CIS 450/550: Database and Information Systems

Course Project Fall 2019

Table of Contents

A. Overview

1. Description

The goal of this project is to identify at least two large, overlapping datasets of interest, import and integrate them into a relational database, then create a web-enabled application of your own choosing over the database. You will work in teams of four, and you can use any relational database technology and any web-development stack.

Our goal for the project is to test whether you're able to apply many of the concepts you're learning about in lecture to solve unstructured problems with limited hands-on guidance. Specifically, the project will require you to practice:

- wrangling and cleaning data,
- performing entity resolution,
- designing schema,
- writing SQL queries,
- choosing database indexes,
- and optimizing SQL queries.

If you use this as an opportunity to build something you're proud of, you can use the final product to demonstrate your skills to employers.

We also want you to enjoy completing the project. This is one of the most open-ended projects in the Penn computer science curriculum. You can build just about any game, application, or website you want, as long as it demonstrates your database skills (Grading section). For example, past students have built trading card games, recipe recommendation platforms, and many other interesting apps. So have fun with it and be creative! Check out the resources section for more inspiration.

In the remainder of this introduction section, we list the deliverables you'll produce while completing the project and call out important dates/deadlines for you to save in your calendar. In the next section, we describe each milestone in detail. In the third section, we explain our grading criteria and identify several sources of extra credit. In the final section, we provide several resources that might benefit you, including dataset sources, descriptions of past students' projects, and links to useful tools.

As always, we're available on Piazza and in office hours to answer your questions. You will also be assigned a project mentor who will be your primary point-of-contact on staff for questions about your project. Good luck and have fun!

2. Deliverables

The table below provides a brief description of each deliverable you'll need to submit to complete the project. Please see the milestone section for more detailed instructions.

Deliverable Name	**Description**	**Milestone**
Project Proposal	A 1-2 page document that identifies the datasets you will use, gives a rough idea of what your application might do, and demonstrates you have performed some basic descriptive analysis on the data.	Milestone 1
Project Outline	A 2-3 page document that describes what your application will do, explains its significance, and gives the schema your database will implement.	Milestone 2
SQL Queries	A list of SQL queries that can run on your database with short explanations of what each query is supposed to do	Milestone 3
Final Report	A 5-8 page document that thoroughly describes the problem you tried to solve, your application functionality, your database design, your query optimization efforts, and more.	Milestone 5
Application Code	A zip file that contains your application code, a list of dependencies, instructions for building the application and any code you wrote to populate the database or wrangle data.	Milestone 5
Application Demo	A 2-4 minute screen-captured video that demonstrates your application's main functionalities and includes narration from at least one group member.	Milestone 5
Final Presentation	A slide deck you will present to several other project groups and several course staff members that gives background on the project (e.g. descriptions of problem, functionality, tech stack, etc.).	NA

3. Important Dates

The table below highlights important dates for the project. We recommend you save these in your calendar. But we will certainly post reminders on PIazza before each date and deadline.

Event	**Date**	**Description**
Project Introduction Recitations	**9/26-9/27**	Recitation where TAs we will discuss project deliverables, give advice based on their experiences, and answer your questions
Milestone 1 Deadline	**10/10 11:59PM**	Deadline for submitting the project proposal and forming groups
Milestone 2 Deadline	**10/25 11:59PM**	Deadline for submitting the project outline
Milestone 3 Deadline	**11/7 11:59PM**	Deadline for populating database and submitting list of queries for the database
Milestone 4 Deadline	**11/25 11:59 PM**	Deadline for meeting with project mentor to demonstrate implementation of some basic functionality
Milestone 5 Deadline	**12/9 11:59 PM**	Deadline for submitting video application demo, application code, and final report
Final Presentations	**12/10-12/13**	Days where final presentations will take place.

B. Milestones and final presentation

1. Milestone 1 - Project Proposal (Oct 10)

Form a team of 4 and develop an initial idea.

The initial step is to select your teammates, then brainstorm ideas applications and search for datasets. We will provide guidance on how to select a dataset and application idea in the project introduction recitation (9/26-9/27). **Remember your datasets must be:**

1. **Large** -- As a rule of thumb, your final database should contain on the order of tens of thousands of instances or more (depending on the size of each instance).
2. **Overlapping** -- Your application will need to include queries that require information from both datasets. This means the datasets need to be related and probably need to contain references to the same entities.

Before finalizing your dataset choices, you should conduct some basic exploratory analysis. Poke around, compute summary statistics, and try to get a sense of how clean, large, and complete the data is. These factors will affect how difficult it will be to clean/pre-process the data, create and query a database, and perform entity resolution later.

