CIS 450/550: Database and Information Systems
SPRING 2017-Course Project

PROJECT OVERVIEW

The goal of this project is to create an application of your own choosing over any dataset. You may choose any publicly available dataset, you may also find supplemental data sources to complement it.

As part of developing this project, you will exercise: schema design, cloud hosting, data extraction, entity resolution, SQL queries, NoSQL queries, query optimization and performance considerations. The project is designed to offer flexibility in choosing features to implement and to think about how to make these implementations efficient.

You can develop any application/website/game, so be creative! The intent of the project is to help you

- understand the importance of good database design
- to introduce you to techniques for extracting information from partly structured sources
- to learn database and application hosting on Amazon Web Services platforms

You should build your application using a relational database technology and augment your basic design with non-relational (NoSQL) technologies.

Please remember you are allowed to use any available dataset but you must clear the idea with your assigned TA first before moving ahead. You must also make sure you meet all the project requirements listed under section 2 of this handout.

PROJECT REQUIREMENTS

Your project should have the following features:

- Information drawn from at least 3 datasources
- Diversity of data types: Datasets might be available in many formats e.g. CSV, .xls, .json, .xml, HTML, Pdf make sure you use and extract data in at least 2 different formats.
- Data cleaning: frequently, online data is incorrect and incomplete. As you import the data, you should run simple scripts to check for errors.
- Entity resolution: Since you are importing data from multiple sources, you will need to "connect" the data. Note the datasets might be from different sources and field values may not match or that there may be misspellings of names (data cleaning issues!).
- Normalized relational schema with more than four relations.
- More than one interaction page with the database.
- Complex SQL queries, i.e. with multiple joins, subqueries, aggregation etc. These can be in the application queries, or part of the schema design (e.g. triggers).
- Consideration of performance, including indexing. (See experimental validation requirement in Milestone 4.)

EXTRA CREDIT

- Trigger Bing Search, see http://datamarket.azure.com/dataset/bing/search, to return additional information.
- Import login and user information from Facebook, Twitter, Google, or Microsoft authentication services.
- View-based access control (e.g. if personal data is part of the application and there are privacy concerns)
- Adding streaming data (e.g. from Twitter feeds)
- Anything else you think is intuitive and adds interest to the application.

PROJECT MILESTONES

Milestone 1 : Due Date- 02/16/2017

Form a team (size 3 or 4), develop initial idea, and set up infrastructure. The initial step is to select your teammates and do the following:

1. Develop an initial idea, and determine the technologies you wish to standardize on as a group. Amazon Web Services should be used for hosting your database and deploying your application.

2. Provide 6-10 questions (in English) that someone might want to ask about the domain of your intended application. (You will be permitted to revise these questions later if needed.)

3. Setup Subversion/Git to share source code and starter data files. See http://www.seas.upenn.edu/cets/answers/subversion.html for details, and be sure that whoever sets it up grants access to everyone in the group. You should also add your assigned TA and Professor Naik to it so we can see what you are doing.

4. One group member should then upload a PDF document via Canvas, stating who is in the group, what your initial idea is, and what technologies you plan to use. The document should also include a timeline for the different milestones of your project, and a preliminary division of responsibilities.

Based on this description, we will assign each group a TA who will follow your progress throughout the remainder of the semester. You should consult them early and often, and get their input as you refine your ideas of the features to be implemented in your project (Milestone 2). Each group member should apply for AWS Educate, which grants $100 in usage credits. You need to use your .upenn.edu email address to register. The approving process may take about one week, so apply early! With a total group amount of about $300, you should have enough to complete the project. However, if you exceed this amount through carelessness you will be responsible for overages. By this, we mean that you should turn off instances whenever they are not being used and NEVER publicly share your id and password or put them somewhere that they can be compromised. We have had multiple incidents in the past where AWS Keys were shared publicly, the accounts were hacked and a bill of more than 1000$ was generated.

NOTE: Please be careful to remove your Database Access credentials before pushing the code to GitHub. GitHub provides private repositories to student accounts you may use that too.

Milestone 2 Due Date- 03/16/2017

Project outline and schema design.

In this phase, you will explain your project idea in more detail, how the datasets shall be used and features that will be implemented. Your project idea should contain the following:

- Motivation for the idea
- Features that will definitely be implemented in the application
- Features that might implemented in the application, given enough time

● Technology and tools to be used

● Description of the complimentary sources you intend to use for data, and how you intend to ingest the data into your database

● Member responsibility for project components

It is important to establish early on specific project component responsibilities – each group member should have aspects of the project that they "own" and are responsible for. "Own" does not necessarily mean they will be doing all the coding / development, but rather that they are responsible for making sure the feature is complete.

You should also design a relational schema for your application, as well as a description of the NoSQL component, if you choose to add one. **Your schema should be based on the application rather than a straightforward copy of the dataset provided**. The relational schema should be represented as an ER diagram, as well as through (normalized) SQL DDL. For the web technology, we prefer you to use Node.js – however, if your team feels this is too difficult, then you may use something simpler (e.g. PHP). **It is most important for you to be able to complete and present the project at the demo**.

For this milestone, you should submit a file with the information above, along with the relational schema and NoSQL description.

Milestone 3 Due Date- 03/28/2017

Populate the database.

Now that you have a baseline schema to work on, the next part is to populate the database. You must extract the data from datasets and import them into your databases.

