

Project - Climate Action Sample

IT 120 - Databases

Kit Transue

Contents

Online Version Differences	1
Support	1
Starting framework	1
Special Care	2
Project	3
Overview	3
Artifacts	5
Design	5
SQL	5
Create Database	7
To turn in:	7
Populate database	9
To turn in	9
SQL Query and Update	11
To turn in	11
Climate Change Database Sample Interview	13
Sample Documents	15
Members	15
Merchandise	15
Sales	15
Donations	16
Events	16
Resources	17
Comments	18
Recap of the Database Development Lifecycle	19
Installation refresher	19

Is database server running?	19
Listing databases	19
Dropping/creating a database	19
commandline	19
pgAdmin	20
SQL	20
Datatypes	20
Syntax	20
CREATE TABLE	20
Design-to-SQL	20

Online Version Differences

I haven't figured out how to make a collaboration space in this fully-online, totally asynchronous format, so I will try to provide much of the support a collaboration would have given you.

The good part about a group project is that no one should be overwhelmed by the work: someone will be there to support you. I will perform that role. Let me know how you're doing!

Something I like about this adapted format for the project is that everyone has the chance to see the richness of SQL: instead of one member grabbing the hardest or easiest or most-fun, you can all see those parts. I think this can be a very good experience.

Support

I understand this is a high-pressure time in the quarter, and you have a short time to work on this project. My goal is to make sure you get a chance to practice the pieces I have talked about in class. If you feel you are short on time, we can talk about the parts of the project that would be most rich for you.

I am nearly 100% available during this project. If you want help or have questions, please email me a time that works for you and I will set up a Zoom for that time or will propose an alternative.

I will also happy to answer questions via email. Sometimes the questions are harder to frame in text, but it's nice having the answers in writing. This works.

Starting framework

I will provide (in a ZIP file) a framework that is a good first pass at implementation, but that has holes that you must fill in. These are marked with `FIXME`. (Use `'grep FIXME'` in a nix environment or PowerShell, or `"findstr FIXME"` in CMD.EXE)

You should unzip this into its own workspace in VSCode and then work between the editor and the psql interpreter as I have in my videos.

I encourage you to commit the files in Git if you are comfortable with it. Regular commits make experimentation easier: things don't have to be working for you to make a commit. If you need help going back to an older version, I can help you do that if you saved the work in a commit.

Special Care

While I have provided a lot of the work, some of the remaining work is still hard. Notably:

- a Sale has a many-to-many relationship with Merchandise (one order can be for more than one item, and different customers may order Merchandise at different times on different orders)
- be careful that the “Quantity” field is attached to the right row

Project

Overview

The project is to construct a database for a sample organization working on climate action. The process of developing the database will follow the model presented in class: you will suggest requirements on the basis of an initial interview; then you create a data model that will go on to become the structure of the database. You will write SQL to create, populate, update, and report on the database to address the customer's needs.

Artifacts

Here are the artifacts I will be looking for in the project.

Design

For design: ultimately: an Entity-Relationship Diagram that reflects your plan for the physical layout of the database. This physical layout should include keys and should detail your plan for addressing any many-to-many relationships that might need to be expressed using a join table.

I suggest starting with a high-level design that captures only the entities and their relations, and then goes on to a mid-level design with the attributes identified. Seek my feedback at each level of iteration: this will save effort of filling in (“cementing,” if you will) a design that might not work well. If you take the tiered approach, I suggest you identify each stage explicitly and make a copy of the design at that stage for your project history/portfolio/detail.

You may use Draw.io or another diagramming tool (including Lucid Charts or Visio) if that works better for you. The tool should be able to show attributes and identify keys, and to show relationships with the 1 (hash) or many (crowsfoot) endpoints.

SQL

The project will require several batches of SQL; each will have a different responsibility and will be needed at different stages of the database’s lifetime. These should be kept separate from each other and all should be included with the project materials. There should be:

- SQL statements to create the table structure from an empty database
- SQL to populate the database with the initial data (COPY from CSV OK, or SQL INSERT)
- queries to answer questions the organization is interested in

Create Database

Create the database in PostgreSQL. Write the SQL script to create the tables for the entities in your Entity-Relationship Diagram. Your script should have all the tables, and the attributes with appropriate data types and the proper primary keys and foreign keys.

Some notes:

- use INTEGER for surrogate keys (no need to go all UUID here, and INTEGER is more efficient than the VARCHAR type)
- do not worry about expressing foreign key constraints (they complicate loading and experimentation at this stage, and it is easy to look up how to add them later)
- for surrogate keys, we will manage them explicitly (do not use SERIAL or other auto-assigned values: this is for ease in exchanging data if we need to, but a regular database would use SERIAL to provide DEFAULT values for surrogate keys)

To turn in:

- the SQL script with the necessary CREATE TABLEs
- one of:
 - psql running on the database, with the output of the “\dt” meta-command showing the tables (ideally from the VSCode panel)
 - a screen shot of the database in pgAdmin, showing the tables in the database hierarchy

Populate database

Enter the sample data in Climate Action Database Sample Data into your database using psql statements. psql statements may be either COPY from CSV or INSERT INTO.

