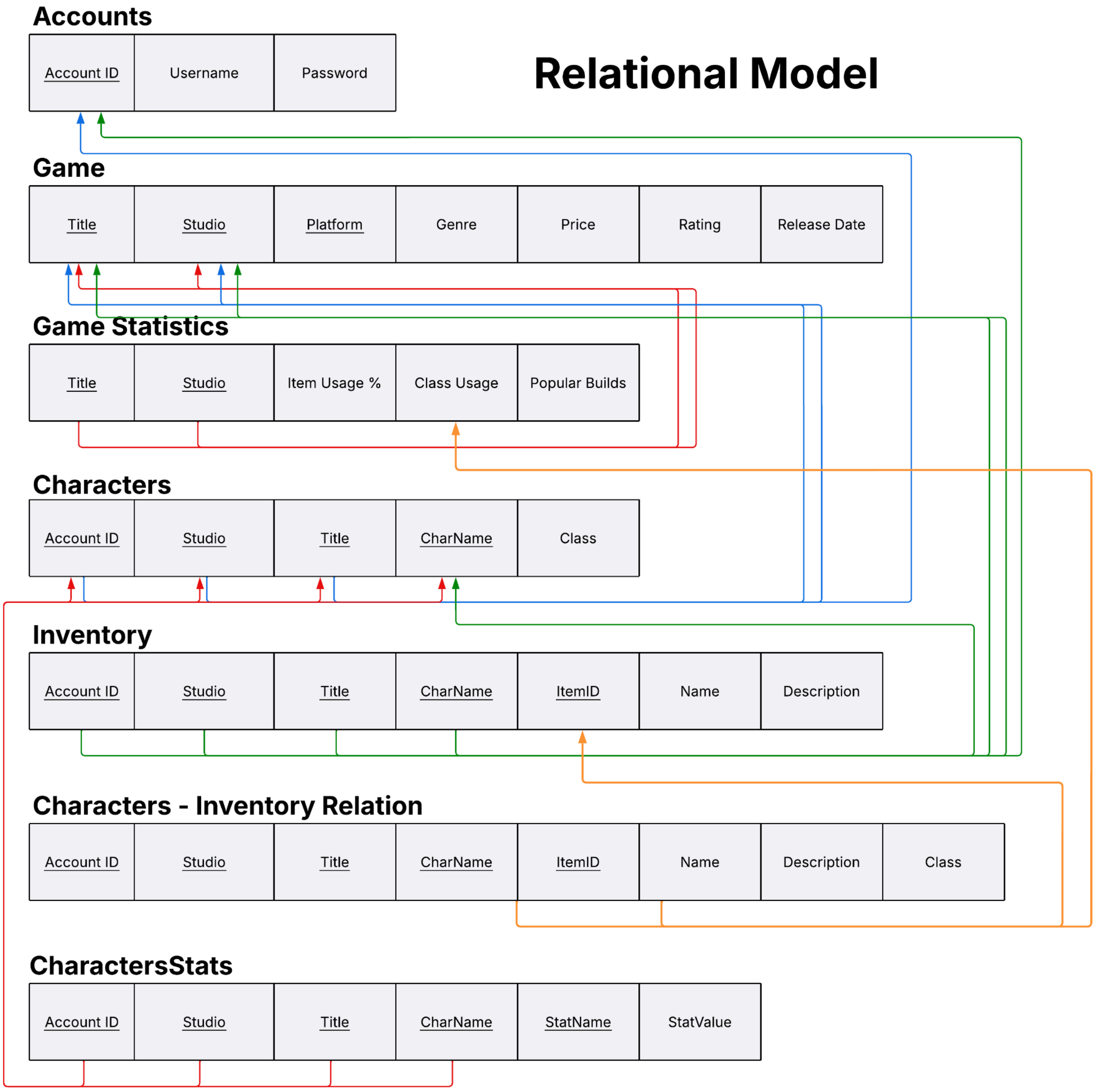
# TW5 Normalize Schema

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

## Schema:

**Figure 1 - Player Data Manager Relational Model**



Relational Model

Accounts

| | | |
|---|---|---|
| Account ID | Username | Password |

Game

| | | | | | | |
|---|---|---|---|---|---|---|
| Title | Studio | Platform | Genre | Price | Rating | Release Date |

Game Statistics

| | | | | |
|---|---|---|---|---|
| Title | Studio | Item Usage % | Class Usage | Popular Builds |

Characters

| | | | | |
|---|---|---|---|---|
| Account ID | Studio | Title | CharName | Class |

Inventory

| | | | | | | |
|---|---|---|---|---|---|---|
| Account ID | Studio | Title | CharName | ItemID | Name | Description |

Characters - Inventory Relation

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Account ID | Studio | Title | CharName | ItemID | Name | Description | Class |

CharactersStats

| | | | | | |
|---|---|---|---|---|---|
| Account ID | Studio | Title | CharName | StatName | StatValue |

# Normalized Schema

## Accounts

PK = (Account ID)
Accounts = ((Account ID), Username, Password)

### 1NF:

Satisfied, no attribute is multi-valued.