Next, write your project proposal. The project proposal should contain the following:

1. A List of group members and email addresses
2. A Description of application/website idea
3. For each dataset you've chosen:
    a. A 1-2 sentence description of the dataset
    b. A link to where you found the dataset
    c. If you're scraping the data, a description of how you will scrape it
    d. If you're not scraping the data, relevant size statistics (e.g. For a table, mb/gb, number of rows, and number of attributes. For a graph, mb, number of nodes, and number of edges)
    e. If you're not scraping the data, summary statistics of several attributes (e.g. report mean, standard deviation)
4. A list of at least 5 queries (in natural language) you could write for your datasets. Some of these should require complex SQL (aggregations, subqueries, joins, etc.)

Submission Instructions: One member should upload the proposal as PDF to Gradescope and add the other three members of the group to the submission as teammates.

Based on your project proposal, we will assign each group a TA who will serve as your project mentor for the remainder of the semester. Your project mentor will email your group with feedback on your proposal by Oct. 17.

2. Milestone 2 - Project Outline (Oct 25)

Detailed functionality description, schema design, and initial set up.

In this phase, you will set up your version control environment, explain your project idea in more detail, and assign responsibilities to each group member.

First, have one group member create a private Github repository to share code and data files.[1] Be sure to share the repository with everyone in your group and your project mentor before the milestone deadline.

Next, write your project outline. The outline should contain the following:

1. Motivation for the idea/description of the problem the application solves
2. List of features you will definitely implement in the application
3. List of features you might implement in the application, given enough time
4. List of pages the application will have and a 1-2 sentence description of each page
5. Relational schema as an ER diagram
6. SQL DDL for creating the database
7. Explanation of how you will clean, pre-process, and ingest the data into the database
8. List of technologies you will use
9. Description of what each group member will be responsible for

Submission Instructions: One group member should then upload the project outline to Gradescope as a PDF and add all other group members to the submission as teammates

If you choose to use AWS: 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 had an incident a few years ago where hackers used the AWS Keys and deployed several EC2 instances and a bill of more than $1000 was generated.

[1] Click here to register for the Github student development pack, which allows you to create unlimited private repositories for free.

3. Milestone 3 - Database Population (Nov 7)

Clean your data, perform entity resolution, and populate the database.

Now that you have a schema, populate the database with the datasets you have chosen. You should clean and format your data, perform entity resolution as needed, then ingest the data.

Next, create a .txt containing the following:

1. **Queries** -- A list of 5-10 SQL/NoSQL queries for your database that you may want to run in your application. Ensure that some of your queries are complex (e.g. involve interesting subqueries, joins, aggregations, etc.)
2. **Descriptions** -- A 1-2 sentence description of what each query is supposed to do
3. **Credentials** -- If you're hosting your database on a cloud platform, instructions and guest credentials for accessing the database. (e.g. For AWS RDS, a full JDBC/SQLPLUS connect string, including guest user ID, password, and database schema name)

If you're using AWS: Refer to the AWS Getting Started handout on Canvas to create your own MySQL (cheaper but slower!) or Oracle database on Amazon RDS.

If you're not hosting the database on a cloud platform: Reach out to your project mentor by Nov 7 to schedule a meeting. You will need to demonstrate to your mentor that you have populated the database and that the queries you submitted work.

Submission Instructions: One group member should upload the .txt file of SQL queries to Gradescope and add all other group members to the submission as teammates

4. Milestone 4 - Project Mentor Check-In (Nov 25)

Demo basic functionality

Schedule a 15-minute meeting with your project mentor to demonstrate that you have implemented a portion of the necessary features you listed in the project outline.

You don't need to prepare a formal presentation for this meeting. Just plan to walk your project mentor through the features you've implemented.

Your mentor will give you feedback on the features you've built and indicate whether they think your team is on track to finish a polished application. If there are technical challenges your team has been unable to resolve, this is a good opportunity to ask your mentor for advice.

5. Milestone 5 - Final Report, Demo Video, and Code (Dec 9)

Finish the application, record a demo of it, and write your report

Create a zip file containing all the code for the application. In addition to application code, ensure the zip file contains:

1. All code used for cleaning, wrangling, and ingesting the data
2. A list of dependencies
3. If the application isn't hosted online, instructions for building it locally

Use screen capture software to record a 2-4 minute video demo of your application. The instructors will only watch the first 4 minutes of your video, so don't make it longer. Your video should include demonstrations of all your application's pages and features. One or more of your group members should narrate it and describe the queries and optimizations associated with each feature. Describe how much each optimization affected performance.

