Brian Laccone
CS 340
3/15/2018

Final Project

Outline

The topic that was chosen for this project is a database representing a simple strategy game. The user will be able to create fighters with different names, races, and classes. The user will also be able to obtain powerful weapons and armor that their fighters can equip. Fighter's can be organized into squads and have a chance to obtain power-ups throughout their adventures. While this topic could involve several different tables, five entities have been chosen for this project, mainly to represent how fighter's will interact with objects in the game.

Database Outline

Fighter Table

This table is the most complicated table out of the six tables created for this project. It will be using "id" as the primary key and will have name, race, and class as varchar(255). These won't be requiring any constraints or foreign keys because, since this is a video game, you can choose to name every character the same name as well as have them the same race or class. The last three attributes are squad_id, weapon_id, and armor_id. These are foreign keys that reference their respective "id" attributes in the squad, weapon, and armor tables. They can also be null. Weapon_id and armor_id will have unique constraints on them because only one weapon and one armor can be equipped to a fighter. A fighter can not use the same weapon or armor as another fighter. Squad_id will not have a unique constraint because squads can hold many fighters and many fighters can be in the same squad as other fighters.

Squad Table

The squad table is a simple table that holds the id (as the primary key), name, and gold amount of each squad. Since it is a video game there will not be a unique constraint on the name attribute. A squad can also exist without any fighter in it because it needs to exist in the game before the user assigns fighters to it. The squad can also exist after all the fighters in the squad die because the user can then buy more fighters.

Weapon Table

The weapon table will be holding the information of each weapon. It consists of id (as the primary key), name, type, quality, durability, and damage. Name, type and durability will be varchar and durability and damage will be ints. Name is not unique because multiple items with the same name can drop during the game. Weapons exist without being equipped to a fighter.

Armor Table

This table is the exact same as the weapon table except that damage is changed to damage_reduction. Armor exists without being equipped to a fighter.

Power-Up Table

The power up table consists of id, name, and type. Since it is a video game there will not be a unique constraint on the name attribute.

