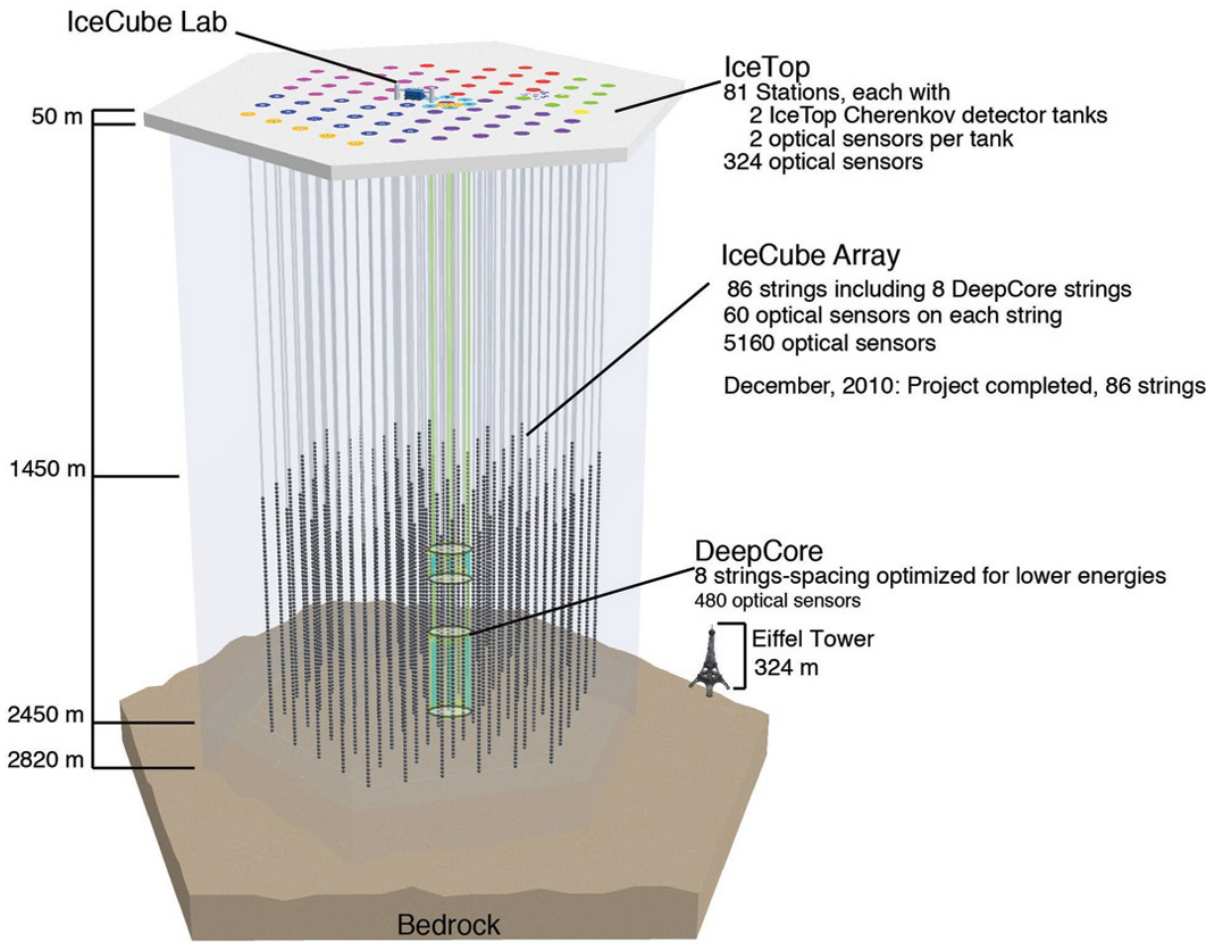
