Lecture 1:

- Course Overview
- Basic DB Concepts and Terms

Logistics

1. Course webpage: www.cs.gonzaga.edu/bowers/courses/cpsc321
2. Piazza: for Q&A, announcements (see invite)
3. GitHub: for homework
4. Blackboard: for tracking points
5. Office Hours: Tu/Th 12:30–1:30, Wed 1–3

See webpage for syllabus, weekly schedule, homework assignments, etc.
Course Overview

Course Topics:
- Relational model
- Basic and intermediate SQL (in MySQL)
- Logical database design (ER, normalization)
- Physical database design (indexes)
- Transactions and recovery

Focus is on app development (versus database system implementation)

Will be a mix of basic theory and hands-on practice

Grading
- HW assignments (8) & final project 380 points
- Problem sets (5) 50 points
- Quizzes (5) 50 points
- Exams (3) 160 points
- Attendance / participation (30) 60 points

Must score at least 60% on homework (assignments, final project, problem sets) and 60% on tests (quizzes and exams)

Please read the syllabus for additional information
Course Overview (cont)

Expectations:

• Engage and participate in class (including doing your own work!)
• Start assignments early, give yourself enough time to succeed
• Assume you have everything you need (ask when in doubt)

(Some) collaboration is encouraged ... e.g., discussing assignments
• but avoid plagiarism & other related issues — e.g., no code sharing

Hints:
• carefully read and follow instructions
• do not use youtube, google, stackoverflow, etc., to "learn" material
• study for quizzes, exams, etc
• come to office hours and frequently check piazza

Basic Concepts and Terms

1. Persistence
• store data beyond (running of) application that created it

2. Database
• an organized (structured) data store for information
  directory of photos (data store)
  when/where photos taken, who is in each photo, etc. (structured)
• organization makes it easier to obtain info and answer questions
  how many pictures taken in Spokane from 2020–2022?

3. Database Management System (DBMS)
• software system for managing databases
4. **DBMS supported user operations**

- design an organization for data (called a “schema”)
- update data by adding, removing, modifying
- retrieve data (via a “query language”)
- manage who has permission to access data

5. **Relational DBMSs and SQL**

- store data using the “relational model” ... tables of rows and columns
- SQL (structured query language) typically used for relational DBMSs

6. **SQL Command (Sub) Languages**

- Data Definition Language (DDL) ... CREATE, ALTER, DROP
- Data Manipulation Language (DML) ... INSERT, UPDATE, DELETE
- Data Query Language (DQL) ... SELECT, WITH
- Data Control Language (DCL) ... GRANT, REVOKE (accounts, privileges)

*Note:* other data models exist as well

- e.g., document, tree, graph, key-value, time-series models
- with their own data languages (or even with SQL)
7. **Transaction Processing (aka “OLTP”)**
   - supports “daily operations” of a business (in real time)
   - many insert, update, delete, and simple queries
   - fast response times and many concurrent sessions

8. **Analytics (aka “OLAP”)**
   - supports “decision making” of a business
   - more complicated queries over large portion of database
   - read heavy (with fewer inserts, updates, deletes)
   - often connected with data warehouses ...

9. **Transactional concurrency**
   - a single transaction can involve multiple data reads and writes
   - most DBMSs (seamlessly) handle multiple transactions at once
   - leads to system design and implementation challenges ...

10. **ACID (transaction concurrency) properties**
    - **Atomicity**: all or nothing execution of transaction
    - **Consistency**: transactions expected to preserve constraints of DB
    - **Isolation**: transactions execute as if no others execute at same time
    - **Durability**: transaction effect on DB must not be lost after it completes

    *Note: Relational DBMSs typically ACID compliant (vs “NoSQL” systems)*