Payday 2 Character Setup

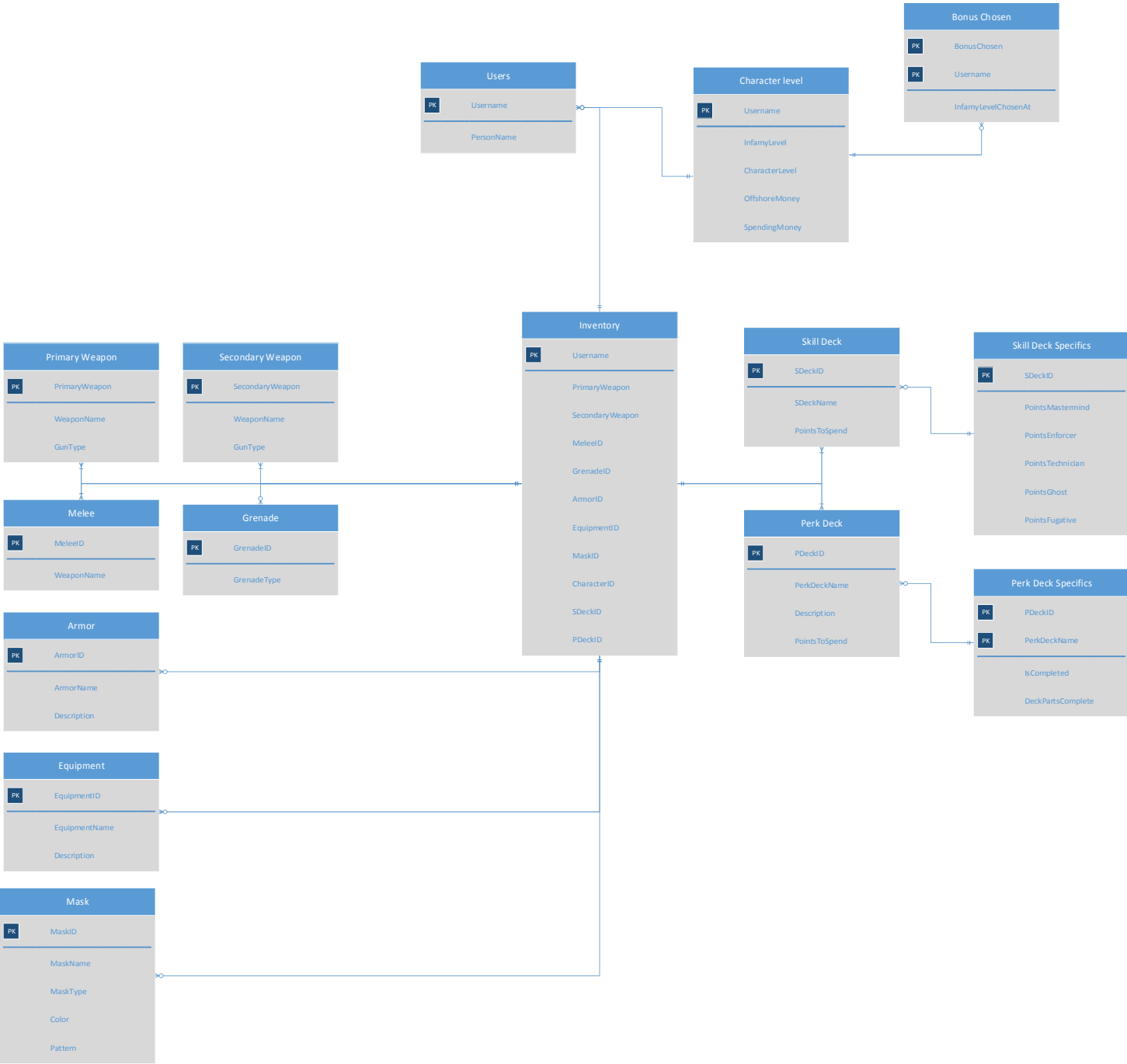# Created by Michael O'Rourke

Table of Contents

Executive Summary

With this document the design and implementation of the Payday 2 Character Setup can be explained and worked through in a simple and efficient manner. The Payday 2 Character Setup helps to clarify and explain what options the user has when getting ready for their next big heist.