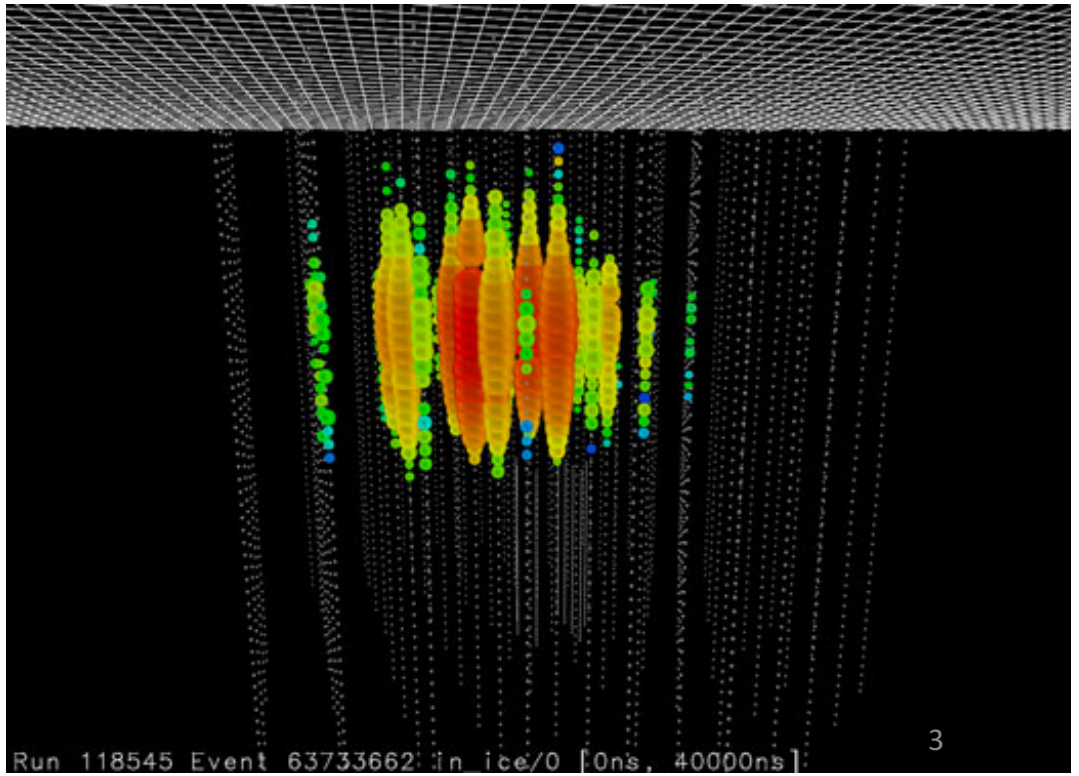
Intro to Database Management Systems

