University of Toronto
CSC343, Fall 2021

# Term Project

## Overview of the whole 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.

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.

## Phase 1: Dataset and Relational Schema

Partner declaration Due: Thursday, 23 September, by 8:00 pm
Phase 1 Due: Thursday, 30 September, by 8:00 pm

### 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 2, 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 very laborious. I strongly recommend picking 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. (Try Googling "open data xxx" where xxx is a location or a topic.)
  - 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: The final schema must have at least 4 tables and at least 3 referential integrity constraints. (You won't know whether or not your schema will have this many until you at least sketch out the schema.) Don't impose 4 tables on the data just to meet this criterion; division into tables should be for a reason, such as dealing with missing values or avoiding redundancy.

  If you find you don't have enough structure, think about whether there is some other data you could connect to what you have. For example, if you have dates, could you find another dataset that records the weather by date, and might the weather be relevant? Maybe you'll find that there is a relationship between the weather and the other data that you have. Or if you have postal codes, can you find out average household income or population density or education level by postal code? Use your imagination to identify other relevant data. Note that you are welcome to use multiple sources of data

### Define your investigative questions

Now come up with 3 specific questions that you would like to answer using your dataset(s). Each question should be specific, not just a general area to explore. But it should also be somewhat open-ended – we want the results of your first query to naturally lead to follow-up questions. In the end, each investigative question will be addressed by a *series* of queries.

We want 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. That is, avoid a schema that results in the same information being repeated. For instance, suppose the movies database we have discussed in class had a table like this:

| mID | title | director | year | length | actor | nationality | role |
|---|---|---|---|---|---|---|---|
| 1 | Shining | Kubrick | 1980 | 146 | Nicholson | American | Jack Torrance |
| 6 | American Graffiti | Lucas | 1973 | 110 | Ford | American | Bob Falfa |
| 5 | Star Wars IV | Lucas | 1977 | 126 | Ford | American | Han Solo |
| 5 | Star Wars IV | Lucas | 1977 | 126 | Fisher | American | Princess Leia Organa |
| 8 | Star Wars V | Lucas | 1980 | 126 | Ford | American | Han Solo |

  There is a lot of redundancy in this table. (Can you identify it?) It can be avoided by splitting the data into tables, as in the schema on the worksheet.

- Avoid designing your schema in such a way that there are attributes that won't always have a value, either because the data doesn't exist or is just missing in the dataset. 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; for them no spouse exists. We would instead put that information in a different table, with a row

only for those students who do have a spouse. Or suppose we have customers who we identify by their email address, and some of them have told us their phone number. We wouldn't want to include a column for phone number since we often have no value for it (even if a value actually exists). Again, we would instead put that information in a different table, with a row only for those customers whose phone number is known.

- Use constraints to prevent data that is clearly nonsensical from being included in your database. Don't forget this! Last year, quite a few students did.
- 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 Justification of your schema, and later in your presentation.

Your dataset may come organized into csv files (or in some other format) that can be translated directly into a relational schema that is already a good design. This is fine. However, you must justify why it is a good design nonetheless.

### Document your schema

There are 3 parts to documenting your schema:

1. Comments: Include a comment for every relation that explains exactly what a row in the relation means in your domain. I will provide a schema from an old assignment as a good example of this. Pick good attribute names that will make it easy to understand what each attribute is.
2. Data dictionary: In the example schema, where needed, we elaborate on the attributes in the relation comments. For the project schema, you will instead explain your attributes by defining a data dictionary, in tabular form, for each relation. For each attribute, it must include: the attribute name, a description of what it represents in your domain, its data type (you can give this in plain English and later translate it into a SQL type), whether or not a value will always be known (this should always be "yes" if you structure the schema well), a default value if one exists, and what are the allowable values. Figure 1 gives an example based on the Submission table from the example schema. Use the columns and content shown in figure 1, and include one of these for each relation in your schema.

   Your data dictionary will not only record your understanding of the data, but will help you think about the kinds of constraints that you will need when you implement the schema in SQL.
3. Justification: Provide a justification of why you divided the data into tables in the way that you did. If you translated the structure of the dataset directly without change, explain why the result is a good design. If you split things up that were together, or put things together that were apart, explain why. If you had to invent a key to uniquely identify items, explain this also. Include anything else that your TA will need in order to understand your choices.

Figure 1: Example of a data dictionary

**Submission**

| Attribute | Description | Type | Required | Default |
|---|---|---|---|---|
| sID | The ID of a file submission on MarkUS | INT | Yes | |
| fileName | The name of the file that was submitted | TEXT | Yes | |
| userName | The user name of the user who submitted the file. | TEXT | Yes | |
| gID | The group ID of the group for which the file was submitted | INT | Yes | |
| when | The date and time when the file was submitted | TIMESTAMP | Yes | |

### What to hand in

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(s) 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.
- Schema.
  - Relational schema
  - Data dictionary
  - Justification of design

## The Remaining Phases

The other parts of the project will have their own handouts, but to give you a sense of what to expect, they are outlined here.

### Phase 2: Schema Implementation and Data Cleaning

For phase 2, you will implement the schema from phase 1 in SQL. Then you will clean your data and import it into the database. Your schema may evolve between phases 1 and 2, based on what you've learned. This is welcome.

### Phase 3: Queries and Results

In this phase, for each of your investigative questions, you will write a series of SQL queries to answer it. Your questions may evolve between phases 1 and phase 3, and you may even decide to improve your schema at this point. This is all welcome.

### Phase 4: Presentation

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 20 minutes long. In the presentation, you will summarize your project, including the domain, data cleaning steps, queries and results, and challenges you faced.

## Resources for data cleaning

The pandas library for python may be very helpful for cleaning your data in phase 2. I can also help you now, when designing your schema, since it will allow you to view and query your data in ways that are not as easy to do with tools such as Excel. Here are some pointers to help you learn about pandas:

- To begin, look through the Getting Started page. There are tutorials included on the page that will teach you the basics of working with data using pandas dataframes. There are also guides that will help you understand how concepts from other tools, such as SQL and Excel, translate to pandas. Not everything in the tutorials will be relevant to what you need to do with your data so don't worry about trying to understand everything you read.
- The 10 minutes to pandas tutorial will give you a quick overview of important pandas concepts.
- The cheat sheet is a handy reference.

You are not required to use the pandas library, but it is helpful and well documented. Please remember that data cleaning is not the point of the project, and can be very time-consuming. I encourage you to pick a dataset that will require little to none of that.

## Thinking ahead to the presentation

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 tradeoffs and challenges from the very first phase of the project. You can use 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, you must declare that partnership on MarkUs. The deadline for doing so is Thursday, 23 September, by 8:00 pm. Anyone who has not found a partner by then will be assigned a random partner from among the remaining students.

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, 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.
- 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!