The Entity Relationship Diagram (E/R Diagram) gives a visual representation of the database table organization and connections. The E/R Diagram is the basis in which the database was structured and formed and was referenced during the creation of the database.

This document additionally references current problems with the database and future goals. These are problems and enhancements to fix in later updates.

Entity Relationship Diagram

Users

PK	Username
	PersonName

Character level

PK	Username
	InfamyLevel
	CharacterLevel
	OffshoreMoney
	SpendingMoney

Bonus Chosen

PK	BonusChosen
PK	Username
	InfamyLevelChosenAt

Primary Weapon

PK	PrimaryWeapon
	WeaponName
	GunType

Secondary Weapon

PK	SecondaryWeapon
	WeaponName
	GunType

Melee

PK	MeleeID
	WeaponName

Grenade

PK	GrenadeID
	GrenadeType

Armor

PK	ArmorID
	ArmorName
	Description

Equipment

PK	EquipmentID
	EquipmentName
	Description

Mask

PK	MaskID
	MaskName
	MaskType
	Color
	Pattern

Inventory

PK	Username
	PrimaryWeapon
	SecondaryWeapon
	MeleeID
	GrenadeID
	ArmorID
	EquipmentID
	MaskID
	CharacterID
	SDeckID
	PDeckID

Skill Deck

PK	SDeckID
	SDeckName
	PointsToSpend

Skill Deck Specifics

PK	SDeckID
	PointsMastermind
	PointsEnforcer
	PointsTechnician
	PointsGhost
	PointsFugative

Perk Deck

PK	PDeckID
	PerkDeckName
	Description
	PointsToSpend

Perk Deck Specifics

PK	PDeckID
PK	PerkDeckName
	IsCompleted
	DeckPartsComplete

Tables

Users

This is a simple table holding the username and the real name of the players.

Create Table Users (
      Username   text Not Null,
      PersonName   text Not Null,
      Primary Key(Username)
);

Functional Dependencies are based on Username as it is the primary key.

	username text	personname text
1	MOR3	Mike ORourke
2	J007	James Bond
3	GenericGuy	John Smith

Character Level

Create Table Character_Level (
      Username   text Not Null,
      InfamyLevel   int Not Null,
      CharacterLevel   int Not Null,
      OffshoreMoney   int Not Null,
      SpendingMoney   int Not Null,
      Primary Key(Username)
);

The Functional Dependencies are based on Username as it is the primary key.

	username text	infamylevel integer	characterlevel integer	offshoremoney integer	spendingmoney integer
1	MOR3	3	72	3000000	4000000
2	J007	7	52	5000000	3500000
3	GenericGuy	15	97	1000000	9000000

Bonus_Chosen

```
Create Table Bonus_Chosen (
        BonusChosen          text Not Null,
        Username             text Not Null,
        InfamyLevelChosenAt       int Not Null,
        Primary Key (BonusChosen, Username)
);
```

The Functional Dependencies are based on a composite key of BonusChosen and Username as both are important to document what bonus was chosen at what level.

	bonuschosen text	username text	infamylevelchosenat integer
1	unlocking the infamy tree	MOR3	1
2	The Assasin Set	MOR3	2
3	The Slaughterhouse Set	MOR3	3
4	unlocking the infamy tree	J007	1
5	The Assasin Set	J007	2
6	Set of the Corrupted	J007	3
7	The Slaughterhouse Set	J007	4
8	Set of the Ingenious	J007	5
9	More xp1	J007	6
10	More xp2	J007	7
11	unlocking the infamy tree	GenericGuy	1
12	The Assasin Set	GenericGuy	2
13	Set of the Ingenious	GenericGuy	3
14	Set of the Corrupted	GenericGuy	4
15	The Slaughterhouse Set	GenericGuy	5
16	More xp1	GenericGuy	6
17	More xp2	GenericGuy	7
18	More xp3	GenericGuy	8
19	More xp4	GenericGuy	9
20	More xp5	GenericGuy	10
21	More xp6	GenericGuy	11
22	More xp7	GenericGuy	12
23	More xp8	GenericGuy	13
24	Lurker	GenericGuy	14
25	DemonShank	GenericGuy	15

