1. Describe what data is stored in the database. (Where is the data from, what attributes and information would be stored?)

The database we chose is the Steam Game Database. This data stores many attributes for each game and it is a combination of information from Steam APIs and steamspy.com. The data has an attribute called QueryID which is the original ID of the game in the "idlist.csv", it also has information about the game in media such as MovieCount, which counts the movies/videos associated with the game. The database also has information about the pricing and availability of the games, such as attributes: IsFree, PriceInitial, and SubscriptionAvail. Furthermore, for each game, the database has information about platform and system requirements, as well as the category, genre and tags.

2. What are the basic functions of your web application? (What can users of this website do? Which simple and complex features are there?)

Some basic functions that our web application can do is allow users to look up specific data in the database, such as searching for games released during a certain year or range of years. Users can also search for games based on criteria that they like, such as the genre or if the game is free. Users can also search up games depending on their name and should be able to view details about the game itself such as developer, price, system requirements and the reviews for the game.

Some complex features that we can do for our game would be to allow users to compare multiple games side-by-side. Users would be able to compare prices, reviews, release date, compatibility, etc. Since we are using Python as our coding language, it is also possible for us to incorporate Numpy and Pyplot features to generate correlation graphs and plots for users to look at such as sale numbers vs time for each genre. These graphs can be dynamic in the sense that a user can input certain parameters, and data can be visualized based on those parameters. This wouldn't be too different from https://waf.cs.illinois.edu/ , where students can compare GPAs of different classes. Another complex feature would also be a questionnaire that a user can fill out based on their preferences and the website can match the games that they would most likely enjoy playing.

3. What would be a good creative component (function) that can improve the functionality of your application? (What is something cool that you want to include? How are you planning to achieve it?)

For a creative component we can create a trend prediction to predict the sales of a certain game from the user. To approach this, we would gather data for each game such as release date, genre, price, developer information, controller support, compatibility etc. Then, we could also gather data such as the time since the game's release, seasonal factors and depending on what genre is popular. After that we can implement an appropriate model/algorithm that can predict the sales based on that data.

In addition, we could also use the Natural Language Processing feature of Python by utilizing Python packages like NLTK to further aid the prediction of the sales number. This is primarily achieved by isolating certain keywords from the game of interest and using NLP to check the sales and performance statistics of previous games with similar keywords from the database. However, since NLP is a relatively complicated subject, this part of the feature would probably remain rudimentary.

4. Project Title

   Game Developer's Database.

5. Project Summary: It should be a 1-2 paragraph description of what your project is.

Our project is a data analysis website on the Steam Game Database. Here users can search up specific data within the dataset, such as looking at most popular games by genre or year. Users can search up games based on criteria that they like, such as the genre or if the game is free. For each game a user should be able to see all the details related to said game. A user can also compare games side by side.

Furthermore A user can visually interpret the data with dynamic graphs, where a user can input parameters and have data visualized based on these parameters. Another complex feature would be a game recommendation system, based on a user's interests. Moreover we plan on implementing a game popularity prediction feature to aid game developers. This feature will utilize NLP and other data analysis techniques.

6. **Description** of an application of your choice. State as clearly as possible what you want to do. What problem do you want to solve, etc.?

Our application is a website-based database application built around the Steam Game Database. We want to provide data about games in the past to future game developers so they would have a better understanding of the game industry. This application would allow them to use filters and keywords to isolate games of a certain genre or producer and look at its sales trend in certain regions in the form of graphs, plots, or charts.

We first want to use SQL to filter and select data from the dataset based on the user's input. After the data is filtered, the selected data is analyzed via Python. We are mainly planning on using Python packages like Numpy, Matplotlib.pyplot, and Pandas to analyze the data and produce graphs. For the Natural Language Processing of title names, we are planning to use the NLTK package of Python. These data analysis tools should help us to select and analyze the data efficiently and present results with better readability to the game developers.

7. **Usefulness**. Explain as clearly as possible why your chosen application is useful. Make sure to answer the following questions: Are there any similar websites/applications out there? If so, what are they, and how is yours different?

Another steam game data-based website is the steamspy database (https://steamspy.com/). However, sites like steamspy are more focused towards providing data

to game users/players by incorporating features like listing the current deals. Also, steamspy extrapolates data from user profiles rather than from the official SteamDB database, so it would be less accurate in terms of data.

On the other hand, our website is generally geared towards game developers to provide them with the insight of the past and current game industry trend and hopefully predict the future prospect based on that trend. Thus, our website incorporates more "Post-processed" features like game data analysis, trend graph generation and sales prediction rather than just presenting raw data from the database. Our purpose is to use this website to help game developers to make wise decisions based on what topics/genres were and will be more popular. This is something that the currently available steam game databases lack in functionality.

8. **Realness**. Describe what your data is and where you will get it.

Our data on Steam games include data for each game on sales, launch year, genre, producer, game title, player numbers (current and peak), metacritic scores, recommendation count, supported languages, platforms they are available on, and whether the game is free, subscription-based, or purchase-based.

We select this dataset from one of the TA-proposed datasets from the project track 1 datasets documentation on Canvas. The data itself comes from data.world website and was gathered and cleaned by Craig Kelly. The author was inspired by original steamdb.info and combined publicly available Steam API data and data from steamspy.com to create this dataset.

