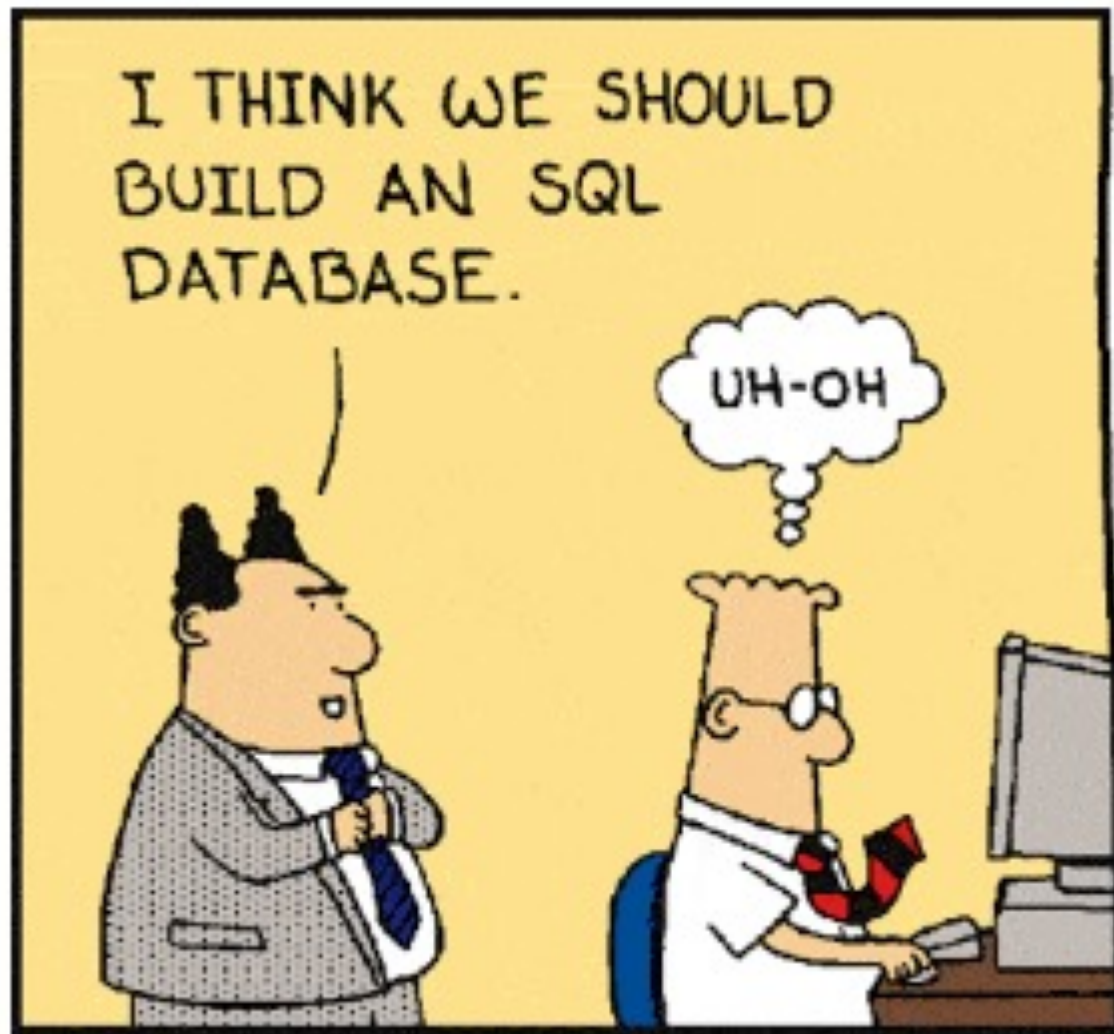


S. Adams E-mail: SCOTTADAMS@AOL.COM

11/17 © 1995 United Feature Syndicate, Inc.(NYC)

I THINK WE SHOULD
BUILD AN SQL
DATABASE.

UH-OH





WHAT COLOR DO YOU
WANT THAT DATABASE?

I THINK
MAUVE HAS
THE MOST
RAM.



Databases

Administrivia

- Course info:
 - Webpage will be the main source of knowledge.
 - Contact me through email or Slack (Slack is better for real-time discussion.)
- Pre-req: CS241 and/or CS251
 - Some data structure knowledge will be helpful, some command-line knowledge will be helpful.
- Coursework:
 - Homework, group project, midterm, final
- Sign up for the CS Slack channel.
- Be prepared to bring laptops every so often.