### 2NF:

Satisfied, there are no partial dependencies.

### 3NF:

Satisfied, there are no transitive dependencies.

## Game

PK = (Title, Studio, Platform)
Games = ((Title, Studio, Platform), Genre, Price, Rating, Release Date)

### 1NF:

Satisfied, no attribute is multi-valued.

### 2NF:

GameInfo(Title, Studio, Genre, Price, Rating, Release Date)
PlatformRelase(Title, Studio, Platform)

### 3NF:

Game(GameID, Title, Studio, Genre, Rating)
PlatformRelease(ReleaseID, GameID, Platform, Price, ReleaseDate)

## Game Stats

PK = (Title, Studio)
GameStatistics = ((Title, Studio), Item Usage %, Class Usage, Popular Builds)

### 1NF:

GameStats(Title, Studio)
ItemUsage(Title, Studio, ItemName, UsagePercent)
ClassUsage(Title, Studio, ClassName, UsagePercent)
PopularBuillds(Title, Studio, BuildID, Description)

### 2NF:

Satisfied, there are no partial dependencies.

### 3NF:

GameStats(Title, Studio)
ItemUsage(Title, Studio, ItemID, UsagePercent)
Item(ItemID, Name, Description)
ClassUsage(Title, Studio, ClassName, UsagePercent)
Classes(ClassName, Description)
PopularBuilds(Title, Studio, BuildID)
Builds(BuildID, Description)

## Characters

PK = (Account ID, Studio, Title, CharName)
Characters = ((Account ID, Studio, Title, CharName), Class, Stats)

### 1NF:

Satisfied, no attribute is multivalued.

### 2NF:

*The Title is dependent on the Studio, in the case that multiple studios have a game with the same title or vice versa. The Class and Name both depend on the Title and Studio attributes in the case that only certain classes or certain names can be used in certain Titles from certain Studios. Finally, the context of Stats depends on Title and Studio and the relevant account depends on Account ID. These can be separated out into their own groups.*

Game(Title, Studio)
GameClass(Title, Studio, Class)
GameName(Title, Studio, Name).
AccountStats(Account ID, Title, Studio, Stats).

### 3NF:

*The context of Stats depends on Title and Studio and the relevant account depends on Account ID. This can also be separated out into its own group.*

Game(Title, Studio)
GameClass(Title, Studio, Class)
GameName(Title, Studio, Name).
AccountStats(Account ID, Title, Studio, Stats).

## Inventory

PK = (AccountID, Studio, Title, CharName, ItemID)
Inventory((AccountID, Studio, Title, CharName, ItemID), Name, Description)

### 1NF:

Satisfied, no attribute is multivalued.

### 2NF:

*Item name and description are dependent only on ItemID, so break out.*

Item(ItemID, Name, Description)
Inventory(AccountID, Studio, Title, CharName, ItemID)

### 3NF:

Satisfied, there are no transitive dependencies.

## Characters - Inventory Relation

PK = (AccountID, Studio, Title, CharName, ItemID)
CharacterInven((AccountID, Studio, Title, CharName, ItemID), Name, Description, Class)

### 1NF:

Satisfied, no attribute is multivalued.

### 2NF:

*Item name and description are dependent only on ItemID, so break out. Class is dependent on Account ID, Studio, Title, and CharName.*

Item(ItemID, Name, Description)

Characters((AccountID, Studio, Title, CharName), Class)
CharacterInventory(AccountID, Studio, Title, CharName, ItemID)

### 3NF:

Satisfied, there are no transitive dependencies.

## CharacterStats

PK = (AccountID, Studio, Title, CharName, StatName)
CharacterStats((AccountID, Studio, Title, CharName, StatName), StatValue)

### 1NF:

Satisfied, no attribute is multivalued.

### 2NF:

*StatTypes is a new relation that uses StatName as the PK and has a description, this is used to validate stat types. In Character Stats, StatName is a foreign key that references StatTypes. Primary key in stat types makes sure the stat is only in that game by referencing Studio and Title along with the stat name, this is so two games can have the stat "Health" with different descriptions.*

StatTypes((Studio, Title, StatName), Description)
CharacterStats((AccountID, Studio, Title, CharName, StatName), StatValue)

### 3NF:

Satisfied, there are no transitive dependencies.