**IFT458/598 – Project 2**

Yuan Li & Edward Halper

**Introduction**

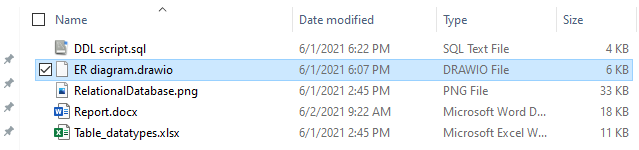
The problem faced for this deliverable is to create a backend database for the non-profit organization IMPACT’s website in which the “bare bones” were created in the previous deliverable. The database must include data entry for officers, volunteers, and events. Specific attributes for officer include a unique ID, names, age, address, description of responsibilities, the office, and start and end dates at the office. Volunteer attributes will match the officer attributes without the specific job attributes. Event attributes will include a unique code, name of the event, officer, description of mission and objectives, date of volunteer, number of attendees, address, and phone number. Finally, the database will be able to record a voluntary event of interest, request dates, and end dates. As an approach to building this database we have started with creating the tables with attributes using ER diagrams, datatype table, relational diagrams, and producing SQL script to create and relate the tables needed in the backend database. This report will cover the description of work, a user manual of the files included in planning and executing the database, and conclusion.

**Description of your work**

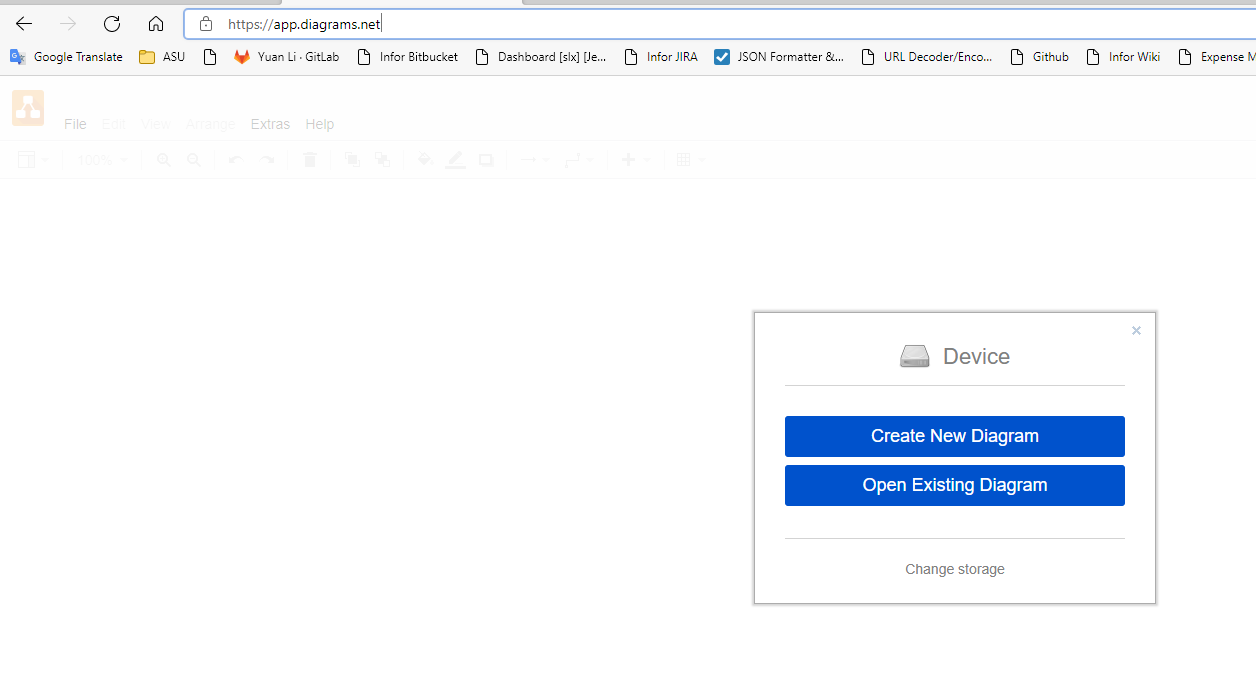
|  |  |  |
| --- | --- | --- |
| Name | Item | description |
| Yuan Li | ER diagram | I used <https://app.diagrams.net/> to draw the conceptual ER diagram.  First: I Created simple entities, single valued attribute.  Second: I Created binary relationships between two entities  Third: I added attributes to the relationships  Fourth: I mapped the ER diagram to tables |
| Yuan Li | SQL script | I used MySQL to define the DDL script.  I installed MySQL on my PC and started define tables, columns, foreign keys, constrains according to the ER diagram and my partner’s relational diagram. |
| Edward Halper | Relational Diagram | I used [www.erdplus.com](http://www.erdplus.com) to draw the relational diagram and show the constraints between tables. Starting with tables I added attributes from the conceptual datatype table, that was created in excel, designating datatypes, primary keys, foreign keys, and unique keys. Pictures and explanation are in the User manual section. |
| Edward Halper | Datatype table | For the datatype table I used excel to create a list of the tables with attributes. Identifying datatypes for each attribute along with the different keys, as in primary, foreign, and unique. Pictures and explanations are located in the user manual section. |

