

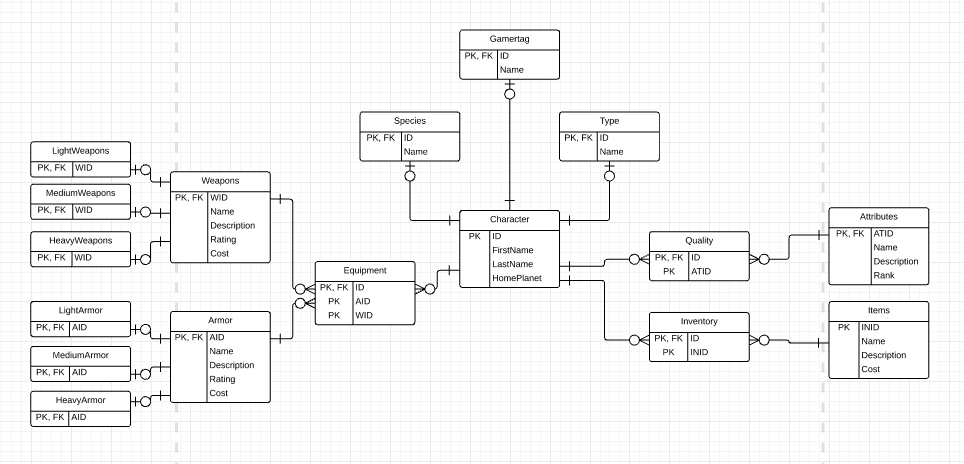
**Database Design Project**

By: Matthew Rahtelli

**Executive Summary**

This database design is intended for the relationships surrounding a Star Wars character based video game. This design will include all of the items, armor, and weapons that can be found in the game. The user is able to see what types of character that they can play, as well as what species they can be. This database also shows the relationships between the character and the items, armor, and weapons that they will find.

This design will go into detail about the relationships each table has in the database. It will also include future expectations and already known problems that exist with the design.

**Entity Relationship Diagram**

**Tables**

Character Table:

CREATE TABLE Character (

ID char(4) not null,

FirstName TEXT,

LastName TEXT,

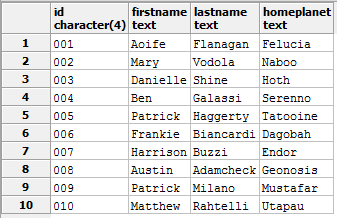
HomePlanet TEXT,

primary key(ID)

);

This table lists all of the characters that have been created or will store any of the characters that will be created in the future. This will list their ID, name, and planet of origin.

Function Dependencies: ID 🡪 FirstName, LastName, HomePlanet



Species Table:

CREATE TABLE Species (

ID char(4) not null REFERENCES Character(ID),

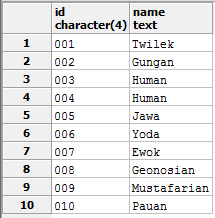
Name TEXT,

primary key(ID)

);

This table lists all of the species that a character can be in the video game. If any more species are created or available to be chosen, it will be listed in this table.

Functional Dependencies: ID 🡪 Name



Type Table

CREATE TABLE Type (

ID char(4) not null REFERENCeS Character(ID),

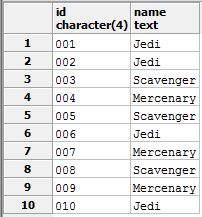
Name TEXT,

primary key(ID)

);

This table lists all of the types that the player can be during the game. This will affect in the future what types of items that they can wield, or developing different types of abilities.

Functional Dependencies: ID 🡪 Name



GamerTag Table

CREATE TABLE Gamertag (

ID char(4) not null REFERENCeS Character(ID),

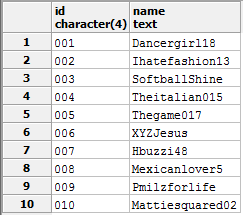
Name TEXT,

primary key(ID)

);

This table lists all of the gamertags of the players who have created characters in the game. This links the character directly with the players account name so users may be able to play together in the future.