Lecture 1: Overview

“Data is the Future”



IceCube Neutrino Observatory.



Big science is data driven.

DeepDive

All of society is online.

Data analysis in the fight against human trafficking.

New York DA use MEMEX Data for all trafficking investigations this year.

Increasingly many companies see themselves as **data driven**.

Even more “traditional” companies…

https://www.youtube.com/watch?v=OvfU1NpCJQQ
https://www.youtube.com/watch?v=3xGoBlI_fdg
https://www.youtube.com/watch?v=OpDIEJrog3s

The world is increasingly **driven by data…**

This class teaches **the basics** of how to use & manage data.

Today's Lecture

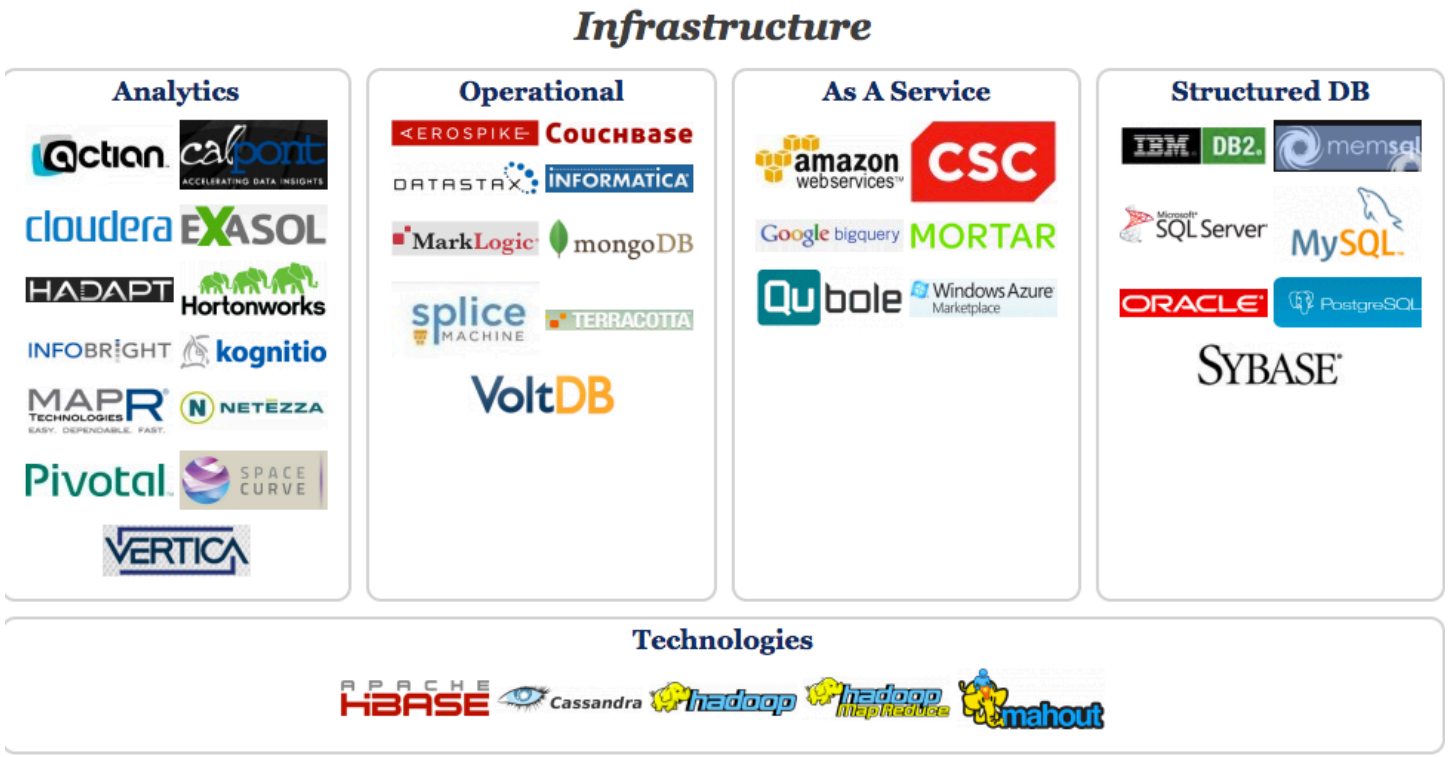
1. Introduction, admin & setup
   - ACTIVITY: Jupyter "Hello World!"

2. Overview of the relational data model
   - ACTIVITY: SQL in Jupyter

3. Overview of DBMS topics: Key concepts & challenges

1. Introduction, admin & setup

Big Data Landscape… Infrastructure is Changing



New tech. *Same* Principles.

http://www.bigdatalandscape.com/

Why should **you** study databases?

- **Mercenary- make more $$$:**
  - Startups need DB talent right away = low employee #
  - Massive industry…

- **Intellectual**:
  - Science: data poor to data rich
    - No idea how to handle the data!
  - Fundamental ideas to/from all of CS:
    - Systems, theory, AI, logic, stats, analysis….

Many great computer systems ideas started in DB.

