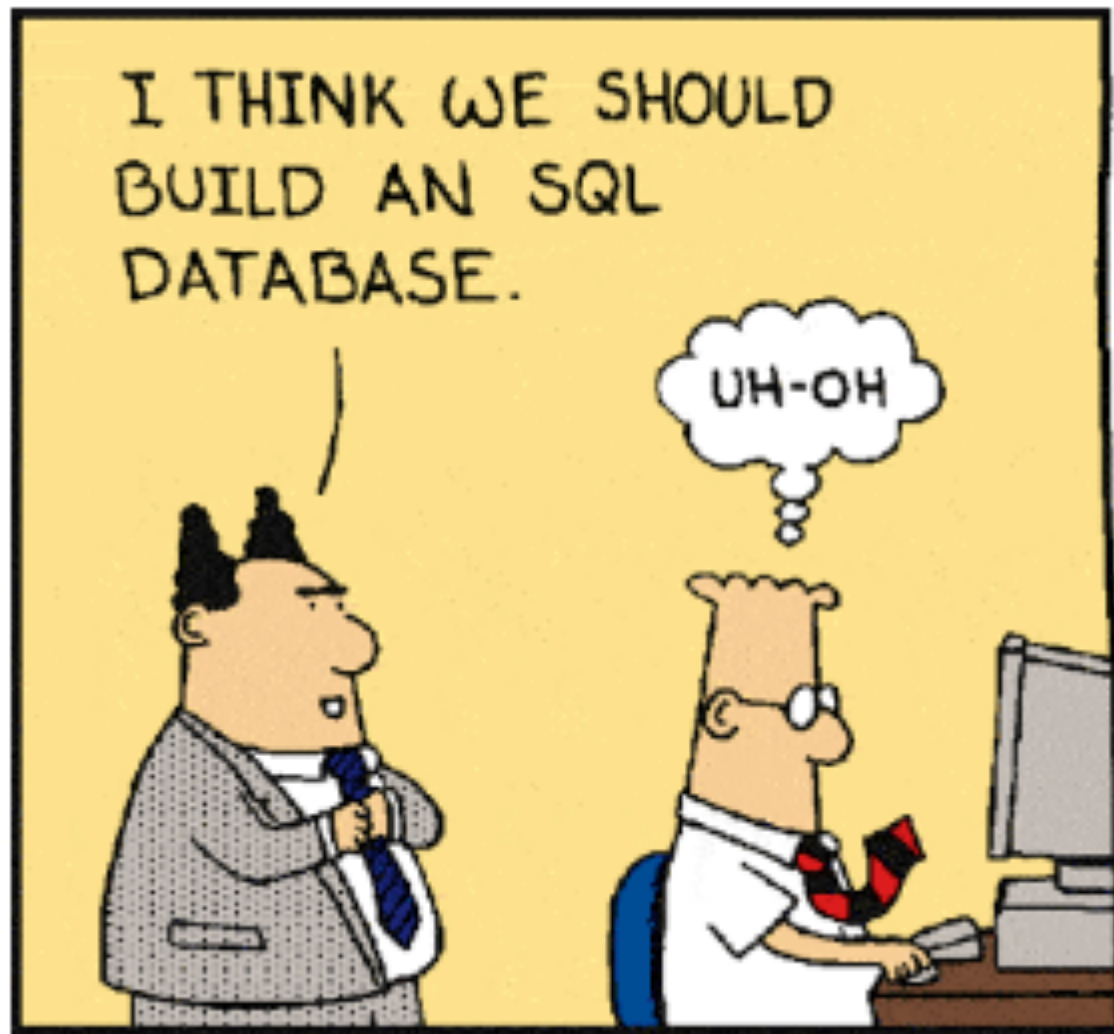


I THINK WE SHOULD
BUILD AN SQL
DATABASE.

UH-OH





WHAT COLOR DO YOU
WANT THAT DATABASE?

I THINK
MAUVE HAS
THE MOST
RAM.



Databases

Standard stuff

- Class webpage: `cs.rhodes.edu/db`
- Textbook: get it somewhere; used is fine
 - Stay up with reading!
- Prerequisite: CS 241
- Coursework:
 - Homework, group project, midterm, final
- Be prepared to bring laptops every so often.

Class conduct

- Be on time.
- Raise your hand to ask a question.
 - Corollary: Raise your hand a lot!
- Please raise your hand to be excused.
- Turn off the computer screens when asked.

Group project

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

How to succeed

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

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?
- 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 BannerWeb

- Professors offer classes, students sign up, get grades
- What are some questions we (students or faculty) could ask?
 - Find my GPA.
 - ...
- Why are security, concurrency, and atomicity important here?


Obvious solution: Folders

- Advantages?
- Disadvantages?