To turn in

- file containing SQL to populate tables
- accompanying CSV file, if used

SQL Query and Update

Provide an SQL file for each of the following that performs the desired action or creates the requested report.

1. Newsletter: Assuming a newsletter date of 2022-03-10, show upcoming events after that date.
2. INSERT a member event suggestion (not approved)
3. List all the members who were added in 2020 (Use the EXTRACT function to get the year), newest first
4. Compute the total value of the donations listed (SUM)
5. Compute the average value of a donation (AVG)
6. Report donors who have given more than \$20 but are not members (SUM, GROUP)
7. Compute the total value of the sales (This involves a JOIN and multiplying the units sold * the price per unit for the merchandise.)
8. Create a sales receipt for a given order (This will be two queries, run sequentially: one for the header and one for the line items.)
9. Give membership to donors who have contributed (UPDATE)
10. Adjust inventory by sales (be concerned about idempotence) (UPDATE)

To turn in

SQL files that will query or update according to the above.

Climate Change Database

Sample Interview

In this imagined interview, Hannah is in the role of the database designer tasked with deploying a database to help a grassroots organization interested in climate change.

Hannah: Thanks for meeting with me. The goal for today is to get an overview of the database. We don't need to go into all the detail, just get a good general picture of what you need. What do you see as the main purpose of the database?

Megan: Our goal is to create a climate action site and we need the database to drive it.

Hannah: Ok, can you describe some of what you want to happen on that site?

Megan: Sure. We want to provide resources—links to sites and documents with useful information. We will generate some materials of our own. We also want to provide a calendar of events, some of which we will offer and some offered by other organizations and institutions. We want to suggest things people can do to make a difference.

Hannah: Are you anticipating any fundraising.

Megan: Yes, we are hoping to see items such as T-shirts, cups, books, posters, etc. to raise enough money to support the site. We will also take donations. Anyone that donates over 20 dollars becomes a member automatically, though you do not have to donate to become a member.

Hannah: How do you anticipate the site operations? Can anyone contribute articles, or can only certain people? How do you think it will work?

Megan: We will want people to register in order to contribute articles or links to events. There will be a small staff that validates the links to make sure they are legitimate, and to oversee donations and such. People who register can agree to receive newsletters that will update them about new information on the side.

Hannah: Will only members be allowed to view the data?

Megan: No. Anyone should be able to view the resources and events and be able to buy merchandise or donate. Only members can contribute events or resources and only members can add comments.

Hannah: Oh, what will they be able to comment on?

Megan: Events and resources, mainly.

Hannah: OK. Is there anything else you want to add?

Megan: I can't think of anything else at the moment.

Hannah: Well, if anything occurs to you: let me know,

Sample Documents

Existing documents provide insight into both requirements and existing processes. Below is an example of the documents that might have been collected during interviews with an imagined grassroots organization interested in climate change.

Members

Member Name	Email	Phone	Date added
Jill Keller	JillKileer@gmail.com	2065551245	3/29/2019
Mark Lowen	lowen@msn.com		4/19/2020
Wendy Nelson	wnelson@yahoo.com	4565551265	5/1/2020
Krystal Brown	krystalball@hotmail.com	9045552211	5/20/2021
Bob Danielson	bd2035@gmail.com	3605552030	6/10/2020
Ta Nguyen	tnguyen304@gmail.com		7/12/2020
Nichole Bradley	nicholeb@harvest.org	2225552323	7/15/2020

Merchandise

Name	Description	Units	Price per unit	Quantity
Earth Rise T-Shirt	T-shirt with earth rising over moon	1 T-Shirt	15.50	20
Earth Rise Puzzle	1000 piece jigsaw puzzle of earth rising over moon	1 box	9.35	15
Endangered Habitats	Tabletop book	1 book	35.30	25
Last Views	4k UHD documentary on endangered species	1 disc	23.90	10
Call to Action	Book on climate change action	1 book	6.45	30

Sales

Name email	Item	Quantity	Item2	Quantity2	Billing ad- dress	Shipping ad- dress	Date or- dered	Date shipped
Lindseylp@msn.com Pe- ter- son	Earth Rise puz- zle	2	Earth Rise T- shirt	1	161 Brown Street, Chicago, IL, 80092	same	10/2/2020	10/4/2020
KrystalBrown	Krystalball@earthlink.net rise T- shirt	3	Earth Rise T- shirt	3	303 South Mar- ket Seattle, WA WA, 98112	120 Pine Street Seattle, WA 98122	10/09/2020	10/12/2019
Marcusmtellerman@yahoo.com Teller- man	Earth views	1	Earth Rise T- shirt	1	2020 North Street San Fran- cisco, CA. 99122		10/30/2020	