Functional Dependencies: ID 🡪 Name



Weapons Table

CREATE TABLE Weapons (

WID char(4) not null,

NAME TEXT,

Description TEXT,

Rating Char(8) NOT NULL,

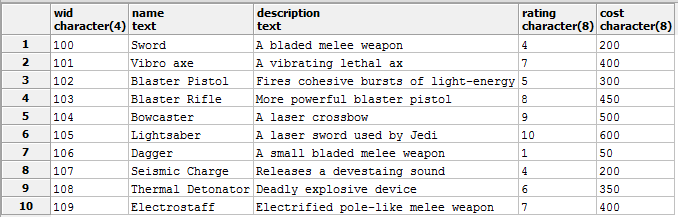
Cost CHAR(8) NOT NULL,

primary key(WID)

);

This table lists all of the weapons that are in the game. It gives a description of what each weapon is. It also shows the rating of the weapon which will affect the skill level of the player. It also determines the cost of the weapon that the player may chose to sell.

Functional Dependencies: WID 🡪 Name, Description, Rating, Cost



LightWeapons Table

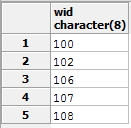
CREATE TABLE LightWeapons (

WID char(8) not NULL REFERENCES Weapons(WID),

primary key(WID)

);

This table lists all of the weapons that are classified as “light”. In the future, some of the types of players or certain species might not able to wield anything but light weapons.



MediumWeapons Table

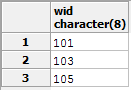
CREATE TABLE MediumWeapons (

WID char(8) not NULL REFERENCES Weapons(WID),

primary key(WID)

);

This table lists all of the weapons that are classified as “medium”. In the future, a character might not be able to wield a medium type weapon.



HeavyWeapons Table

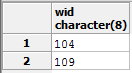
CREATE TABLE HeavyWeapons (

WID char(8) not NULL REFERENCES Weapons(WID),

primary key(WID)

);

This table lists all of the weapons that are classified as “heavy”. In the future, some types or species will be able to wield this type of weapon rather other species or types may not.



Armor Table

CREATE TABLE Armor (

AID char(4) not null,

NAME TEXT,

Description TEXT,

Rating Char(8) NOT NULL,

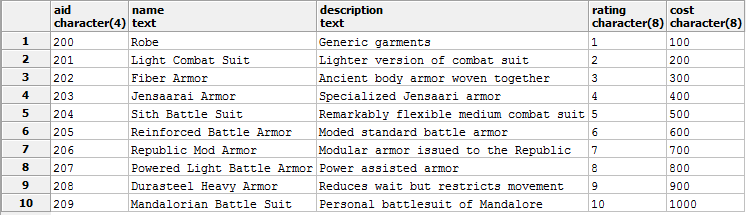
Cost CHAR(8) NOT NULL,

primary key(AID)

);

This table lists all of the armor that are in the game. It gives a description of what the armor is. It also shows the rating of the armor and the affect it will have on the player’s defense. It also determines the cost of the armor that the player may chose to sell.

Functional Dependencies: AID 🡪 Name, Description, Rating, Cost



LightArmor Table

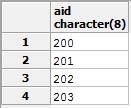
CREATE TABLE LightArmor (

AID char(8) not NULL REFERENCES Armor(AID),

primary key(AID)

);

This table lists all of the armor that is classified as “light armor”. In the future, some types of characters may only be allowed to wear light armor.



MediumArmor Table

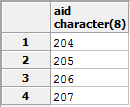
CREATE TABLE MediumArmor (

AID char(8) not NULL REFERENCES Armor(AID),

primary key(AID)

);

This table lists all of the armor that is classified as “medium armor”. In the future, some types of characters may not be permitted or allowed to wear medium armor.



HeavyArmor Table

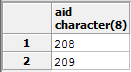
CREATE TABLE HeavyArmor (

AID char(8) not NULL REFERENCES Armor(AID),

primary key(AID)

);

This table lists all of the armor that is classified as “heavy armor”. In the future, some types of characters may not be permitted or allowed to wear heavy armor.



Attributes Table

CREATE TABLE Attributes (

ATID char(4) not null,

NAME TEXT,

Description TEXT,

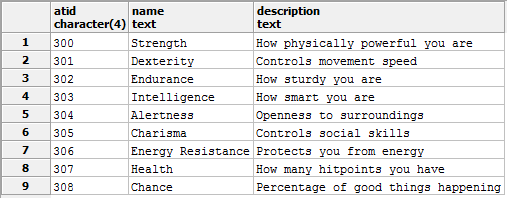
Rank Char(8) NOT NULL,

primary key(ATID)

);

This table lists all of the attributes that apply to the character in the game. These attributes will affect the player’s skill level. In the future, only certain types will be able to access certain attributes.

