Mass Effect 2 Character Generator Database

Designed By: Robert Coords

Table of Contents

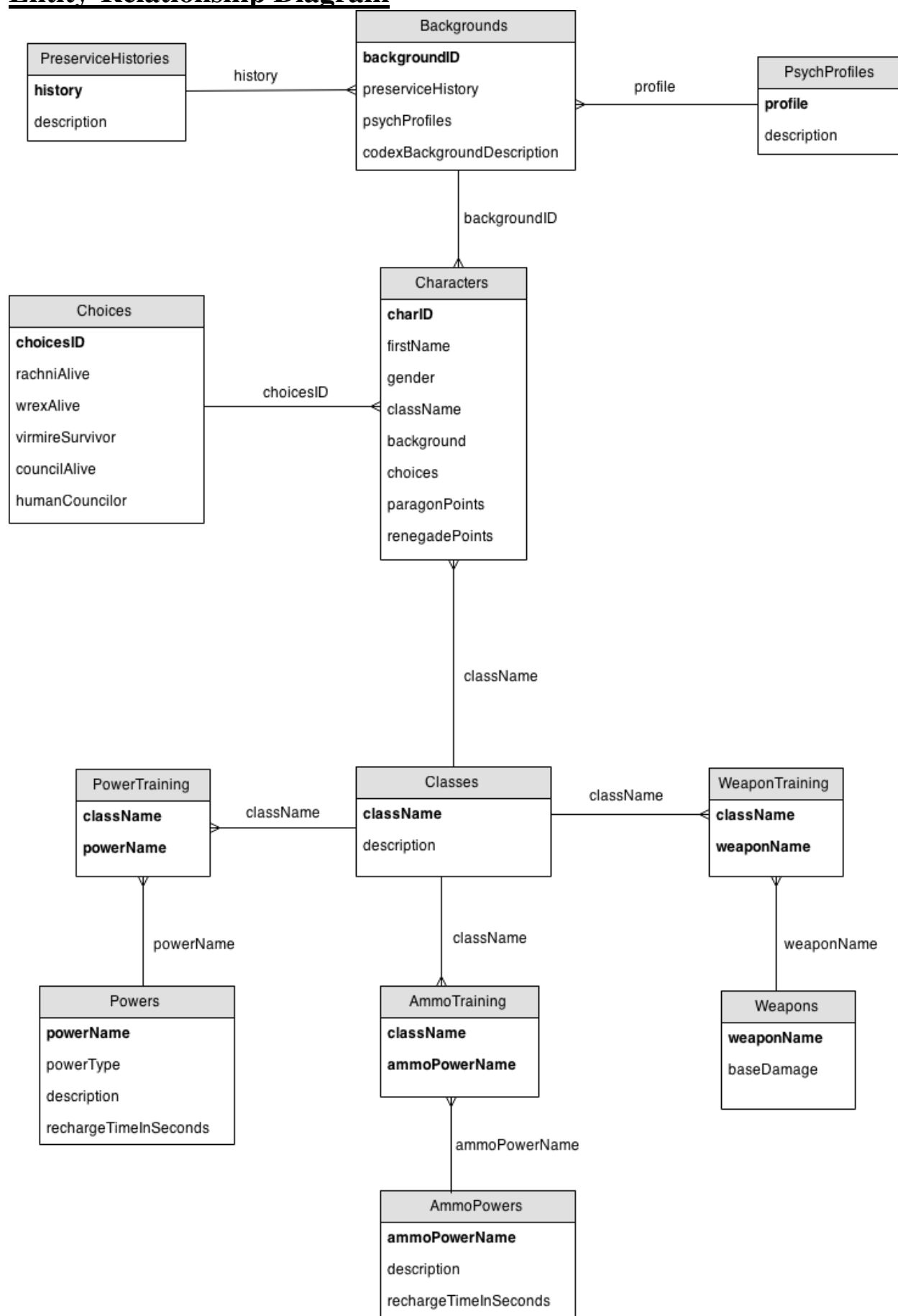## **Executive Summary**

This document represents the design and implementation of a database for the Mass Effect 2 character creator. The database is an organized way to see one's Mass Effect 2 character, while also providing a way to compare one's character to other player's characters. Potential users of this database would be gamers, specifically those interested in the Mass Effect franchise.