**User Manual**

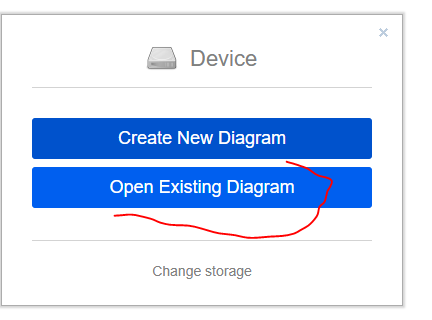
Step1: In the zip file, there is a file called ER diagram.drawio. That’s the ER diagram file



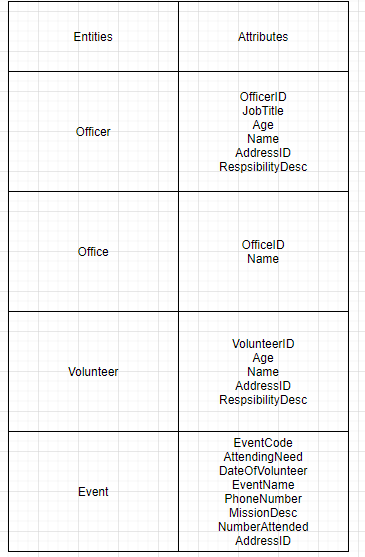
Step2: To be able to open it, you have to go to <http://app.diagrams.net>



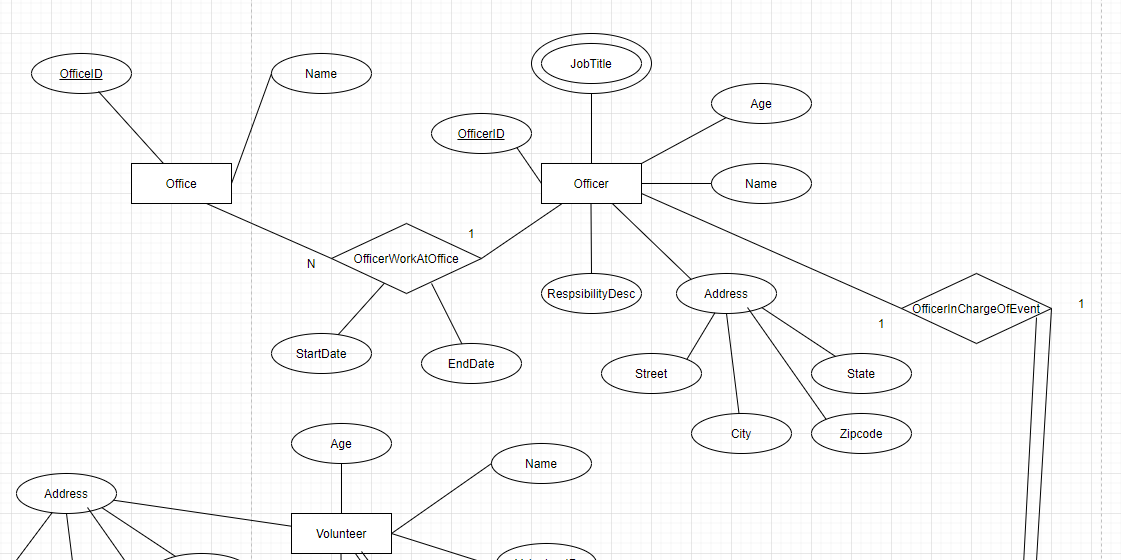
Step3: Then select Open Existing Diagram, and select ER diagram.drawio in your folder

