

CPSC 304 Project Cover Page

Milestone #: 2

Date: October 20, 2023

Group Number: 85

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Andy Li	65134645	o4k0h	andyzli333@gmail.com
Yilian Zhao	38891495	f7g3p	yilianz368@gmail.com
Akshat Singla	33275280	k6w2b	akshat250901@gmail.com

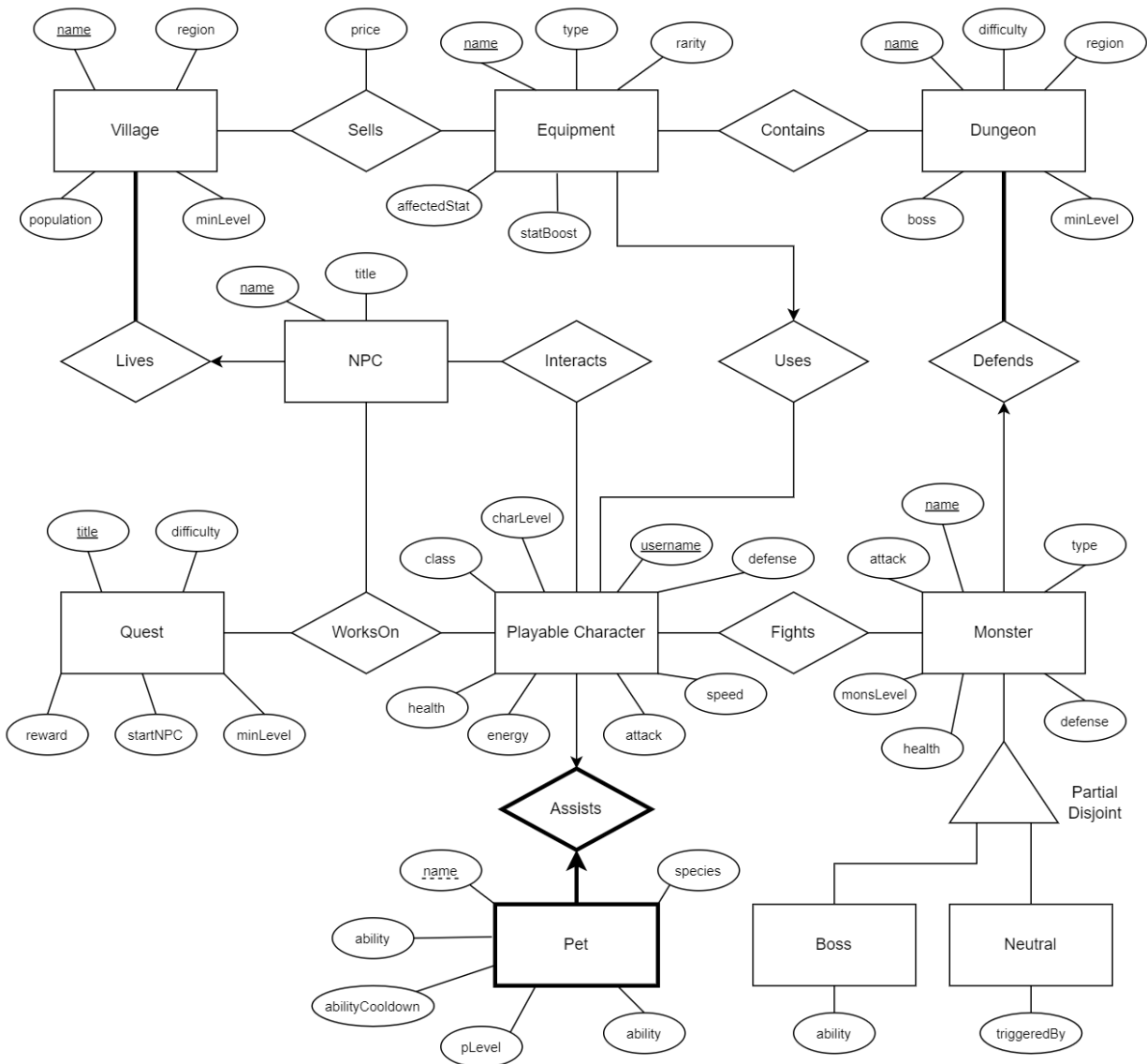
By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

Project Summary

Our database models the general structure of a role playing game, or RPG for short, which is targeted for anyone interested in exploring how RPGs work at a basic level. Users of this database can see the important entities that make up any RPG as well as how they interact with each other. Users can also figure out how to reach or obtain something they want, such as a specific piece of equipment or dungeon.

