

THE GEORGE WASHINGTON UNIVERSITY

WASHINGTON, DC

1. Course Overview

CSCI 2541 Database Systems & Team Projects



CS 2541: Database Systems & Team Projects

Spring 2022

Professor: Tim Wood

TAs: Deep Bhattacharya and Chih-Chen "Joan" Wu

Support Staff: Catherine Meadows, Ethan Baron,
Jett Jacobs, and Alex Coleman

<https://cs2541-22s.github.io>

Tim Wood

I teach: Software Engineering, Operating Systems, Sr. Design

I like: distributed systems, networks, building cool things



Course Staff

Grad TAs: Labs, office hours, project mentoring, grading

- Deep and Joan

Undergraduate team: Labs, office hours, project mentoring

- Cat, Ethan, Alex, and Jett

(full intros in future classes)

Who are you?

We are looking forward to getting to know all of you in the coming weeks!

If possible, please enable video in Zoom

- Helps us!
- Helps you!
- +1 if I can recognize you when we are back in person

If you can't, be sure to set a zoom profile picture

These days, more than ever, we need human interaction!

What is a Database System?

A Database is a collection of related data

- Typically carefully structured with well defined relations
- Models entities and relations

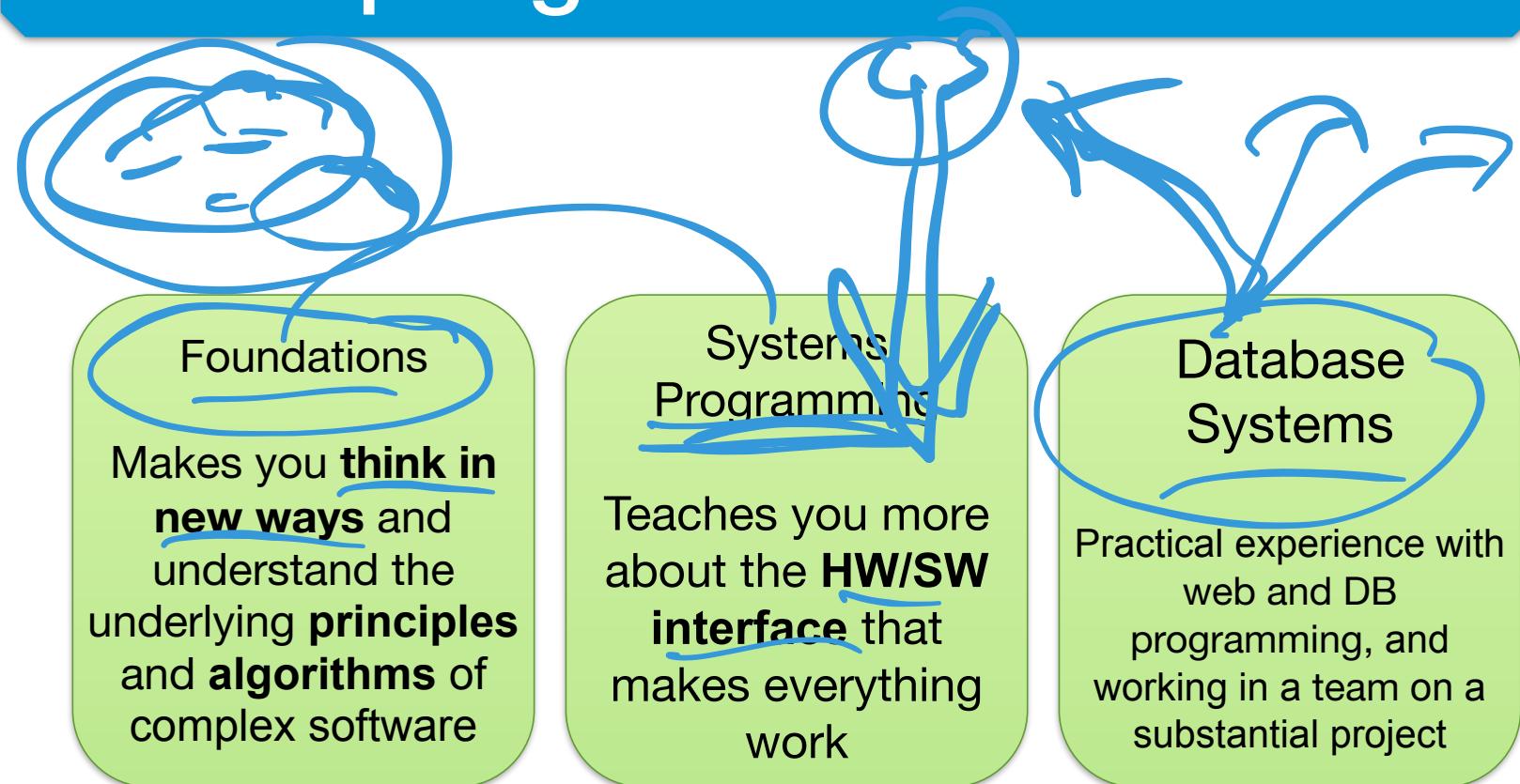
A Database Management System (**DBMS**) is the software system to store/retrieve/manage the database.