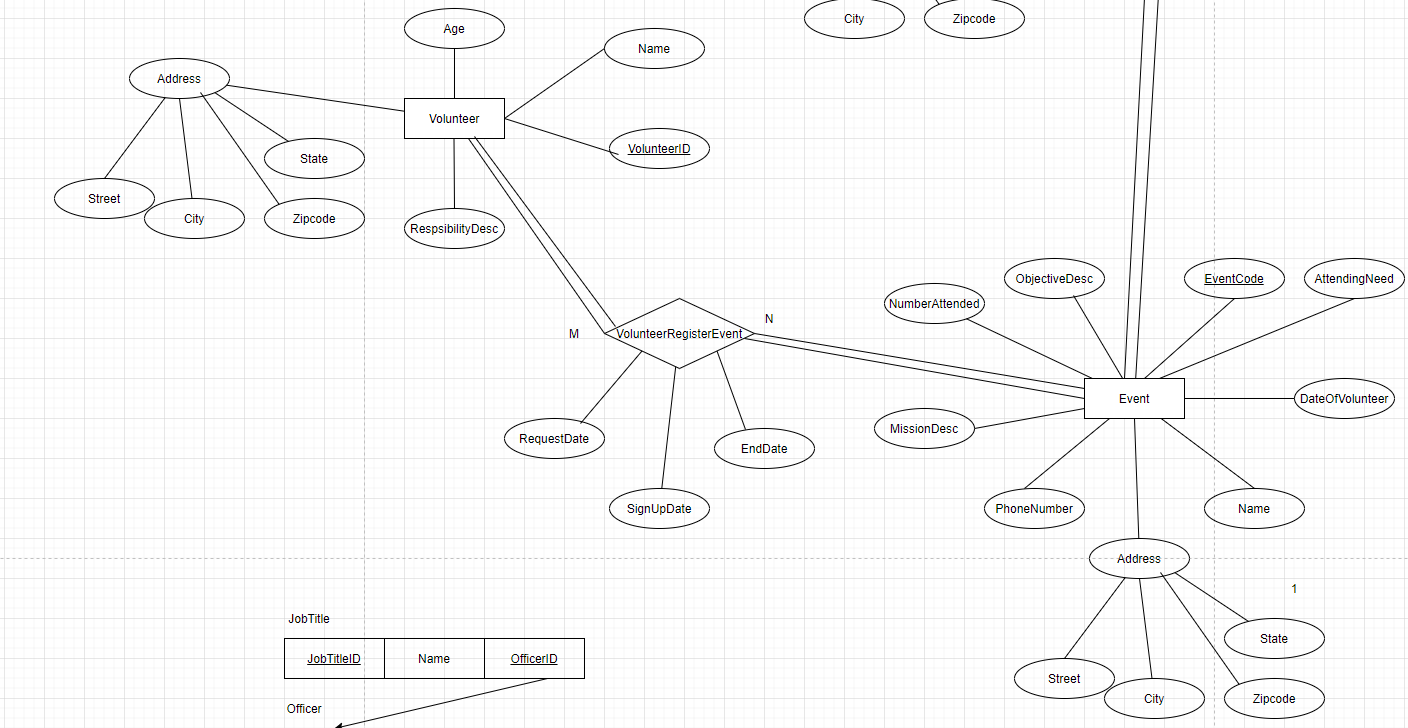


Step4: After you open the file, you will see the ER diagram, on top there is a simple table which defines simple entities and attributes.

