Lecture Notes for Lecture 1 of CS 5200 (Database Management Systems) for the Summer 1, 2019 session at the Northeastern University Silicon Valley Campus.

Course Overview

Philip Gust,
Clinical Instructor
Department of Computer Science

Course Overview

Instructor, Course and Presentation URLs

- The instructor's website is:
  http://www.ccis.northeastern.edu/home/pgust
- The course website is:
  http://www.ccis.northeastern.edu/home/pgust/classes/cs5200/2019/Summer1

Course Overview

Goals

- This course introduces database management systems (DBMS) as a class of software systems and prepares students to be sophisticated users of database management systems.
- The course will study two widely-used types of DBMS that present different models of information storage and organization:
  - Relational Database Management Systems (RDBMS)
  - Non-relational database management systems (NoSQL)

Course Overview

About Relational Databases

- A relational database management system (RDBMS) is based on the relational model invented by Edgar F. Codd at IBM's San Jose Research Laboratory.
- RDBMSs have been a common choice for the storage of information in databases used for financial records, manufacturing and logistical information, personnel data, and other applications since the 1980s.
- Relational databases have generally replaced legacy hierarchical databases and network databases because they were easier to implement and administer.

Course Overview

About Non-Relational Databases

- A non-relational database is a database that does not incorporate the table/key model that relational database management systems (RDBMS) promote.
- These kinds of databases require data manipulation techniques and processes designed to provide solutions to big data problems that big companies face.
- The most popular emerging non-relational database is called NoSQL (Not Only SQL).
- Due to the expanse of technologies, such as horizontal scaling of computer clusters, NoSQL databases have become popular as an alternative to RDBMS databases.

Course Overview

About About the Course

- This course includes:
  - ***Lectures*** that focus on giving you a conceptual understanding of database design & usage, as well as implementation.
  - ***Homework*** will ask you to apply their conceptual knowledge via problems and code implementation.
  - ***Quizzes*** will be given periodically to help you assess their mastery of recent course material.
  - ***Team Project*** will allow you to work in two-person teams to complete a RDBMS and (time permitting) a NoSQL database application.
- There will be no midterm or final exam.

Course Overview

Grading Policy

- Your grade for this course will be based on:
  - Individual assignments: 50%
  - Quizzes: 15%
  - RDBMS project: 30%
  - Instructor's discretion: 5%

Course Overview

Grading Policy

- The final grade for a student is calculated as the weighted average of the preceding list and rounded up to the nearest integer. Letter grade uses the following scale:
  - A [95, 100]
  - A- [90, 94]
  - B+ [85, 89]
  - B [80, 84]
  - B- [75, 79]
  - C+ [70, 74]
  - C [65, 69]
  - C- [60, 64]
  - D [0, 59]

Course Overview

Teaching Assistants

- Two teaching assistants (TAs) will be assisting the instructor. We will announce office hours once we all have a chance to meet and discuss dividing responsibilities.
  - Qing Liao (liao.qing@husky.neu.edu)
  - Sanchit Saini (saini.sa@husky.neu.edu)

Course Overview

Piazza Online Forum

- This class will use NEU's instance of Piazza online forum:

  http://piazza.com/ccs.neu.edu/summer2019/cs5200siliconvalley/home

- The course staff have registered you with the system, and you should already have receive a confirmation email..
- You can communicate with your fellow students, and ask questions of the instructors and TA. You may make private posts that will only be seen by the instructor and TA.
- The course staff will also make announcements through this forum.

Course Overview

GitHub

- The class will use the CCIS GitHub instance to distribute programming examples, and check in homework and class projects.
- Each of you will have a repository for each assignment, quiz, or project: 2019S1CS5200SV/assignment-1-*ccisid* using your CCIS ID.
- You can use GitHub's drag-and-drop user interface to check your code into your repositories.

