Introduction

There currently is no centralized database for the Pennsylvania School System. We aim to build an easy to use database system focused on fiscal data, bringing valuable information to the general public and officials alike. With detailed and specialized information on how different Counties, Districts, and Schools operate, individuals can come to conclusions about how to optimize learning.

Table of Contents

Overview

The application that we are developing will be able to quickly and easily sort through various School/District based datasets, while keeping all information in one central place for the Users. This application will be used to analyze and compare fiscal data for personal and official use. The focus of the application is to take relevant data on the financial facts of Pennsylvania schools and put it in a centralized and organized single application. The application will place an emphasis on being user friendly and intuitive.

Project Requirement Summary

- 3 Visualizations
- Display data and visualization of Schools and Districts in an easy to use format
- Be able to compare between two Entities

- Multi-platform support(Computer and Mobile)
- Store data from the given datasets

Considerations and Parameters

Assumptions, Constraints and Dependencies

Development Guideline and Practices

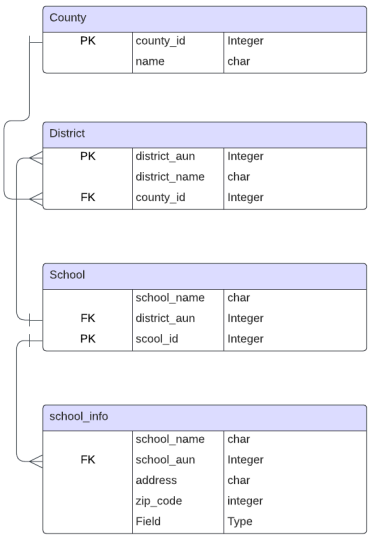
As this is a project that is going to take an extended period of time we understand that many of our initial ideas and practices will be subject to change if the need comes. So with that in mind if/when we find inadequacies with our design we will reconvene to figure out how to pivot into something that is more desired by the team and project. For matters of actual development we will meet together on a weekly basis and discuss what work we have done, any problems we've run into and what we plan to do in the coming weeks.

Design

The format for the design is as follows. A home screen that will have a function to search for specific data, and one screen for each category County, District, School. The home screen will be a simple HTML based page with basic information on the functionality of the application along with the search bar. Then the three other pages will be structured the same way to keep consistency. The entity name will be at the top of the page and underneath will be a brief grid containing the other entities within the first object, as well as some information. This grid will have links to each item's respective page so the user can traverse down a relation. Underneath the grid of entities will be the different tables and the data they provide. From there the user can select different years to change the tables to represent that otherwise all information will be as up to date as possible.

Data Design and Structure

The basic structure of our database is designed through Django and SQL. Django makes the addition of new data and tables simpler than normal sql. This allows us to quickly make changes and additions. The design is rather simple to reduce redundancy. Entities(Schools,Districts, and Counties) will be split up into their own table with their identifying information and links to related entities. This allows the database to save time when querying through the concept of normalization. The primary key of these entities is their disting 9 digit AUN number. This number is then used to join to get user desired information from related



relations.

UI

The UI is based off of HTML and Bootstrap, creating a modern feel to the system. The basic interaction between the user and the website is to receive data on desired Entities. To do this the user can navigate to the County page to find their desired County, by searching using AUN, name or using an a-z list. On selection of the County there will be data on said county as well as a link to the districts in that county. This is the basic user behavior that is repeated from district to school. This simple approach of piecing together nested pages allows for rapid familiarization for the user and visual, logical connection. This connection/familiarization makes learning the database simple as it follows human reasoning.

Some additional interactions between the user and the interface is a year selection filter and a comparison feature. The year selection is on all individual entities pages. Once on the page there is an option to change the year of the data to see how the data changed over time. To better compare data through the years a comparison feature will be implemented to allow for

the selection of two different comparable tuples to view at the same time. This allows for rapid comparisons and is a key feature of the system.

Conclusion