CS340 Final Project
Nick Wong

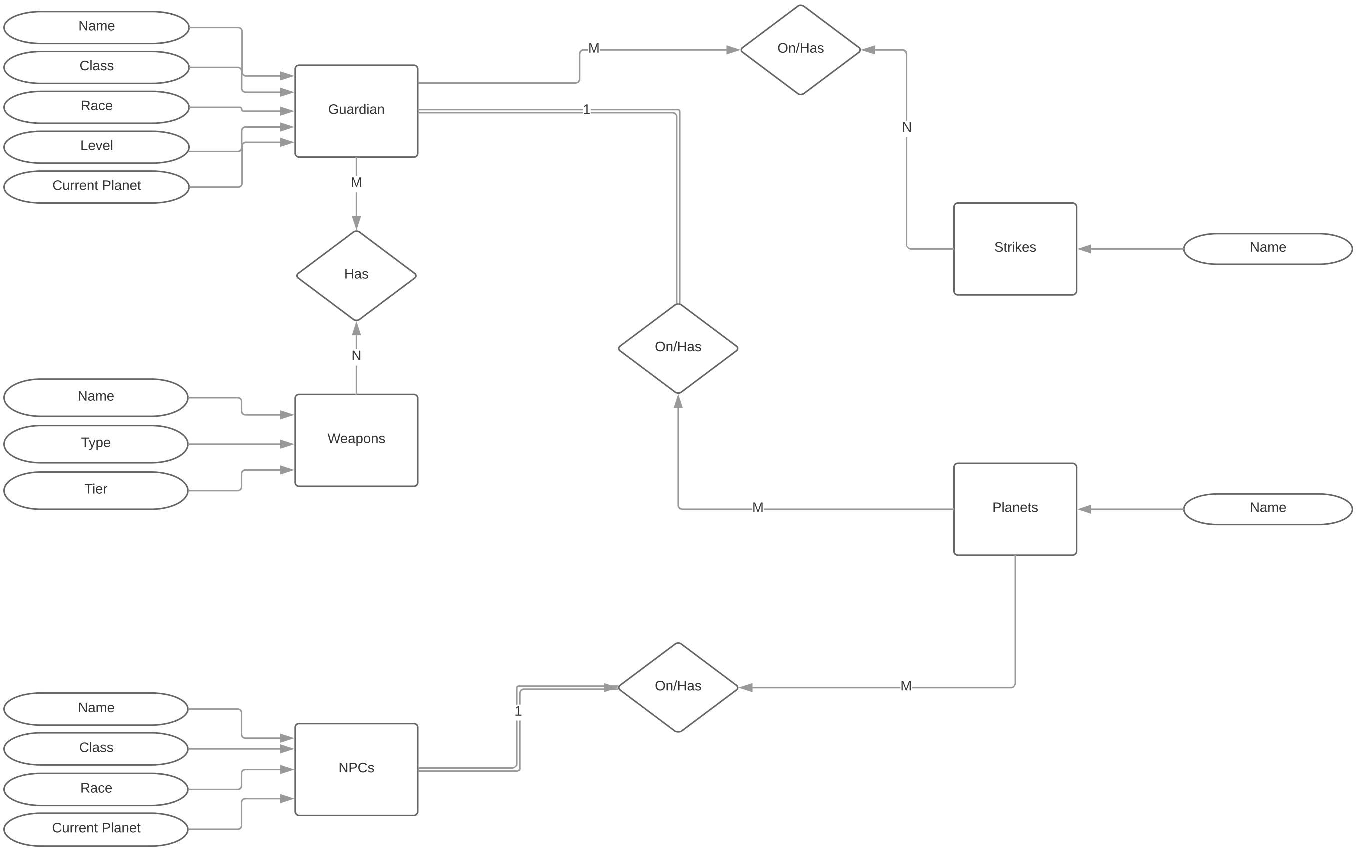
Outline:
I decided to do my final project on the game Destiny. I thought it was fitting because there are many entities which have relationships to each other. The game Destiny is a MMOFPS (massively multiplayer online first person shooter). Players (or Guardians) will complete strikes or other missions to progress throughout the game. They will collect weapons along their gameplay that they can equip in their inventory. This project is only a small model of what the Destiny game appears to be. I originally thought that NPC's would assign strikes to the players, but as I progressed through the project I realized that this didn't make much sense according to the game, so I left the strikes page in the NPCs page.

