CS 564: DATABASE MANAGEMENT SYSTEMS

Spring 2025

COURSE LOGISTICS

INSTRUCTOR

Paris Koutris

- paris@cs.wisc.edu

- Office hours
    - **WHERE***: office CS 4363
    - **WHEN***: Mondays 1:00-2:30 pm
    - *Zoom is possible if necessary*

ABOUT ME

- undergrad in Athens, Greece

- Ph.D. in University of Washington (the other UW)

- at UW-Madison since Fall 2015!

Research Interests

- massively parallel processing

- data pricing

- managing uncertain data

- data structures for query processing

MEET YOUR TAS

Dev Ahluwalia

- office hours: **Thursday 2:30 pm - 4:30 pm @ CS 1334**
- email: dahluwalia@wisc.edu

Simon Frisk

- office hours: **Wednesday 10:00 am - 12:00 pm @ CS 1334**
- email: simon.frisk@wisc.edu

Skylar Hou

- office hours: **Friday 10:00 am - 12:00 pm @ CS 1334**
- email: whou25@wisc.edu

Zhiwei Song

- office hours: **Tuesday 2:30 pm - 4:30 pm @ CS 1334**
- email: zsong96@wisc.edu

COURSE FORMAT

- Lectures **M+W+F** 2:30-3:45 pm

- Homework (7 assignments)
- Midterm Exam
- Final Exam

CANVAS HAS EVERYTHING!

We will use Canvas for everything

- Lecture slides

- Assignments

- Extra material

- Lecture recordings

COMMUNICATION

Piazza: access through Canvas

- all course announcements

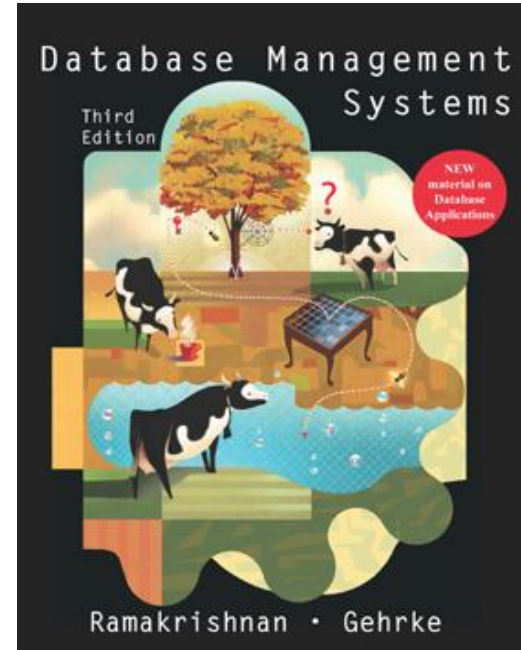- questions (answer each other's questions!!)

- discussions

TODO #1: Enroll to Piazza!

LECTURES

- Lectures will be in person @ Psychology 105
- All lectures will be recorded and uploaded to Kaltura

TEXTBOOK

- Database Management Systems (3d edition)

- **Come to the lectures!**
  - take notes
  - ask questions
  - participate

PREREQUISITES

- Data structures and algorithm background

- For the programming projects
  - C++ will be used for the database internals
  - Python is also required

C++

Brush up your C++ skills!

- [CS 368: C++ for Java Programmers](#)

- [C++ tutorial](#)

- [Another](#) C++ tutorial

GRADE DISTRIBUTION

- 7 Assignments: 50%

- Midterm: 20%

- Final: 30%

ASSIGNMENTS

SQL
Due Feb 9 at 11:59pm | 100 pts

ER Modeling & Schema Design
Due Feb 23 at 11:59pm | 100 pts

Normalization & Relational Algebra
Due Mar 2 at 11:59pm | 50 pts

Buffer Manager
Due Feb 16 at 11:59pm | 100 pts

Heapfile Manager
Due Apr 6 at 11:59pm | 100 pts

Query Operators
Due Apr 20 at 11:59pm | 100 pts

Query Optimization
Due May 2 at 11:59pm | 50 pts

INDIVIDUAL ASSIGNMENTS

Python + Jupyter notebooks

- **Assignment #1**
  - *SQL queries*
- **Assignment #3**
  - *Normalization & Relational Algebra*
- **Assignment #7**
  - *Query Optimization*

TEAM ASSIGNMENTS

In groups of 3: Python and C++

- **Assignment #2**
  - *ER Modeling & Schema Design*
- **Assignment #4**
  - *Buffer Manager*
- **Assignment #5**
  - *Heapfile Manager*
- **Assignment #6**
  - *Query Operators*

EXAMS

- **Midterm Exam**
  - *when*: Wednesday, March 12 (during class)
  - *where*: Psychology 105

- **Final Exam**
  - *When*: Friday, May 9 (2:45 – 4:45)
  - *Where*: TBD

WHAT IS EXPECTED FROM YOU

- Attend the lectures

- Participate and ask questions

- Do the assignments (start early!)

- Study for the exams

PLAGIARISM

- You may not copy source code or your answers from other students or the web

- Plagiarism will not be tolerated!