ER Diagram



ER Diagram Changes:

Added attributes to the following entities in order to create meaningful FDs, some of which can be normalized. The following list shows the new attributes for their corresponding entities:

- Village: population, minLevel
- Dungeon: minLevel, boss
- Quest: reward, length, minLevel, startNPC
- Pet: ability, abilityCooldown
- Monster: health, attack, defense

Other changes:

- "level" reserved in SQL already, so PlayableCharacter's "level" renamed to "charLevel", Monster's "level" renamed to "monsLevel", and Pet's "level" renamed to "pLevel"
- "trigger" reserved in SQL already, so Neutral's "trigger" renamed to "triggeredBy"

Schemas with Functional Dependencies (FDs)

Note that the CKs (candidate keys) in each relation exclude the PK (primary key), and each CK is separated by parentheses.

Underlined attributes are the PK, bolded attributes are foreign keys (FKs)

- Village(name: varchar, region: varchar, population: int, minLevel: int)
 - CKs: (region, population, minLevel)
 - not null: region, population, minLevel
 - FDs
 - name → region, population, minLevel
 - region, population, minLevel → name
- Sells(village: varchar, **equipment**: varchar, price: int)
 - not null: price
 - FDs
 - village, equipment → price
- Equipment(name: varchar, type: varchar, rarity: varchar, affectedStat: varchar, statBoost: int, **usedBy**: varchar)
 - not null: type, rarity, affectedStat, statBoost

- FDs
 - name → type, rarity, affectedStat, statBoost, usedBy
 - type, rarity, affectedStat, statBoost → name
 - type, rarity, affectedStat → statBoost
 - affectedStat → type
- Contains(**equipment**: varchar, **dungeon**: varchar)
- Dungeon(name: varchar, difficulty: int, region: varchar, minLevel: int, boss: varchar)
 - CKs: (boss, difficulty, region)
 - not null: difficulty, region, boss
 - FDs
 - name → difficulty, region, minLevel, boss
 - boss, difficulty, region → name, minLevel
 - difficulty, boss → region
 - minLevel → difficulty
- NPC(name: varchar, title: varchar, **village**: varchar)
 - FDs
 - name → title, village
- Interacts(**NPC**: varchar, **playableCharacter**: varchar)
- Quest(title: varchar, difficulty: int, reward: int, length: int, minLevel: int, **startNPC**: varchar)
 - CKs: (minLevel, difficulty)
 - not null: difficulty, reward, length, minLevel
 - FDs
 - title → difficulty, reward, length, minLevel, startNPC
 - minLevel, difficulty → title, reward, length, startNPC
- WorksOn(**quest**: varchar, **NPC**: varchar, **playableCharacter**: varchar)
- PlayableCharacter(username: varchar, class: varchar, charLevel: int, health: int, energy: int, attack: int, defense: int, speed: int, **pet**: varchar)
 - not null: class, charLevel, health, energy, attack, defense, speed
 - FDs
 - username → class, charLevel, health, energy, attack, defense, pet
- Pet(name: varchar, pLevel: int, species: varchar, ability: varchar, abilityCooldown: int, **owner**: varchar)

- not null: pLevel, species, ability
- FDs
 - name → pLevel, species, ability, abilityCooldown, owner
 - species → ability
 - species, ability, abilityCooldown → pLevel
- Fights(playableCharacter: varchar, monster: varchar)
- Monster(name: varchar, type: varchar, monsLevel: int, health: int, attack: int, defense: int, **defends**: varchar)
 - not null: type, monsLevel, health, attack, defense
 - FDs
 - name → type, monsLevel, health, attack, defense, defends
- Boss(name: varchar, ability: varchar)
 - unique: ability
 - FDs
 - name → ability
- Neutral(name: varchar, triggeredBy: varchar)
 - not null: triggeredBy
 - FDs
 - name → triggeredBy

Normalization into 3NF

Note that Village, Sells, Contains, NPC, Interacts, Quest, WorksOn, PlayableCharacter, Fights, Monster, Boss, and Neutral are already in 3NF.

