Northeastern University

**Course Syllabus**
CS6020

# CS6020 – Collecting, Storing, and Retrieving Data

Spring 2015


**Instructor:** *Martin Schedlbauer, Ph.D.*
**E-Mail:** *m.schedlbauer@neu.edu*
**Phone Number:** *617.373.2229*


## Required Texts

*Eric Redmond, Jim R. Wilson. Seven Databases in Seven Weeks: A Guide to Modern Databases and the NoSQL Movement Paperback. Pragmatic Bookshelf, 2012. ISBN 978-1934356920.*

*Abedin, Jaynal. Data Manipulation with R. Packt Publishing, March 2015. Available as a Kindle Edition or as paperback.*

*Churcher, Clare. Beginning Database Design. Second Edition. Apress, 2012. Available as a Kindle Edition or as paperback.*

Additional online resources and free electronic versions of books provided on Blackboard.

## Course Prerequisites

An undergraduate course in statistics is required. Exposure to modern programming languages is helpful.

## Course Description

In this course students will learn how to build large-scale information repositories of different types of information objects so that they can be selected, retrieved, and transformed for analytics and discovery, including statistical analysis. Students will become knowledgeable about traditional approaches to data storage and how they can be applied alongside modern approaches that use non-relational or NoSQL architectures. Through case studies, readings on background theory, and hands-on experimentation, students will have the opportunity to learn how to select, plan, and implement storage, search and retrieval components of large-scale structured and unstructured information repositories. In particular, students will be conversant in the tools and techniques used to assess and recommend efficient and effective large-scale information storage and retrieval components that provide data scientists with properly structured, accurate, and reliable access to information needed for investigation.

Northeastern University

**Course Syllabus**
CS6020

## Course Outcomes

After completing this course, students will be able to:

- Classify information and data storage approaches based on object type and retrieval requirements
- Select an appropriate information storage structured depending on object type and analysis goals
- Plan an information repository for data analysis and discovery
- Collect data from online sources using R
- Clean and transform data into effective storage structures in R
- Discuss how NoSQL databases can be applied to store and retrieve unstructured data
- Complete simple implementations of structured and unstructured data repositories using R
- Use SQL to store and retrieve data from relational databases
- Transform data objects into representations that can be transferred to data analysis platforms
- Distinguish between storage needs for statistical and non-statistical analysis of data
- Outline tiered information architectures for efficient data retrieval and search
- Apply knowledge from case studies to select, plan, and implement information repositories

## Learning Assessment

Achievement of learning outcomes will be assessed and graded through:

- Contributions to case study discussions, code walkthroughs, and topic discussions (10%)
- Completion of small to medium exercises involving scripting in R, NoSQL, and relational databases (50%)
- Completion of a term project in which students will analyze a specific application domain, plan an information repository, explore a "big data" technology, or implement a prototype of a data repository (30%)
- Completion of online tests ensuring progression through the lessons (10%)

## Schedule

| *Week* | Topic | Assignments |
|---|---|---|
| *1* | Introduction & Essential Concepts in Big Data and Data Science | Readings; Short Reports & Questions |
| *2* | Programming in R | Short Programming Exercises in R; Setting up GitHub; Loading Files |
| *3* | Basic Data Shaping | Shaping and wrangling data in R |
| *4* | Data Import | Loading text files; processing XML |
| *5* | Data Collection Practicum I | Collecting data from files and XML |

| | | |
|---|---|---|
| *6* | Data Collection through Web Scraping | Collect data from web sites by parsing HTML |
| *7* | Data Collection Practicum II | Collecting data from HTML |
| *8* | Data Collection through Web APIs | Collecting data from web APIs |
| *9* | Break | |
| *10* | The Relational Data Model | Installing relational data engines; designing relational data stores |
| *11* | Relational Data Storage | Implementation of relational data stores in R through mySQL, SQLite |
| *12* | Data Retrieval via SQL | SQL programming |
| *13* | Key-Value & Columnar Databases | NoSQL databases I |
| *14* | Document & Graph Databases | NoSQL databases II |
| *14* | Storage Design | Storage Design & Term Project |

## Course Methodology

Each week, students are expected to:

1. Review the week's learning objectives
2. Complete all assigned readings
3. Complete all lessons for the week
4. Participate in online discussions and collaborative code walks
5. Complete and submit all assignments and assessments by the due date
6. Complete the weekly quiz by the due date

## Participation/Discussion Board

Interaction occurs primarily through the Blackboard discussion board and Blackboard Collaborate. Each week, students are expected to:

- Post their questions in the discussion board
- Respond or comment on other students' posts
- Join the online interactive sessions

## Communication

Communication between instructor and students is through

- E-mail via the Blackboard distribution list
- Announcements posted on Blackboard
- Notes posted on the Blackboard discussion board
- Private email exchanges
- Google Hangout for private communication

## Submission of Work

All work for the course is expected to be completed by the due date and time and must be submitted in the Assignments folder. No email submissions are accepted. In the Assignments folder, click on the Assignment link to view an assignment. Attach your files or documents along with explanatory comments and click Submit to turn them in. Once an assignment has been graded, students will be able to view the grade and feedback by clicking on My Grades.

## Academic Integrity Policy

The University views academic dishonesty as one of the most serious offenses that a student can commit while in college and imposes appropriate punitive sanctions on violators. Here are some examples of academic dishonesty. While this is not an all-inclusive list, we hope this will help you to understand some of the things instructors look for. The following is excerpted from the University's policy on academic integrity; the complete policy is available in the Student Handbook.

*Cheating* – intentionally using or attempting to use unauthorized materials, information or study aids in an academic exercise

*Fabrication* – intentional and unauthorized falsification, misrepresentation, or invention of any data, or citation in an academic exercise

*Plagiarism* – intentionally representing the words, ideas, or data of another as one's own in any academic exercise without providing proper citation

*Unauthorized collaboration* – instances when students submit individual academic works that are substantially similar to one another; while several students may have the same source material, the analysis, interpretation, and reporting of the data must be each individual's independent work.

*Participation in academically dishonest activities* – any action taken by a student with the intent of gaining an unfair advantage

*Facilitating academic dishonesty* – intentionally or knowingly helping or attempting to violate any provision of this policy