This document begins by showing an Entity-Relationship Diagram for the database, which gives an overall view of the database and shows how its tables relate to one another. We then dive into the database's individual tables, explaining their purpose and showing their SQL code, functional dependencies, and some sample data. The tables are followed by a series of reports, showing queries that can be executed on these tables. We continue with a look into the database's views and stored procedures, followed by a glimpse into the security measures implemented to keep the data in the Mass Effect 2 character creator database safe. Finally, this document closes with a few notes from the designer regarding implementation, the known problems within this database, and a preview of future enhancements that may be made to this database.

Entity-Relationship Diagram

PreserviceHistories
- **history**
- description

— history —<

Backgrounds
- **backgroundID**
- preserviceHistory
- psychProfiles
- codexBackgroundDescription

— profile —<

PsychProfiles
- **profile**
- description

backgroundID

Characters
- **charID**
- firstName
- gender
- className
- background
- choices
- paragonPoints
- renegadePoints

Choices
- **choicesID**
- rachniAlive
- wrexAlive
- virmireSurvivor
- councilAlive
- humanCouncilor

— choicesID —<

className

PowerTraining
- **className**
- **powerName**

— className —<

Classes
- **className**
- description

— className —

WeaponTraining
- **className**
- **weaponName**

powerName

Powers
- **powerName**
- powerType
- description
- rechargeTimeInSeconds

className

AmmoTraining
- **className**
- **ammoPowerName**

weaponName

Weapons
- **weaponName**
- baseDamage

ammoPowerName

AmmoPowers
- **ammoPowerName**
- description
- rechargeTimeInSeconds

Tables

Characters Table

The Characters table holds information on any player-created character in Mass Effect 2.

```
-- Characters --
create table characters (
        charID              char(4) not null,
        firstName           text not null,
        gender              text check (gender='male' or gender='female'),
        className           text not null references classes(className),
        background          char(3) not null references backgrounds(backgroundID),
        choices             char(4) references choices(choicesID),
        paragonPoints       integer default 0,
        renegadePoints      integer default 0,
        primary key (charID)
);
```

**Note: All player-created characters in Mass Effect 2 have the same last name (which is 'Shepard'). Each player chooses his own first name for his character. This is why last name is not stored in the database, but first name is.

Functional Dependencies:

charID → firstName, gender, className, background, choices, paragonPoints, renegadePoints

Sample Data:

	charid character(4)	firstname text	gender text	classname text	background character(3)	choices character(4)	paragonpoints integer	renegadepoints integer
1	c000	John	male	Soldier	b00	ch00	0	0
2	c001	Jane	female	Soldier	b00	ch00	0	0
3	c003	R.	male	Vanguard	b07	ch01	110	40
4	c004	Jane	female	Infiltrator	b04	ch03	170	20
5	c006	Jennifer	female	Adept	b03	ch04	150	40
6	c010	Mark	male	Infiltrator	b04	ch10	30	100
7	c021	Alex	male	Engineer	b08	ch15	95	95

Classes Table

The Classes table stores basic information regarding the playable classes in Mass Effect 2. More detailed information on each class's individual powers can be found in the PowerTraining, AmmoTraining, and WeaponTraining tables.

```
-- Classes --
create table classes (
        className    text check (className='Adept' or className='Soldier' or
                     className='Engineer' or className='Vanguard' or className='Sentinel'
                     or className='Infiltrator'),
        description  text,
        primary key (className)
);
```

Functional Dependencies:

className → description

Sample Data:

	classname text	description text
1	Adept	Adepts are outfitted with L5x implants that spawn micro-singularities, blocking enemies and pullin them into the air.
2	Soldier	High-level operatives are outfitted with an ocular synaptic processor that allows them to focus on targets with lethal accuracy.
3	Engineer	Engineers can spawn a combat drone to harass enemies or force them out of entrenched cover positions.
4	Vanguard	Vanguards are outfitted with L5n implants that use biotic charges to knock down or stun opponents. This gives the Vanguard
5	Sentinel	Sentinels are equipped with the most advanced ablation armor system to keep the Sentinel safe. If overloaded, the system stuns
6	Infiltrator	Infiltrators are equipped with a tactical cloak system that allows them to avoid detection for short periods of time, gaining

