



# Welcome!

## EECS-3421A:

### *“Introduction to Database Systems”*

### Fall 2016

**Parke Godfrey**

class site

# Introduction

# What is a database?

- A *systematic* collection of data.
- Which begs the question, how is the data organized?
  - We need some general notion of *data model*.
  - For a *given* collection of data, we need to specify *its* organization.

We will call this a *schema*.

- How do we retrieve specific data out of a database?
- How do we *add* and *delete* information?

# How much data?

smartphone calendar	30,000 bytes
university course database	30,000,000 bytes
audio collection of Beethoven	30,000,000,000 bytes
printed collection of US Library of Congress	30,000,000,000,000 bytes
data processed by Google *daily*	30,000,000,000,000,000 bytes
total global Internet traffic per month	30,000,000,000,000,000,000 bytes

<http://en.wikipedia.org/wiki/Petabyte>

<http://en.wikipedia.org/wiki/Exabyte>

[http://www.jameshuggins.com/h/tek1/how\\_big.htm](http://www.jameshuggins.com/h/tek1/how_big.htm)

# Universality of Database Management

Almost all of the above — all except the last — are stored in databases.

And these are “queried” using a common *query language*!

---

How universal, we will have to see...

# What is a database *system*?

- A system to manage — *create, update, and query* — databases for us.
- 
- What more *functionality* should such a system provide?

# A database system should support...

- **management:** *manage* databases (and the data in them)
  - *create* databases
  - *update* the data
  - *query* the data
- **querying:** provide a powerful, easy way for users & applications to *retrieve* given data from a database
  - What should this “API” look like?
  - For users?  
For application programs?
- **permanence / durability:** *ensure* data is never lost
- **scalability:** *scale up* painlessly to *very large* databases
- **integrity:** *ensure* correctness (*quality*) of the data
- **concurrency:** handle many users & applications *at the same time*
- **security:** *manage* and *enforce* privileges for which users & applications can *see* and *manipulate* which data

---

And do this all *very* efficiently, of course!

# A different perspective!

- *Databases* is a *data-centric* way of looking at things.
- This is quite different than what most of us are used to!
- The main paradigm is the *query*.



# A history of database systems

## Tabulating Machine Company

- It is hard to trace the origins of an idea.
  - But one could make the case that the field of *databases* all started in 1896...
- 



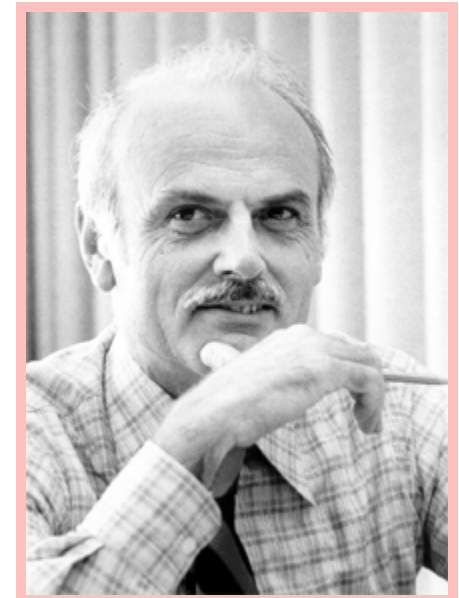
“IBM” was founded in 1896 as TMC — Tabulating Machine Company — by Herman Hollerith.

# The relational data model

- No...When did modern *computer* database systems come about?!
  - In the 1960's.
  - But our story starts in 1970.
- 

Edgar F. Codd invents the *relational data model*, and its *first-order theory*.  
[1970]

An IBM team implements *System R* (the first *cost-based relational query optimizer*).



# Oracle



Larry Ellison implements *Oracle* from the System R paper, and markets Oracle. [1978]

---

IBM wakes up, markets DB2. [1983]

# Big Table

Google implements its own *Big Table* to store the entire WWW.

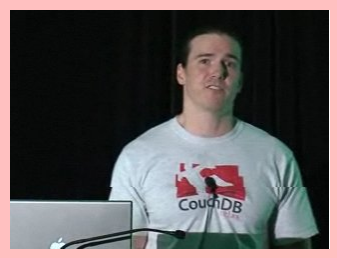
Big Table was designed and implemented by Jeffery Dean and Sanjay Ghemawat.  
[1978]

---

This marks the true start of “big data”.



# NoSQL



Facebook and eBay deploys a radically different family of data storage engines, known as NoSQL (“Not only” SQL).

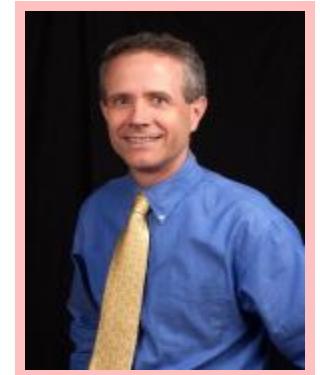
*CouchDB* is implemented by Damien Katz, a former IBM engineer. [2005]

---

*Scale out versus scale up.*

# Embedded: data in the small

Relational database engines are scaled down to be embedded in mobile devices: Android and iOS.



*SQLite* is used by both smart phone OSes.

SQLite was implemented by Richard Hipp, for on-board data management of Navy missile system. [2008]

---

*Sybase Anywhere* started in 1992! A Waterloo-Kitchner company.

Now part of SAP.

**This course**

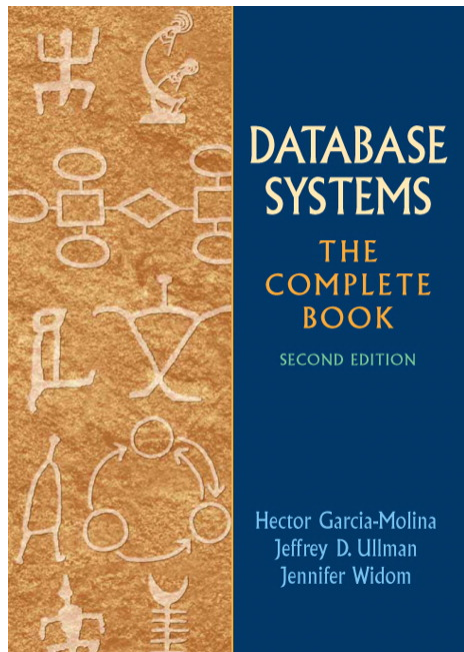
# Outline

- data models
- design
- queries (SQL)
- applications

- [syllabus: schedule](#)
- [class site](#)



# Textbook



H. Garcia-Molina, J. Ullman, & J. Widom  
Database Systems: The Complete Book  
Pearson / Prentice Hall, 2nd Edition

[@amazon.ca](https://www.amazon.ca)

# Structure

- Four projects (25% in all)
- two tests (20%  $\times$  2)
- final exam (35%)

- [syllabus: criteria](#)
- [class site](#)

# What this course is *not* about

- This course is *not* about *implementing* database systems.
  - It is about “implementing” (*designing*) databases (*schema*), querying them, and building applications with databases.
  - EECS-4411, *Database Management Systems*, is about *building* database systems.
- This course is *not* about *big data*, per se, which is about *scale out*.
  - A 4<sup>th</sup>-year course is being developed, *Big Data Systems*, that will be about that.