- Provides an interface over the database
- Examples: Oracle, MySQL, MongoDB, Postgres, Dynamo...

A Database System = DBMS + Data + Application

- In this course, we will use MySQL + Python (Flask web app framework)

Your Spring Semester



What is this course?

Database systems design and implementation

- Theory of relational database design and query languages
 - Relational Model, Relational algebra, SQL
- Application development using Relational DBMS (MySQL), with PHP
Python web apps

Intro to database models for unstructured data (Big data)

- Overview of NoSQL database models

but wait there's more!

What is this course?

Database systems design and implementation

- Theory of relational database design and query languages
 - Relational Model, Relational algebra, SQL
- Application development using Relational DBMS (MySQL), with PHP
Python web apps

Intro to database models for unstructured data (Big data)

- Overview of NoSQL database models

Database System Project: Full stack development

Teamwork – SW development in teams

- Project (SW) integration

Improving technical communication skills:

- Writing in the disciplines (WID)* in tandem with CS2501

*Course is not just about Database design – you have to learn and participate in the other two course objectives (WID, Team SW).

What is this course?

Week 1: Intro to DBs, HTML, CSS

- Also: Learn Python **on your own!**

Weeks 2-3: Relational Database Model and SQL

Weeks 4-5: Good Database Design Practices

Week 6: EXAM!

Week 7: Getting ready for the project

Weeks 8+: Team project time!

- Phase 1: Build a web app
- Phase 2: Integrate your web apps

What is this course?

One of the most **useful** and
applicable courses you will take
while at GW!

(I hope)

What about you?

How much do you know
about **HTML/CSS**?

How much do you know
about **Python**?

(Poll)

Intro to Databases & Database Management Systems

Before We Start...

How are you going to succeed in this class?

Pay attention

- Make your own notes by slide number. Slides will be posted online, but are insufficient on their own!

Participate

- Ask at least one question per week (either here or later on Slack)
- Will count as part of your grade (not just attendance)

Use Zoom's "Raise Hand" if you have a question, or type in chat

Databases in the Real-World

Databases are everywhere in the real-world even though you do not often interact with the DBMS directly.

- ~\$50 billion annual industry

Examples:

- Retailers manage their products and sales using a database.
 - Wal-Mart has one of the largest databases in the world ~40 Petabytes !
- Online web sites such as Amazon, eBay, etc..
- Social media sites: Facebook adds >500 Terabytes of new data per day!
- The university maintains all your registration information in a database.
- Mobile apps need to store your local data somewhere

DBMS Examples

There are many different Database Management Systems

- In this class we will (mostly) use **MySQL**



Have you heard of any other database software platforms?

(type in chat)



An Example: RestaurantDB

Your aunt and uncle want you to help track the top customers at their restaurant

- When are upcoming reservations?
- Which customer visits the most often?
- Who spends the most money?
- How to reward customers who order 10+ dishes?
- What is the most popular dish?
- What is the most popular dish on Tuesdays?