Inventory

```
Create Table Inventory (
        Username        text Not Null,
        PrimaryWeapon          text Not Null,
        SecondaryWeapon     text Not Null,
        MeleeID                text Not Null,
        GrenadeID      text Not Null,
        ArmorID                text Not Null,
        EquipmentID  text Not Null,
        MaskID                 text Not Null,
        CharacterID    text Not Null,
        SDeckID                text Not Null,
        PDeckID                text Not Null,
        Primary Key(Username)
);
```

The Functional Dependencies are based on Username as it is the primary key which holds the active selections for the user.

	username text	primaryweapon text	secondaryweapon text	meleeid text
1	MOR3	Thanatos .50 cal Sniper Rifle	Compact-5 Submachine Gun	Survival Tomahawk
2	J007	Akimbo Bernetti 9 Pistols	Signature .40 Pistol	Weapon Butt
3	GenericGuy	Golden AK.762 Rifle	Street Sweeper Shotgun	Bayonett Knife

grenadeid text	armorid text	equipmentid text	maskid text	characterid text	sdeckid text	pdeckid text
Frag Grenade	Improved Combined Tactical Vest	Ammo Bag	Surprised Santa	Dragan	Techforcer	Gambler
Frag Grenade	Two-Piece Suit	ECM Jammer	The Heat	Hoxton	Stealth	Hitman
Molotov Cocktail	Combined Tactical Vest	Sentry Gun	Mr. Mannequin	Dallas	Tech build	Muscle

Skill Deck

Create Table Skill_Deck (
        SDeckID   text Not Null,
        SDeckName   text Not Null,
        PointsToSpend   int Not Null,
        Primary Key(SDeckID)
);

The Functional Dependencies are based on SDeckID as it is the primary key

	sdeckid text	sdeckname text	pointstospend integer
1	MOR3-1	Techforcer	2
2	MOR3-2	Saw	0
3	MOR3-3	Stealth	4
4	MOR3-4	Build #4	86
5	J007-1	Undercover (stealth)	0
6	J007-2	Dodge Mastermind	2
7	J007-3	Loud Armor	3
8	J007-4	Dodge Tech	6
9	J007-5	Undercover (tech)	1
10	GenericGuy-1	Loud	7
11	GenericGuy-2	Quiet	4
12	GenericGuy-3	Loud Tech	3
13	GenericGuy-4	Quiet Saw	2
14	GenericGuy-5	Loud Saw	6

Perk Deck

Create Table Perk_Deck (
        PDeckID   text Not Null,
        PerkDeckName   text,
        Description   text,
        PointsToSpend   int Not Null,
        Primary Key(PDeckID)
);

The Functional Dependencies are based on PDeckID as it is the primary key.

	pdeckid text	perkdeckname text	description text	pointstospend integer
1	MOR-10	Gambler	When collecting ammo drops player gets	0
2	J007-5	Hitman	Increased concealment	0
3	GenericGuy-2	Muscle	Intimidation and health increase	0

Skill Deck Specifics

```
Create Table Skill_Deck_Specifics (
        SDeckID                 text Not Null,
        PointsMastermind        int Not Null,
        PointsEnforcer              int Not Null,
        PointsTechnician        int Not Null,
        PointsGhost             int Not Null,
        PointsFugative              int Not Null,
        Primary Key(SDeckID)
);
```

The Functional Dependencies are based on SDeckID as it is the primary key.

	sdeckid text	pointsmastermind integer	pointsenforcer integer	pointstechnician integer	pointsghost integer	pointsfugative integer
1	MOR3-1	6	56	10	6	3
2	MOR3-2	6	26	15	35	15
3	MOR3-3	6	6	0	45	25
4	MOR3-4	0	0	0	0	0
5	J007-1	6	6	0	44	25
6	J007-2	30	6	0	10	50
7	J007-3	6	8	48	6	3
8	J007-4	6	8	35	6	46
9	J007-5	6	54	20	4	6
10	GenericGuy-1	6	8	48	6	3
11	GenericGuy-2	6	6	0	45	25
12	GenericGuy-3	6	8	35	6	46
13	GenericGuy-4	6	26	15	35	15
14	GenericGuy-5	6	56	10	6	3

