Database Systems I

Instructor: Ouldooz Baghban Karimi

CMPT 354 - Summer 2019
About Me

Ouldooz Baghban Karimi

Contact
Email: ouldooz@sfu.ca
Office: SUR 4142 [We can also meet at BBY TASC1 9215 - Shared Space]
Web: https://www.sfu.ca/computing/people/faculty/ouldoozbaghbankarimi.html

Education
PhD, Computer Science, Simon Fraser University
MSc, Computer Engineering, Iran University of Science and Technology
BSc, Computer Engineering, University of Tehran

Interests
Networks and Systems
Virtualization
Data, Security, and Privacy
Teaching Assistants

Nadeegi Fernando  
Email: nhfernan@sfu.ca  
Education: Computing Science MSc  
Interests: Program analysis  
         Fault localisation

Padmanabhan Rajendrakumar  
Email: rajendra@sfu.ca  
Education: Computing Science MSc (Big Data)  
Interests: Database Systems  
         Data Warehousing  
         Big Data Analytics

Ryan McBride  
Email: rom2@sfu.ca  
Web: www.sfu.ca/~rom2/  
Education: Computing Science PhD Candidate  
Interests: Data-Mining for Electrical Systems  
         Risk Management  
         Public Good
Why Do We Study Database Systems?

• Many important computer applications must manage, update, and query databases (e.g. Banks, fleet control, search, school system)

• Data quality, quantity and timeliness becoming more important with AI & Machine Leaning (algorithms that generalize from data)

• Growth of user and machine generated data
Why Do We Study Database Systems?

- Challenges of data-intensive systems
  - Reliability
  - Concurrency
  - Performance
  - Accessibility
  - Security
Why Do We Study Database Systems?

• Most popular type of data-intensive system: Relational Databases (e.g. MySQL, Oracle)

• What is a database?
  • A collection of files that store related data

• What is a DBMS?
  • A piece of software designed to store and manage databases
Course Goals

• Learn to query a database

• Learn to design a database

• Learn working basics and important issues of Database Management Systems
Course Overview

- Introduction and history of data-intensive systems
- Data models & Relational Model of Data
- SQL
- Relational algebra
- Entity relationship model
- Query optimization & execution
- Database design theory
- Database applications
- Database systems implementation (storage management, query processing, transaction processing)
- Transaction processing (concurrency, recovery, security & privacy)
- Brief discussion of advanced topics
Grading

• **Assignments** (4 homework assignments, 2 mini-projects): 20%
  - -25% each late day

• **Quizzes** (5): 10%

• **Midterm (June 27, 2019)**: 20%

• **Final (August 12, 2019)**: 50%

• **Bonus Points**: Up to 2%
Office Hours

• Ouldooz
  Tuesdays & Thursdays 8-10am BBY TASC1 9215 (by appointment)
  Wednesdays 12-2pm, SRY 4142 (by appointment)

• Ryan
  Thursdays 10am-12pm (drop-in)

• Nadeegi
  Tuesdays 10am-12pm (drop-in)
  Tuesdays 3.30-4.30pm (by appointment)

• Padmanabhan
  Tuesdays 10:30am – 12:30pm (drop-in)
  Mondays 11:30am – 12:30pm (by appointment)
Course Resources

• Course Web Page
  • https://coursys.sfu.ca/2019su-cmpt-354-d1/

• Discussion Board, Assignments, Additional Learning Materials: Canvas

• Course mailing lists
  • cmpt-354-teaching (Teaching Team)
  • cmpt-354: Announcements (registered automatically)

• Email policy (cmpt-354-teaching)
  • Teaching Team Mailing list
    • [Urgent] Brief Description
    • [Office Appointment Request] Brief Description
    • [Request] Brief Description

• Anonymous feedback form
  • https://forms.gle/pwbZpn6BcECDXapE8
Textbooks

• **Database Management Systems**, 3rd Edition
  Raghu Ramakrishnan, Johannes Gehrke

• **A First Course in Database Systems**, 3rd Edition
  Jeffrey D. Ullman, Jennifer Widom
  Pearson, 2007

  Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer Widom
  Prentice Hall, 2009

  Michael Kifer, Arthur J. Bernstein, Philip M. Lewis
  Pearson, 2005
Brief History of Database Systems

1959
• CODASYL (Committee on Data Systems Languages) introduces COBOL
  • Common Business Oriented Language

