class period due for credit (no exceptions)
- **Missing class**: more than 4 absences may result in a V grade

Textbook:

Academic Integrity

Basic rules

• Quizzes and exams must be *your own work*
• You are responsible for understanding course content
• You will be tested on material from assignments (reading & hw)
• Okay to work with others on assignments (but see above)
• Must turn in homework individually

If you do work together on assignments

• List who you worked with in your assignment summary
• Use “pairing” and not “divide and conquer”
• The goal is learning, not just finishing an assignment
Quizzes

We’ll have short weekly quizzes

• Cannot “make up” a quiz
• Given at the beginning of class
• Each quiz should take 5-10 minutes

Why quizzes?

• Give you a feel for the types of questions I ask on exams
• Give you feedback on your understanding of material
• Give me feedback on your understanding

Be sure to study for quizzes (notes, readings, homework, etc)
Assignments

Reading Questions

- Most weeks there will be a reading component
- Read the material, answer a few short questions

Hands-On Work

- Majority of assignments will be writing database queries
- Some database design assignments
- A larger group project (implement end-to-end database application)

For Most Assignments

- Cover sheet
- Assignment summary (around 1-page written reflection)
- Testing strategies and results
Goals and Course Topics

Goals for the Course

1. Know when and how to use database systems
2. Read and write SQL
3. Design databases, including basic normal forms and indexes
4. Use database systems within applications (basics)
5. Gain hands-on experience with a DBMS (MySQL) and dynamic SQL

We’ll Cover Various Topics

- Relational database systems and SQL
- Logical database design (ER, normalization)
- Touch on database internals (storage)
- Physical database design (indexes)
- Transactions and recovery (as time allows)

In class exercise ...
Why study database management?

Data is important!

- E.g., to business, science, government, etc.
  - Critical to existence of many companies
- Important skills for computer scientists / developers today
  - Many tech companies built on managing data
  - E.g., as a business asset: Google, Amazon, Facebook, Twitter, ...
  - Or as a business product: Oracle, Microsoft, IBM, SAP, ...

Database systems span many areas in CS

- File, memory, process management (Operating Systems)
- Languages, algorithms, complexity (Theory)
- Information modeling, formal logic (Artificial Intelligence)
- Application development (Software Engineering)
- Optimization, concurrency, distribution (Systems Programming)
Database management systems

We’ll focus on **Relational Database Management Systems**

- Abbreviated RDBMS or just DBMS
- Data stored in structured **relations** (i.e., tables)
- **Structured Query Language** – SQL – to manage and retrieve (query) data
  - Pronounced “es que el”
  - Sometimes as “sequel” (SQL’s “predecessor” …)

Some characteristics of a typical (R)DBMS:

- **Data is persistent**
  - DBMS can be stopped and restarted without losing data
  - Data stored across applications
  - Requires reliability

- **Data accessed by many users**
  - Data accessed concurrently by many users
  - E.g., flight reservations, product catalogs, facebook pages
  - Often requires safe “transactions”

- **Data impractical, inefficient, complicated to manage in-memory**
  - E.g., by programming to data stored in files
  - Lose benefits of a DBMS … e.g., central repository, query support/optimization, APIs, recovery, security, etc.

- **Data sets may be large**
  - practical today to store small to large data sets in a DBMS

- DBMSs used by a huge number of companies