What this lecture is (and is not)

- Discuss **fundamentals of data management**
  - How to design databases, query databases, build applications with them.
  - How to debug them when they go wrong!
  - Not how to be a DBA or how to tune Oracle 12g.

- We'll cover **how database management systems work**

- But not **the principles of how to build** them ☹

Lectures

- Lecture slides cover **essential material**
  - This is your best reference.

- Try to cover same thing in **many ways**: Lecture, lecture notes, homework, exams (no shock)
  - Attendance makes your life easier…

Lectures: A note about format of notes

Take note!!

These are asides / notes (still need to know these in general!)

Definitions in blue with **concept being defined** bold & underlined

Main point of slide / key takeaway at bottom

Warnings- pay attention here!

Jupyter Notebook “Hello World”

- Jupyter notebooks are interactive shells which **save output in a nice notebook format**
  - They also can display markdown, LaTeX, HTML, js…

FYI: “Jupyter Notebook” are also called iPython notebooks but they handle other languages too.

- You’ll use these for
  - in-class activities
  - interactive lecture supplements/recaps
  - homeworks, projects, etc.- if helpful!

Note: you **do need to know or learn python** for this course!

Jupyter Notebook Setup

1. **HIGHLY RECOMMENDED.** Install **on your laptop** via the instructions on the next slide / Piazza

2. Other options running via one of the alternative methods:
   1. **Ubuntu VM.**

Please help out your peers by posting issues / solutions on Piazza!

Jupyter Notebook Setup

https://github.com/HazyResearch/cs145-notebooks-2016/blob/master/jupyter_install.md

2. Overview of the relational data model

What is a DBMS?

- A large, integrated collection of data

- Models a real-world *enterprise*
  - *Entities* (e.g., Students, Courses)
  - *Relationships* (e.g., Alice is enrolled in 145)

A **Database Management System (DBMS)** is a piece of software designed to store and manage databases

A Motivating, Running Example

- Consider building a course management system (**CMS**):
  - Students
  - Courses
  - Professors

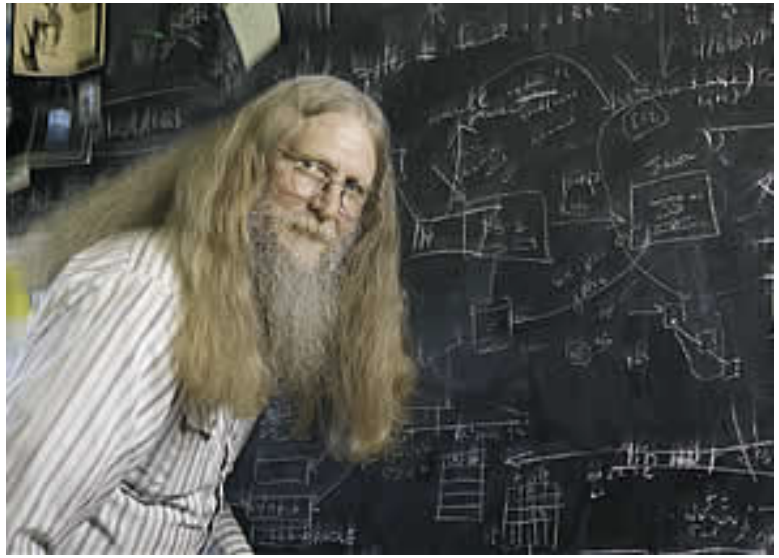
  } *Entities*

  - Who takes what
  - Who teaches what

  } *Relationships*

Data models

- A **data model** is a collection of concepts for describing data
  - The relational model of data is the most widely used model today
    - Main Concept: the *relation*- essentially, a table
- A **schema** is a description of a particular collection of data, **using the given data model**
  - E.g. every *relation* in a relational data model has a *schema* describing types, etc.