Why not just use Excel?

A spreadsheet can easily store data

- Use columns to structure information
- Enter a new row for each restaurant customer
- Can use formulas to calculate answers to some queries or sort/filter table

But...

Why will Excel have problems with this?
What will be difficult?

Customers.xlsx						
first_name	last_name	email	phone	reservation	birthday	
Kelli	Perris	kperris0@nifty.com	963-930-8531	1/6/2020	9/12/1958	
Goddart	Braams	gbraams1@ted.com	534-300-7372	1/26/2020	1/18/1979	
Merrel	Clere	mclere2@blogger.com	194-430-7153	1/25/2020	2/12/1957	
Towney	Bratcher	tbratcher3@narod.ru	304-227-0235	1/5/2020	7/10/1977	
Latia	Peete	lpeete4@w3.org	448-368-1546	1/28/2020	3/6/1964	
Hadria	Rann	hrann5@cbsnews.com	206-421-4913	1/24/2020	1/5/1976	
Bastian	Clother	bclothier6@microsoft.com	104-598-7586	1/25/2020	9/15/1965	
Corene	Attoe	cattoe7@soup.io	819-616-3261	1/20/2020	3/7/1946	
Sara-ann	Creeboe	screeboe8@theatlantic.com	831-348-1941	1/13/2020	4/15/1998	

Why not just use Excel?

A spreadsheet can easily store data

- Use columns to structure information
- Enter a new row for each restaurant customer
- Can use formulas to calculate answers to some queries or sort/filter table

But...

- Some calculations may not be easy (or possible) to write as Excel formulas
- Sorting/filtering tables to see different results will get messy
- Need to be careful about entering data (no validation of format/data type)
- Can accidentally erase old data since data storage and processing are combined in one place

Why not just use files?

You already know how to read and write data to a file...

- Could store data in Excel, then export to CSV (comma separated value file)
- Program could read file's lines and store into a data structure e.g., a Linked List
- Different functions could calculate answers for different queries

Why will file processing have problems with this?
What will be difficult?

first_name, last_name, email, phone, reservation, birthday
Kelli, Perris, kperris0@nifty.com, 963-930-8531, 1/6/2020, 9
Goddart, Braams, gbraams1@ted.com, 534-300-7372, 1/26/2020,
Merrel, Clere, mclere2@blogger.com, 194-430-7153, 1/25/2020
Towney, Bratcher, tbratcher3@narod.ru, 304-227-0235, 1/5/2020
Latia, Peete, lpeete4@w3.org, 448-368-1546, 1/28/2020, 3/6/2020
Hadria, Rann, hrann5@cbsnews.com, 206-421-4913, 1/24/2020, 1
Bastian, Clother, bclothert6@microsoft.com, 104-598-7586, 1/20/2020
Corene, Attoe, cattoe7@soup.io, 819-616-3261, 1/20/2020, 3/7/2020
Sara-ann, Creeboe, screeboe8@theatlantic.com, 831-348-1941
Conny, Matthius, cmatthius9@sphinn.com, 117-195-6721, 1/13/2020

Why not just use files?

You already know how to read and write data to a file...

- Could store data in Excel, then export to CSV (comma separated value file)
- Program could read file's lines and store into a data structure e.g., a Linked List
- Different functions could calculate answers for different queries

But...

- File system doesn't know anything about file format used in our program - we need to implement all parsing ourselves
- Calculations are tightly tied to data format - if we change format we need to rewrite all the code
- Redundancy - many similar programs would all have code for parsing files
- Hard to support multiple simultaneous users

Scaling up

What if we have a chain of restaurants?

What becomes more complex?

Can Excel/file parsing work in this environment?

Scaling up

What if we have a chain of restaurants?

Uhoh...