Equipment

Put FDs into standard form

name → type

name → rarity

name → affectedStat

name → statBoost

name → usedBy

type, rarity, affectedStat, statBoost → name

type, rarity, affectedStat → statBoost

$\text{affectedStat} \rightarrow \text{type}$

Minimize LHS of each FD

Remove type from $(\text{type}, \text{rarity}, \text{affectedStat} \rightarrow \text{statBoost})$ because we have $(\text{affectedStat} \rightarrow \text{type})$

Remove type and statBoost from $(\text{type}, \text{rarity}, \text{affectedStat}, \text{statBoost} \rightarrow \text{name})$ because we have $(\text{affectedStat} \rightarrow \text{type})$ and $(\text{rarity}, \text{affectedStat} \rightarrow \text{statBoost})$

$\text{name} \rightarrow \text{type}$

$\text{name} \rightarrow \text{rarity}$

$\text{name} \rightarrow \text{affectedStat}$

$\text{name} \rightarrow \text{statBoost}$

$\text{name} \rightarrow \text{usedBy}$

$\text{rarity}, \text{affectedStat} \rightarrow \text{name}$

$\text{rarity}, \text{affectedStat} \rightarrow \text{statBoost}$

$\text{affectedStat} \rightarrow \text{type}$

Delete Redundant FDs

Remove $(\text{name} \rightarrow \text{type})$ because we have $(\text{name} \rightarrow \text{affectedStat} \rightarrow \text{type})$

Remove $(\text{name} \rightarrow \text{statBoost})$ because we have $(\text{name} \rightarrow \text{affectedStat})$, $(\text{name} \rightarrow \text{rarity})$, and $(\text{rarity}, \text{affectedStat} \rightarrow \text{statBoost})$

Minimal Cover:

$\text{name} \rightarrow \text{rarity}$

$\text{name} \rightarrow \text{affectedStat}$

$\text{name} \rightarrow \text{usedBy}$

$\text{rarity}, \text{affectedStat} \rightarrow \text{name}$

$\text{rarity}, \text{affectedStat} \rightarrow \text{statBoost}$

$\text{affectedStat} \rightarrow \text{type}$

Decompose by Synthesis

Add a relation for each FD in minimal cover:

EquipmentRarity(name, rarity)

EquipmentAffectedStat(name, affectedStat)

EquipmentUser(name, usedBy)
EquipmentName(rarity, affectedStat, name)
EquipmentStatBoost(rarity, affectedStat, statBoost)
EquipmentType(affectedStat, type)

Key is already included in at least one relation.

Dungeon

Put FDs into standard form

name \rightarrow difficulty
name \rightarrow region
name \rightarrow minLevel
name \rightarrow boss
boss, difficulty, region \rightarrow name
boss, difficulty, region \rightarrow minLevel
difficulty, boss \rightarrow region
minLevel \rightarrow difficulty

Minimize LHS of each FD

Remove region from (boss, difficulty, region \rightarrow name) and (boss, difficulty, region \rightarrow minLevel) because we have (difficulty, boss \rightarrow region)

name \rightarrow difficulty
name \rightarrow region
name \rightarrow minLevel
name \rightarrow boss
boss, difficulty \rightarrow name
boss, difficulty \rightarrow minLevel
difficulty, boss \rightarrow region
minLevel \rightarrow difficulty