Group project

- You will design and implement your own database-driven website.
- Ideas: shopping, auctions, write a better course registration system, library/bibliography system, reviews (Yelp), bank, finance/stocks, job postings, social networking (Instagram, Twitter), recipes, movies, apartments (AirBnB), ...
- Groups: probably 4-5 people, formed on your own.
- Spread out over the whole semester; check-ins along the way.

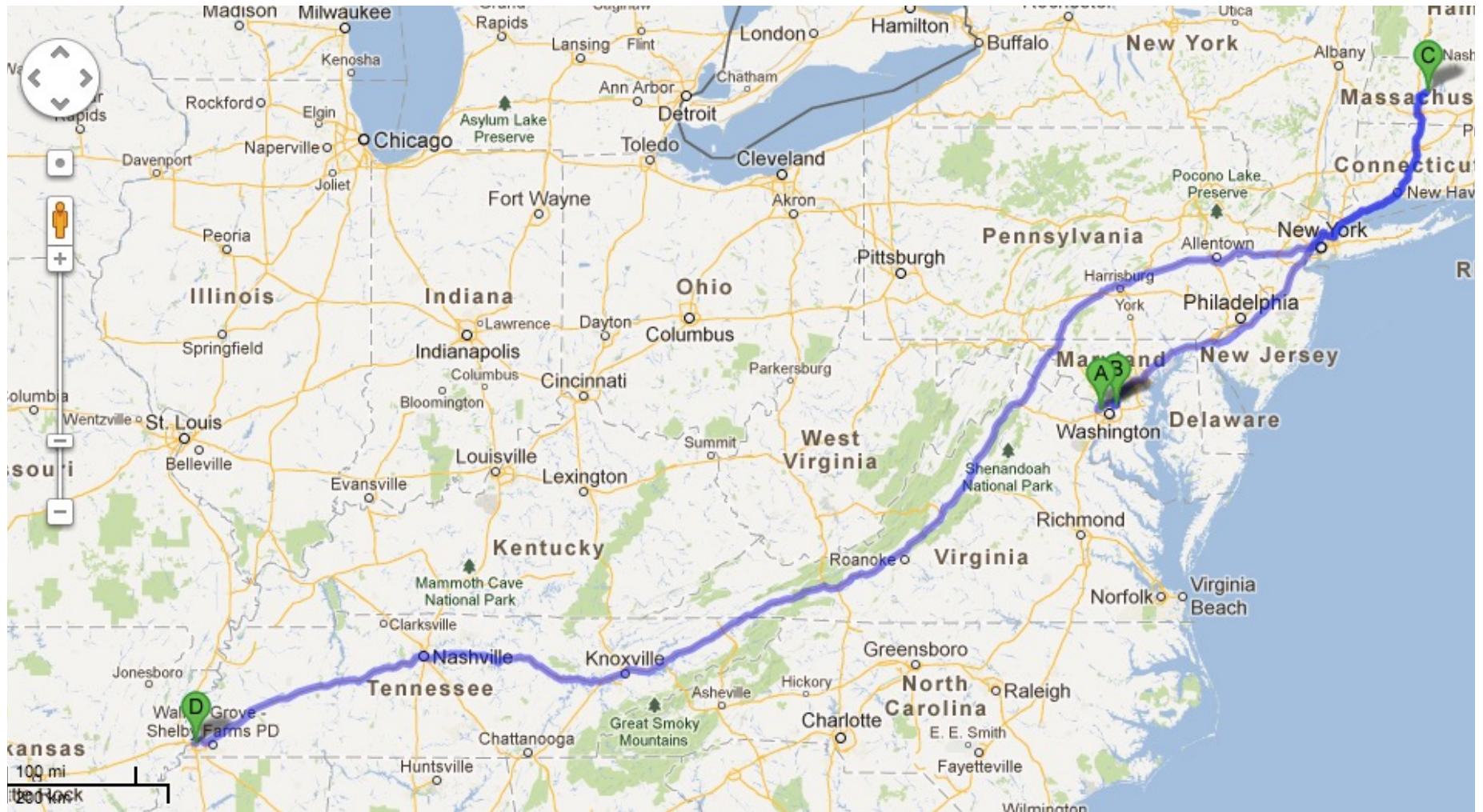
Office Hours

- Please come! These are a resource for you. Typically they are underutilized.
- Also send me email or a Slack message for "unscheduled" office hours.
 - I can't promise I'll respond immediately, but I'll always get back to you within a day, and we can most likely schedule a time to meet.