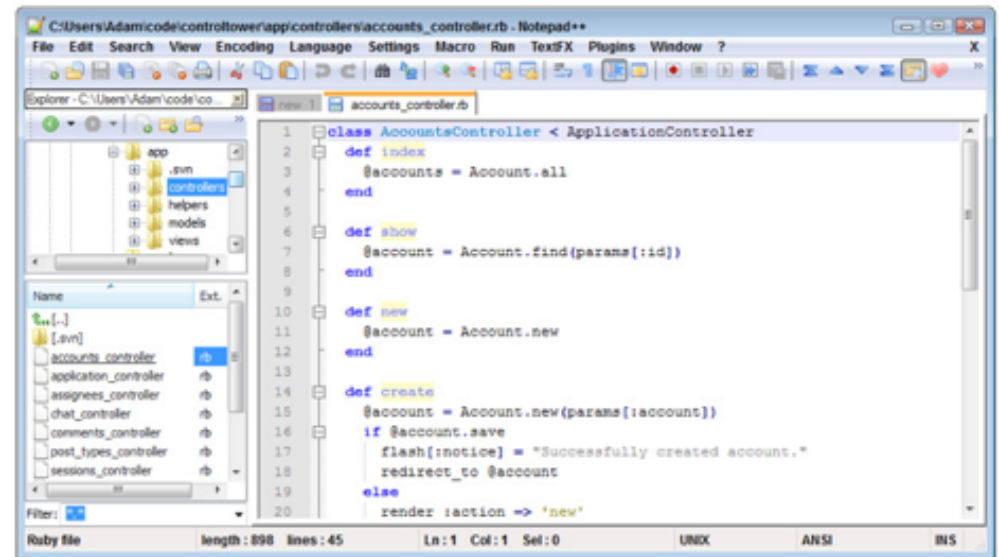


Obvious solution++

- Text files and Python/C++/Java programs



```
Sat Sep 11 18:42:26 1999 CDT
Dadditide teleports in.
Dadditide says, "Beep"
Dadditide vigorously wiggles his network.
Kira says, "howdy."
Dadditide smiles.
Dadditide says, "Spin idlin'?"
Sat Sep 11 18:47:26 1999 CDT
Kira says, "i think so."
Dadditide nods.
Dadditide has the worst luck...
Kira says, "ah?"
Dadditide finally got his starter kit from his bank for online banking the same
day that intuit's online banking server goes kookoo...
Dadditide sighs.
Dadditide says, "They're supposed to have it fixed by Monday, though. But I'll
believe it when I see it."
Dadditide has been real impressed with the Quicken folks of late...
Dadditide got a free upgrade from the Quicken that he purchased (version 4, if I
remember correctly) to Quicken Deluxe 98.
Sat Sep 11 18:52:26 1999 CDT
Kira says, "He. I should try quicken, sebbe. Dunno. Right now I have my
checkbook in filemaker and it works just fine for me ;)"
Dadditide doesn't have Filemaker.
Dadditide could always whip something together in AppleWorks, now that he has
it.
Dadditide says, "I really like Quicken, because it saves me a lot of typing by
resembling past entries."
Dadditide says, "Just type "w" instead of wal-mart, for example."
Dadditide spends a lot of money at good ol' Nally World. :)
```



```
C:\Users\Adam\code\controltowerapp\controllers\accounts_controller.rb - Notepad++
File Edit Search View Encoding Language Settings Macro Run TextFX Plugins Window ?
Explorer - C:\Users\Adam\code\co
accounts_controller.rb
1 class AccountsController < ApplicationController
2
3   def index
4     @accounts = Account.all
5   end
6
7   def show
8     @account = Account.find(params[:id])
9   end
10
11  def new
12    @account = Account.new
13  end
14
15  def create
16    @account = Account.new(params[:account])
17    if @account.save
18      flash[:notice] = "Successfully created account."
19      redirect_to @account
20    else
21      render :action => 'new'
22    end
23  end
24 end
```

Obvious solution++

- Let's use CSV:



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

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, R123

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

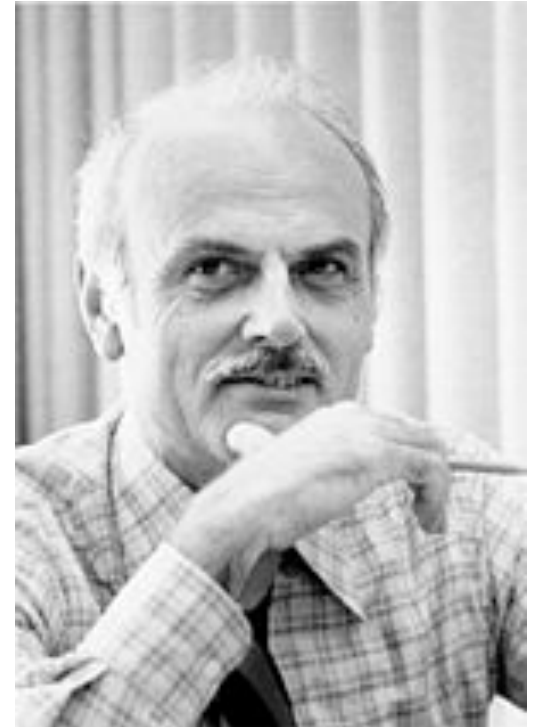
- Inconvenient – need to know Python/C++/Java to get at data!
- Redundancy/inconsistency
- Integrity problems
- Atomicity problems
- Concurrent access problems
- Security problems

Why are there problems?

- Two main reasons:
 - The description of how the files are laid out is buried within the Python/C++/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: combining multiple tables, optimizing the query.
- The *transaction manager* handles all the details of atomicity and concurrency.

On to the real stuff now...