Delete Redundant FDs

Remove (name \rightarrow region) because we have (name \rightarrow difficulty, boss \rightarrow region)
Remove (name \rightarrow minLevel) because we have (name \rightarrow boss, difficulty \rightarrow minLevel)

name \rightarrow difficulty

name \rightarrow boss

boss, difficulty \rightarrow name

boss, difficulty \rightarrow minLevel

difficulty, boss \rightarrow region

minLevel \rightarrow difficulty

Decompose by Synthesis

Add a relation for each FD in minimal cover:

DungeonDifficulty(name, difficulty)

DungeonBoss(name, boss)

DungeonName(boss, difficulty, name)

DungeonMinLevel(boss, difficulty, minLevel)

DungeonRegion(boss, difficulty, region)

DungeonMinLevelToDifficulty(minLevel, difficulty)

Keys are already included in at least one relation.

Pet

Put FDs into standard form

name \rightarrow pLevel

name \rightarrow species

name \rightarrow ability

name \rightarrow abilityCooldown

name \rightarrow owner

species \rightarrow ability

species, ability, abilityCooldown \rightarrow pLevel

Minimize LHS of each FD

Remove ability from (species, ability, abilityCooldown \rightarrow pLevel) because we have (species \rightarrow ability)

name \rightarrow pLevel

name \rightarrow species
name \rightarrow ability
name \rightarrow abilityCooldown
name \rightarrow owner
species \rightarrow ability
species, abilityCooldown \rightarrow pLevel

Delete Redundant FDs

Remove (name \rightarrow pLevel) because we have (name \rightarrow species, abilityCooldown \rightarrow pLevel)

Remove (name \rightarrow ability) because we have (name \rightarrow species \rightarrow ability)

Remove (name \rightarrow abilityCooldown) because we have (name \rightarrow species \rightarrow abilityCooldown)

name \rightarrow species
name \rightarrow owner
species \rightarrow ability
species, abilityCooldown \rightarrow pLevel

Decompose by Synthesis

Add a relation for each FD in minimal cover:

PetSpecies(name, species)

PetOwner(name, owner)

PetAbility(species, ability)

PetLevel(species, abilityCooldown, pLevel)

Key is already included in at least one relation.

Post-Normalization Tables

- Village(name: varchar, region: varchar, population: int, minLevel: int)
 - CKs: (region, population, minLevel)
- Sells(village: varchar, equipment: varchar, price: int)
- EquipmentRarity(name: varchar, rarity: varchar)
- EquipmentAffectedStat(name: varchar, **affectedStat**: varchar)
- EquipmentUser(name: varchar, **usedBy**: varchar)
- EquipmentName(rarity: varchar, **affectedStat**: varchar, **name**: varchar)
- EquipmentStatBoost(rarity: varchar, **affectedStat**: varchar, statBoost: int)
- EquipmentType(affectedStat: varchar, type: varchar)
- Contains(equipment: varchar, dungeon: varchar)
- DungeonDifficulty(name: varchar, difficulty: int)
- DungeonBoss(name: varchar, **boss**: varchar)
- DungeonName(boss: varchar, difficulty: int, **name**: varchar)
- DungeonMinLevel(boss: varchar, difficulty: int, minLevel: int)
- DungeonRegion(boss: varchar, difficulty: int, region: varchar)
- DungeonMinLevelToDifficulty(minLevel: int, difficulty: int)
- NPC(name: varchar, title: varchar, **village**: varchar)
- Interacts(NPC: varchar, playableCharacter: varchar)
- Quest(title: varchar, difficulty: int, reward: int, length: int, minLevel: int, **startNPC**: varchar)
 - CKs: (minLevel, difficulty)
- WorksOn(quest: varchar, NPC: varchar, playableCharacter: varchar)
- PlayableCharacter(username: varchar, class: varchar, charLevel: int, health: int, energy: int, attack: int, defense: int, speed: int, **pet**: varchar)
- PetSpecies(name: varchar, **species**: varchar)
- PetOwner(name: varchar, **owner**: varchar)
- PetAbility(species: varchar, ability: varchar)
- PetLevel(species: varchar, abilityCooldown: int, pLevel: int)
- Fights(playableCharacter: varchar, monster: varchar)
- Monster(name: varchar, type: varchar, monsLevel: int, health: int, attack: int, defense: int, **defends**: varchar)
- Boss(name: varchar, ability: varchar)
- Neutral(name: varchar, triggeredBy: varchar)