How to succeed

- Come to class.
- Ask questions when you are confused: in class or office hours.
- Take notes.
- Do not leave readings, homework, projects to the last minute. You can't BS (most) of these.

A little about me



Why study databases?

- Academic reasons
- Programming reasons
- Business (get a job) reasons
- Student reasons

What will you learn?

- Database design
 - How do you model your data so it can be stored in a database?
- Database programming
 - How do I use a database to ask it questions about my data (*queries*)?
 - How do I use a programming language to interact with a database?
- Database implementation
 - How does the database itself work; i.e., how does it store, find, and retrieve data efficiently?

What is the goal of a database?

- Electronic record-keeping, enabling **fast** and **convenient** access to the information inside.
- DBMS = Database management system
 - Software that stores individual databases and knows how to search the information inside.
 - RDBMS = Relational DBMS
 - Examples: Oracle, MS SQL Server, MS Access, MySQL, PostgreSQL, IBM DB2, SQLite

DBMS Features

- Support massive amounts of data
 - Giga-, tera-, petabytes
- Persistent storage
 - Data continues to live long after program finishes.
- Efficient and convenient access
 - Efficient: don't search the entire thing to answer a question!
 - Convenient: allow users to ask questions as easily as possible.
- Secure, concurrent, and atomic access

Example: build a better Workday

- Professors offer classes, students register, get grades
- What are some questions we (students or faculty) could ask of this database system?
 - Find my GPA.
 - ...

Example: build a better Workday

- Why are security, concurrency, and atomicity important here?