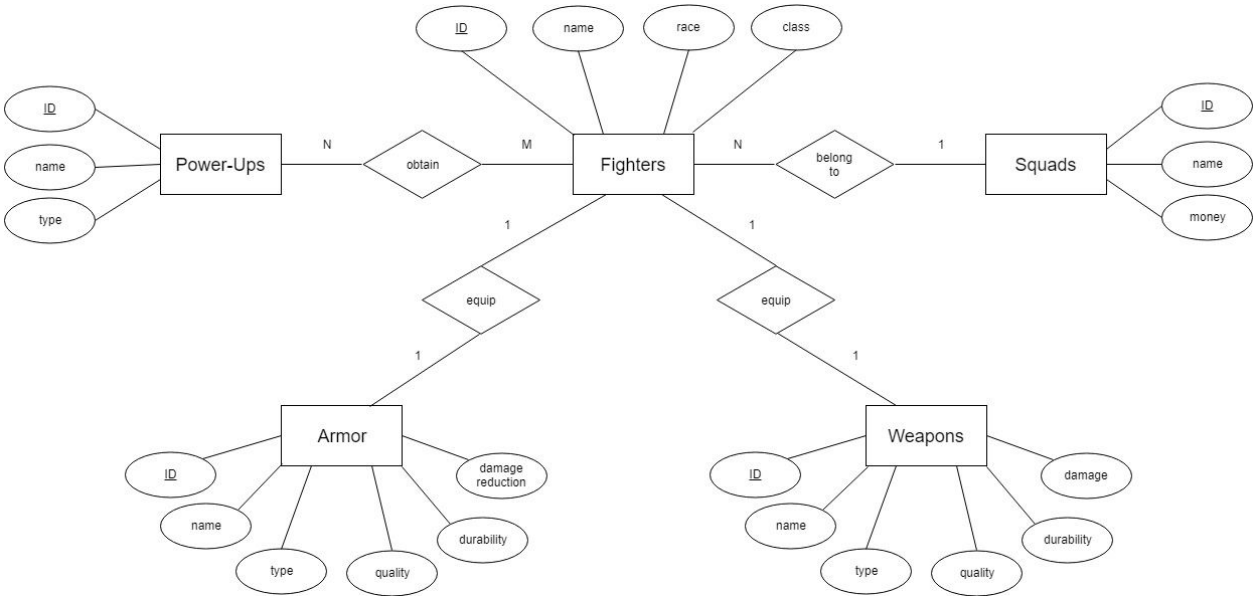
Fighter - Power-Up Table

This table consists of id (as the primary key), fighter_id, and power_up_id. The fighter_id is a foreign key that is a reference to an id value in the fighter table and power_up_id is a foreign key that is a reference to an id value in the power_up table. Id will be used as the primary key instead of a combination of just fighter_id and power_up_id because this is a many to many relationship between fighter and

power_up. The id is necessary because the same fighter can get the same power-up multiple times. The power-ups stack in the game.

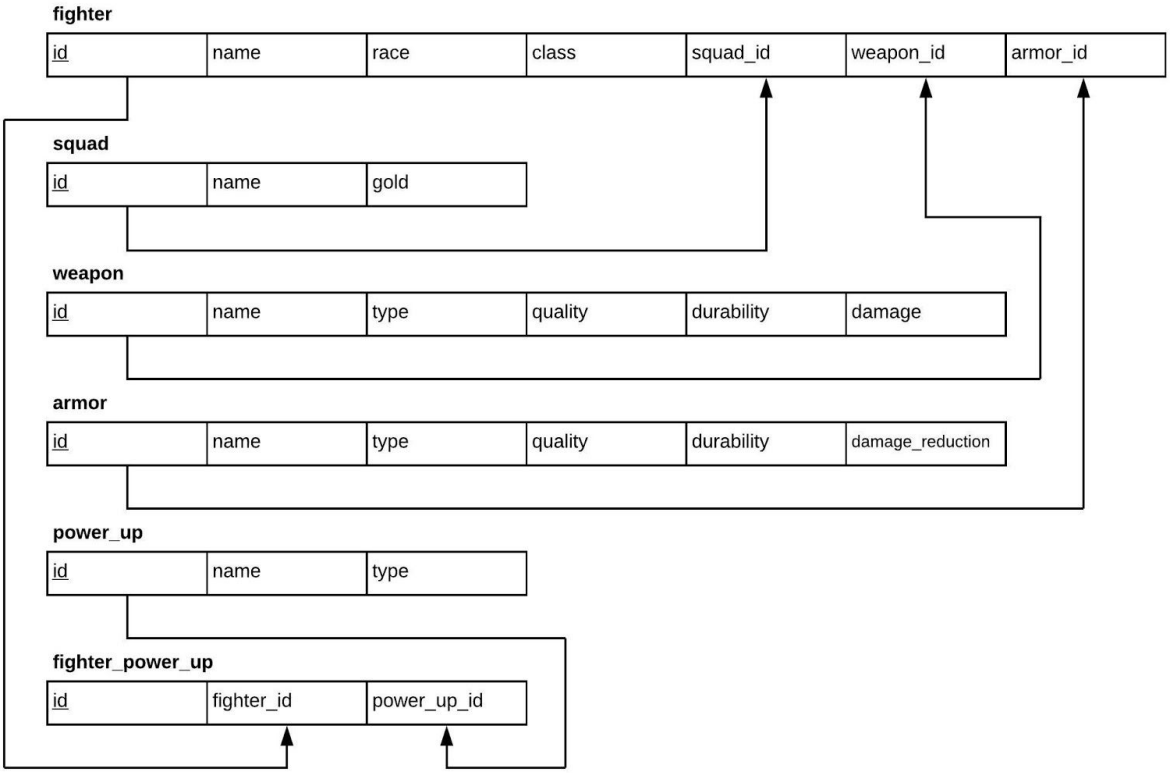
ER Diagram

The ER diagram is a little bit small so a jpeg of it is in the zip file that was uploaded. It is called “ER Diagram.jpeg”.