“Relational databases form the bedrock of western civilization”

- Bruce Lindsay, IBM Research

Modeling the CMS

- *Logical Schema*
  - Students(sid: *string*, name: *string*, gpa: *float*)
  - Courses(cid: *string*, cname: *string*, credits: *int*)
  - Enrolled(sid: *string,* cid*: string,* grade*: string*)

Relations

sid	Name	Gpa
101	Bob	3.2
123	Mary	3.8

Students

cid	cname	credits
564	564-2	4
308	417	2

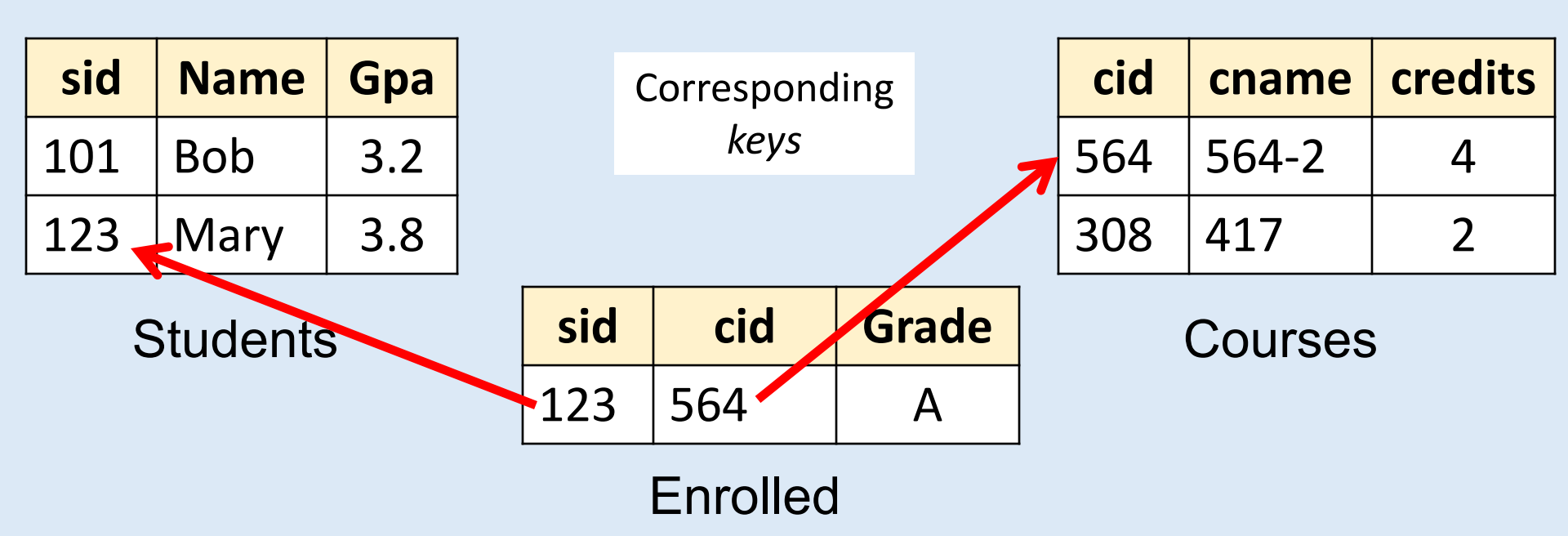
Courses

sid	cid	Grade
123	564	A

Enrolled

Modeling the CMS

- *Logical Schema*
  - Students(sid: *string*, name: *string*, gpa: *float*)
  - Courses(cid: *string*, cname: *string*, credits: *int*)
  - Enrolled(sid: *string,* cid*: string,* grade*: string*)



Data independence

Concept: Applications do not need to worry about *how the data is structured and stored*

Logical data independence:
protection from changes in the *logical structure of the data*

I.e. should not need to ask: can we add a new entity or attribute without rewriting the application?

Physical data independence:
protection from *physical layout changes*

I.e. should not need to ask: which disks are the data stored on? Is the data indexed?

One of the most important reasons to use a DBMS