



Player Data Manager

Simon Little, Teja Dasari, Aaron Quashnock,
Drew Rigby, Thariq Zubair

What is the Player Data Manager?

- Allows players of different games to come together and create builds
- Facilitates developers to have global access to game trends and strength of builds
- Uses REST APIs that games use to expose said items, characters, and builds
- Allows producers and executives to see what fosters growth in their games

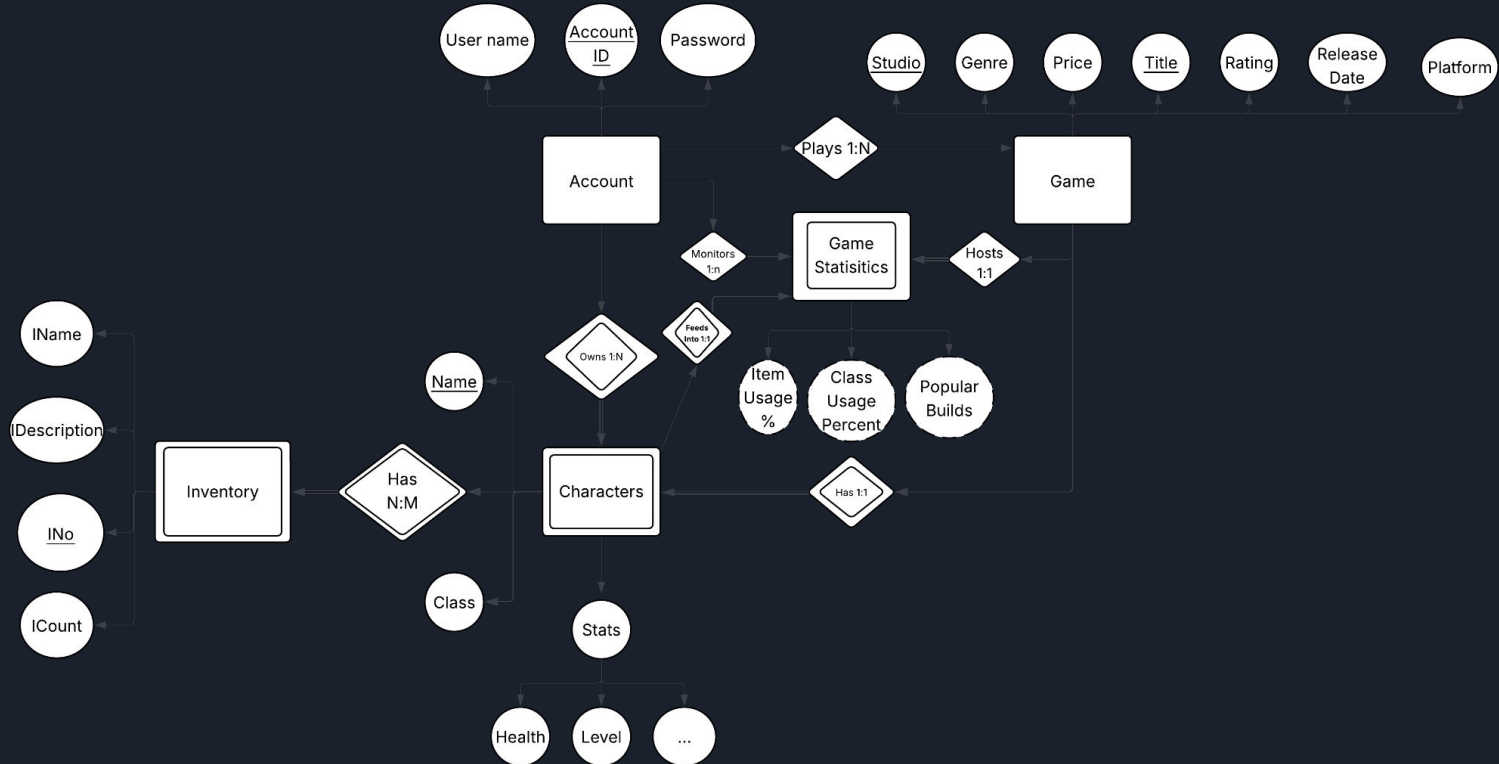




What are the Entities?

- Accounts
 - Account_ID, Username, Password
- Game
 - Title, Studio, Platform, Genre, Price, Rating, Release_Date
- Characters
 - (Title, AccountId, CharName), Class, Stats
- Game Statistics
 - (Title, Studio), Characters
 - Derived from all character: (Item Usage %, Class Usage %, Popular Builds Ranked)
- Inventory
 - (AccountId, Title, Studio, CharName, ItemID), Name, Description

Entity Relationship Diagram



Relational Model

Accounts

<u>Account ID</u>	Username	Password
-------------------	----------	----------

Game

<u>Title</u>	<u>Studio</u>	<u>Platform</u>	Genre	Price	Rating	Release Date
--------------	---------------	-----------------	-------	-------	--------	--------------

Game Statistics

<u>Title</u>	<u>Studio</u>	Item Usage %	Class Usage	Popular Builds
--------------	---------------	--------------	-------------	----------------

