University of Toronto
CSC343, Fall 2020

Term Project

In the term project, you will use the skills developed in this course to investigate questions that you are curious about in a domain of interest to you. The project will be done in pairs.

Your project will culminate in a short presentation to course TAs, giving you a chance to develop and practise your presentation skills. To help you stay on track, you will hand in work at several points along the way.

Learning Goals

By the end of this project you should be able to:

- Investigate a non-trivial, real dataset to determine its precise semantics and choose a subset of the data that is sufficient to answer questions of interest.
- Use common-sense principles to design a relational schema capable of representing the dataset. This includes identifying alternatives and making trade-offs, and will leave you primed for learning a more principled approach to design later.
- Implement a relational schema using the SQL Data Definition Language.
- Clean a real dataset to prepare it for importing into a postgreSQL database.
- Use the SQL Data Manipulation Language to explore the dataset and answer questions about it.
- Present technical material to a technical audience.

Note: This is the first time we've had this project in csc343. Some details about the requirements may evolve as we progress through and I learn from the early phases.

Phase 1: Dataset and Relational Schema (due Thu 8 Oct, by 8pm)

Identify your domain and dataset

The purpose of the project is to give you practical experience answering real questions in a domain that you care about. Begin by identifying some domains that interest you. This could be anything, for example habitat preservation, small businesses, economic inequality, government spending on education, or climate change. With a couple of ideas in mind, search for open datasets related to your interests.

Here are some criteria that should influence your choice of dataset.

- Pick data that will be easy to retrieve.
  In Phase 3, you will retrieve the data. For now, just start thinking about how you will do that. You could get your data by writing code that uses an API to retrieve information from an online site, but this is beyond what is expected of you. It is fine to use data that has already been assembled, and is in a format that you know how to work with, such as csv or json, or even just a formatted text file.
- Pick a dataset that is easy to interpret.
  Sometimes a dataset comes with a "data dictionary" that explains the format and meaning of the data. Other times, that is defined poorly or not at all. It's possible to infer quite a bit about a dataset's meaning. For example, by reading the content of one table, it may seem that a certain attribute is a reference to something else, or that an attribute is a key. Once the data is loaded, you could test out these sorts of hypotheses by running queries to look for exceptions. This whole process can raise other uncertainties and end up being quite laborious, hence the advice to pick a dataset that is easy to interpret.

- Pick a dataset that is rich enough.
  It will take a bit of effort to find a dataset that has enough in it to do something interesting with. Here are some requirements for your dataset:
  - It must be open data.
  - It must not already be structured into database tables.
  - It does not have to be large in quantity (number of rows), but shouldn't be so small that you could answer your investigative questions just by looking at the data.
  - It must be rich in structure (number of columns and constraints). The final schema must have at least 6 tables and at least 4 referential integrity constraints.

Define your investigative questions

Now come up with 3 specific questions that you would like to answer using this dataset. Each question should be specific, not just a general area to explore. Aim to define questions that will require you to dig deeply into the data. Ideally, the answers you find would be of interest to someone whose work involves this data.

Design your schema

Then, design a relational schema for your domain, written using relational notation (like the schema in the Relational Algebra worksheet). There are many possible schemas for any interesting dataset, so you will have to make design choices. Later on, we'll learn a formal design process. For now, use your common sense and follow as many of these general principles as you can:

- Avoid redundancy.
- Avoid designing your schema in such a way that there are attributes that won't always have a value. For example, in a relation about students, we wouldn't want to have an attribute that identifies their spouse, since many or most students are not married. We would instead put that information in a different table, with a row only for those students who do have a spouse.
- Use constraints to prevent data that is clearly nonsensical from being included in your database.
- Define a key for every relation.

You may find there is tension between some of these principles. Where that occurs, use your judgment to make a trade-off. Keep a written record of important trade-offs and other decisions made; you will use that in your presentation.

Hand in a single file called `phase1.pdf` containing the following sections:

- Domain. The domain you have chosen for your project.
- Dataset. A description of this dataset including:
  - a link to the dataset that you have identified.
  - what information in it is relevant to your project (there may be lots of irrelevant extra data too)
  - any learning you will have to do in order to interpret the data
  - any cleaning up you think you will have to do in order to use the data
- Questions. Your three investigative questions that you plan to answer using this dataset. It's okay if these evolve as you explore the data.
- Schema. Your relational schema. It's also okay if this evolves over the phases of the project.

Phase 2: Schema Implementation (due Thu 29 Oct, by 8pm)

Now that you know the Data Definition Language for SQL, you are ready to implement your schema. Follow these guidelines:

- Define a primary key for every table.
- Wherever data in one table references another table, define a foreign key in SQL.
- Consider a NOT NULL constraint for every attribute in every table. It doesn't always make sense, but it usually does — especially if your design is good.
- Express any other constraints that make sense for your domain.

To facilitate repeated importing of the schema as you correct and revise it, begin your DDL file with our standard three lines:

```
drop schema if exists projectschema cascade; -- You can choose a different schema name.
create schema projectschema;
set search_path to projectschema;
```

Be sure that you can import your schema without errors.

Hand in the following:

- A file called `schema.ddl` containing the schema expressed in the SQL Data Definition Language.
- A file called `demo.txt` containing an example interaction with the postgreSQL shell showing you loading the schema successfully.

Phase 3: Data Cleaning and Import (due Thu 5 Nov, by 8pm)

In this phase, you will create the SQL statements necessary to import your data.