Schema

The Schema is a little bit small so a jpeg of it is in the zip file that was uploaded. It is called “Schema.jpeg”.



Data Definition Queries

The data definitions queries are also located in the zip file that was uploaded. It is called “definitions.sql”.

```
DROP TABLE IF EXISTS `fighter_power_up`;
DROP TABLE IF EXISTS `fighter`;
DROP TABLE IF EXISTS `squad`;
DROP TABLE IF EXISTS `weapon`;
DROP TABLE IF EXISTS `armor`;
DROP TABLE IF EXISTS `power_up`;

CREATE TABLE squad (
    id INT AUTO_INCREMENT,
    name varchar(255) NOT NULL,
    gold INT NOT NULL,
    PRIMARY KEY(id)
) ENGINE=InnoDB;

CREATE TABLE weapon (
    id INT AUTO_INCREMENT,
    name varchar(255) NOT NULL,
    type varchar(255) NOT NULL,
    quality varchar(255) NOT NULL,
    durability INT NOT NULL,
    damage INT NOT NULL,
    PRIMARY KEY(id)
) ENGINE=InnoDB;
```

```
CREATE TABLE armor (
    id INT AUTO_INCREMENT,
    name varchar(255) NOT NULL,
    type varchar(255) NOT NULL,
    quality varchar(255) NOT NULL,
    durability INT NOT NULL,
    damage_reduction INT NOT NULL,
    PRIMARY KEY(id)
) ENGINE=InnoDB;

CREATE TABLE fighter (
    id INT AUTO_INCREMENT,
    name varchar(255) NOT NULL,
    race varchar(255) NOT NULL,
    class varchar(255) NOT NULL,
    squad_id INT,
    weapon_id INT,
    armor_id INT,
    CONSTRAINT UC_weapon UNIQUE (weapon_id),
    CONSTRAINT UC_armor UNIQUE (armor_id),
    PRIMARY KEY(id),
    FOREIGN KEY (squad_id) REFERENCES squad(id) ON DELETE SET NULL ON
UPDATE CASCADE,
    FOREIGN KEY (weapon_id) REFERENCES weapon(id) ON DELETE SET NULL ON
UPDATE CASCADE,
    FOREIGN KEY (armor_id) REFERENCES armor(id) ON DELETE SET NULL ON
UPDATE CASCADE
) ENGINE=InnoDB;
```

```
CREATE TABLE power_up (
    id INT AUTO_INCREMENT,
    name varchar(255) NOT NULL,
    type varchar(255) NOT NULL,
    PRIMARY KEY(id)
) ENGINE=InnoDB;

CREATE TABLE fighter_power_up (
    id INT AUTO_INCREMENT,
    fighter_id INT,
    power_up_id INT,
    PRIMARY KEY(id),
    FOREIGN KEY (fighter_id) REFERENCES fighter(id) ON DELETE CASCADE ON
UPDATE CASCADE,
    FOREIGN KEY (power_up_id) REFERENCES power_up(id) ON DELETE CASCADE
ON UPDATE CASCADE
) ENGINE=InnoDB;
```