Course Overview

Software

- For the RDBMS portion of this course, we will use Java with the following software:
  - JavaDB/Derby. This is an embedded RDBMS product that is provided as an optional component of Java that is now provided by the Apache foundation. This will be our primary RDBMS.
  - MySQL/MariaDB. This is a server based RDBMs product that is now provided by Oracle. MariaDB is a non-commercial version of MySQL. We may use this as a second database to study certain features.
  - JDBC. This is the Java binding to relational database management systems. It the Java implementation of the Open Database Connectivity standard for RDBMS, and a standard part of the J2EE.

Course Overview

Software

- For the NoSQL portion of this course, we will use JavaScript with the following software:
  - MongoDB. This is a NoSQL database management system that is based on a document model. It is widely used in web applications because of its JSON data model and JavaScript API.
  - NodeJS. This is a container for running Javascript applications. It is widely used in web applications.
  - MongooseJS. This is a JavaScript binding for MongoDB that is designed to operate within a NodeJS container.

Course Overview

JavaDB/Derby

- JavaDB/Derby is a relational database management system (RDBMS) maintained by the Apache Software Foundation.
- Derby is a pure Java implementation of an RDBMs that can be embedded in an application or run as a server. There are a few non-standard features and features not implemented.
- Derby functionality includes:
  - Embedded engine with JDBC drivers
  - Network Server
  - Network client JDBC drivers
  - Command line tools: ij (SQL scripting), dblook (schema dump) and sysinfo (system info)

Course Overview

JavaDB/Derby

- Download the latest version of Derby at
  https://db.apache.org/derby/derby_downloads.html
- On MacOS, and Linux, recommend creating /usr/local/Apache and unpacking the ZIP file into this directory.
- On Windows, recommend creating C:/Apache and unpacking the ZIP file into this directory.
- In Eclipse or other IDE, import required derby JAR files from the release lib directory.

Course Overview

MySQL/MariaDB

- MySQL is an open-source relational database management system (RDBMS) that was developed by MySQL AB, and was recently acquired by Oracle
- It is written in C and C++. Its SQL parser is written in yacc, but it uses a home-brewed lexical analyzer.
- The server software itself and the client libraries use dual-licensing distribution: GPL 2 and proprietary.
- MariaDB is a community-developed fork of MySQL, intended to remain free under the GNU GPL.
- Some of the original developers forked the code due to concerns over its acquisition by Oracle Corporation.

Course Overview

MySQL

- Download the latest version MySQL Community Server
  - https://dev.mysql.com/downloads/
- You will need the MySQL community server and the MySQL JDBC Connector/J.
- On MacOS, MySQL can also be installed with HomeBrew:
  - https://gist.github.com/nrollr/3f57fc15ded7dddddcc4e82fe137b58e

Course Overview

MariaDB

- Download version of MariaDB at:
  - http://go.mariadb.com/search-download-MariaDB-server.html
- Download the MariaDB version of Connector/J at
  - https://mariadb.com/kb/en/library/about-mariadb-connector-j/
- On MacOS, MariaDB can also be installed with HomeBrew:
  - https://mariadb.com/kb/en/library/installing-mariadb-on-macos-using-homebrew/

Course Overview

MongoDB

- MongoDB is a free open-source cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents and schemata.
- MongoDB is developed by MongoDB Inc., and is published under a combination of the GNU Affero General Public License and the Apache License.
- MongoDB provides a JavaScript API for interacting with the database. It is a popular choice for web programming.

Course Overview

MongoDB

- Download version of the MongoDB Community Server at
  https://www.mongodb.com/download-center#community
- On MacOS, MongoDB can also be installed with HomeBrew:
  https://docs.mongodb.com/manual/tutorial/install-mongodb-on-os-x

Course Overview

NodeJS

- Node.js is an open-source, cross-platform JavaScript run-time environment that runs JavaScript code outside of a browser.
- Historically, JavaScript was used primarily on the client-side with JavaScript are embedded in a webpage's HTML and run client-side by a JavaScript engine in the user's web browser.
- Node.js lets developers use JavaScript for server-side scripting to produce dynamic web page content before the page is sent to the user's web browser.
- Node.js represents a "JavaScript everywhere" paradigm, unifying web application development around a single programming language.