Perk Deck Specifics

Create Table Perk_Deck_Specifics (
        PDeckID   text Not Null,
        PerkDeckName   text Not Null,
        Description   text Not Null,
        IsCompleted   boolean Not Null,
        DeckPartsComplete int Not Null,
        Primary Key(PDeckID, PerkDeckName)
);

The Functional Dependencies are based on the composite key PDeckID and PerkDeckName as there can be many perk decks of the same name as they can not be altered in game so they are attatched to the users perk deck ID making them unique.

	pdeckid text	perkdeckname text	description text	iscompleted boolean	deckpartscomplete integer
1	MOR-1	Crew Cheif	Increases user health and gives team bonuses	t	9
2	MOR-2	Muscle	Intimidation and health increase	t	9
3	MOR-3	Armorer	Improved armor and increased armor recovery time	f	0
4	MOR-4	Rouge	Increase dodge chance and weapon switching	t	9
5	MOR-5	Hitman	Increased concealment	t	9
6	MOR-6	Crook	Increases dodge chance with ballistic vests	t	9
7	MOR-7	Burglar	General bonuses to stealth oriented skills	f	5
8	MOR-8	Infiltrator	Increases damage when surrounded by enemies	t	9
9	MOR-9	Sociopath	increases armor and armor regen in close quarters combat	f	0
10	MOR-10	Gambler	When collecting ammo drops player gets	t	9
11	J007-1	Crew Cheif	Increases user health and gives team bonuses	t	9
12	J007-2	Muscle	Intimidation and health increase	t	9
13	J007-3	Armorer	Improved armor and increased armor recovery time	t	9
14	J007-4	Rouge	Increase dodge chance and weapon switching	t	9
15	J007-5	Hitman	Increased concealment	t	9
16	J007-6	Crook	Increases dodge chance with ballistic vests	t	9
17	J007-7	Burglar	General bonuses to stealth oriented skills	t	9
18	J007-8	Infiltrator	Increases damage when surrounded by enemies	t	9
19	J007-9	Sociopath	increases armor and armor regen in close quarters combat	t	9
20	J007-10	Gambler	When collecting ammo drops player gets	t	9
21	GenericGuy-1	Crew Cheif	Increases user health and gives team bonuses	t	9
22	GenericGuy-2	Muscle	Intimidation and health increase	t	9
23	GenericGuy-3	Armorer	Improved armor and increased armor recovery time	t	9
24	GenericGuy-4	Rouge	Increase dodge chance and weapon switching	t	9
25	GenericGuy-5	Hitman	Increased concealment	t	9
26	GenericGuy-6	Crook	Increases dodge chance with ballistic vests	t	9
27	GenericGuy-7	Burglar	General bonuses to stealth oriented skills	t	9
28	GenericGuy-8	Infiltrator	Increases damage when surrounded by enemies	t	9
29	GenericGuy-9	Sociopath	increases armor and armor regen in close quarters combat	t	9
30	GenericGuy-10	Gambler	When collecting ammo drops player gets	t	9

Primary Weapon

Create Table Primary_Weapon (
      PrimaryWeapon   text Not Null,
      WeaponName text Not Null,
      GunType   text Not Null,
      Primary Key(PrimaryWeapon)
);

The Functional Dependencies are based on PrimaryWeapon as it is the primary key.

	primaryweapon text	weaponname text	guntype text
1	PW1	Thanatos .50 cal Sniper Rifle	Sniper
2	PW2	R93	Sniper
3	PW3	Golden AK.762 Rifle	Assault Rifle
4	PW4	AMR-16	Assualt Rifles
5	PW5	Car-4	Assault Rifles
6	PW6	M308	Assault Rifle
7	PW7	Gewehr 3	Assault Rifle
8	PW8	Buzzsaw 42	Light Machine Guns
9	PW9	RPK	Light Machine Guns
10	PW10	OVE9000	Special
11	PW11	FlameThrower	Special
12	PW12	HRL-7	Special
13	PW13	BJoceline O/U 12G	Shotguns
14	PW14	Locomotive 12G	Shotguns
15	PW15	Akimbo Bernetti 9 Pistols	Akimbo Pistols
16	PW16	Akimbo Deagle	Akimbo Pistols
17	PW17	Akimbo Interceptor 45	Akimbo Pistols

