CSE 344

MARCH 26TH - INTRODUCTION
WELCOME!

- CSE 344
- Today’s lecture
  - Course administration
  - What to expect
  - Introduction and motivation
COURSE FORMAT

Lectures
• Location: MLR 301

Sections:
• Content: exercises, tutorials, questions, new materials (occasionally)
• Locations: see web
• Please attend
• Bring your laptop

8 homework assignments
7 web quizzes

Midterm and final
GRADING

Homeworks 30%
Web quizzes 10%
Midterm 25%
Final 35%

This is all subject to change
ADMINISTRATION


- Syllabus (course information)
- Lecture/section notes will be available there
- Homework assignments will be available there
- Link to web quizzes is there

Piazza

- Questions and clarification; place to give and get help
- NOT office hours, code can be difficult to debug remotely

Gitlab

- Account created this week, for submitting HW assignments

NewGradiance

- Autograded online quizzes, good for practice, unlimited attempts
TEXTBOOK


Good reference and alternative explanation

Also, good source for practice problems
EIGHT HOMEWORK ASSIGNMENTS

H1: Sqlite intro (Out today)
H2: Sqlite basics
H3: Advanced SQL on Azure
H4: Datalog and Relational Algebra
H5: NoSQL: Json/SQL++
H6: Spark on AWS
H7: Schema Design
H8: Transactional Application

Submit via git
ABOUT THE ASSIGNMENTS

You will learn/practice the course material:
  • SQL, RA, parallel db, transactions, ...

You will also learn lots of new technology
  • Cloud computing: Azure, Cloud9, AWS
  • NoSQL: AsterixDB, Souflle
  • Git

The time spent learning the new technology is very useful: write everything on your CV!
DEADLINES AND LATE DAYS

Assignments are expected to be done on time, but things happen, so…

You have up to 3 late days

- No more than 2 on any one assignment
- Used in 24-hour chunks

Late days = safety net, not convenience!

- You should not plan on using them
- If you use all 3 you are doing it wrong
SEVEN WEB QUIZZES

- http://newgradiance.com/
- Create account; please make sure you use your UW first/last name
- Token to be provided to course email

Short tests, take many times, best score counts

No late days – closes at 11:00 deadline

Provides explanations for wrong answers
LECTURES

- Slides contain vital information for exams
  - May emphasize tricks or problem types off slides
- Posted after lecture
- Associated readings
  - Good for alternate explanations
  - (also I get a lot of inspiration for exam questions)
EXAMS

• Dates
  • Midterm (TBA – Late April/Early May)
  • Final, Wednesday, June 6th, 8:30 – 10:20

• Preparation
  • Exam review
ABOUT ME

• Evan McCarty (ejmcc@cs.washington.edu)
• Theory and Algorithms research
• Data Scientist for Partners for Our Children
• Lecture notes posted after class
• Part-time Faculty
  • On campus MWF
  • Available by email
ABOUT STAFF

• TAs
  • Sravan Konda
  • Ariel Lin
  • Xi Liu
  • Michelle Prawiro
  • Jason Tan

• First resource for coding / setup problems
• Office hours posted on Wednesday (start next week)
EXPECTATIONS ABOUT YOU

• CSE majors
• Half-asleep
• (Hopefully) registered
  • If not, talk with me after
• Academic Honesty and Participation
• Piazza and help
The world is drowning in data!

Need computer scientists to help manage this data

• Help domain scientists achieve new discoveries
• Help companies provide better services (e.g., Facebook)
• Help governments (and universities!) become more efficient

Welcome to 344: Introduction to Data Management

• Existing tools PLUS data management principles
• This is not just a class on SQL!
WHY DATABASE MANAGEMENT?

• This course was my least favorite topic in undergrad
WHY DATABASE MANAGEMENT?

• This course was my least favorite topic in undergrad
• Now, I work with databases
WHY DATABASE MANAGEMENT?

• This course was my least favorite topic in undergrad

• Now, I work with databases
  • Intelligent design and organization of data allows important work and research to occur efficiently and correctly
WHY DATABASE MANAGEMENT?

• This course was my least favorite topic in undergrad

• Now, I work with databases
  • Intelligent design and organization of data allows important work and research to occur **efficiently** and **correctly**

• Organizations need a diverse set of skills, you may not ever need to manage a DB, but you will certainly be interfacing with one
WHY DATABASE MANAGEMENT?