- We need to support **concurrent** updates/reads to the data
- We need to ensure data remains **consistent** despite simultaneous access
- Google Sheets might make it easier for multiple users to make edits, but doesn't guarantee these properties!
- Expanding our custom file parser to support network access is a major effort!

So what can we conclude thus far....

Excel is limited in **scale** and **flexibility**

File processing is not a **portable** or **efficient** solution

Need a “database approach” that provides **data**
independence from the processing acting on it

Need to support **simultaneous access** while retaining data
integrity

So how do we specify **business rules** of the data,
relationships within the data, **who gets access** to what
data... **How to organize and manage the data ?**

A DBMS should provide...

Structure that
is **independent**
of the
underlying file
formats

Queries to
flexibly read,
update, and
delete
information

Transactions
that provide
guarantees
about **multi-**
user
consistency

DBMS: How to structure the data?

Structure that is independent of the underlying file formats

What is the data needed?

- Eg: What do we need to store to uniquely identify a restaurant customer?

How to store & organize the data?

- How many attributes are really needed about a student/course/faculty
- What is an efficient way to organize the data?
- This is why we will need to study schema design and Normal forms

Data Models and Representation

Structure that is independent of the underlying file formats

A **data model** is a formal framework for describing data.

- Data objects, relationships, constraints (business rules)
- Provides primitives for data manipulation and data definition
- Provides us with the mathematical basis to prove/assert properties and show correctness of algorithms

The **relational model** was the first model of data that is **independent** of its data structures and implementation

- Data organized as **relations** (“tables”)

Not the only way!

- NoSQL databases model unstructured and big data without requiring strict relations
- Other data models: network, hierarchical, Object Oriented...
- Relational model is inefficient for many such applications

DBMS: How to provide abstraction?

Structure that is independent of the underlying file formats

The major problem with developing applications based on files is that the application is **dependent** on the file structure.

There is no **program-data independence** separating the application from the data it is manipulating.

- If the data file changes, the code that accesses the file may require changes to the application.



A major advantage of DBMS is they provide **data abstraction**.

Data abstraction allows the internal definition of an object to change without affecting programs that use the object through an external definition.

Database Schema

Structure that is independent of the underlying file formats

Similar to types and variables in programming languages

Schema – **structure** of the database

- Ex: database contains information about Students and Courses and the relationships between them
- Defines columns and the type of data that can be stored in them

Occurs at multiple levels:

- **Logical Level:** Database design, definition of structure and relations
- **Physical Level:** Implementation of how data is stored on disk

Customers Relation (Table)

first_name	last_name	email	phone	reservation	birthday
Kelli	Perris	kperri0@nifty.com	963-930-8531	1/6/2020	9/12/1958
Goddart	Braams	gbraams1@ted.com	534-300-7372	1/26/2020	1/18/1979
Merrel	Clere	mcclere2@blogger.com	194-430-7153	1/25/2020	2/12/1957
Towney	Bratcher	tbratcher3@narod.ru	304-227-0235	1/5/2020	7/10/1977
Latia	Peete	lpeete4@w3.org	448-368-1546	1/28/2020	3/6/1964
Hadria	Rann	hrann5@cbsnews.com	206-421-4913	1/24/2020	1/5/1976
Bastian	Clother	bclother6@microsoft.com	104-598-7586	1/25/2020	9/15/1965
Corene	Attoe	cattoe7@soup.io	819-616-3261	1/20/2020	3/7/1946
Sara-ann	Creeboe	screeboe8@theatlantic.com	831-348-1941	1/13/2020	4/15/1998

Levels of Data Modeling

Structure that is independent of the underlying file formats

Logical Level: describes data stored in the database and the relationship between them

ex: type customer {

name: string

email: string

birthday: date }

Physical Level: describes how a record is stored (i.e., how is data organized on the disk)

- Ex: sorting, page alignment, index



Big Idea: Logical and Physical level independence

- Can change one without changing the other!!

