

SUMMARY OF THE DATA

We have chosen data based on Destiny 2 Game. This game as it's style is a first-person shooter game which has simulation of role playing and MMO elements. However, we came to know that this game is very different from regular MMO games as it lacks most of the characteristics of a basic MMO game. We have chosen this data as this has a very unique data set and fascinating data analyzing which has a sufficient amount of conceptual data model. Furthermore, from this data we get to know the representation of how the game works. This data is not a form of typical database and moreover is very user friendly to find out the information about the game.

Basically, the dataset consists of 15 different tables which most of them are interconnected to each other. The main table of the dataset is the Weapons table, that's where we get access to all of the different kinds of Weapons and different kinds of abilities of the Weapons. Weapons will have different types of weapons, different types of ammo, different types of frame and many more. Each Weapon will have the state where we can get range, handling, zoom, and so on. We have ID's for all of the tables in large numbers because we are using ID's directly from the API.

We have a large database consisting of 40000 records into 15 different tables. Data set consist of various entities, starting with the weapons which has foreign keys from different entities. After That, we have perks, armours which are being used to operate the weapons. Also, this data consists of weapon type, canMode, canRoll and armourForm.

We have extracted data using Bungie API. The data was in Bungie API in JSON format and we have converted them to sql format. The file populateDatabase.js will convert the API to the sql supported statements. We have provided the database with all of the inserted statements and all necessary lines of codes that will need to create the database.

DATA MODEL

The database was split into tables to simplify it from the more monolithic format that it was provided in by the Bungie API. The Bungie API provides the data in the same way that the game uses it, and that is somewhat different from how the data is interpreted. For instance, all of Weapon Frames, Weapon mods and Perks are all under the SandboxPerk umbrella. (Weapon mods here referring to the effect of the mod and not the actual mod itself which is classified as an InventoryItem) So in order to reduce confusion for the subset of data that we are using, the tables were split into what are presented to better format the data provided.

The largest difficulty was sorting through the data provided by the API, as mentioned, many seemingly independent things are all classified under the same umbrella, so trying to find logical splits in the data was difficult. Additionally, some connected things in-game are not connected via the database so workarounds had to be utilized for those.

SUMMARY OF THE DATA MODEL

- List of each of the final tables, along with its cardinality and arity

Cardinality and arity of all tables is provided below.

Table Name	Cardinality	Arity
AmmoType	5	2
Armor	4158	12
Armorform	1803	2
Armormod	9815	2
Canmod	1397	2
Canroll	433	2
Catalysts		
DamageType	6	2
Frame	494	2
Mods	4548	2
Perks	2832	3
Source	323	2
TierType	7	2
Weapons	906	17
Weapontype	30	2

LIST OF QUERIES

We have more than 10 queries, of which one or two are quite simple but others are a little complex queries. We have used GROUP BY and ORDER BY in two of the following queries.

1. Find all void weapons
2. Find all submachine guns
3. Find all void submachine guns
4. Find all void submachine guns with a precision frame
5. Find all weapons that use the adaptive frame
6. Find all weapons that use the adaptive frame that are solar
7. Find all exotic armor with “Helm” in its name
8. Find all weapons that have an aim assist of >90, and are energy hand cannons
9. Count the number of each weapon type ordered most to least
10. Count the number of armor pieces by rarity
11. And a number of person-chosen queries, created via the interface.

SUMMARY OF THE INTERFACE

Our interface mainly contains HTML, CSS, JAVASCRIPT and we have done most of the work in javascript. We have created an interface by using javascript language and we have used the NPM library to run the connection files. We are using a postgresSQL for the backend to create the database and for the connection of the database to the API. From our API users will be able to search the Weapons by their name if they want to get the specific weapon type or Tier type then we have the option to select the weapons from such types. Our API will give the options to search the weapons of different stats where it will be Aim assistance, Stability, impact. We have set up in a way where users will have a provided hint on how to select those options and search Weapons.

Our interface will have pre-made queries with simple english description where users will be able to select the english description and by searching that up, they should be able to see the tables. Our pre-made queries are using the order by and group by functionalities. Users can set the limit of the table size if they want to see just 10 results; they can change the number in the search limit box. Premade query will show all of the columns for now with all of the foreign key attached to it.