Powers Table

The Powers table stores basic information on the powers that can be used by players in Mass
Effect 2.

```
-- Powers --
create table powers (
        powerName                       text not null,
        powerType                       text check (powerType='Combat' or
                                        powerType='Tech' or powerType='Biotic'),
        description                     text not null,
        rechargeTimeInSeconds           numeric(4,2) not null,
        primary key (powerName)
);
```

Functional Dependencies:

powerName → powerType, description, rechargeTimeInSeconds

Sample Data:

	powername text	powertype text	description text	rechargetimeinseconds numeric(4,2)
1	Adrenaline Rush	Combat	Accelerate reflexes, granting time to line up the perfect shot.	5.00
2	Concussive Shot	Combat	Flatten your enemy with a precise blast at short or long range.	6.00
3	AI Hacking	Tech	Allows the player to hack robotic enemies.	6.00
4	Cryo Blast	Tech	Flash-freeze and shatter unprotected enemies. Slow down the rest.	4.50
5	Combat Drone	Tech	Deploy this attack drone to stun targets and draw enemy fire.	3.00
6	Incinerate	Tech	Burn your opponents and incinerate their armor.	6.00
7	Overload	Tech	Overload electronics with this power surge, stunning your enemy.	6.00
8	Tactical Cloak	Tech	Become invisible. Gain a massive damage bonus when breaking from cloak to attack.	6.00
9	Tech Armor	Tech	Protect yourself with this holographic armor or detonate it to damage nearby enemies.	12.00
10	Charge	Biotic	Smash into a target while encased in this biotic barrier, leveling your opponents.	6.00
11	Pull	Biotic	Yank an opponent helplessly off the ground.	3.00
12	Singularity	Biotic	Create a sphere of dark energy that traps and dangles enemies caught in its field.	4.50
13	Shockwave	Biotic	Topple a row of enemies with this cascading shockwave.	6.00
14	Throw	Biotic	Toss your enemy through the air with this biotic blast.	3.00
15	Warp	Biotic	Rip your enemy apart at a molecular level.	6.00

AmmoPowers Table

The AmmoPowers table stores basic information on the ammunition power upgrades that can be used by players in Mass Effect 2.

```
-- AmmoPowers --
create table ammoPowers (
        ammoPowerName                           text not null,
        description                             text not null,
        rechargeTimeInSeconds                   numeric(4,2) not null,
        primary key (ammoPowerName)
);
```

Functional Dependencies:

ammoPowerName → description, rechargeTimeInSeconds

Sample Data:

	ammopowername text	description text	rechargetimeinseconds numeric(4,2)
1	Cryo Ammo	Flash freeze and shatter unprotected enemies.	1.50
2	Disruptor Ammo	Bring down the barrier and shields of your opponents.	1.50
3	Incendiary Ammo	Shoot and your enemies will burst into flames.	1.50

Weapons Table

The Weapons table stores basic information on the weapons used by players in Mass Effect 2.

```
-- Weapons--
create table weapons (
        weaponName          text not null,
        baseDamage          numeric(5,2),
        primary key (weaponName)
);
```

<u>Functional Dependencies:</u>

weaponName → baseDamage

<u>Sample Data:</u>

	weaponname text	basedamage numeric(5,2)
1	Assault Rifle	10.80
2	Sniper Rifle	263.10
3	Shotgun	27.50
4	Submachine Gun	20.50
5	Heavy Pistol	37.20
6	Heavy Weapon	500.00

Power Training

The PowerTraining table links the powers from the Powers table with the classes from the
Classes table, based on which class is trained to use which power.