You should clean and format data from your main and complementary sources, as needed, and perform entity resolution. For each of your 6-10 questions (from Milestone 1) provide its translation to an SQL query. (If you needed to change any of your original questions, provide the new questions in English and explain why you needed to change or replace them.)

You should use the AWS Getting Started handout to create your own MySQL (cheaper!) or Oracle database on Amazon RDS. For the milestone you should submit a text file with a full JDBC/SQLPLUS connect string, including guest user ID and password and database schema name, to us via Canvas. (From this we should be able to dump your SQL tables.)

Milestone 4 Due Date- 04/06/2017

Demo basic functionality.

In this phase, you should have a running application with some basic features. Submit the source code and a brief document of the list of features through Canvas; you should also set up a time to demo what you did to your overseeing TA to get their feedback.

Final Milestone

Project Demonstrations Date- 04/27-28

Your final demo should contain all the basic and/or advanced features mentioned in your report along with any extra credit implemented. You should also give the instructors an updated copy of your project description (Milestone 2) prior to starting the demo.

FINAL DELIVERABLES TO BE SUBMITTED

A. EXPERIMENTAL VALIDATION AND REPORT

A modern software infrastructure project isn't done until you understand how it performs, and where the bottlenecks are. Instrument your application to collect timings on various aspects. You should at least be able to determine what the latency in handling each request is, and extra credit will be awarded if you can also see what happens under multiple concurrent requests. Your final report should include a write-up of:

1. Introduction and project goals

2. Basic architecture (not a dump of the classes)

3. Key features of the project.

4. Technical challenges and how they were overcome

5. Description of your complementary data and how you extracted it

6. Performance evaluation

7. Potential future extensions

B. CODE DUE DATE: 04/30/2017

The entire project code along with the final report should be zipped and submitted on Canvas.

DATASETS

You're open to use any publicly available data. You may use any dataset that is directly available on the internet or you can scrape them of the web to suit your application needs. But depending on your application idea, you will need to extract at least one of the datasets to make your application more informative (and interesting). You will need to use content extractors (e.g., the Jackson JSON parser, the Tika reader for many file formats or Beautifulsoup for Python) that read the contents of the raw data items, interpret the file format, and extract the required information. You will likely want some of the following libraries to extract data from (partly) structured files, such as html (Wikipedia, World Factbook, DBpedia), CSV, XML, JSON, and text files.

- tika.apache.org (reads Word docx, PDF, … text and headers)
- github.com/FasterXML/jackson(reads JSON)
- commons.apache.org/proper/commons-csv/ (reads comma separated)
- jsoup.org/ (reads html)
- www.crummy.com/software/BeautifulSoup/ (reads HTML)

Here are some examples of data sources:

- Wikidata, https://www.wikidata.org/wiki/Wikidata:Main_Page
- DBpedia, http://wiki.dbpedia.org/
- Wikipedia, https://en.wikipedia.org/wiki/Main_Page
- World Bank Open Data, http://data.worldbank.org/
- World Factbook, https://www.cia.gov/library/publications/the-world-factbook/
- tableau: https://public.tableau.com/s/resources?qt-overview_resources=1
- openflights: http://openflights.org/data.html
- Greatest Sports Nation, http://www.greatestsportingnation.com/
- sqlbelle: https://sqlbelle.com/2015/01/16/data-sets-for-bianalyticsvisualization-projects/
- Github: https://github.com/caesar0301/awesome-public-datasets
- Yelp: https://www.yelp.com/dataset_challenge
- fivethirtyeight :https://github.com/fivethirtyeight/data
- http://data.philly.com/
- https://nycopendata.socrata.com/
- http://www.zillow.com/research/data/

However, there are many other sources of interesting data that you may choose instead.

SAMPLE PROJECT IDEAS

- Music Suggestion and Similarity app:

Using the data available in cp.jku.at/datasets/musiclef/index and labrosa.ee.columbia.edu/millionsong/ you may build a music suggestion app. The datasets contains information about Songs, albums, artists. It also contains tags associated with each song. You may use the concepts used in the course to design a Schema that make songs suggestion based on your inputs.

- World Bank database Factbook:

Using the data available in data.worldbank.org/data-catalog/world-development-indicators . This dataset is a compilation of development indicators, compiled from officially-recognized international sources. It presents the most current and accurate global development data available, and includes national, regional and global estimates. You can build an interface that lets the users choose the countries or regions on select indicators. You can them display the obtained data using interesting visualizations

- World Travel Guide:

Using https://code.google.com/archive/p/worlddb/,https://www.maxmind.com/en/free-world-cities-database You can build your own database that contains data scraped from the around the web. You may then build an interface that lets the user choose features, to see the list of places that satisfy their travel needs.

- Soccer Fantasy Team:

You may even you github.com/jokecamp/FootballData along with your own extracted data to build a Soccer fantasy league. You can build interfaces that lets user choose and build their own team. You can implement user and roles based features learnt from the course providing different levels of features to different users.

Plagiarism Policy

You can refer to web or any other resource for ideas, but you are STRICTLY NOT ALLOWED to use other people’s code directly. In case you would like to use some code or snippets, please consult your mentoring TA before you do so. Please make sure that you cite the original author/source if you are approved to use it. If you are caught under Plagiarism, academic measures will be taken as directed by: http://gethelp.library.upenn.edu/PORT/documentation/plagiarism_policy.html.