You will have learned how to insert a row into a table using an INSERT INTO statement such as this:

```
INSERT INTO Student VALUES (00157, 'Leilani', 'Lakemeyer', 'UTM', 'lani@cs', 3.42);
```

You could populate an entire database with a long series of these statements, however there is an overhead cost associated with executing a SQL statement, and you will incur that cost for every individual INSERT INTO. A more efficient approach is to use the postrgreSQL command `\COPY`. It lets you load all the rows of a table in one statement, so you incur the overhead cost only once, for the whole table. This is not only faster than INSERT INTO, it is also more convenient. You probably already have your data in a csv or formatted text file, and `\COPY` lets you load that data directly (rather than having to covert the data into a series of INSERT INTO statements). For instance, if you had data in a comma-separated csv file called `data.csv`, you might say:

```
\COPY Student from data.csv with csv
```

Cleaning the data

As you do the importing, you may find the data doesn't perfectly follow its specifications (or your guess as to what its specifications would be if someone had written that down). As a result, it may sometimes violate constraints that you have expressed. You will have to make decisions about how to handle this. For example, if a foreign key constraint is violated, you could remove the constraint so that SQL won't complain, keep the valid references where available, and replace the invalid references with NULL. If a NOT NULL constraint is violated, you might remove it. Or in either of these cases, you might decide to remove any rows that would violate. Of course this

affects that answers you will get to some queries and introduces questions about the validity of any conclusions you make. But that's okay. This is a database project, not a research project. The point is to learn about database design and implementation rather than to come to highly accurate conclusions about your domain.

One way to find the constraint violations is to define all the constraints, import the data, and watch the errors fly by. But an early error can influence subsequent errors, making the process laborious. An alternative is to omit some or all of the constraints from the schema at first, import the data, and then run queries to find data that would violate if the constraint were present. Once you have resolved all the issues, you can clear out the database, import the full schema with constraints, and then import the cleaned up data.

If the data is really huge, you may need to cut it down in order not to overload our database server. Be aware that this may violate some constraints, for example, if you remove data that is referred to from elsewhere. See above for how to deal with violated constraints.

Depending on your dataset, data cleaning can turn out to be a significant task. It is fine to pick a dataset that doesn't need a lot of this work. In any case, keep track of all the decisions you make and cleaning steps you take. You will talk about these in your final presentation.

Hand in the following:

- The data itself: details of what to hand in will be provided later.
- A file called `demo.txt` containing an example interaction with the postgreSQL shell where you (a) load the schema and data successfully, and (b) run a SELECT * query on each of the tables.
- A file called `decisions.pdf` containing a short description of the decisions you made and cleaning steps you took.

Phase 4: Queries and Results (due Thu 19 Nov, by 8pm)

Now comes the payoff for all your hard work! Write SQL queries to find the answers to your questions. This will likely be an exploratory process. As you discover things, you will have follow-up questions that may take you in new directions. This is fine. Enjoy exploring!

Hand in the following:

- A file called `queries.sql` containing the SQL queries you have written.
- A file called `demo.txt` containing an example interaction with the postgreSQL shell where you (a) load the schema and data successfully, (b) run a SELECT * query on each of the tables, and (c) run each query and show its result.
- A file called `discussion.pdf` containing a written discussion (at least one paragraph and up to 500 words) of what you learned about your domain.

Phase 5: Presentation (between Mon 23 Nov and Fri 4 Dec)

In the final phase, you will give a short presentation to two csc343 TAs, via Zoom. You will sign up for a time slot, which I expect to be roughly 15 minutes long.

Include one slide for each of these topics:

- Domain: The domain you explored
- Questions: The questions you set out to answer
- Results: What you learned about your domain

- Challenges and Lessons: One or two challenges you faced or lessons you learned about how to do this kind of work
- Questions?: Put this slide up while you answer questions from the TAs.

How your project will be graded

All phases before the presentation will be graded based on you having submitted all the pieces, and each of these being complete and showing a good effort. Marking resources are limited, but the TAs will try to give you feedback to let you know whether you are on the right track. I encourage you to use the course office hours to get any additional feedback or assistance you need in order to keep progressing successfully through the project.

I will say more about the rubric for the final presentation later. For now, keep in mind that these are the kinds of things we will be looking for:

- Insight into design tradeoffs. Did the team recognize alternative designs and make reasonable choices? Could they justify their choices?
- Insight into challenges faced. For instance, did an early design decision have negative consequences later? Did the team find a reasonable resolution and did they learn something from it about design?
- Clarity of presentation.
- Following the presentation requirements and timeframe.
- Answering questions well.

A great idea would be to start taking some notes on these issues from the very first phase of the project. You can pull on those when you put together the presentation and prepare for questions from the TAs.

About working in a pair

Once you have begun to work with a partner, please declare that partnership on MarkUs. In the unlikely event that your partner drops the course at some point during the term, you will be allowed to continue alone or to find another solo person to pair with, in which case you would choose one of the two projects to proceed with. Either way, you would contact us through the course account so that we can redefine your group in MarkUs.

Because this project will continue through the rest of the term, your choice of partner is important. I recommend that you consider these factors:

- Available times to work: You should aim to work together, at the same time, on a regular basis. This will be easier if you both like to work at similar times. Don't forget to factor in time zones!
- Work habits: Do you like to start early or do you tend to procrastinate? Do you like to work steadily over a long period of time, or put in a big push closer to the due date?
- Goals: Are you aiming to create something you will be very proud to show off, or just looking to get a decent mark?

I hope you enjoy this project and learn a lot through it!