Secondary Weapon

Create Table Secondary_Weapon (
        SecondaryWeapon text Not Null,
        WeaponName text Not Null,
        GunType   text Not Null,
        Primary Key(SecondaryWeapon)
);

The Functional Dependencies are based on SecondaryWeapon as it is the primary key.

	secondaryweapon text	weaponname text	guntype text
1	SW1	Interceptor 45	Pistol
2	SW2	Chimano Compact	Pistol
3	SW3	Bernetti 9	Pistol
4	SW4	Deagle	Pistol
5	SW5	Compact-5	Submachine guns
6	SW6	Uzi	Submachine guns
7	SW7	Patchett L2A1	Submachine guns
8	SW8	Mark 10	Submachine guns
9	SW9	The Judge	Shotguns
10	SW10	Street Sweeper	Shotguns

Melee

Create Table Melee (
      MeleeID   text Not Null,
      WeaponName text Not Null,
      Primary Key(MeleeID)
);

The Functional Dependencies are based on MeleeID as it is the primary key.

	meleeid text	weaponname text
1	MW1	Weapon Butt
2	MW2	350K Brass Knuckles
3	MW3	Money Bundle
4	MW4	Psycho Knife
5	MW5	Spear of Freedom
6	MW6	Surival Tomahawk

Grenade

Create Table Grenade (
      GrenadeID   text Not Null,
      GrenadeType text Not Null,
      Primary Key(GrenadeID)
);

The Functional Dependencies are based on GrenadeID as it is the primary key.

	grenadeid text	grenadetype text
1	G1	Frag Grenade
2	G2	Molotov Cocktail

Armor

Create Table Armor (
        ArmorID   text Not Null,
        ArmorName   text Not Null,
        Description   text Not Null,
        Primary Key(ArmorID)
);

The Functional Dependencies are based on ArmorID as it is the primary key.

	armorid text	armorname text	description text
1	AR1	Two-Piece Suit	This is the standard two-piece suit that any self-respecting criminal should wear
2	AR2	Ballistic Vest	It offers good protection from small-caliber handgun and shotgun projectiles, making it
3	AR3	Lightweight Ballistic Vest	It offers a specially developed trauma liner which provides effective blunt trauma prot
4	AR4	Heavy Ballistic Vest	It is a modern body armor combining a ballistic vests with shoulder and side protection
5	AR5	Flak Jacket	The flak jacket by GenSec Security Industries is a modern take on the classic version.
6	AR6	Combined Tactical Vest	It is made of high performance ballistic materials and is designed to carry heavy load
7	AR7	Improved Combined Tactical Vest	The perfectly fitted interlocking fiber layers rightly woven together with metal plate

Equipment

Create Table Equipment (
        EquipmentID text Not Null,
        EquipmentName   text Not Null,
        Description   text Not Null,
        Primary Key(EquipmentID)
);

The Functional Dependencies are based on EquipmentID as it is the primary key.

	equipmentid text	equipmentname text	description text
1	EQ1	Ammo Bag	The ammo bag is a portable bag, originally intended to allow soldiers to carry extra ammunition
2	EQ2	Armor Bag	The Armor Bag is a portable bag used to transport an armor of your choice that can be used during a h
3	EQ3	Body Bag Case	The body bag case is a case containing body bags that are designed to contain a human body, used for
4	EQ4	Doctor Bag	The doctor bag is a portable bag, usually used by a physician or other medical progression to transpo
5	EQ5	ECM Jammer	An electronic countermeasure (ECM) is an electronic device designed to trick radar, sonar and other d
6	EQ6	First Aid Kit	The first aid kit is a collection of supplies and equipment for use in giving first aid
7	EQ7	Sentry Gun	A sentry gun is a gun that automatically aims and fires at targets that are detected by its sensors.
8	EQ8	Trip Mine	The trip mine is an explosive device and is designed to destroy safes and disable enemy targets as th

```
Create Table Mask (
        MaskID              text Not Null,
        MaskName    text Not Null,
        MaskType    text Not Null,
        Color           text Not Null,
        Pattern        text Not Null,
        Primary Key(MaskID)
);
```

The Functional Dependencies are based on MaskID as it is the primary key.

	maskid text	maskname text	masktype text	color text	pattern text
1	M1	Silent Night	Happy Santa	Default	Default
2	M2	Anonymous	Mr. Mannequin	White	Money
3	M3	Nixon	The 37th	Green and Blue	Monkey
4	M4	Flames	The Devil	Red	Default
5	M5	Not So Silent Night	Surprised Santa	Default	Luchador

Views

Name and Level

This view is used to display the username, name and levels of the user for a quick view of their overall game progress.