Characters

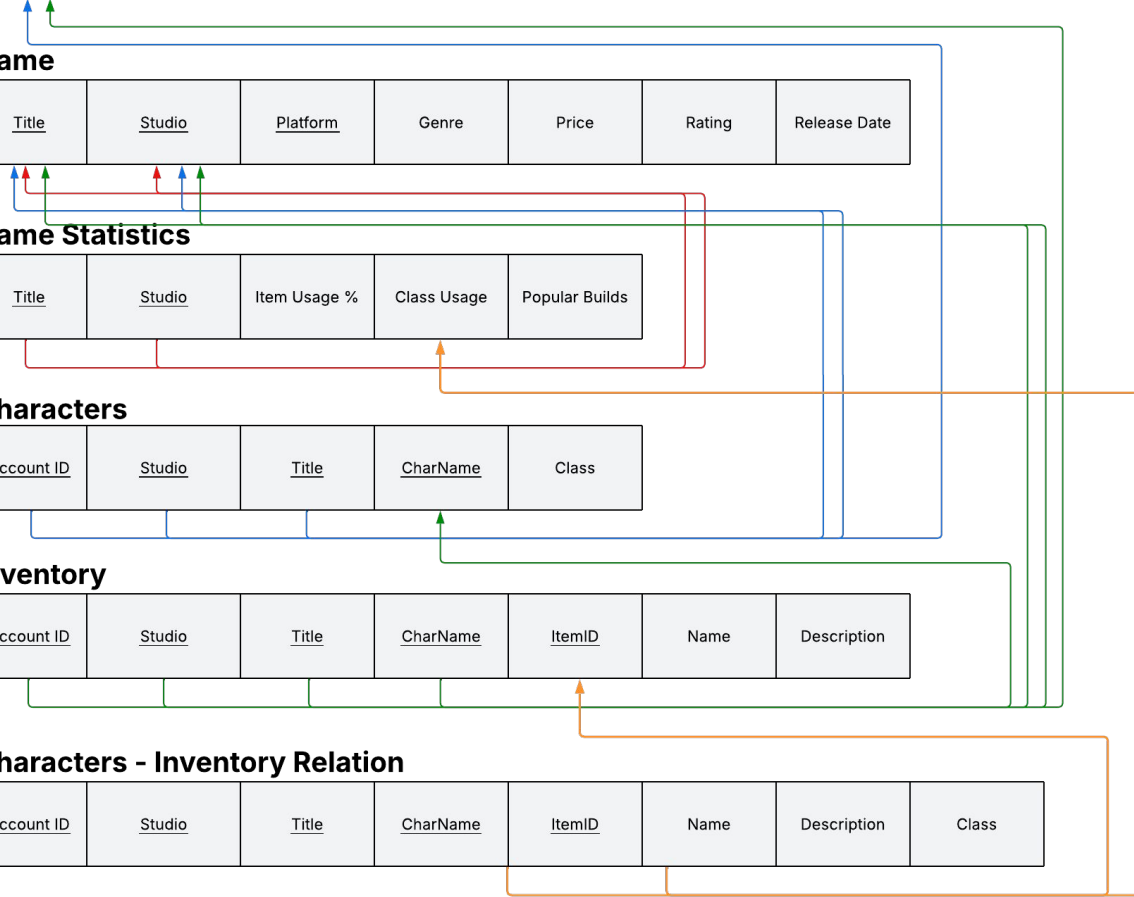
<u>Account ID</u>	<u>Studio</u>	<u>Title</u>	<u>CharName</u>	Class
-------------------	---------------	--------------	-----------------	-------

Inventory

<u>Account ID</u>	<u>Studio</u>	<u>Title</u>	<u>CharName</u>	<u>ItemID</u>	Name	Description
-------------------	---------------	--------------	-----------------	---------------	------	-------------

Characters - Inventory Relation

<u>Account ID</u>	<u>Studio</u>	<u>Title</u>	<u>CharName</u>	<u>ItemID</u>	Name	Description	Class
-------------------	---------------	--------------	-----------------	---------------	------	-------------	-------





Possible Constraints

- Domain Constraints
 - Game Genre should come from a predefined list
 - Item usage percent should be a float
 - Popular builds should be a dictionary with the build and the usage where one side of the value is the build and the other is the percentage as a float
 - Rating would have to be 1-5
 - Release date in a standard format
- Entity Integrity Constraints
 - Stats per game should be defined
 - No keys should be NULL
 - In games without classes class CAN be NULL
 - Game must have Stats, Items, and Characters
- Referential Integrity Constraints
 - If you remove a player the corresponding character must be removed
- Uniqueness Constraints
 - All Keys must be unique



Data Probing (Different users)

- Players (of a specific game, looking to play that game, etc.)
- Developers (of a specific game, looking for inspiration, etc.)
- People who follow that game (build theorycrafters, fans of the series, etc.)
- Marketing department (of game studios so they understand how to market the game and items post release)



Data Probing (Users questions)

- Which items provide the best synergy for certain character classes?
- What are the most used items for each character role?
- How does the effectiveness of certain item builds change over time (e.g., with patches or updates)?
- What are the most effective item builds for top-performing players?
- Which item builds lead to the highest win rates in specific RPGs?
- How do item builds differ between casual players and competitive players?
- Are there item builds that are more successful for certain playstyles (e.g., aggressive vs. defensive)?



Future Plans

- Refine the game statistics entity type
- Add more derived attributes
- Add support for more games
- Have more detailed character statistics



Questions?