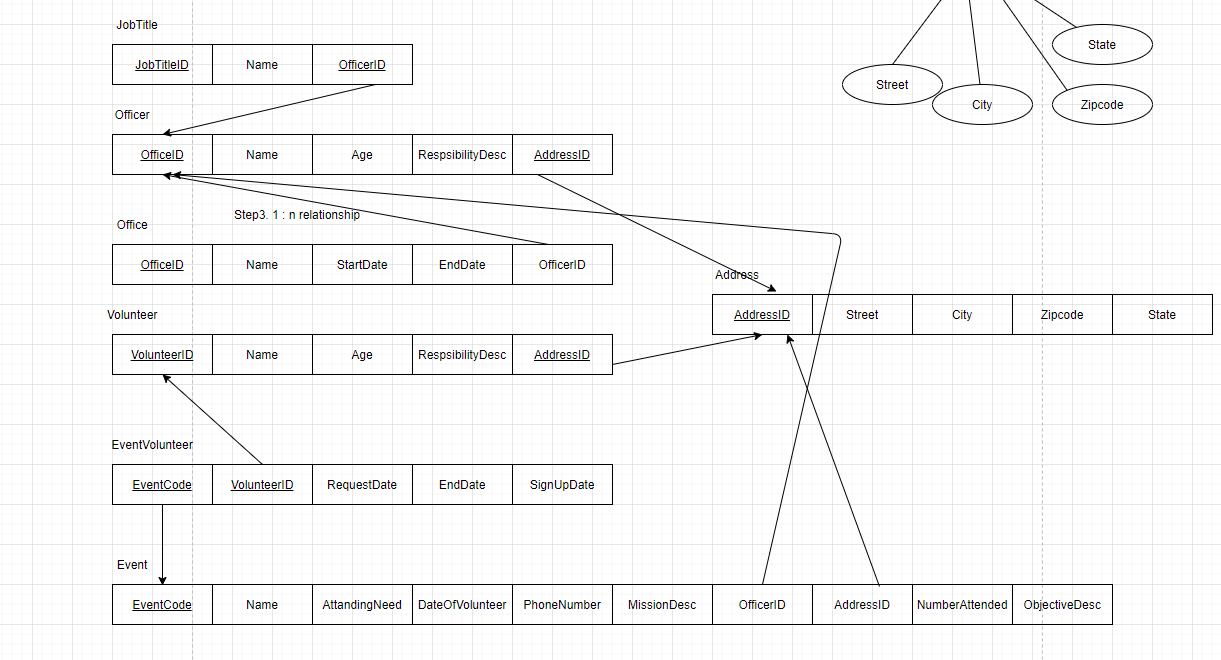


Step5: Scroll down, you will see the ER diagram including entities, attributes, and relations