• This course was my least favorite topic in undergrad
• Now, I work with databases
  • Intelligent design and organization of data allows important work and research to occur efficiently and correctly
• Organizations need a diverse set of skills, you may not ever need to manage a DB, but you will certainly be interfacing with one
• Decisions made in setting up a DB (or even a query) can affect performance going forward
WHY DATABASE MANAGEMENT?

• Disk and magnetic tape are linear storage
  • We can access elements throughout them, but there is a continuous serialization of this data.
  • Data itself is rarely one dimensional
  • Imagine storing all data about UW students on disk
WHY DATABASE MANAGEMENT?

- Disk and magnetic tape are linear storage
  - We can access elements throughout them, but there is a continuous serialization of this data.
  - Data itself is rarely one dimensional
    - Imagine storing all data about UW students on disk
  - What is their order? Are students related?
WHY DATABASE MANAGEMENT?

• Disk and magnetic tape are linear storage
  • We can access elements throughout them, but there is a continuous serialization of this data.
  • Data itself is rarely one dimensional
  • Imagine storing all data about UW students on disk
• What is their order? Are students related?
  • Related relative to other data?
  • Why store “students” at all?
What is a database?
What is a database?
A collection of files storing related data

Give examples of databases
DATABASE

What is a database?
A collection of files storing related data

Give examples of databases
Accounts database; payroll database; UW’s students database; Amazon’s products database; airline reservation database
What is a DBMS?
What is a DBMS?

A big program written by someone else that allows us to manage efficiently a large database and allows it to persist over long periods of time.

Examples of DBMSs

- Oracle, IBM DB2, Microsoft SQL Server, Vertica, Teradata
- Open source: MySQL (Sun/Oracle), PostgreSQL, CouchDB
- Open source library: SQLite

We will focus on relational DBMSs most quarter.
AN EXAMPLE: ONLINE BOOKSELLER

What data do we need?
AN EXAMPLE: ONLINE BOOKSELLER

What data do we need?

• Data about books, customers, pending orders, order histories, trends, preferences, etc.
• Data about sessions (clicks, pages, searches)
• Note: data must be persistent! Outlive application
• Also note that data is large… won’t fit all in memory
What data do we need?

- Data about books, customers, pending orders, order histories, trends, preferences, etc.
- Data about sessions (clicks, pages, searches)
- Note: data must be persistent! Outlive application
- Also note that data is large… won’t fit all in memory

What capabilities on the data do we need?
AN EXAMPLE: ONLINE BOOKSELLER

What data do we need?

- Data about books, customers, pending orders, order histories, trends, preferences, etc.
- Data about sessions (clicks, pages, searches)
- Note: data must be persistent! Outlive application
- Also note that data is large… won’t fit all in memory

What capabilities on the data do we need?

- Insert/remove books, find books by author/title/etc., analyze past order history, recommend books, …
- Data must be accessed efficiently, by many users
- Data must be safe from failures and malicious users
AN EXAMPLE: ONLINE BOOKSELLER

• What can go wrong?
AN EXAMPLE: ONLINE BOOKSELLER

- What can go wrong?
  - *It depends on how well you store the data*
  - Suppose we store everything we need in a big text file (or a .csv if we get fancy)
AN EXAMPLE: ONLINE BOOKSELLER

- What can go wrong?
  - *It depends on how well you store the data*
  - Suppose we store everything we need in a big text file (or a .csv if we get fancy)
    - Related data?
    - Concurrent access?
    - Consistency?
    - Runtime?
    - Planning?
WHAT A DBMS DOES

Describe real-world entities in terms of stored data
Persistently store large datasets
Efficiently query & update
  • Must handle complex questions about data
  • Must handle sophisticated updates
  • Performance matters
Change structure (e.g., add attributes)
Concurrency control: enable simultaneous updates
Crash recovery
Security and integrity
THE PLAYERS

DB application developer: writes programs that query and modify data (344)

DB designer: establishes schema (344)

DB administrator: loads data, tunes system, keeps whole thing running (344, 444)

Data analyst: data mining, data integration (344, 446)

DBMS implementor: builds the DBMS (444)
WHAT IS THIS CLASS ABOUT?

Unit 1: Intro (today)
Unit 2: Relational Data Models and Query Languages
Unit 3: Non-relational data
Unit 4: RDMBS internals and query optimization
Unit 5: Parallel query processing
Unit 6: DBMS usability, conceptual design
Unit 7: Transactions
Unit 8: Advanced topics (time permitting)
WHAT TO EXPECT SOON

• Course Website
• Syllabus
• Git tutorial / help
• The first HW assignment
• Piazza page
• Canvas page
• Link for online quizzes