SQL DDL Create

Note: Most of our foreign keys that use ON DELETE CASCADE should also use ON UPDATE CASCADE but Oracle does not support ON UPDATE. In RPGs in general, anything that is created is rarely updated or modified, so we will omit ON UPDATE entirely instead.

```
CREATE TABLE Village (  
  name VARCHAR PRIMARY KEY,  
  region VARCHAR NOT NULL,  
  population INTEGER NOT NULL,  
  minLevel INTEGER NOT NULL  
);
```

```
CREATE TABLE PetAbility(  
  species VARCHAR PRIMARY KEY,  
  ability VARCHAR NOT NULL  
);
```

```
CREATE TABLE PetSpecies(  
  name VARCHAR PRIMARY KEY,  
  species VARCHAR NOT NULL,  
  FOREIGN KEY (species) REFERENCES PetAbility(species)  
  ON DELETE CASCADE  
);
```

```
CREATE TABLE PlayableCharacter(  
  username VARCHAR PRIMARY KEY,  
  class VARCHAR NOT NULL,  
  charLevel INTEGER NOT NULL,  
  health INTEGER NOT NULL,  
  energy INTEGER NOT NULL,  
  attack INTEGER NOT NULL,  
  defense INTEGER NOT NULL,  
  speed INTEGER NOT NULL,  
  pet VARCHAR,  
  FOREIGN KEY (pet) REFERENCES PetSpecies(name)
```

```
    ON DELETE SET NULL  
);
```

```
CREATE TABLE EquipmentRarity (  
    name VARCHAR PRIMARY KEY,  
    rarity VARCHAR NOT NULL  
);
```

```
CREATE TABLE EquipmentType(  
    affectedStat VARCHAR PRIMARY KEY,  
    type VARCHAR NOT NULL  
);
```

```
CREATE TABLE Sells (  
    village VARCHAR,  
    equipment VARCHAR,  
    price INTEGER NOT NULL,  
    PRIMARY KEY (village, equipment),  
    FOREIGN KEY (village) REFERENCES Village(name)  
        ON DELETE CASCADE,  
    FOREIGN KEY (equipment) REFERENCES EquipmentRarity(name)  
        ON DELETE CASCADE  
);
```

```
CREATE TABLE EquipmentAffectedStat (  
    name VARCHAR PRIMARY KEY,  
    affectedStat VARCHAR NOT NULL,  
    FOREIGN KEY (name) REFERENCES EquipmentRarity(name)  
        ON DELETE CASCADE,  
    FOREIGN KEY (affectedStat) REFERENCES EquipmentType(affectedStat)  
        ON DELETE CASCADE  
);
```

```
CREATE TABLE EquipmentUser (  
    name VARCHAR PRIMARY KEY,  
    usedBy VARCHAR,  
    FOREIGN KEY (name) REFERENCES EquipmentRarity(name)
```

```
    ON DELETE CASCADE,  
    FOREIGN KEY (usedBy) REFERENCES PlayableCharacter(username)  
    ON DELETE SET NULL  
);
```

```
CREATE TABLE EquipmentName (  
    rarity VARCHAR,  
    affectedStat VARCHAR,  
    name VARCHAR NOT NULL,  
    PRIMARY KEY (rarity, affectedStat),  
    FOREIGN KEY (name) REFERENCES EquipmentRarity(name)  
    ON DELETE CASCADE,  
    FOREIGN KEY (affectedStat) REFERENCES EquipmentType(affectedStat)  
    ON DELETE CASCADE  
);
```

```
CREATE TABLE EquipmentStatBoost (  
    rarity VARCHAR,  
    affectedStat VARCHAR,  
    statBoost INTEGER NOT NULL,  
    PRIMARY KEY (rarity, affectedStat),  
    FOREIGN KEY (affectedStat) REFERENCES EquipmentType(affectedStat)  
    ON DELETE CASCADE  
);
```

```
CREATE TABLE DungeonDifficulty (  
    name VARCHAR PRIMARY KEY,  
    difficulty INTEGER NOT NULL  
);
```

```
CREATE TABLE Contains(  
    equipment VARCHAR,  
    dungeon VARCHAR,  
    PRIMARY KEY (equipment, dungeon),  
    FOREIGN KEY (equipment) REFERENCES EquipmentRarity(name)  
    ON DELETE CASCADE,  
    FOREIGN KEY (dungeon) REFERENCES DungeonDifficulty(name)
```

```
    ON DELETE CASCADE  
);
```

```
CREATE TABLE DungeonName(  
    boss VARCHAR,  
    difficulty INTEGER,  
    name VARCHAR NOT NULL,  
    PRIMARY KEY (boss, difficulty),  
    FOREIGN KEY (name) REFERENCES DungeonDifficulty(name)  
    ON DELETE CASCADE  
);
```

```
CREATE TABLE DungeonBoss(  
    name VARCHAR PRIMARY KEY,  
    boss VARCHAR NOT NULL,  
    FOREIGN KEY (name) REFERENCES DungeonDifficulty(name)  
    ON DELETE CASCADE,  
    FOREIGN KEY (boss) REFERENCES DungeonName(boss, difficulty)  
    ON DELETE CASCADE  
);
```

```
CREATE TABLE DungeonMinLevelToDifficulty(  
    minLevel INTEGER PRIMARY KEY,  
    difficulty INTEGER NOT NULL  
);
```

```
CREATE TABLE DungeonMinLevel(  
    boss VARCHAR,  
    difficulty INTEGER,  
    minLevel INTEGER,  
    PRIMARY KEY (boss, difficulty),  
    FOREIGN KEY (boss, difficulty) REFERENCES DungeonName(boss, difficulty)  
    ON DELETE CASCADE  
);
```

```
CREATE TABLE DungeonRegion(  
    boss VARCHAR,
```

```
difficulty INTEGER,  
region VARCHAR NOT NULL,  
PRIMARY KEY (boss, difficulty),  
FOREIGN KEY (boss, difficulty) REFERENCES DungeonName(boss, difficulty)  
ON DELETE CASCADE  
);
```

```
CREATE TABLE NPC(  
name VARCHAR PRIMARY KEY,  
title VARCHAR,  
village VARCHAR,  
FOREIGN KEY (village) REFERENCES Village(name)  
ON DELETE SET NULL  
);
```

```
CREATE TABLE Interacts(  
npc VARCHAR,  
playableCharacter VARCHAR,  
PRIMARY KEY (npc, playableCharacter),  
FOREIGN KEY (npc) REFERENCES NPC(name)  
ON DELETE CASCADE,  
FOREIGN KEY (playableCharacter) REFERENCES PlayableCharacter(username)  
ON DELETE CASCADE  
);
```

```
CREATE TABLE Quest(  
title VARCHAR PRIMARY KEY,  
difficulty INTEGER NOT NULL,  
reward INTEGER NOT NULL,  
length INTEGER NOT NULL,  
minLevel INTEGER NOT NULL,  
startNPC VARCHAR,  
FOREIGN KEY (startNPC) REFERENCES NPC(name)  
ON DELETE CASCADE  
);
```

```
CREATE TABLE WorksOn(  