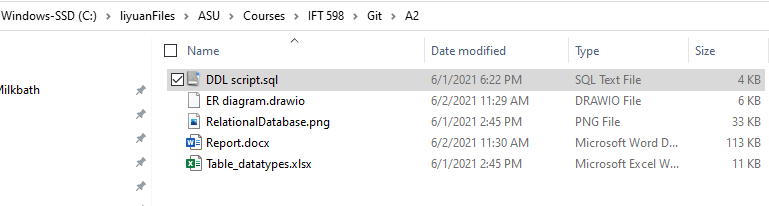




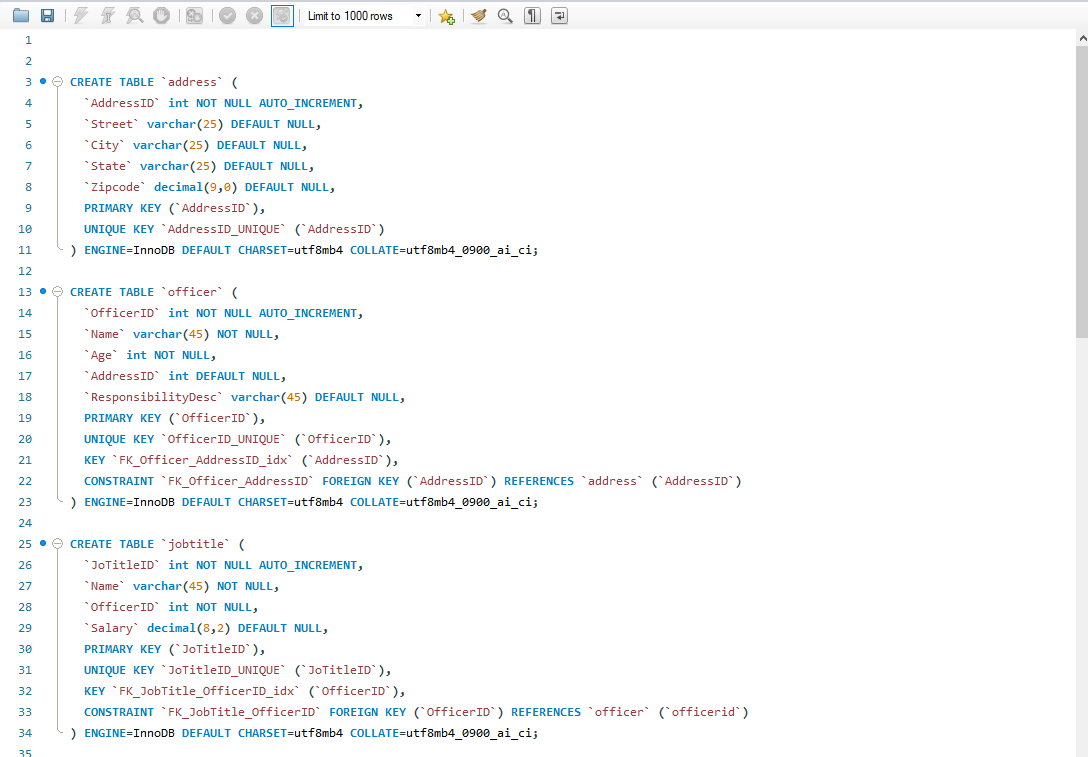
Step6: Continue scrolling down, you will see the relation diagram based on ER diagram. We first only had 4 simple entities but ended up with 7 entities. That’s because we have one M:N relationship and two multivalued attributes



Step7: You can open the DDL scrip to check the detailed implementation.