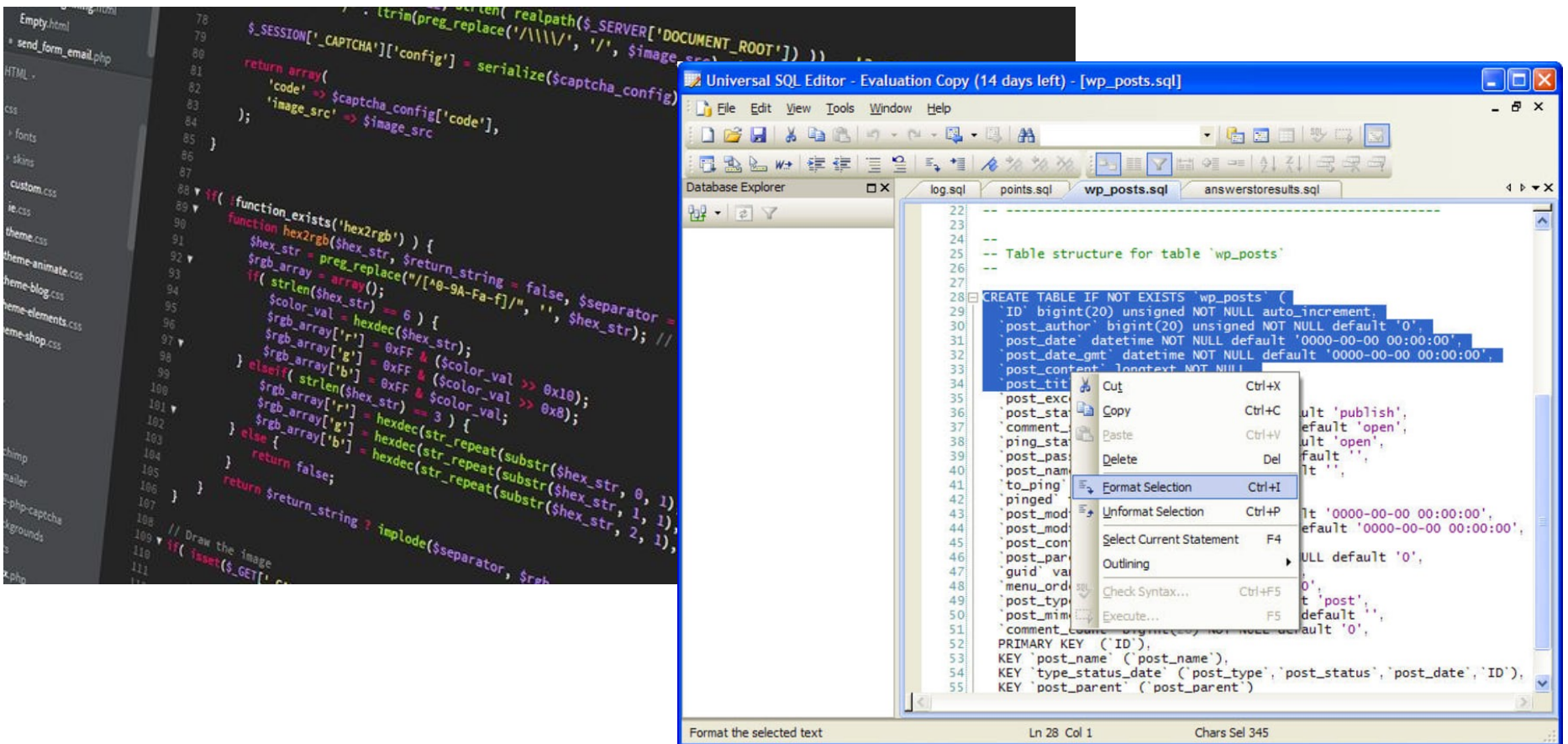
Solution 1

- Advantages?
- Disadvantages?



Solution 2

- Text files and Python/Java programs



Solution 3

Let's use CSV:
(comma-separated
values)



```
Hermione,Granger,R123,Potions,A  
Draco,Malfoy,R111,Potions,B  
Harry,Potter,R234,Potions,A  
Ronald,Weasley,R345,Potions,C
```

What's the issue here?

Hermione, Granger, R123, Potions, A

Draco, Malfoy, R111, Potions, B

Harry, Potter, R234, Potions, A

Ronald, Weasley, R345, Potions, C

Harry, Potter, R234, Herbology, B

Hermione, Granger, R123, Herbology, A

File 1:

Hermione, Granger, R12

Draco, Malfoy, R111

Harry, Potter, R234

Ronald, Weasley, R345

File 2:

R123, Potions, A

R111, Potions, B

R234, Potions, A

R345, Potions, C

R234, Herbology, B

R123, Herbology, A

Problems

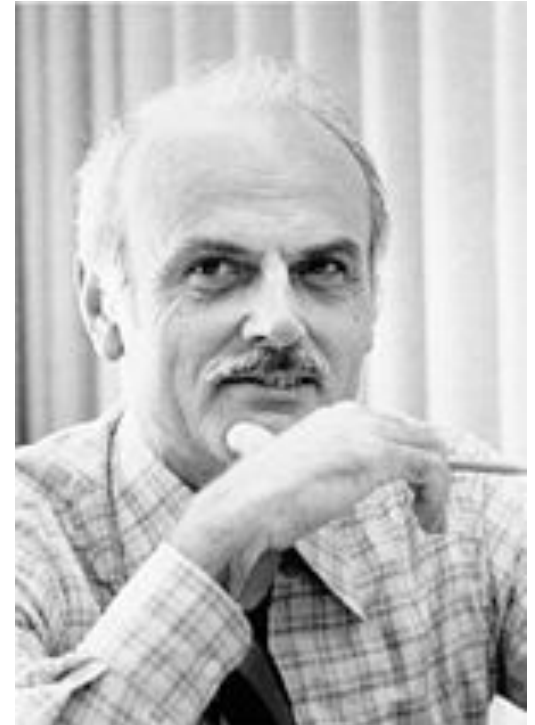
- Inconvenient – need to know Python/Java to get at data!
- Redundancy/inconsistency
- Integrity problems
- Atomicity problems
- Concurrent access problems
- Security problems

Why are there problems?

- Two main reasons:
 - The description of how the files are laid out is buried within the Python/Java code itself (if it's documented at all)
 - There is no support for **transactions** (supporting concurrency, atomicity, integrity, and recovery)
- **DBMSs handle exactly these two problems.**

Relational database systems

- Edgar F. Codd was a researcher at IBM who conceived a new way of organizing data based on the mathematical concept of a *relation*. (1970)
- Relation: a set of ordered tuples (oh, no, CS172 stuff...)



Highlights of RDBMS

- (R)DBMS = relational database management system.
- Data is stored in *relations*, which resemble tables:

First	Last	Course	Grade
Hermione	Granger	Potions	A
Draco	Malfoy	Potions	B
Harry	Potter	Potions	A
Ronald	Weasley	Potions	C

- Underlying data structures are more complicated.

Highlights of RDBMS

- Users issue ***queries*** to the DBMS, which are handled by the ***query processor***.
 - Behind the scenes: *query optimizer* handles all the details of figuring out the most efficient way to answer the query, which might involve combining multiple tables, sorting the data, selecting only a subset of it, ...
- The ***transaction manager*** handles all the details of atomicity and concurrency.

On to the real stuff now...