```

```
quest VARCHAR,  
npc VARCHAR,  
playableCharacter VARCHAR,  
PRIMARY KEY (quest, npc, playableCharacter),  
FOREIGN KEY (quest) REFERENCES Quest(title)  
    ON DELETE CASCADE,  
FOREIGN KEY (npc) REFERENCES NPC(name)  
    ON DELETE CASCADE,  
FOREIGN KEY (playableCharacter) REFERENCES PlayableCharacter(username)  
    ON DELETE CASCADE  
);
```

```
CREATE TABLE PetOwner(  
    name VARCHAR PRIMARY KEY,  
    owner VARCHAR,  
    FOREIGN KEY (name) REFERENCES PetSpecies(name)  
        ON DELETE CASCADE,  
    FOREIGN KEY (owner) REFERENCES PlayableCharacter(username)  
        ON DELETE CASCADE  
);
```

```
CREATE TABLE PetLevel(  
    species VARCHAR,  
    abilityCooldown INTEGER,  
    pLevel INTEGER NOT NULL,  
    PRIMARY KEY (species, abilityCooldown),  
    FOREIGN KEY (species) REFERENCES PetSpecies(name)  
        ON DELETE CASCADE  
);
```

```
CREATE TABLE Monster(  
    name VARCHAR PRIMARY KEY,  
    type VARCHAR NOT NULL,  
    monsLevel INTEGER NOT NULL,  
    health INTEGER NOT NULL,  
    attack INTEGER NOT NULL,  
    defense INTEGER NOT NULL,
```



```
defends VARCHAR,  
FOREIGN KEY (defends) REFERENCES DungeonDifficulty(name)  
ON DELETE SET NULL  
);
```

```
CREATE TABLE Fights(  
playableCharacter VARCHAR,  
monster VARCHAR,  
PRIMARY KEY (playableCharacter, monster),  
FOREIGN KEY (playableCharacter) REFERENCES PlayableCharacter(username)  
ON DELETE CASCADE,  
FOREIGN KEY (monster) REFERENCES Monster(name)  
ON DELETE CASCADE  
);
```

```
CREATE TABLE Boss(  
name VARCHAR PRIMARY KEY,  
ability VARCHAR UNIQUE,  
FOREIGN KEY (name) REFERENCES Monster(name)  
ON DELETE CASCADE  
);
```

```
CREATE TABLE Neutral(  
name VARCHAR PRIMARY KEY,  
triggeredBy VARCHAR NOT NULL,  
FOREIGN KEY (name) REFERENCES Monster(name)  
ON DELETE CASCADE  
);
```