PLAGIARISM AND AI

The use of generative AI technology is discouraged in this course, although not prohibited. In instances when you choose to use AI, you are required to submit a follow-up document in the comments section of the assignment that documents the extent to which you utilized AI. This will make it easy for the instructor and the course grader to see how you used the results to develop or brainstorm your own response and will prove that there was no intention to engage in plagiarism. Providing this log ensures transparency and does not jeopardize academic integrity.

Use of AI technology is allowed for proofreading assignments or checking them for grammatical errors. Students are requested to indicate this at the bottom of the submission with a statement such as the following:

This document utilized an AI-powered grammar checker tool, [Name of tool], to identify and correct potential grammatical errors, ensuring clarity and accuracy in the writing process. All suggested changes were reviewed and manually implemented by the author, who retains full responsibility for the final content.

If the instructor suspects use of AI in an assignment and the student has not disclosed it, they will be contacted directly to arrange a meeting and address these concerns. Use of AI without transparency may be considered academic misconduct.

COURSE OVERVIEW

Part A: Databases from the user's perspective

- Module **A1**: SQL

- Module **A2**: Database Design

- Module **A3**: Relational Algebra

COURSE OVERVIEW

Part B: Database internals

- Module **B1**: Basics of DB Internals

- Module **B2**: Indexes

- Module **B3**: Query Processing

- Module **B4**: Transactions

BEFORE WE START

JUPYTER NOTEBOOK

- Jupyter notebooks are interactive shells which save output in a nice notebook format

- You'll use these for
  - in-class activities
  - interactive lecture supplements/recaps
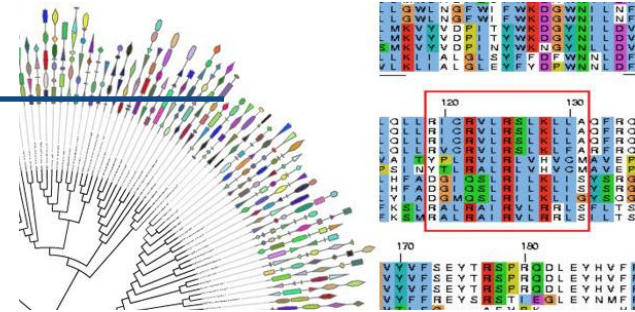 - problem sets, projects, …

JUPYTER NOTEBOOK SETUP

- Install on your laptop!

- We will be using **Python 3**

- Alternative methods are to use:
  - Ubuntu VM
  - CS Machines

- Come to office hours if you need help with the installation!

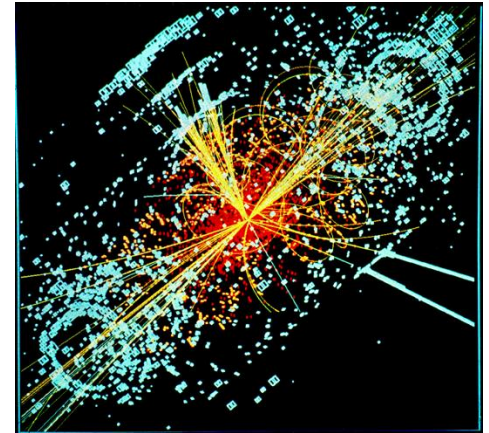**TODO #2**: try Activity 1 to familiarize yourself with Jupyter!

DATABASES: A SHORT INTRO

DATA IS EVERYWHERE!

- Our world is increasingly data driven
  - scientific discoveries
  - online services (social networks, online retailers)
  - decision making

- **Databases** are the core technology

WHAT IS THIS CLASS ABOUT?

The fundamentals of data management

- how we design and query a database?
- how do database management systems work?
- how do we build a DBMS?

DATABASES

*What is a database (**DB**)?*

an organized collection of structured data

What are examples of databases?

- payroll information
- product information (e.g. Amazon)
- airline reservations
- ...

DBMS

*What is a Database Management System (**DBMS**)?*

a program that allows us to manage efficiently a large DB and allows data to persist over long periods of time

What are examples of DBMSs?

- SQL Server, Microsoft Access (Microsoft)

- DB2 (IBM)

- Oracle

- MySQL, PostgreSQL, SQLite

EXAMPLE: ONLINE BOOKSTORE

What data do we need to store?

- information on books: title, author, ...
- customer information
- orders: date, items, payment method
- user preferences and behavior
- ...

EXAMPLE: ONLINE BOOKSTORE

How will we use the data stored?

- search for a specific book

- make an order

- recommendation engine

- generate order history

- produce sales figures

- ...

WHY NOT USE A FILE?

We could store the information about books in a CSV file that the application manages

Problems:

- How can we find a particular book fast?

- What if two threads try to write at the same time?

- What happens if there is a crash during an update?

- How do we ensure that an attribute has a valid value?

WHAT CAN A DBMS DO?

- Automate a lot of boring operations on data
  - don't have to program over and over
  - write complex data manipulations in a few lines
- Make data retrieval very fast
- Scale up to very large data sets (100s of GB)
- Safely allow concurrent access to the data
- Protect from system crashes