CS480: Database Systems

Introduction and Overview

Abolfazl Asudeh, asudeh@uic.edu

Aug. 2025

Instructor & TA

Instructor:

- Abolfazl Asudeh
- https://www.cs.uic.edu/∼asudeh/
- Email address: asudeh@uic.edu
- Drop-In Office Hours (in-person): **Mondays, 12:30 PM to 2:30 PM**
- Office number: 5452 CDRLC

TA:

- Mohsen Dehghankar
- https://mohsendehghankar.github.io/
- Email address: mdehgh2@uic.edu
- Office Hours: **Wednesdays, 3:30 PM to 5:30 PM**
- Office number: CDRLC 2404

Course Information

What (Is a Database)?
Who (Uses DB)?
Why (Should you care)?)

What Do Databases Do?

Core Functionality: Online Transaction Processing (**OLTP**)

- **"Define"** Data and their relationship
- **"Manipulate"** Data
  - **Insert**
  - **Update**
  - **Delete**

What Do Databases Do?[1]

- Provide **persistent storage**
- Efficient **declarative** access to data (aka querying)
- **Atomicity** of updates: Protection from hardware/software **failures**
- **Concurrent** access by **multiple users**
- **Secure** and **Safe** access to data

[1] Some of the slides, including this one, are taken from Prof. Glavic's slides.

What Do Databases Do?

ACID Property

- **Atomicity**: Ensures that a transaction is treated as a single, indivisible unit of work. **All parts** of the transaction must complete successfully, or the entire transaction is rolled back, leaving the database in its original state.
- **Consistency**: Guarantees that a transaction brings the database from one valid state to another.
- **Isolation**: Ensures that the ongoing operations of one transaction do not interfere with other concurrently running transactions.
- **Durability**: Guarantees that once a transaction has been committed, its changes are permanent and will survive any subsequent system failures, such as power outages

Brainstorming/curiosity question

How would you do it?

Who Uses Databases?

- **Most big software systems involve DBs!**
  - Web-based systems
  - Business Intelligence; e.g., IBM Cognos
  - . . .
- **Desktop / Mobile software (You!)**
  - Your music player; e.g., Amarok
  - Your email client
  - Half of the apps on your phone use SQLite
  - . . .
- **Every big organization**
  - Banks, Google, . . .
  - Government
  - . . .

Who Produces Database Systems?

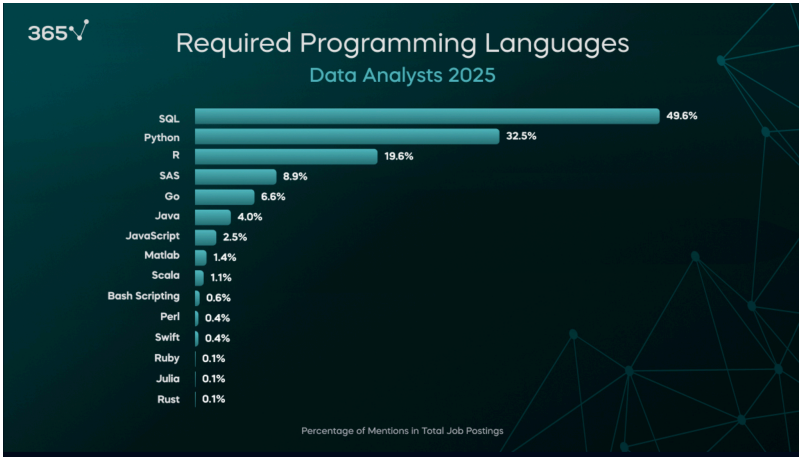
- **Traditional relational database systems is big business**
  - IBM → DB2
  - Oracle → Oracle
  - Microsoft → SQLServer
  - Open Source → MySQL, Postgres, SQLite, DuckDB, . . .
- **Distributed systems with DB characteristics**
  - Cloud storage and Key-value stores → Amazon S3, Google Big Table, . . .
  - Big Data Analytics → Hadoop, Google Map&Reduce, . . .
  - SQL on Distributed Platforms → Spark, Tenzing, . . .

Why Should You Care?

49 Years of Queries + 2 years
Don Chamberlin, SIGMOD Keynote 2023

Abstract: "The relational data model, proposed by **Ted Codd** in the 1970s, has been **the dominant paradigm** for storing and accessing business data for *several decades*."

51 years of SQL, still at the top! (365datascience.com)



Course Information

Course Description
Outline of topics
Textbook

Course Description

This course has been updated to include the *recent trends and technological advancements* in AI and Computer Science. It is organized into two major parts.

- Part I: the core principles underpinning the discipline of database design and utilization.
- Part II: modern extensions of the course in the AI era.

Outline of topics

Part I: **Relational Database Core** (2/3rd of the semester).

- Introduction to Databases & DBMS
- Data Modeling with ER Diagrams
- Relational Model and Constraints
- SQL: Basic Queries & Schema Definition
- SQL: Advanced Queries
- Functional Dependencies & Normalization

Outline of topics

Part II: **Modern Extensions** (1/3rd of the semester).