Step8: Open the DDL script

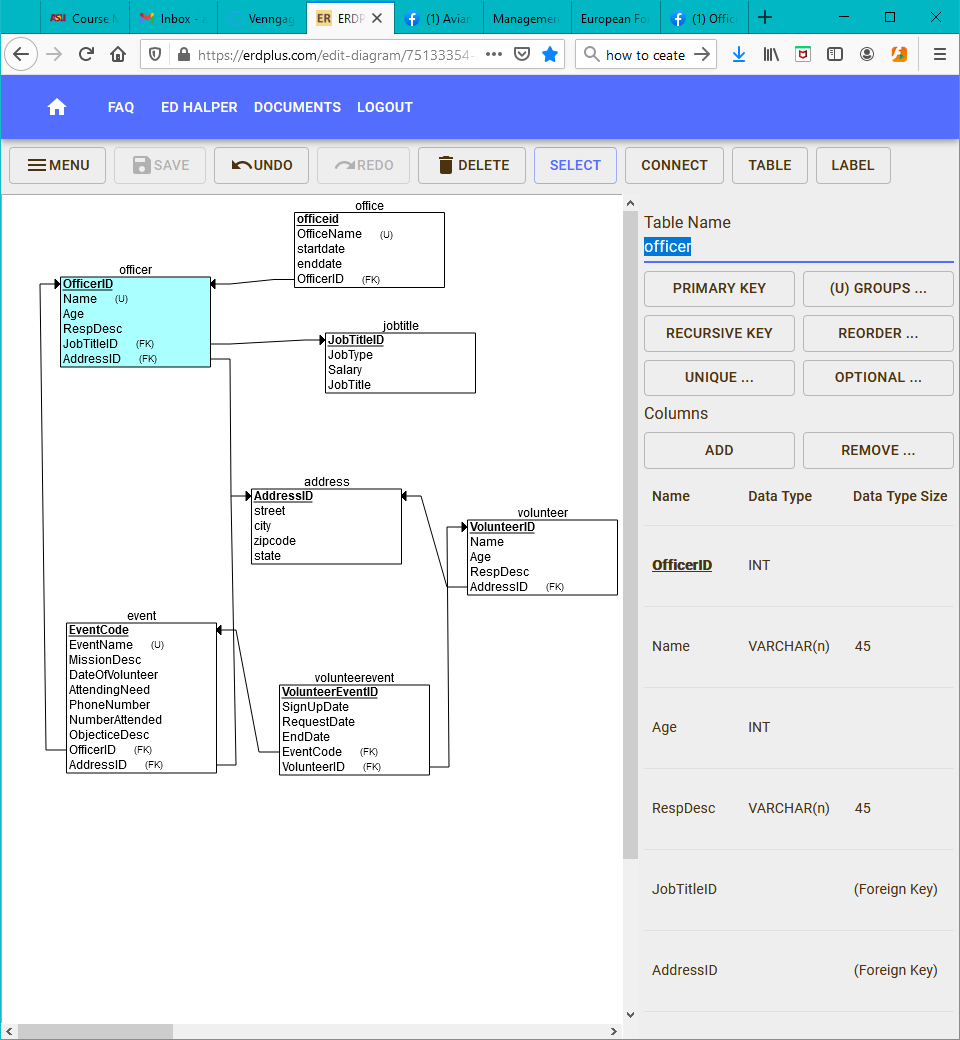






Relational Diagram

The relational diagram shows the tables with attributes and the constraints between the tables. As you can see from the image below using erd I was able to accomplish input of tables, attributes, constraints, and datatypes. Primary keys are designated by being bold and underlined while foreign keys have an FK in parenthesis and the Unique designator is a U in parenthesis. Datatypes are listed and in view from the tab which was not exported but matched to the datatype table.