Course Overview

NodeJS

- Download version of NodeJS at

  https://www.nodejs.org/en/download

- On MacOS, NodeJS can also be installed with HomeBrew. Here is an article describing the process.

  https://medium.com/@kkostov/how-to-install-node-and-npm-on-macos-using-homebrew-708e2c3877bd

- NodeJS includes its own package manager, *npm*, for installing supporting packages into a NodeJS instance.

Course Overview

MongooseJS

- Mongoose provides JavaScript API for MongoDB, written in node.js. It provides management policies that are not provided directly by MongoDB, similar to provided by SQL:
  - nested, typed schemas
  - flexible schema field specs
  - restricted values for fields
  - schema for allowing arbitrarily nested documents.
  - transactional integrity

Course Overview

MongooseJS

- Install MongooseJS using the NodeJS *npm* (NodeJS Package Manager).

  https://mongoosejs.com/docs/

Course Overview

What is a Database?

- A database is an organized collection of data, stored and accessed electronically.
- Database designers organize the data to model aspects of reality in a way that supports processes requiring information.
- An example is modelling the availability of rooms in hotels in a way that supports finding a hotel with vacancies.

Course Overview

What is a Database Management System?

- A database management system (DBMS) is software that interacts with end users, applications, and the database itself to capture and analyze data.
- A general-purpose DBMS allows the definition, creation, querying, update, and administration of databases.
- A database is generally stored in a DBMS-specific format which is not portable, but different DBMSs can share data by using standards such as SQL and ODBC or JDBC.
- The sum total of the database, the DBMS and its associated applications can be referred to as a "database system".

Course Overview

What is a Database Management System?

- Computer scientists may classify database-management systems according to the database models that they support. We will be studying two major types.
  - *Relational databases* became dominant in the 1980s. These model data as rows and columns in a series of tables, and the vast majority use SQL for writing and querying data.
  - *Non-relational databases* became popular in the 2000s. The are also referred to as *NoSQL* because they use different query languages.

Course Overview

Terminology and Overview

- A "database" refers to a set of related data and the way it is organized.
- Access to this data is usually provided by a "database management system" (DBMS) consisting of an integrated set of computer software.
- A DBMS allows users to interact with one or more databases and provides access to the data contained in the database (restrictions may exist that limit access to particular data).
- The DBMS provides various functions that allow entry, storage and retrieval of large quantities of information and provides ways to manage how that information is organized.

Course Overview

Terminology and Overview

- Existing DBMSs provide functions that allow management of a database and its data which can be classified into four main groups, which are also subsystems in most DBMSs:
  - **Data definition** – Creation, modification and removal of definitions that define the organization of the data.
  - **Update** – Insertion, modification, and deletion of the actual data.
  - **Retrieval** – Providing information in a form directly usable or for further processing by other applications.
  - **Administration** – Registering and monitoring users, enforcing data security, monitoring performance, maintaining data integrity, dealing with concurrency control, and recovering information that has been corrupted by some event such as an unexpected system failure.

Course Overview

Terminology and Overview

- Physically, database servers are dedicated computers that hold the actual databases and run only the DBMS and related software.
- Database servers are usually multiprocessor computers, with generous memory and RAID disk arrays used for stable storage.
- DBMSs may be built around a custom multitasking kernel with built-in networking support, but modern DBMSs typically rely on a standard operating system to provide these.

Course Overview

History

- 1960s – Navigational DBMS based on B-trees to provide navigational paths through the data.
- 1970s – Relational database systems (RDBMS) based on work by Edgar Codd at IBM San Jose based on relational algebra
- 1980s – Object-Relational mapping bridged the gap between object-oriented languages and relational databases.
- 1990s – Object-oriented databases that directly implemented object-oriented storage models for engineering and science.
- 2000s – NoSQL and NewSQL took a document-oriented storage approach based growth of XML and JSON.

Course Overview

Plan of Study

- Beginning with lecture 2, we will begin exploring relational databases and the relational model.
- We will consider some of the basic principals of data management that are foundational to databases in general and to relational databases in particular.
- We will also look at common software architectures that are used to create data-centric applications.
- Finally, we will begin to explore the relational model that Edgar Codd developed and the underlying relational algebra on which it is based.