SQL DDL Insert

Note: We are aware that bulk inserts may not work on Oracle, but its notation is easier to read. We will split all inserts apart if we finalize on using Oracle.

```
INSERT
INTO      Village(name, region, population, minLevel)
VALUES    ('Tutorial Town', 'Lowlands', 25, 1),
          ('Farms', 'Lowlands', 40, 5),
          ('Aria Falls', 'Wicked Forest', 20, 20),
          ('Shipton', 'Beach', 50, 30),
          ('Frostford', 'Mount Veritas', 25, 40);
```

```
INSERT
INTO      PetAbility(species, ability)
VALUES    ('Cat', 'Heal'),
          ('Dog', 'Bite'),
          ('Snake', 'Stun'),
          ('Bird', 'Fly'),
          ('Horse', 'Gallop');
```

```
INSERT
INTO      PetSpecies(name, species)
VALUES    ('Bob', 'Cat'),
          ('Larry', 'Dog'),
          ('Tabby', 'Cat'),
          ('Edwin', 'Bird'),
          ('Stacy', 'Horse');
```

```
INSERT
INTO      PlayableCharacter(username, class, charLevel, health, energy, attack,
                             defense, speed, pet)
VALUES    ('Jay', 'Warrior', 5, 400, 100, 30, 15, 10, 'Bob'),
          ('Elle', 'Mage', 10, 1200, 110, 40, 25, 15, 'Larry'),
          ('Simon', 'Priest', 20, 1000, 150, 80, 50, 25, 'Tabby'),
          ('Alice', 'Warrior', 30, 2000, 270, 100, 80, 30, 'Edwin');
```

```
(‘Steven’, ‘Assassin’, 40, 3000, 370, 150, 110, 35, ‘Stacy’),  
(‘Felix’, ‘Scout’, 15, 350, 100, 40, 20, 30, NULL);
```

```
INSERT  
INTO EquipmentRarity(name, rarity)  
VALUES (‘Short Sword’, ‘Common’),  
(‘Chainmail’, ‘Common’),  
(‘Demonic Staff’, ‘Rare’),  
(‘Magic Boots’, ‘Common’),  
(‘Frost Glaive’, ‘Epic’),  
(‘Light Bow’, ‘Common’),  
(‘Shadow Beads’, ‘Epic’),  
(‘Ritual Grail’, ‘Epic’),  
(‘Singed Wand’, ‘Common’),  
(‘Soulless Sabre’, ‘Epic’);
```

```
INSERT  
INTO EquipmentType(affectedStat, type)  
VALUES (‘Attack’, ‘Sword’),  
(‘Defense’, ‘Armour’),  
(‘Attack’, ‘Staff’),  
(‘Speed’, ‘Boots’),  
(‘Health’, ‘Helmet’);
```

```
INSERT  
INTO Sells(village, equipment, price)  
VALUES (‘Tutorial Town’, ‘Short Sword’, 100),  
(‘Farms’, ‘Chainmail’, 250),  
(‘Aria Falls’, ‘Demonic Staff’, 800),  
(‘Shipton’, ‘Magic Boots’, 500),  
(‘Frostford’, ‘Frost Glaive’, 900);
```

```
INSERT  
INTO EquipmentAffectedStat(name, affectedStat)  
VALUES (‘Short Sword’, ‘Attack’),  
(‘Chainmail’, ‘Defense’),
```