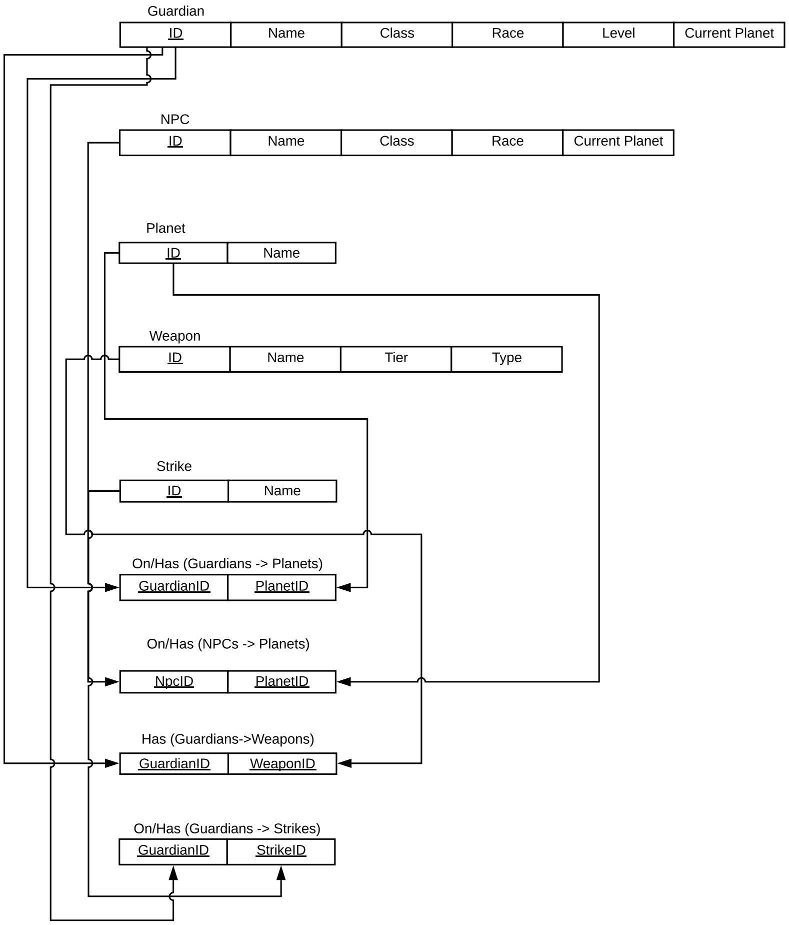
Database Outline:
Entities:
The entities that are used in this project are guardians (which are the players), strikes, weapons, planets, and NPCs (non-playable characters).
Attributes of the Entities:

- Guardians have a name, class, race level, and current planet
- Strikes have a name
- Weapons have a name, type, and tier
- Planets have a name
- NPCs have a name, class, race, and current planet

In addition to these entities, I have two other tables which define the many-to-many relationship between:
Guardians -> Strikes
Guardians -> Weapons
These tables are not defined in the ERD diagram because they are not entities within my database, however they are defined within the Schema Diagram.
Relationships:
The player can have many weapons, while a weapon can also be owned by many players.
The player can be on a current planet, while a planet can have many players.
The player can be on many strikes, while a strike can have many players.
The NPC can be on a current planet, while a planet can have many NPCs.
Along with these relationships are certain restrictions:
The player, and NPC must be on a current planet.
The rest of the relationships don't have a restriction.
Every entity has an ID, so in the ERD diagram, because I didn't have enough room to display the ID attribute on certain entities, I omitted them however they are in the Schema diagram.

Database ERD (Also included in zip)





Guardian

ID | Name | Class | Race | Level | Current Planet

NPC

ID | Name | Class | Race | Current Planet

Planet

ID | Name

Weapon

ID | Name | Tier | Type

Strike

ID | Name

On/Has (Guardians -> Planets)

GuardianID | PlanetID

On/Has (NPCs -> Planets)

NpcID | PlanetID

Has (Guardians->Weapons)

GuardianID | WeaponID

On/Has (Guardians -> Strikes)

GuardianID | StrikeID

Database ERD (Also included in zip)