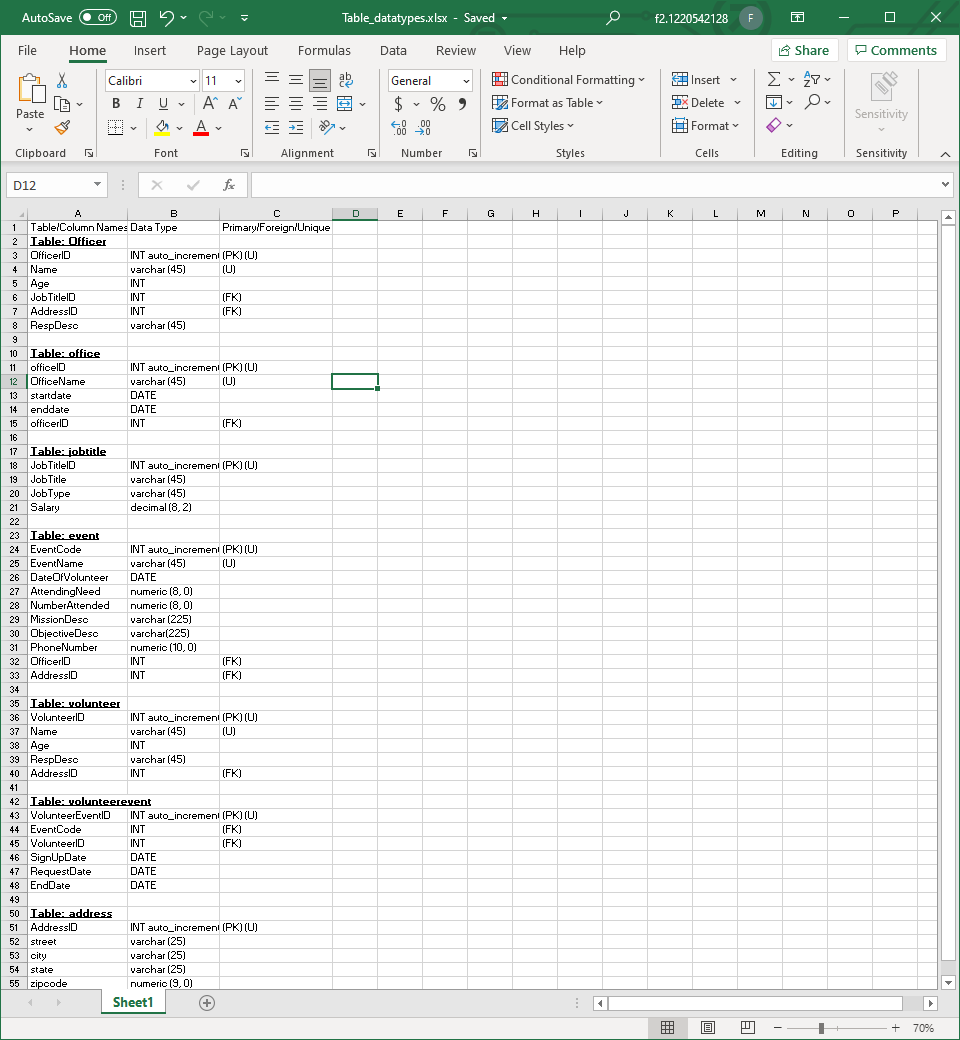
The below image is the final product for the relational diagram. Each line and arrow shows a constraint from one table to another. The arrows are pointing from the foreign key to the primary key of the related table.

Diagram

Description automatically generated

Datatype Table

The image below shows the excel produced datatype table. This table shows the table name, entities for each table, and attributes of those entities. Following common identifiers for primary key (PK), foreign key (FK), and Unique (U). The datatype is denoted in the second column of the table for integer, varchar with number of characters, numeric (p s), decimal (p s), and date.



**Yuan’s Conclusion:**

I learned how to design a database from the concept to implementation. I used to design the database directly from database, and then generate the diagram from the database, which is kind of a backwards process. I never followed the conceptual algorithm to design the ER diagram step by step. I think the step by step design will help catch all the requirements. I also learned the relationship attribute; a lot of attributes are only existing when the relationship exists, so we can add as many as attributes to the relationship entity/table.

Now if you give me an ER diagram, I think I will understand it and can implement it in a database accordingly. I also learned MySQL, I never used MySql before. I installed MySQL and designed tables and created the DDL scripts. MySQL is very similar to SQL Server, it’s actually easy to pick it up. The thing to improve I should spend more time to look for a better tool, because the current one I am using is an online ER tool. You have to open their website to draw your diagram and open your existing diagram, and there is no an smart way to organize your diagram layout besides dragging and dropping.

**Edward’s Conclusion:**

In conclusion I believe that we have achieved a good start to a database structure and my partner was able to quickly put together the sql to create the database for the backend of the website. I was able to take the information from the deliverable document and follow along to create the necessary tables. Pulling from some of my previous experience from work and school I decided to split the address table into a separate table from the officer and volunteer, I believe this could allow for certain lines of data to keep from being redundant but may take a little more tweaking to put the correct information in the correct tables to accomplish this. I learned how to use erdplus as I have not gotten a chance to use this product before. I found it to be easy especially after creating the datatype table. I went in order from creating the tables with entities to creating the connections and constraints. I also enjoyed working with my partner on this deliverable because I believe we are starting to communicate well and accomplish tasks faster together. As for improvements I believe in the process we could have continued through the development loop and refined the tables and entities in each one. I seem to remember going through 4 iterations to checking database designs but with such a simple database do not believe it to add too much value in this instance.