-- PowerTraining --
create table powerTraining (
        className   text not null references classes(className),
        powerName   text not null references powers(powerName),
      primary key(className, powerName)
);

Functional Dependencies:

className, powerName →

Sample Data:

	classname text	powername text
1	Soldier	Adrenaline Rush
2	Soldier	Concussive Shot
3	Engineer	AI Hacking
4	Infiltrator	AI Hacking
5	Engineer	Cryo Blast
6	Sentinel	Cryo Blast
7	Engineer	Combat Drone
8	Engineer	Incinerate
9	Infiltrator	Incinerate
10	Engineer	Overload
11	Sentinel	Overload
12	Infiltrator	Tactical Cloak
13	Sentinel	Tech Armor
14	Vanguard	Charge
15	Adept	Pull
16	Vanguard	Pull
17	Adept	Singularity
18	Adept	Shockwave
19	Vanguard	Shockwave
20	Adept	Throw
21	Sentinel	Throw
22	Adept	Warp
23	Sentinel	Warp

AmmoTraining Table

The AmmoTraining table links the ammo powers from the AmmoPowers table with the classes from the Classes table, based on which class is trained to use which ammo power.

-- AmmoTraining --
create table ammoTraining (
        className   text not null references classes(className),
        ammoPowerName   text not null references ammoPowers(ammoPowerName),
        primary key (className, ammoPowerName)
);

Functional Dependencies:

className, ammoPowerName →

Sample Data:

	classname text	ammopowername text
1	Infiltrator	Cryo Ammo
2	Soldier	Cryo Ammo
3	Vanguard	Cryo Ammo
4	Infiltrator	Disruptor Ammo
5	Soldier	Disruptor Ammo
6	Soldier	Incendiary Ammo
7	Vanguard	Incendiary Ammo

WeaponTraining Table

The WeaponTraining table links the weapons from the Weapons table with the classes from the Classes table, based on which class is trained to use which weapon.

-- WeaponTraining --
create table weaponTraining (
      className   text not null references classes(className),
      weaponName   text not null references weapons(weaponName),
    primary key (className, weaponName)
);

Functional Dependencies:

className, weaponName →

Sample Data:

	classname text	weaponname text
1	Soldier	Assault Rifle
2	Soldier	Sniper Rifle
3	Infiltrator	Sniper Rifle
4	Adept	Submachine Gun
5	Engineer	Submachine Gun
6	Infiltrator	Submachine Gun
7	Vanguard	Submachine Gun
8	Sentinel	Submachine Gun
9	Soldier	Shotgun
10	Vanguard	Shotgun
11	Adept	Heavy Pistol
12	Engineer	Heavy Pistol
13	Infiltrator	Heavy Pistol
14	Vanguard	Heavy Pistol
15	Sentinel	Heavy Pistol
16	Soldier	Heavy Pistol
17	Adept	Heavy Weapon
18	Engineer	Heavy Weapon
19	Infiltrator	Heavy Weapon
20	Vanguard	Heavy Weapon
21	Sentinel	Heavy Weapon
22	Soldier	Heavy Weapon

Backgrounds

The Backgrounds table stores basic information on the player's background, using information from the PreserviceHistories and PsychProfiles tables.

```
-- Backgrounds --
create table backgrounds (
        backgroundID                    char(3) not null,
        preserviceHistory               text not null references preserviceHistories(history),
        psychProfile                    text not null references psychProfiles(profile),
        codexBackgroundDescription      text not null,
        primary key (backgroundID)
);
```

Functional Dependencies:

backgroundID → preserviceHistory, psychProfile, codexBackgroundDescription

Sample Data:

	backgroundid character(3)	preservicehistory text	psychprofile text	codexbackgrounddescription text
1	b00	Spacer	Sole Survivor	Both of your parents were in the Alliance military. Your childhood was spent on ships and stations as they transferred
2	b01	Spacer	War Hero	Born into a naval family, you spent your childhood on ships and stations. You moved from posting to posting as your parents were
3	b02	Spacer	Ruthless	Born into a naval family, you spent your childhood on ships and stations. You moved from posting to posting as your parents were
4	b03	Colonist	Sole Survivor	You were raised on Mindoir on the fringes of the Attican Traverse. When you were sixteen, the colony was raided by slavers. The
5	b04	Colonist	War Hero	You were raised on Mindoir on the fringes of the Attican Traverse. When you were sixteen, the colony was raided by slavers. The entire
6	b05	Colonist	Ruthless	You were raised on Mindoir on the fringes of the Attican Traverse. When you were sixteen, the colony was raided by slavers. The entire
7	b06	Earthborn	Sole Survivor	You were born on Earth, but you never knew your parents. A child of the streets, you learned to live by your wits and guts, surviving
8	b07	Earthborn	War Hero	You were born on Earth, but you never knew your parents. A child of the streets, you learned to live by your wits and guts, surviving in the
9	b08	Earthborn	Ruthless	You were born on Earth, but you never knew your parents. A child of the streets, you learned to live by your wits and guts, surviving in the

PreserviceHistories Table

The PreserviceHistories table stores basic information on the preservice histories available for player-created characters in Mass Effect 2.

```
-- PreserviceHistories --
create table preserviceHistories (
        history         text check (history='Spacer' or history='Colonist' or history='Earthborn'),
         description     text,
        primary key (history)
);
```

Functional Dependencies:

history → description

Sample Data:

	history text	description text
1	Spacer	Both of your parents were in the Alliance military. Your childhood was spent on ships and stations as they transferred from posting
2	Earthborn	You were an orphan raised on the streets of the great megatropolises covering Earth. You escaped the life of petty crime and
3	Colonist	You were born and raised on Mindoir, a small border colony in the Attican Traverse. When you were sixteen slavers raided Mindoir,

PsychProfiles Table

The PsychProfiles table storesbasic information on the psychological profiles available for player-created characters in Mass Effect 2.

```
-- PsychProfiles --
create table psychProfiles (
        profile          text check (profile='War Hero' or profile='Sole Survivor' or profile='Ruthless'),
        description      text,
        primary key (profile)
);
```

Functional Dependencies:

profile → description

Sample Data:

	profile text	description text
1	Sole Survivor	During your service, a mission you were on went horribly wrong. Trapped in an extreme survival situation, you had to overcome
2	War Hero	Early in your military career you found yourself facing an overwhelming enemy force. You risked your own life to save your fellow
3	Ruthless	Throughout your military career, you have held fast to one basic rule: get the job done. You have been called cold, calculating, and

Choices Table

The Choices table stores the choices made by players in Mass Effect 1 that were imported into Mass Effect 2 when they imported their character.

```
-- Choices --
create table choices (
        choicesID           char(4) not null,
        rachniAlive         boolean default false,
        wrexAlive           boolean default false,
        virmireSurvivor     text check (virmireSurvivor='Ashley' or virmireSurvivor='Kaidan'),
        councilAlive        boolean default false,
        humanCouncilor      text default 'Udina' check (humanCouncilor='Anderson' or humanCouncilor='Udina'),
        primary key (choicesID)
);
```

Functional Dependencies:

choicesID → rachniAlive, wrexAlive, virmireSurvivor, councilAlive, humanCouncilor

Sample Data:

	choicesid character(4)	rachnialive boolean	wrexalive boolean	virmiresurvivor text	councilalive boolean	humancouncilor text
1	ch00	f	f	Ashley	f	Udina
2	ch01	t	f	Ashley	t	Anderson
3	ch03	t	t	Ashley	t	Anderson
4	ch04	f	t	Kaidan	f	Udina
5	ch10	f	f	Kaidan	t	Anderson
6	ch15	t	f	Ashley	f	Udina

Reports

Characters' First Names with over 100 Paragon Points

Displays all first names of characters who have more than 100 paragon points.

select firstName
from characters
where paragonPoints > 100;

Sample Data:

	firstname text
1	R.
2	Jane
3	Jennifer

Powers used by Adept Class with Recharge Time over 5 Seconds

Displays all information for every power used by the Adept class that has a recharge time that is greater than 5 seconds.

select p.*
from powers p inner join powerTraining pt
  on p.powerName = pt.powerName
where pt.className = 'Adept'
  and p.rechargeTimeInSeconds > 5.00;