Data Definition Queries (These are also included in the zip file in case some characters don’t translate correctly)**:**

```
-- Drop Tables if they Exist:

DROP TABLE IF EXISTS `guardian`;
DROP TABLE IF EXISTS `guardian_on_strike`;
DROP TABLE IF EXISTS `weapon`;
DROP TABLE IF EXISTS `guardians_weapons`;
DROP TABLE IF EXISTS `planet`;
DROP TABLE IF EXISTS `npc`;
DROP TABLE IF EXISTS `strike`;


-- Guardian Table:

CREATE TABLE guardian (
   id int NOT NULL AUTO_INCREMENT,
   weapon_id int,
   planet_id int,
   npc_id int,
   strike_id int,
   current_planet VARCHAR(255),
   name VARCHAR(255) NOT NULL,
   class VARCHAR(255) NOT NULL,
   race VARCHAR(255) NOT NULL,
   level int NOT NULL,
   PRIMARY KEY(id),
   FOREIGN KEY (strike_id) REFERENCES strike(id),
   FOREIGN KEY (weapon_id)  REFERENCES weapon(id),
   FOREIGN KEY (planet_id) REFERENCES planet(id),
   FOREIGN KEY (npc_id) REFERENCES npc(id),
   UNIQUE (name)
);

CREATE TABLE guardians_weapons (
   id int NOT NULL AUTO_INCREMENT,
   guardian_id int,
   weapon_id int,
   PRIMARY KEY(id),
   FOREIGN KEY(guardian_id) REFERENCES guardian(id),
   FOREIGN KEY(weapon_id) REFERENCES weapon(id)
);

CREATE TABLE guardian_on_strike (
   id int NOT NULL AUTO_INCREMENT,
   guardian_id int,
   strike_id int,
   PRIMARY KEY(id),
   FOREIGN KEY (guardian_id) REFERENCES guardian(id),
   FOREIGN KEY (strike_id) REFERENCES strike(id)
);
```

```
-- NPC Table:

CREATE TABLE npc (
   id int NOT NULL AUTO_INCREMENT,
   weapon_id int,
   guardian_id int,
   planet_id int,
   current_planet VARCHAR(255) NOT NULL,
   name VARCHAR(255) NOT NULL,
   class VARCHAR(255) NOT NULL,
   race VARCHAR(255) NOT NULL,
   PRIMARY KEY(id),
   FOREIGN KEY (weapon_id)  REFERENCES weapon(id),
   FOREIGN KEY (guardian_id)  REFERENCES guardian(id),
   FOREIGN KEY (planet_id)  REFERENCES planet(id),
   UNIQUE (name)
);


-- Planet Table:

CREATE TABLE planet (
   id int NOT NULL AUTO_INCREMENT,
   npc_id int,
   guardian_id int,
   name VARCHAR(255) NOT NULL,
   PRIMARY KEY(id),
   FOREIGN KEY (npc_id) REFERENCES npc(id),
   FOREIGN KEY (guardian_id) REFERENCES guardian(id),
   UNIQUE (name)
);

-- Weapon Table:

CREATE TABLE weapon (
   id int NOT NULL AUTO_INCREMENT,
   guardian_id int,
   npc_id int,
   name VARCHAR(255) NOT NULL,
   tier VARCHAR(255) NOT NULL,
   type VARCHAR(255) NOT NULL,
   PRIMARY KEY(id),
   FOREIGN KEY (guardian_id) REFERENCES guardian(id),
   FOREIGN KEY (npc_id) REFERENCES npc(id)
);

-- Strikes Table:

CREATE TABLE strike (
   id int NOT NULL AUTO_INCREMENT,
   guardian_id int,
   npc_id int,
   name VARCHAR(255) NOT NULL,
    PRIMARY KEY(id),
    FOREIGN KEY (guardian_id) REFERENCES guardian(id),
    FOREIGN KEY (npc_id) REFERENCES npc(id)
);


-- Inserting initial planets
INSERT INTO planet (name)
VALUES
('Earth'),
('Io'),
('Titan'),
('The Reef'),
('Nessus');

-- Inserting initial weapons
INSERT INTO weapon(name, tier, type)
VALUES
('MIDA Multi-Tool', 'Exotic', 'Scout Rifle'),
('Gjallarhorn', 'Exotoc', 'Rocket Launcher'),
('Prometheus Lens', 'Exotic', 'Trace Rifle'),
('Sorrow MG2', 'Common', 'Submachine Gun');


-- Inserting Initial Guardians:
INSERT INTO guardian (name, class, race, level, current_planet)
VALUES
('HorsemanSOJelly', 'Hunter', 'Human', '20', 'Earth'),
('kidvi11ain', 'Warlock', 'Awoken', '20', 'The Reef'),
('IMgrenade', 'Titan', 'Human', '12', 'Earth');

-- Inserting initial NPCs
INSERT INTO npc (name, class, race, current_planet)
VALUES
('Cayde-6', 'Hunter', 'EXO', 'Earth'),
('Ikora Rey', 'Warlock', 'Human', 'Earth'),
('Zavala', 'Titan', 'Awoken', 'Earth');


-- Inserting initial strikes
INSERT INTO strike (name, npc_id)
VALUES
('Nightfall', (select id from npc where name = 'Zavala')),
('Exodus Crash', (select id from npc where name = 'Cayde-6')),
('Savathûns Song', (select id from npc where name = 'Zavala')),
('The Inverted Spire', (select id from npc where name = 'Cayde-6')),
('The Pyramidion', (select id from npc where name = 'Ikora Rey'));
```