1960s
• Integrated Data Store: First general purpose DBMS by Charles Bachman at General Electric in early 1960s. Basis for **network data model**

• Information Management System (IMS): Developed by IBM
Brief History of Database Systems

• First important data applications
  • Banking systems
  • Airline reservation systems
  • Corporate record keeping
  • Supplies and parts

• Requirements of such applications
  • Small items of data
  • Many queries or modifications on data
Brief History of Database Systems

1970s

• Previous to this point data models are **navigational**

• Edgar Codd Mathematician at IBM San Jose Research Lab
  Proposed relational data model
    • Store database in simple data structures
    • Decouple high-level structure from the low level storage details

• Early implementation of relation DBMS
  • System R (IBM Research)
  • INGRES (UC Berkeley)
  • Oracle (Larry Ellison)
Brief History of Database Systems

1980s

- SQL developed as part of IBM system R project and standardized in late 1980s
  - IBM DB2 commercialized 1983
- Relational model gained widespread use
- Start of today’s two famous open source DBMS
  - PostgreSQL
  - MySQL

1999

- SQL adopted and Standardized
  - American National Standards Institute (ANSI)
  - International Organization for Standardization (ISO)

SQL still has a very big market
Brief History of Database Systems

2000s

• Internet boom
  • Larger data volume and faster updates that could not be handled by single machine
  • Read-only web to read-write web
  • Scalability, Availability, Flexibility
  • Unstructured & semi-structured data
Brief History of Database Systems

2000s

- **OnLine Transaction Processing (OLTP)** & **OnLine Analytical Processing (OLAP)**
  - **OLTP** (High-frequency updates & small queries)
    - **NoSQL** vs. **Relational DBMS**?
  - **OLAP** (Low frequency updates & big queries)
    - **MapReduce** vs. **Relational DBMS**?
Brief History of Database Systems

2000s

• NoSQL
  • Non-Relational data models (document, key-value)
  • Non ACID Transactions
  • Custom APIs instead of SQL
  • Usually open source

ACID Properties
A: Atomicity (all or nothing)
C: Consistency
I: Isolation
D: Durability
Brief History of Database Systems

2000s

• **Not Only SQL (NoSql)**
  • **Document** stores (Data Model: **JSON**) - Example Systems: SimpleDB, CouchBase, MongoDB
  • **Column** stores (Data Model: **Big Table**) - Example Systems: Hbase, Cassandra, HyperTable
  • **Key-value** stores (Data Model: **Hash**) - Example Systems: DynamoDB, Riak, Redis, Membase
  • **Graph data** stores (Data Model: **Graph**) - Example Systems: Neo4J, InfoGrid, GraphBase

• **NoSQL Limitations**
  • Low-level Language: Simple read & write database operators
  • Weak Consistency (Last, Stale): Eventual Consistency (All updated)
  • Lack of Standardization: 100+ NoSQL systems
Brief History of Database Systems

2000s

• MapReduce
  • Advantages
    • Fault tolerance
    • Complex analytics (SQL, ML, Graph Processing)
    • Heterogeneous storage systems
    • No data loading requirement
  • Limitations
    • Debate on
      • Missing features
      • Incompatibility with DBMS tools
      • Novelty
      • Quality of implementation

MapReduce a Program model rather than a database system
Brief History of Database Systems

2010s

• NewSQL
  • Provide same performance of OLTP workloads as NoSQL DBMS without giving up ACID
    • Relational / SQL
    • Distributed
    • Usually closed source
  • Strong Consistency
  • Features & limitations on flexibility, scalability and availability
  • Systems: ScaleBase, MySQL Cluster, Hekaton SQL Server Clustrix, SAP HANA, Pivotal
Acknowledgements

I have used materials from the following resources in preparation of this course:

• Database Systems: The Complete Book
• Database Systems (Kifer, Bernstein, Lewis)
• Database System Concepts: https://www.db-book.com
• Course offerings
  • W 4111 (Eugene Wu - Columbia): https://w4111.github.io/
  • CS 245 (Matei Zaharia - Stanford): http://web.stanford.edu/class/cs245/
  • CS 186 (Joe Hellerstein - Berkeley): https://sites.google.com/site/cs186fall17/
  • CMU 150721 (Andy Pavlo - CMU): https://15721.courses.cs.cmu.edu/spring2016/