Sample Data:

	powername text	powertype text	description text	rechargetimeinseconds numeric(4,2)
1	Shockwave	Biotic	Topple a row of enemies with this cascading shockwave.	6.00
2	Warp	Biotic	Rip your enemy apart at a molecular level.	6.00

Characters who can use the Shotgun

Displays the charIDs and first names of all characters that are of a class that is trained to use the shotgun.

select c.charID, c.firstName

from characters c inner join weaponTraining w

  on c.className = w.className

where w.weaponName = 'Shotgun';

Sample Data:

	charid character(4)	firstname text
1	c000	John
2	c001	Jane
3	c003	R.

Views

MoralStances View

Displays each character and his paragon/ renegade scores.

-- MoralStances
create view moralStances (charID, firstName, paragonPoints, renegadePoints)
as
      select charID, firstName, paragonPoints, renegadePoints
      from characters;

Sample Data:

	charid character(4)	firstname text	paragonpoints integer	renegadepoints integer
1	c000	John	0	0
2	c001	Jane	0	0
3	c003	R.	110	40
4	c004	Jane	170	20
5	c006	Jennifer	150	40
6	c010	Mark	30	100
7	c021	Alex	95	95

Stored Procedures

PowersForCharacter()

The function returns the powers available to the character with the given id. The refcursor parameter is the name of the returned table.

```
-- PowersForCharacter(id, resultset)
create or replace function PowersForCharacter(char(4), refcursor) returns refcursor as
$$
declare
        id char(4) := $1;
        resultset refcursor := $2;
begin
        open resultset for
                select p.powerName
                from powerTraining p, characters c
                where p.className = c.className
                  and c.charID = id;
        return resultset;
end;
$$
language plpgsql;
```

Sample Data:

```
select powersForCharacter('c003', 'results');
fetch all from "results";
```

	powername text
1	Charge
2	Pull
3	Shockwave

WeaponsForCharacter()

The function returns the weapons available to the character with the given id. The refcursor parameter is the name of the returned table.

```
-- WeaponsForCharacter(id, resultset)
create or replace function WeaponsForCharacter(char(4), refcursor) returns refcursor as
$$
declare
        id char(4) := $1;
        resultset refcursor := $2;
begin
        open resultset for
                select w.weaponName
                from weaponTraining w, characters c
                where w.className = c.className
                  and c.charID = id;
        return resultset;
end;
$$
language plpgsql;
```

Sample Data:

```
select weaponsForCharacter('c003', 'results');
fetch all from "results";
```

	weaponname text
1	Submachine Gun
2	Shotgun
3	Heavy Pistol
4	Heavy Weapon

Security

There are three types of users for this database.

1. The admin, who is given all privileges for database maintenance.

   -- Admin role
   create role admin;
   grant insert, update, delete, select on all tables in schema public to admin;

2. The player, who can view any table in the database. They can also add and delete their player-created characters from the characters table, as well as add and delete their choices from the choices table.

   -- Player role
   create role player;
   grant select on all tables in schema public to player;
   grant insert, delete on characters to player;
   grant insert, delete on choices to player;

3. The public user, who can only view all the tables in the database.

   -- PublicUser role
   create role publicUser;
   grant select on all tables in schema public to publicUser;

Implementation Notes/ Known Problems/ Future Enhancements

The implementation of the database went fairly well. One issue I encountered early on involved whether or not to fully implement the appearance editor for Mass Effect 2, which allows players to customize the appearance of their characters. While the idea of implementing this concept was appealing, I noticed that Bioware had chosen to let the players customize their characters' appearance using a series of sliders, creating a lot of meaningless data that would be hard to give context to within the database. In the end, I chose not to reflect this part of the character creator in the database's current version.

There is room for this database to expand and improve, beyond simply fixing these known problems. One future enhancement that would benefit the Mass Effect 2 database would be the addition of a series of tables allowing the players to customize their characters' armor. Additionally, more player choices imported from Mass Effect 1 can be added to the Choices table. These are just 2 enhancements that will help to improve the Mass Effect 2 database in future years.