CS425 – Summer 2016
Jason Arnold
Course Information

Modified from:

Database System Concepts, 6th Ed.

Why are Databases Important?

- **What do Databases do?**

1. Provide persistent storage
2. Efficient declarative access to data -> Querying
3. Protection from hardware/software failures
4. Safe concurrent access to data

Who uses Databases?

- Most big software systems involve DBs!
  - Business Intelligence ⇒ e.g., IBM Cognos
  - Web based systems
  - …
- **You!** (desktop software)
  - Your music player ⇒ e.g., Amarok
  - Your Web Content Management System
  - Your email client
  - …
- Every big company
  - Banks
  - Insurance
  - Government
  - Google, …
  - …

Who Produces Databases?

- **Traditional relational database systems is big business**
  - IBM ⇒ DB2
  - Oracle ⇒ Oracle ☺
  - Microsoft ⇒ SQLServer
  - Open Source ⇒ MySQL, Postgres, …
- **Emerging distributed systems with DB characteristics and Big Data**
  - Cloud storage and Key-value stores ⇒Amazon S3, Google Big Table, . . .
  - Big Data Analytics ⇒Hadoop, Google Map & Reduce, . . .
  - SQL over Distributed Platforms ⇒ Hive, Tenzing, …

Why are Database Interesting (for Students)?

- **The pragmatic perspective**
  - Background in databases make you competitive in the job market ;-)
- **Systems and theoretical research**
  - Database research has a strong systems aspect
    - Hacking complex and large systems
    - Low-level optimization
      - cache-conscious algorithms
      - Exploit modern hardware
  - Databases have a strong theoretical foundation
    - Complexity of query answering
    - Expressiveness of query languages
    - Concurrency theory
    - …

Why are Database Interesting (for Students)?

- Connection to many CS fields
  - Distributed systems
    - Getting more and more important
  - Compilers
  - Modeling
  - AI and machine learning
    - Data mining
  - Operating and file systems
  - Hardware
    - Hardware-software co-design

Webpage and Faculty

- **Course Info**
  - **Course Webpage**: http://cs.iit.edu/~cs425
  - **Blackboard**
  - **Syllabus:** http://cs.iit.edu/~cs425/files/syllabus.pdf
- **Faculty**
  - **Jason Arnold**
  - **Email:** jarnold6@hawk.iit.edu
  - **Phone**: 847-987-0179
  - **Office**: Stuart Building, room 204 (here) ☺
  - **Office Hours**: I will stick around after class as long as people need to talk to me…

Workload and Grading

- **Exams**
  - Midterm (25%)
  - Final – Cummulative (25%)
- **Homework Assignments** (preparation for exams!)
  - HW1 (Relational algebra – 5%)
  - HW2 (SQL – 5%)
  - HW3 (Database Application Programming – 15%)
- **Course Project (25%)**
  - Individual project
  - Goal is to learn about database tuning
  - … and have fun doing so…
  - Chances to earn extra credit

Course Objectives

- Understand the underlying ideas of database systems
- Understand the **relational data model**
- Be able to write and understand **SQL** queries and data definition statements
- Understand **relational algebra** and its connection to SQL
- Understand how to **write programs that access a database server**
- Understand the **ER model** used in database design
- Understand **normalization** of database schemata
- Be able to **create a database design** from a requirement analysis for a specific domain
- Know basic **index structures** and understand their importance
- Have a basic understanding of relational database concepts such as **concurrency control**, **recovery**, **query processing**, and **access control**

Fraud and Late Assignments

- All work has to be original!
  - Cheating = 0 points for assignment/exam
  - Possibly E in course and further administrative sanctions
  - Every dishonesty will be reported to office of academic honesty
- Late policy:
  - 50% off of grade

Required Textbook

- **Textbook:** Silberschatz, Korth and Sudarsham
  - ***Database System Concepts, 6th edition***
  - McGraw Hill
  - publication date:2006,
  - ISBN 0-13-0-13-142938-8.

Outline

- Introduction
- Relational Data Model
- Formal Relational Languages (relational algebra)
- SQL
- Indexing and Hashing
- Database Design
- Transaction Processing, Recovery, and Concurrency Control
- Storage and File Structures