- Vector Databases & Embeddings
- RAG: Retrieval Augmented Generation
- Fundamentals of NL-to-SQL with LLMs

Excluded (yet important) topics – beyond the scope of this course

- **Relational Algebra**
- **System & Performance**
  - Indexing Structures & Query Processing
  - Transactions & Concurrency Control
  - Physical Database Design & Storage
- . . .

Course Materials: Part I: Relational Database Core

1. (main textbook: `https://db-book.com/`)
   Silberschatz, Korth, and Sudarshan, **Database System Concepts**, McGraw Hill, 2010, ISBN-13: 978-1260084504
2. Elmasri and Navathe, **Fundamentals of Database Systems**, 7th edition, Addison-Wesley, 2006, ISBN: 9780133971224
3. Ramakrishnan and Gehrke, Database Management Systems, 3nd Edition, McGraw-Hill, 2002, ISBN-13: 978-0072465631
4. Garcia-Molina, Ullman, and Widom, **Database Systems: The Complete Book**, 2nd Edition, Prentice Hall, 2008, ISBN-13: 978-0131873254

Course Materials: Part II: Modern Extensions

1. **Vector Databases**:
   - — Online materials for Vector Database
   - — pgvector documentations and tutorials:
     - GitHub (official documentation): https://github.com/pgvector/pgvector
     - “PostgreSQL as a Vector Database: A Pgvector Tutorial”
   - — HNSW: Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs.
   - — HENN: A Hierarchical Epsilon Net Navigation Graph for Approximate Nearest Neighbor Search. (The TA’s paper)
2. **RAG**: TBA
3. **NL-to-SQL**: (a recent survey) Liu, Xinyu, et al. ”A Survey of Text-to-SQL in the Era of LLMs: Where are we, and where are we going?.” IEEE Transactions on Knowledge and Data Engineering (2025)

Preformance Evaluation

Grading Policy
Points Breakdown
Project

Grading Policy

1. **Assignments** 15% = 5%+5%+5
2. **Midterm** 30%
3. **Project** 20%
4. **Final exam** 35%

- A: > 85
- B: > 70
- C: > 55
- D: > 40
- F: < 40

Academic Honesty & Late Submission Policy

Cheating will NOT be tolerated!

- **All work has to be original.**
- **Cheating = 0 points for assignment/exam**. Possibly F in the course and **further administrative sanctions**.
- Every dishonesty **will be reported** to office of academic dishonesty
- Unless explicitly specified, you are **NOT ALLOWED** to use Generative AI and Large Language Models (such as ChatGPT).

Late policy for assignments and the project:

- **-20% per day**
- No exceptions!

Project

- Create **groups of at most three**
  - Your responsibility!
  - Inform me + TA
- **Git repositories**
  - Create an account on GitHub, using your UIC email
  - Use it to exchange code with your fellow group members
  - The project has to be submitted via the group repository
- More details TBA!

Project

- Every student has to contribute to every phase of the project!
- Don’t let others freeload on your hard work!
- **Inform me or a TA ASAP**

UIC

Software

Git
Docker
PostgreSQL

Self Study

I expect you to learn by yourself how to effectively use the following technologies:

- **Git** - a version control system
  - Projects are submitted through git
  - We provide some useful examples/scripts through git
- **Docker** - a virtualization platform (think VMs, but more lightweight)
  - The easiest way to get Postgres running is by using the Docker image provided by Dr. Glavic's Github Repo (link below)
- **PostgreSQL**
  - I expect you to learn how to start/stop/configure a Postgres server and how to connect to a running Postgres server

Further Instruction

Follow `https://github.com/lordpretzel/cs480`

Postgres Overview

- Client/Server Architecture
- Postgres Cluster
  - A directory on the machine running the server that stores data and configuration files
- Postgres Server
  - A Postgres server handles the data of a single cluster
  - Clients connect to the server via network (TCP/IP)
    - Send commands and receive results

Postgres Overview

- Clients
  - GUI clients: e.g., PGAdmin (https://www.pgadmin.org/)
  - CLI clients: e.g., the built-in psql tool
  - Programming Language Libraries
    - Java: JDBC (https://jdbc.postgresql.org/)
    - Python: asyncpg (https://github.com/MagicStack/asyncpg), pyscopg (http://initd.org/psycopg/)

Get Your Hands Dirty

Get a working version of the PostgreSQL server. Feel free to use LLMs (like ChatGPT) as a help for installing Postgres and getting it to work. Your options:

- **Install locally**
  - Installer packages for Windows exists
  - Linux distributions have a Postgres package
  - Installation from source is not that hard
- **Get our docker image** (docker pull Dr. Glavic's Github Repo)
  - It's an extension of the official Postgres image which loads our running example university database

Get Your Hands Dirty

- Create a database cluster (the directory PostgreSQL uses to store data)
- Check that you can start/stop the server
- Check that you can connect to the running server using psql or any other client

An AI-assisted tool for ERD design

Dr. Rooshenas has created a tool for designing ERD, which provides an AI assistant. You can use this for designing your ERD and ensuring its correctness.

Follow `http://aiassist.cs.uic.edu/`.
How to use: [link]
You will need a user name to use this – will be done soon.

Thank you!