Introduction to Data Management

Lecture #1
(The Course “Trailer”)

Instructor: Mike Carey
mjcarey@ics.uci.edu

Today’s Topics

- Welcome to my biggest class ever!
- Read (and live by) the course wiki page:
  - http://www.ics.uci.edu/~cs122a/
- Also follow (and live by) the Piazza page:
  - https://piazza.com/uci/winter2017/cs122a/home
- Let’s peek at the wiki page, and then let’s also preview what lies ahead...
- Note: There will be a quiz in this week’s initial discussion sessions...!
What is a Database System?

- What’s a **database**?
  - A very large, integrated collection of data

- Usually a model of a **real-world enterprise**
  - **Entities** (e.g., students, courses, Facebook users, …) with attributes (e.g., name, birthdate, GPA, …)
  - **Relationships** (e.g., Susan is *taking* CS 234, Susan is a *friend* of Lynn, …)

- What’s a **database management system** (DBMS)?
  - A software system designed to store, manage, and provide access to one or more databases

---

File Systems vs. DBMS

- Application programs must sometimes *stage large datasets* between main memory and secondary storage (for buffering huge data sets, getting page-oriented access, etc.)
- *Special code needed* for different queries, and that code must be (stay) correct and efficient
- Must *protect data from inconsistency* due to multiple concurrent users
- *Crash recovery* is important since data is now the currency of the day (corporate jewels)
- *Security and access control* are also important(!)
Evolution of DBMS

- Early DBMS Technologies
  - Records and pointers
  - Large, carefully tuned data access programs that have dependencies on physical access paths, indexes, etc.
  - Majority of application development effort goes towards building and then maintaining data access logic

- CODASYL/IMS
  - Manual Coding
    - Byte streams

- Relational DB Systems
  - Declarative approach
  - Tables and views bring “data independence”
  - Details left to system
  - Designed to simplify data-centric application development

Why Use a DBMS?

- Data independence.
- Efficient data access.
- Reduced application development time.
- Data integrity and security.
- Uniform data administration.
- Concurrent access, recovery from crashes.
Why Study Databases?

- **Shift from computation to information**
  - At the “low end”: explosion of the web (a mess!)
  - At the “high end”: scientific applications, social data analytics, ...

- **Datasets increasing in diversity and volume**
  - Digital libraries, interactive video, Human Genome project, EOS project, the Web itself, ...
  - Mobile devices, Internet of Things, ...
  - ... need for DBMS exploding!

- **DBMS field encompasses most of CS!!**
  - OS, languages, theory, AI, multimedia, logic, ...

---

Why Study Databases (Really)?

- **Big Data! 😊**
Data Models

- A *data model* is a collection of concepts for describing data
- A *schema* is a description of a particular collection of data, using a given data model
- The *relational model* is (still) the most widely used data model today
  - *Relation* – basically a table with rows and (named) columns
  - *Schema* – describes the tables and their columns

Levels of Abstraction

- Many *views* of one *conceptual (logical) schema* and an underlying *physical schema*
  - Views describe how different users see the data.
  - Conceptual schema defines the logical structure of the database
  - Physical schema describes the files and indexes used under the covers
**Example: University DB**

- **Conceptual schema:**
  - `Students(sid: string, name: string, login: string, age: integer, gpa: real)`
  - `Courses(cid: string, cname: string, credits: integer)`
  - `Enrolled(sid: string, cid: string, grade: string)`

- **Physical schema:**
  - Relations stored as unordered files
  - Index on first and third columns of `Students`

- **External schema (a.k.a. view):**
  - `CourseInfo(cid: string, cname: string, enrollment: integer)`

---

**To Be Continued – Questions?**