9. Description of the **functionality** that your website offers. This is where you talk about what the website delivers. Talk about how a user would interact with the application (i.e. things that one could create, delete, update, or search for). Read the requirements for stages 4 and 5 to see what other functionalities you want to provide to the users. You should include:

We would implement the create functionality in our application by having users sign up, and then consequently login to access the application. For the update and delete functionality, the users can have a game wishlist and list of friends, where the user can add new friends or new games, and delete friends or games from the specified lists. For the read functionality, we would read the steam game database, but also read the user database in order to accomplish many of the tasks and features of the application. For example, in order to add a new friend to a user's friends list, the application has to read through the user database to select the specified account to add to the list. The search feature would also be widely implemented as a user can search for games, and search for friends.

10. **A low fidelity UI mockup**: What do you imagine your final application's interface might look like? A PowerPoint slide or a pencil sketch on a piece of paper works!

In terms of UI, we are drawing inspiration/reference from existing game-related database sites on other topics, such as this following website https://aoestats.io/. This website is about the game Age of Empires 2, where it analyzes and displays different civilization win rates in

different scenarios. We are currently planning making our website UI similar to the aoestats one, as shown below.

For the individual games, we are planning on making the users able to click on the game's name and the website will redirect them to a subpage where they can see all the detailed stats and graphs we generated, also similar to the layout of this aoestats Franks civilization page shown below.

Highest Win Rates Against

Civilization	Games	Win Rate
Vietnamese	1,521	59.89% ±2.47
Koreans	963	58.98% ±3.11
Britons	4,093	57.56% ±1.51
Chinese	1,799	57.53% ±2.29
Mayans	2,531	56.97% ±1.93

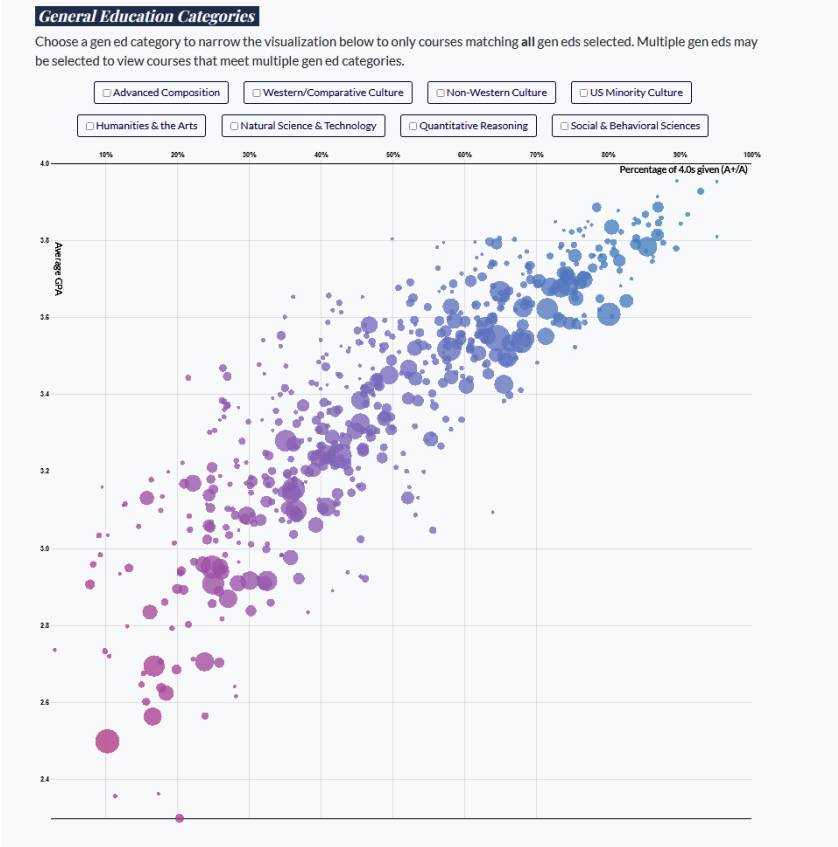
Lowest Win Rates Against

Civilization	Games	Win Rate
Vikings	1,847	48.19% ±2.28
Incas	2,293	48.15% ±2.05
Dravidians	528	47.92% ±4.27
Teutons	2,646	46.45% ±1.90
Romans	0	0.00% ±100.00

Openings

Opening	Picks	Play Rate	Win Rate ▾
Scouts	33,626	51.56%	56.48% ±0.53
Fast Castle	13,481	20.67%	55.71% ±0.84
Fires	580	0.89%	54.83% ±4.06
Archers	4,810	7.37%	51.83% ±1.41
Man-at-Arms	3,858	5.92%	47.74% ±1.58
Trash	7,593	11.64%	45.86% ±1.12
Drush	763	1.17%	45.61% ±3.54
Galleys	164	0.25%	44.51% ±7.69
Towers	346	0.53%	41.33% ±5.21

For the graphs we took inspiration from https://waf.cs.illinois.edu where we can have dynamic graphs like the one below, where the user can check specific parameters and the graph would change accordingly:



11. **Project work distribution**: Who would be responsible for each of the tasks or subtasks? List of the person responsible for which exact functionalities in section 6. Explain how backend systems will be distributed across members. Be as specific as possible as this could be part of the final peer evaluation metrics.

Website design/ui: Arpan Swaroop

Data visualization/graphs: Adriel Taparra

Database programming in SQL: Nour Sarikaya

Creation of trends using data processing, analysis and NLP: Joseph Sun (xsun45)