Create View NameLevel
As
Select Distinct Users.Username, Users.PersonName, Character_Level.InfamyLevel, Character_level.Characterlevel
from Users join
      Character_level on
      Users.Username = Character_Level.Username;

	username text	personname text	infamylevel integer	characterlevel integer
1	GenericGuy	John Smith	15	97
2	J007	James Bond	7	52
3	MOR3	Mike ORourke	3	72

Queries
Primary Gun in use

Queries the gun from options and if a user has it active.
Note: if the guntype is not one that any user has active it will result in a blank query

select Primary_Weapon.PrimaryWeapon, Primary_Weapon.WeaponName, Inventory.Username
from Inventory join
      Primary_Weapon on
      Primary_Weapon.PrimaryWeapon = Inventory.PrimaryWeapon
where Primary_Weapon.guntype = 'Sniper'

primaryweapon text	weaponname text	username text
PW1	Thanatos .50 cal Sniper Rifle	MOR3

Secondary Gun in use

Queries the gun from options and if a user has it active.
Note: if the guntype is not one that any user has active it will result in a blank query

select Secondary_Weapon.SecondaryWeapon, Secondary_Weapon.WeaponName, Inventory.Username
from Inventory join
      Secondary_Weapon on
      Secondary_Weapon.SecondaryWeapon = Inventory.PrimaryWeapon
where Secondary_Weapon.guntype = 'Pistol'

	secondaryweapon text	weaponname text	username text
1	SW2	Signature .40 Pistol	J007

Store Procedures

Change Gun Name

This stored procedure was intended to help the admin change the name of a gun quickly without having to go through the table and change the entity manually.

```
Create or Replace Function GunNameChange (text, refcursor) returns refcursor as $$
declare
        name_gun int := $1;
        resultset refcursor := $2;
begin
        open resultset for
        select PrimaryWeapon, WeaponName
        from PrimaryWeapon;
end;
$$
language plpgsql;
```

NOTE:
Known Problem: Function not quite running correctly.

Security

        The two people accessing this database would be the Admin who has full privilege to do any type of modifying to the database and the users who can view various parts but are unable to change anything.

Admin

Create Role Admin
Grant select, insert, update, delete
on all tables in schema public
to Admin

        The reason the users do not have more access is that all of their changes are done through the game rather than the database itself calling upon changes via in game commands.

Users

Create Role Users
grant select
on all tables in schema public
to Users

Implementation Notes

The creation of the database was smooth with the major bump in the road being how much of the game would make sense to implement for this project. As the game has hundreds of weapon and mask entities it would be difficult to specify them in sample data and thus only a few were chosen. One of the largest parts of the weapons was the modifications that were able to be done in game was left out due to the vast number of tabs and modifications that could be done to each weapon which was outside the scope of this project. However, the mask customizations were few enough that they were able to be placed into the project with a few examples.

Known Problems

The first currently known problems are issues with the stored procedures that were intentionally meant to be implemented were cut out as the in game commands are largely used to modify the database rather than many stored procedures.

A second issue is that there is a lack of views due to the inventory being the largest source of information as this view gives a good overview of active items the user chose in game to use. Due to this being the largest central entity it is difficult to see everything the user has unlocked unless a specific table is viewed rather than a mixture.

The database is lacking triggers due in two parts, first because I was unable to get them to work and rather than have pieces of not working scrap code I cleaned up the sql and removed them. Second was they did not make as much sense in the database because the Inventory table does not need to trigger anything when being changed, rather it needs to just be updated when the user makes a change.

Future Enhancements

  Looking toward the future the addition of the vast amounts of customizations to the weapons would be a great piece to implement into the database adding another layer of complexity. Additionally, a trigger or multiple triggers to activate the addition of a starter character for new users; when a new user logs in for the first time it triggers the database to add them to all the necessary tables.