Donations

Name	Email	Date	Amount
Hamid Brown	hamidb@outlook.com	8/14/2020	10.00
Ha Li	hali@gmail.com	10/15/2020	200.00
Ken Jackson	kj202@yahoo.com	10/15/2020	15.00
Wendy Nelson	windynelson@morningstar.org	10/16/2020	135.00
Ken Jackson	kj202@yahoo.com	3/2/2022	15.00

Events

Convention on Climate Policy
Room 313
Convention Center
200 Pike Street
Seattle, WA, 98122
December 3rd 2022

Price 250.00 per attendee

UN Report on Climate Progress
UN Convention House 2
UN Plaza
New York, NY, 00231
January 13,2022

Price Free, but limited seats

Resources

Partnership for Energy progress

<https://www.pepnw.org/about-us/>

The Partnership for Energy Progress is a collaboration of utilities, farmers, workers, small and large businesses, and community advocates across the Northwest. Our goal is to communicate the work we do to provide reliable, affordable energy to homes and businesses, and highlight the progress we're making to address climate change.

Date added: 9/12/2020

Member who added: Jill Keller

Siemens Company

<https://new.siemens.com/us/en/company/sustainability/environmental-action.html>

Siemens is supporting the Decade of Action to accelerate the adoption of sustainable technologies so our customers, our communities and our employees can lead the U.S. towards a low-carbon future.

Date Added: 10/09/2020

Member added: Jill Keller

UN Environment Program

<https://www.unep-wcmc.org/>

UN Environment Programme World Conservation Monitoring Centre is a world leader in biodiversity knowledge. It works with scientists and policy makers worldwide to place biodiversity at the heart of environment and development decision-making to enable enlightened choices for people and the planet.

Date Added: 10/18/2020

Member added: Bob Danielson

NASA on Climate Change

<https://climate.nasa.gov/>

Explore a real-time data visualization of NASA's Earth-orbiting satellites and the data they collect about climate change.

Date Added: 10/25/2020

Member added: Ta Nguyen

Comments

10/20/2020

Nichole Bradey

[Climate.nasa.gov](https://climate.nasa.gov)

This site is gorgeous and useful. Well worth the time to visit and explore its features.

10/22/2020

Wendy Nelson

I attended the Convention on Climate policy last year and it was quite enlightening. They had discussions from several high ranking scientists and politicians. Worth the price.

Recap of the Database Development Lifecycle

We are covering the last few sections of Database Design, where we go from the “physical model” stage of the ERD (with all relations fully expressed using foreign keys) to the SQL and running databases.

Installation refresher

We need to:

- make sure the database server is running
- create an empty database to work with
- add tables to the database
- populate those tables with data

Is database server running?

These tools can help confirm the database server is running:

```
pgAdmin4
```

```
brew services start postgresql
```

```
pg_ctl start
```

Listing databases

```
psql -l
```

Dropping/creating a database

commandline

This is the preferred way to automate operations

```
dropdb databasename
createdb databasename
```

pgAdmin

Right-click on the Server and select “new database.” Choose “template-0” if asked for a template.

SQL

Datatypes

These are the datatypes that should be sufficient for the project:

- INTEGER (for surrogate keys)
- DATE – for dates
- NUMERIC(14,2) – allows trillion dollar donations
- VARCHAR – for text

Syntax

Keywords and table/column names are case-insensitive (mostly; you can quote if you need to).

Every SQL statement must be terminated with a semicolon.

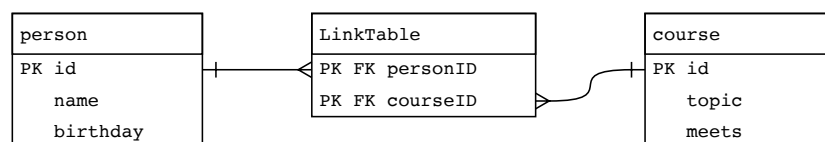
New: Comments begin with – (two hyphens)

CREATE TABLE

```
CREATE TABLE table-name (
  first-column-name INTEGER,
  second-column-name VARCHAR
);
```

Design-to-SQL

Since the designs all have attributes down to the physical model (all foreign key attributes to support relations are shown in the design), writing the SQL is a pretty direct mapping:



becomes:

```
CREATE TABLE person (  
    id INTEGER PRIMARY KEY,  
    name VARCHAR,  
    birthday DATE  
);  
  
CREATE TABLE course (  
    id INTEGER PRIMARY KEY,  
    topic VARCHAR,  
    meets VARCHAR  
);  
  
CREATE TABLE LinkTable (  
    personID INTEGER,  
    courseID INTEGER,  
    PRIMARY KEY (personID, courseID)  
);
```