Data Independence

Structure that is independent of the underlying file formats

Logical data independence

- Protects the user from changes in the logical structure of the data:
- Lets us reorganize the customer “schema” without changing how we query/store it

Physical data independence

- Protects the user from changes in the physical structure of data:
- Lets us change how student data is stored in memory/disk without changing how the user would write the query

Additional Views:

- DB applications hide details of data types and can also hide some information (salary?) for security & privacy purposes

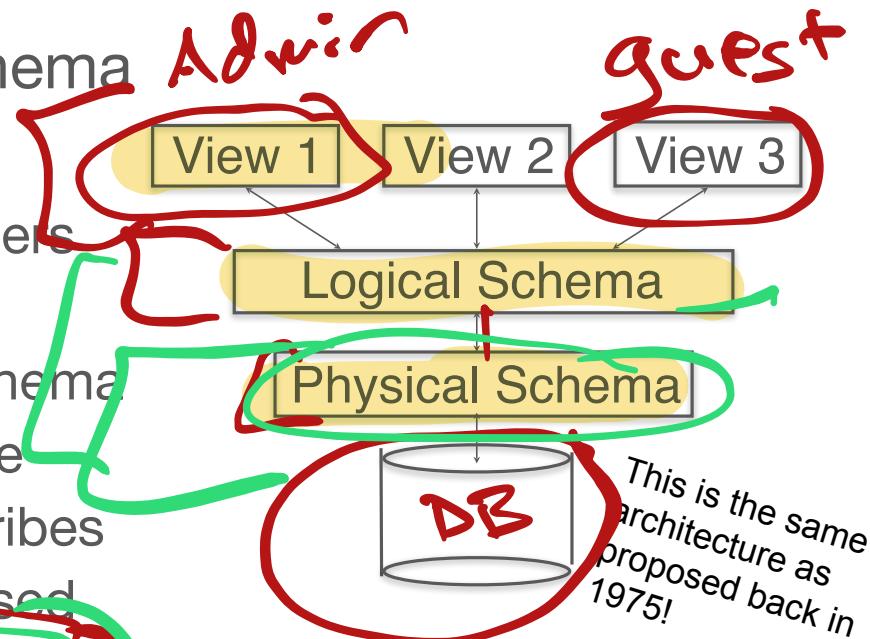
Summary- Levels of Abstraction

Structure that is independent of the underlying file formats

Many views, single conceptual (logical) schema and physical schema

- Views describe how users see the data
- Conceptual/Logical schema defines logical structure
- Physical schema describes the files and indexes used

*Schemas are defined using DDL;
data is modified/queried using DML*



Confusing?
Curious?

Course Resources

Course webpage: will have links to syllabus, lecture notes, online resources (and in-class exercises when applicable)

- <https://cs2541-22s.github.io/>

Blackboard:

- Electronic submission of non-programming homeworks
- Reporting grades

Github:

- Project submissions and team management

Slack:

Message Board

- For class announcements and Q&A
- For team coordination

Replit.com:

- For labs and HW assignments

Next . .

Complete the survey that will be mailed to you by COB Tuesday

- Without this you will NOT be able to do the lab exercises

Make sure you have Github and Replit accounts before next class!

Sign up for the class Slack page

Watch for announcements and engagement opportunities

Attributions

These slides are adapted from materials made by
Prof. Bhagi Narahari

Image attribution:



Created by Wilson Joseph
from Noun Project



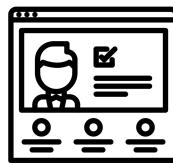
Created by Ditta
from Noun Project



Created by Gregor Cremer
from Noun Project



Created by Wilson Joseph
from Noun Project



Created by lastspark
from Noun Project



Created by Dawid Sobolewski
from Noun Project



Created by Wilson Joseph
from Noun Project



Created by Nataša
from Noun Project



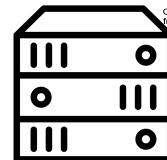
Created by srujanreddy
from Noun Project



Created by jinu.mallick9
from Noun Project



Created by jinu.mallick9
from Noun Project



Created by Senthil Agra
from Noun Project



Created by Yvesse Akans
from Noun Project