Data Manipulation Queries:

Guardians Page:

```
SELECT id, name FROM planet        (Getting planets)

SELECT guardian.id, guardian.name, planet.name AS current_planet, level, class, race FROM guardian INNER JOIN
planet ON current_planet = planet.name      (Getting all Guardians)

SELECT guardian.id, guardian.name, planet.name AS current_planet, level, class, race FROM guardian INNER JOIN
planet ON current_planet = planet.name WHERE guardian.id = ?          (Getting filtered Guardian)

SELECT id, name, class, race, level FROM guardian WHERE id = ?          (Getting single Guardian for updating

SELECT guardian.id AS id, weapon.id AS wid, weapon.name AS Name, weapon.tier AS Tier, weapon.type AS Type \
   FROM guardians_weapons \
   INNER JOIN guardian on guardian.id = guardians_weapons.guardian_id \
   INNER JOIN weapon on weapon.id = guardians_weapons.weapon_id \
   WHERE guardian.id = ? \
   ORDER BY Tier                                                        (Getting single Guardian's weapons)



SELECT id AS wid, name, tier, type FROM weapon                         (Getting all Weapons)

INSERT INTO guardian (name, class, race, current_planet, level) VALUES (?,?,?,(select name from planet where id
= ?),?)                                                                (Adds a Guardian to database)

INSERT INTO guardians_weapons (guardian_id, weapon_id) VALUES (?,?)          (Associates weapon to guardian)

UPDATE guardian SET name=?, class=?, race=?, current_planet=(select name from planet where id = ?), level=?
WHERE id=?                                                             (Updating a Guardian)

DELETE FROM guardians_weapons WHERE guardian_id = ? AND weapon_id = ? (Deleting a weapon from
                                                                       inventory)

DELETE FROM guardian WHERE id = ?                                      (Deleting a Guardian)
```

Weapons Page:

```
SELECT id, name, tier, type FROM weapon                                (Getting all weapons)

INSERT INTO weapon (name, type, tier) VALUES (?,?,?)                   (Adding a weapon to database)

DELETE FROM weapon WHERE id = ?                                        (Deleting a weapon from database)
```

Planets Page:

SELECT id, name FROM planet (Getting all planets)

INSERT INTO planet (name) VALUES (?) (Adding a planet)

DELETE FROM planet WHERE id = ? (Deleting a single planet)

NPC Page:

SELECT id AS gid, name FROM guardian (Getting al guardians)

SELECT guardian_on_strike.guardian_id AS gsid, guardian_on_strike.strike_id AS ssid, guardian.name AS Guardian, strike.name AS Strike FROM guardian_on_strike INNER JOIN guardian on guardian.id = guardian_on_strike.guardian_id INNER JOIN strike on strike.id = guardian_on_strike.strike_id ORDER BY Guardian, Strike (Getting all guardians on strikes)

SELECT id AS sid, name FROM strike (Getting all strikes)

SELECT id AS pid, name FROM planet (Getting all planets)

SELECT npc.id as nid, npc.name, planet.name AS current_planet, class, race FROM npc INNER JOIN planet ON current_planet = planet.name (Getting NPCs and their planets)

INSERT INTO npc (name, class, race, current_planet) VALUES (?,?,?,(select name from planet where id = ?))
(Adds an NPC)

INSERT INTO guardian_on_strike (guardian_id, strike_id) VALUES (?,?) (Assigns a strike to a guardian)

DELETE FROM guardian_on_strike WHERE guardian_id = ? AND strike_id = ?
(Deletes a strike from a guardian)

DELETE FROM npc WHERE id = ? (Deletes an NPC)