ECE 356 Project

Meta Description

This document is a meta-description of the course project. It describes the requirements that are consistent for all projects, regardless of the particular dataset your project team is using. Details that are specific to each particular dataset will be provided within separate documents for the respective datasets.

Overall Description

The course project is a database-design and implementation exercise, together with a data-mining exercise. The starting point for this exercise will be a sizable dataset from a particular domain. The NHL game data is an example of such a dataset, and it contains 1.5 GB of data spread over 13 CSV files with over 100 distinct attributes. The datasets we have acquired for your projects are of similar or greater magnitude and complexity, and are in the following broad areas:

While the particular datasets we have found are of similar size and number of attributes as the NHL Game data, they tend to have far fewer CSV files. Specifically, if a "good" relational design was already evident within the CSV file mix then we tended to exclude that as a possible data source, as creating and justifying a good design is a significant component of the project.

Used Car Sales Data	2020 Election	Movie Data	Recipes
Kaggle Metadata	Traffic Accidents	Smart Meters	COVID19
Education Data	Parking Tickets	Stock Data	MLB
Books and Libraries	Internet Traffic	Demographics	Crime
Airline Arrival Data			

Your first task as a project team is to select your three preferred areas, in order of preference. We will use that information, and that of all of the other project-teams' preferences, to assign each project team a dataset. While we will do everything we can to accommodate your first-choice preference, we cannot guarantee that it will be possible, or even that we will be able to accommodate any of your preferred choices. In cases where none of your choices are possible, and the alternatives are not desirable to you, we are willing to entertain a limited number of proposals for alternate datasets. Any such proposal must identify the dataset source, which must be CSV files of similar or greater magnitude and complexity as the datasets we have selected, and not already be decomposed into a good database design. You must also provide a description of what will be done that is consistent with the overall project requirements. This must be submitted to the course instructor who will either accept it as is, require modifications to some aspects, or reject the proposal.

The most likely reason for an outright rejection of a proposal is if the data source is not commensurate with those being used by other project teams (too small, insufficient distinct attributes, already organized into a structured database). The most likely reason for requiring modifications of the proposal is that the dataset is acceptable but there are weaknesses in the requirements that is inconsistent with the requirements being placed on the other project teams.

There are numerous possible sources of very large amounts of data including, but not limited to, StatsCanada, the US Bureau of Labor Statistics, and, of course, Kaggle.

Once your project team has a confirmed dataset, you should study and understand the domain in the same way as was required in dealing with the NHL dataset.

The requirements for project, then, are as follows:

1. A command-line client application appropriate to the domain
2. An entity-relationship design to model the data

3. A relational schema based on the ER design
4. A data-mining investigation of the dataset

Client Application

The client application is required to be one that is appropriate to the dataset domain. It must allow for two key requirements:

1. Querying the data in a way that a customer in the domain would do
2. Modifying the data in a way that a customer in the domain would do

For example, the used-car sales data would need a client that allows a customer to search for used cars on some reasonable basis that a person looking for a used car would want: by year, by make and/or model, by price, *etc.* Likewise, a person should be able to list a car for sale, modify the listing to change the price and/or add additional information, and remove the listing once the car is sold.

If you wish to do a more sophisticated user interface you are welcome to do so, but you should be cautioned that (a) it will not affect your grade in the project; and (b) it will make creating testcases to demonstrate the quality of your finished product substantially more difficult.

The user-interface need only be a simple command-line interface, even with single-letter commands, as this aspect will not form any part of the grading scheme.

Specific requirements will be drafted in the per-dataset project requirements, but they will say little different than has just been described for the used-car scenario, other than that it will be appropriate to the particular dataset. It is expected of your projected team to work out an appropriate set of things that a user would wish to do, though allowing for the fact that this is a course project and therefore you should scope your project accordingly. In particular, you should decide as a team

1. What you think an ideal client *should* be able to do
2. What you plan to actually implement for your client given the time constraints
3. (At the end of the project, when you write your report) What you actually implemented from your plan, and what you left
4. An explanation justifying each of the above choices

Entity-Relationship Design

You are required to create an entity-relationship design appropriate to the dataset domain. Your design is required to clearly identify:

1. All entity sets, specifying the entity set name and attributes, showing any compound attributes, multivalued attributes, and optional attributes per the methods described in the course
2. All relationship sets, specifying the relationship set name and any attributes it might have
3. All primary keys, cardinality constraints, and attribute domains

4. Any weak, specialized, or aggregations
5. Any other aspects relevant to an ER design

You are required to create an ER diagram for your design, and explain why you choice the entity sets, relationship sets, *etc.* that you chose. Where appropriate, you shoud specify what alternatives you considered and explain why you chose the design that you did rather than the alternative.

We will be covering an approrpiate ER design for the NHL data in the coming weeks, so that you will understand what kind of design choices are available to you.

Relational Schema

You are required to translate your ER design into a relational schema. There are a number of places where there will be choices for you to make in this regard, and in those places you should explain why you made the choices that you made.

In converting your ER design into a relational schema you are expected to write the necessary SQL code to:

1. create the required tables, views, *etc.* for the relational schema
2. create the required primary keys, foreign keys, and integrity constraints
3. create indexes as necessary for the query operations you will do both in the client and in the data-mining exercise
4. load the data from your dataset CSVs into the tables

It is quite likely, as you have already seen with the NHL game data, that the data will contain errors and inconsistencies relative to your design. You are required to handle those issues in appropriate ways, including:

1. fixing obvious data errors
2. removing any duplicate data
3. modifying your design in certain cases

In any situation where you must handle such issues, you should document what you did to handle the issues.

In the coming weeks we will also be describing how we will transform our NHL ER design into a relational schema. While there will likely be similarities to the existing NHL database that you have used for assignment 1, there will be a number of differences owing to design errors in that data design.

Data-Mining Investigation

Given a large set of data, we can determine information from that data. Indeed, this is how all of science proceeds. Data represents facts. We wish to see if those facts allow us to formulate a theory about something, and to validate that

theory if possible. In this course we will teach you three specific techniques: classification, association discovery, and clustering. You will be required to implement one of those techniques and apply it to answering a question appropriate to the domain.

Specifically, you are required to

1. Select a domain-appropriate question that you want data mining to answer
2. Select a technique or techniques that will be appropriate to the question you are investigating
3. Implement said technique *efficiently* to build a data model
4. Determine the validity of your model
5. Report the results of your investigation

Assigned Instruction-Team Member

Each project team will be assigned to one member of the instruction team. A chat group will be set up within Piazza containing just the team members, but with instructor access. You should take advantage of this to ask questions specific to your project during the term.

Deliverables

There are no formal intermediate deliverables for this project. However, it is *strongly* recommended that as you go through the term you report progress to your assigned instruction-team member and request feedback from him/her. There are approximately 10 weeks remaing in the term, and therefore it is suggested you do so roughly every three weeks.

The final project deliverables are as follows:

1. Final Written Report: this should describe the client application, the ER design, the relational schema, and your data-mining investigation, detailing the specific issues required above. In addition, you should include a testcase plan that describes how you test the various code aspects of your project.
2. Code: you are not expected to submit your code but rather store it in a github repository, or equivalent, to which we can have access. The code in the repository should include the following:
   (a) All client code
   (b) The SQL code necessary to implement the relational schema and load the data from the CSV files
   (c) The code, SQL and otherwise, needed to implement your data-mining investigation
   (d) Test cases for the above
3. Video Demo: a 20-minute walk-through/presentation of your project. It should describe all of the aspects of your design, implementation, and results.