Record timings associated with your application's features before and after performing optimizations, such as caching or indexing. For example, record how long it takes to execute queries, respond to client requests, and execute any other common tasks.

Write a polished final report containing the following information:

1. **Title --** Name of your website/application (Be creative!)
2. **Introduction --** Explanation of project goals/target problem, description of application functionality, list of group members
3. **Architecture --** List of technologies, description of system architecture/application
4. **Data --** For each dataset, a link to the source, a description of the data, relevant summary statistics, and an explanation of how you use the data
5. **Database --** Explanation of data ingestion procedure and entity resolution efforts, ER diagram, number of instances in each table, normal form and justification
6. **Queries** -- Examples of at least 5 queries in your application and explanations of how they're used. Please report the queries you think are most complex.
7. **Performance evaluation** -- recorded timings, descriptions of events being timed, and explanations of why caching, indexing, and other optimizations improved timings
8. **Technical challenges** -- List of technical challenges and how you overcame them
9. **Extra credit** -- List of extra credit features you implemented

In most cases, the final report turns out to be about 6-10 pages. Feel free to copy text from your earlier submissions where appropriate, if you think it's sufficiently polished and clear.

Submission Instructions: One group member should upload the report PDF and the code .zip file to Gradescope and add all other group members to the submission as teammates. One group member should upload the video demo to Canvas.

6. Final Presentation (Dec 10 - 13)

Present your application to TAs and other students

Your project mentor will organize a 2 hour presentation session, where all of the groups he or she has been overseeing will give presentations about their application. Barring an irresolvable schedule conflict with an exam, we expect every member of your group to arrive at the start of this session and attend for the entire duration.

Your team will play your video demo and use a slide deck to present the following information:

1. **Title**
2. **Introduction --** Explanation of project goals/target problem, description of application functionality
3. **Data --** For each dataset, a description of the data and relevant summary statistics
4. **Database --** Explanation of data ingestion procedure and entity resolution efforts, ER diagram, normal form of database
5. **Architecture --** List of technologies, description of system architecture/application
6. **Query --** Explanation of your most complex query, the information it retrieves, and how the application uses it
7. **Performance evaluation --** pre/post optimization timings for several queries and descriptions of implemented optimizations
8. **Technical challenges --** description of 1-2 technical challenges and how you overcame them

Including your video demo, the presentation should last no longer than 15 minutes. You will be graded on the content as well as the clarity of the presentation, so be sure to practice several times in advance.

C. Grading

1. Criteria

We will not be disclosing the exact breakdown of the project grade, but the seven following criteria will be used to evaluate your project. The criteria are roughly in order of importance.

1. Technical Quality (Does the application solve a non-trivial problem or answer non-trivial questions? Is the application robust and functional? Are some of the queries complex? Is the database designed well? Is the schema normalized? Was entity resolution and data cleaning performed correctly? Does the application perform interesting joins between data from different sources?)

2. Scope (Does the application implement a sufficient number of features? Does the application have multiple pages? Does each page interact with a database? Are the datasets large?)

3. Optimization (Did the developers attempt to optimize their more complex queries using relevant optimization techniques (e.g. improving the query, caching, indexing)? Do the developers provide reasonable, correct explanations for why each optimization failed or succeeded?)

4. Clarity of presentation (Are documents written using clear and concise language? Are they easy to read? During the presentation and video demo, are the developers easy to hear and understand?)

5. Group Dynamics (Did all members of the team contribute to the project? Were all members of the team given the opportunity to contribute to the project? Did team members work cooperatively and support each other?)

6. Look and feel (Is the application visually appealing? Is the user interface easy to use and robust? Are presentation slides eye-catching and concise? Do documents appear organized and easy to read? Are diagrams and tables formatted in a clear manner?)

7. Code quality (Does code follow a consistent, easy-to-read style? Is the code well-documented? Is code organized into multiple classes, functions, and files as necessary?)

2. Extra credit and awards

Projects are eligible to receive extra credit worth up to 10% of the project grade.

Extra credit features

We may award extra credit for any of the following:

1. Using cloud hosting for the database and/or application server
2. Using NoSQL
3. Integrating with other applications (e.g. Trigger Bing Search to return additional information, or Facebook/Microsoft/Google to authenticate users and import user information)
4. Implementing security safeguards (e.g. password encryption, view-based access control, etc.)
5. Adding real-time streaming data (e.g. from Twitter feeds, weather sources, etc.)