Functional Dependencies: ATID 🡪 Name, Description, Rank



Items Table

CREATE TABLE Items (

INID char(4) not null,

NAME TEXT,

Description TEXT,

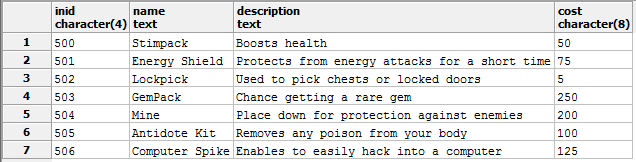
Cost Char(8) NOT NULL,

primary key(INID)

);

This table lists all of the items that can be found in the game. It shows what the item does, what the name of it is, and how much the player can sell it for if the user so chooses. In the future, it will hopefully be able to combine items to weapons or armor to improve them.

Functional Dependencies: INID 🡪 Name, Description, Cost



Equipment Table

CREATE TABLE Equipment (

ID char(4) not null REFERENCeS Character(ID),

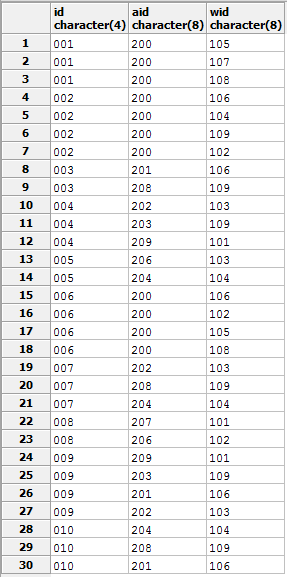
AID char(8) not NULL REFERENCES Armor(AID),

WID char(8) not NULL REFERENCES Weapons(WID),

primary key(ID, AID, WID)

);

This table shows the relationship between the weapons and armor tables to the character table. It shows which weapons and armor are available for the character to find and/or use.



Quality Table

CREATE TABLE Quality (

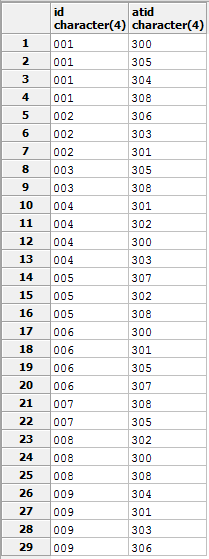
ID char(4) not null REFERENCEs Character(ID),

ATID CHAR(4) NOT NULL REFERENCES Attributes(ATID),

primary KEY(ID, ATID)

);

This table shows the relationship between the character table and the attributes table. This is because of the many to many relationship between each other.



Inventory Table

CREATE TABLE Inventory (

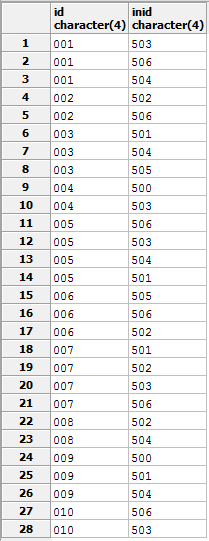
ID char(4) not null REFERENCEs Character(ID),

INID CHAR(4) NOT NULL REFERENCES Items(INID),

primary KEY(ID, INID)

);

This table shows the relationship between the character table and the items table. This is because of the many to many relationship that they have.



**Security**

Administrator: The administrator would be allowed to change content within the database such as altering current content or adding new content.

User: A user would only be able to view the database. The user would not be given permissions to change or add new content.

**Implementation Notes-Known Problems**

The implementation went well but I definitely see a lot of room for improvement. The setup is pretty basic for a regular star wars video game, but there are a lot of possibilities to tackle. One problem to tackle is that each type of character can access all of the armor even if it doesn’t make sense for them to be able to. I think for the design to be so general that there are not too many problems, but to add more details and more enhancements would cause more problems to occur.

**Future Enhancements**

This database could be improved by creating a way for attributes, armor level, and weapon level to increase the skill of the user. This way, the user would be allowed to level up the player and take on harder challenges in the game. It could also be improved by each item in the game having a standard cost and sell price but would change depending on the level of the user to make it hard to buy certain items.

The main update of the design would to only allow certain species to gain access to certain types of weapons and armor. For example, only a jedi should be able to wield a lightsaber or a mercenary to wear heavy armor.