Data Manipulation Queries

The data definitions queries are also located in the zip file that was uploaded. It is called “manipulations.sql”.

```
/* I sometimes have multiple queries which are the same.
   I will not be repeating the queries which are the exact same.
   So, for example, if a query that is in update was already used
```

```
   in another section it will be excluded from update.
*/

/*********************************************************
************************Fighters*************************
*********************************************************/

/* ADD */
SELECT id, name FROM squad;

SELECT w.id, w.name
FROM weapon w
WHERE w.id NOT
IN (
   SELECT f.weapon_id
   FROM fighter f
   WHERE f.weapon_id IS NOT NULL
);

SELECT a.id, a.name
FROM armor a
WHERE a.id NOT
IN (
   SELECT f.armor_id
   FROM fighter f
   WHERE f.armor_id IS NOT NULL
);

/* Display Tables*/
```

```
SELECT id, name, race, class, squad_id, weapon_id,armor_id FROM fighter;

INSERT INTO fighter (name, race, class, squad_id, weapon_id, armor_id)
VALUES ([$name], [race], [class],
(SELECT id FROM squad WHERE id=[squad_id]),
(SELECT id FROM weapon WHERE id=[weapon_id]),
(SELECT id FROM armor WHERE id=[$armor_id]));

/* Update */
SELECT id, name FROM weapon WHERE id=[$weapon_id];

SELECT id, name FROM armor WHERE id=[$armor_id];

UPDATE fighter SET name=[name], race=[race], class=[class],
squad_id=(SELECT id FROM squad WHERE id=[squad_id]),
weapon_id=(SELECT id FROM weapon WHERE id=[weapon_id]),
armor_id=(SELECT id FROM armor WHERE id=[$armor_id])
WHERE id=[id];

/* Delete */
DELETE FROM fighter WHERE id=[id];

/* Search */
SELECT id, name, race, class, squad_id, weapon_id,armor_id FROM fighter WHERE
name=[search];


/*********************************************************
**************************Squads**************************
```

**********/

/* ADD */
INSERT INTO squad (name, gold) VALUES ([name], [gold]);

/* Display Tables*/
SELECT id, name, gold FROM squad;

/* Update */
SELECT name, gold FROM squad WHERE id=[$id];

UPDATE squad SET name=[$name], gold=[gold] WHERE id=[id];

/* Delete */
DELETE FROM squad WHERE id=[id];

/* Search */
SELECT id, name, gold FROM squad WHERE name=[search];

/**********
**********Weapons**********
**********/

/* ADD */
INSERT INTO weapon (name, type, quality, durability, damage)
VALUES ([name], [type], [quality], [durability], [damage])

/* Display Tables*/

```
SELECT id, name, type, quality, durability, damage FROM weapon;

/* Search */
SELECT id, name, type, quality, durability, damage FROM weapon WHERE
name=[search];



/*********************************************************
**************************Armor**************************
*********************************************************/

/* ADD */
INSERT INTO armor (name, type, quality, durability, damage_reduction)
VALUES ([name], [type], [quality], [durability], [damage_reduction])

/* Display Tables*/
SELECT id, name, type, quality, durability, damage_reduction FROM armor;

/* Search */
SELECT id, name, type, quality, durability, damage_reduction FROM armor WHERE
name=[search];



/*********************************************************
*************************Power-Ups************************
*********************************************************/

/* ADD */
INSERT INTO power_up (name, type)
```

```
VALUES ([name], [type]);

/* Display Tables*/
SELECT id, name, type FROM power_up;

/* Search */
SELECT id, name, type FROM power_up WHERE name=[search];



/************************************************************
**********Fighter - Power-Ups**********
************************************************************/

/* ADD */
SELECT id, name FROM fighter;

SELECT id, name FROM power_up;

INSERT INTO fighter_power_up (fighter_id, power_up_id)
VALUES ((SELECT id FROM fighter WHERE id=[fighter_id]),
(SELECT id FROM power_up WHERE id=[power_up_id]));



/* Display Tables*/
SELECT id, fighter_id, power_up_id FROM fighter_power_up;

/* Update */
SELECT fighter_id, power_up_id FROM fighter_power_up WHERE id=[id];
```

```
UPDATE fighter_power_up SET fighter_id=(SELECT id FROM fighter WHERE
id=[fighter]),
power_up_id=(SELECT id FROM power_up WHERE id=[power_up])
WHERE id=[id];

/* Delete */
DELETE FROM fighter_power_up WHERE id=[id];
```

Website Functionality

Website is hosted at: http://web.engr.oregonstate.edu/~lacconeb/display_tables.php