We also reserve the right to award extra credit for any other feature not listed here that we think is exceptionally creative or technically complex.

Awards

In addition, Dr. Davidson will select several projects to receive awards at the end of the semester. Each project that earns an award will receive some extra credit.

The following awards will be distributed:

1. Best all-around application
2. Most beautiful application
3. Most technically complex application

D. Resources

1. Dataset Resources

We've compiled a list of datasets and dataset aggregators to help jumpstart your search for data. But don't feel compelled to choose something off this list. This is just the tip of the iceberg. There's a lot more data out there!

Dataset Aggregators

1. Wikidata
2. DBpedia
3. World Bank Open Data
4. Tableau datasets
5. sqlbelle's list of datasets and dataset aggregators
6. AwesomeData's list of public datasets
7. fivethirtyeight datasets
8. OpenDataPhilly
9. NYC OpenData
10. Zillow Datasets
11. AWS Datasets
12. Goolge Cloud Datasets
13. Reddit Datasets
14. Buzzfeed Datasets
15. Data.world
16. Kaggle
17. Federal Government Datasets
18. UCI ML Repository
19. Academic Torrents

Knowledge Bases

1. World Factbook
2. Wikipedia
3. Greatest Sporting Nation
4. Summer Olympics Medals

Datasets

1. OpenFlights
2. Yelp
3. NYC Taxis

2. Example projects

The list below gives a few examples of projects people have done in the past. We hope these help inspire your group and give you a better idea of what we're looking for. You can use the datasets, but must develop your own original application over the datasets using your own code.

World Bank database Factbook

The data made available by WorldBank presents the most current and accurate global development data available, and it includes national, regional and global estimates. A group of students used this data to make an application that displayed development data for selected countries/regions in interactive visualizations.

World Travel Guide

A group used travel datasets like Google's to build an interface that lets users see information about places they could travel to, based on their travel preferences and needs.

Soccer Fantasy League

A group used this soccer dataset along with data scraped from ESPN to build a Soccer fantasy league. They built interfaces that let users draft their own teams and implemented an account system to save teams and expose different features to different kinds of users.

Olympics App

A group used the Summer Olympics dataset augmented with data from the WorldBank to create an application that helped people learn more about each country, while viewing how well that country has performed at the olympics historically.

Bike rental routes

A group developed a web-app that allowed people to find routes between two points in NYC using a combination of bike, subway and walking. Platforms such as Google Maps assume that the user either carries their own bike or none at all. This project used Citi Bike database JSON and combined that with subway stations database of New York City and calculated the shortest distance using a combination of all three, allowing the user to pick up a Citi Bike and drop it off during the journey.

3. Useful Tools

We've compiled a list of software tools that you may find helpful for scraping, cleaning, and ingesting data or making your demo video. If you know of some convenient tool we didn't include, please let us know on Piazza!

Content Extractors

1. Apache Tika -- Extracts metadata and text from word documents, PDF, powerpoints, and many other types of files
2. Jackson Project -- Efficient, easy-to-use JSON parser for Java
3. Trail -- XML Parser for Java
4. jsoup -- HTML parser for Java
5. Selenium for Python -- A Python library for automating interactions with a web browser to perform web scraping or application testing
6. Beautiful Soup -- A Python library for extracting data from HTML and XML files

Data cleaning and exploration tools

1. Pandas -- A Python library for data analysis that provides DataFrame, a convenient data structure for storing and processing large amounts of data
2. Dora -- A Python library for cleaning data and performing exploratory analysis
3. JupyterLab -- Software that enables you to work with Jupyter notebooks -- documents that contain live runnable Python code and display code output inline

Screen Recording Software

1. ScreenRec -- Free screen recording software for Mac, Windows, and Linux
2. Open Broadcaster Software -- Free screen recording software for Mac, Windows, and Linux

Database Software

1. MongoDB Atlas -- Cloud database hosting service for MongoDB with 512 MB of free storage
2. MongoDB Compass -- GUI for MongoDB
3. MySQL Workbench -- GUI for MySQL

E. Plagiarism Policy

You can refer to the web or any other resource for ideas, but you are STRICTLY NOT ALLOWED to use other people's code directly. If you would like to use some code or snippets, please consult your project mentor and obtain permission 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 violating this policy, you will receive a 0 for the final project and/or the class, and we will refer your case to the Office of Student Conduct (OSC), which may take further disciplinary action.