(‘Demonic Staff’, ‘Attack’),
(‘Magic Boots’, ‘Speed’),
(‘Frost Glaive’, ‘Attack’);

INSERT
INTO EquipmentUser(name, usedBy)
VALUES (‘Short Sword’, NULL),
(‘Chainmail’, ‘Jay’),
(‘Demonic Staff’, ‘Simon’),
(‘Magic Boots’, NULL),
(‘Frost Glaive’, ‘Alice’);

INSERT
INTO EquipmentName(rarity, affectedStat, name)
VALUES (‘Common’, ‘Attack’, ‘Short Sword’),
(‘Common’, ‘Defense’, ‘Chainmail’),
(‘Rare’, ‘Attack’, ‘Demonic Staff’),
(‘Common’, ‘Speed’, ‘Magic Boots’),
(‘Epic’, ‘Attack’, ‘Frost Glaive’);

INSERT
INTO EquipmentStatBoost(rarity, affectedStat, statBoost)
VALUES (‘Common’, ‘Attack’, 5),
(‘Common’, ‘Defense’, 10),
(‘Rare’, ‘Attack’, 40),
(‘Common’, ‘Speed’, 5),
(‘Epic’, ‘Attack’, 75);

INSERT
INTO DungeonDifficulty(name, difficulty)
VALUES (‘The Abandoned Farm’, 1),
(‘Lair of the Perished Mountain’, 4),
(‘The Raging Catacombs’, 3),
(‘The Eternal Cells’, 3),
(‘The Bleak Tunnels’, 2);

```
INSERT
INTO      Contains(equipment, dungeon)
VALUES    ('Light Bow', 'The Abandoned Farm'),
          ('Shadow Beads', 'Lair of the Perished Mountain'),
          ('Ritual Grail', 'The Raging Catacombs'),
          ('Singed Wand', 'The Eternal Cells'),
          ('Soulless Sabre', 'The Bleak Tunnels');
```

```
INSERT
INTO      DungeonName(boss, difficulty, name)
VALUES    ('The Bull', 1, 'The Abandoned Farm'),
          ('The Frost Dragon', 4, 'Lair of the Perished Mountain'),
          ('Skulls of Fear', 3, 'The Raging Catacombs'),
          ('The Prison Guard', 3, 'The Eternal Cells'),
          ('The Tree of Evil', 2, 'The Bleak Tunnels');
```

```
INSERT
INTO      DungeonBoss(name, boss)
VALUES    ('The Abandoned Farm', 'The Bull'),
          ('Lair of the Perished Mountain', 'The Frost Dragon'),
          ('The Raging Catacombs', 3, 'Skulls of Fear'),
          ('The Eternal Cells', 'The Prison Guard'),
          ('The Bleak Tunnels', 'The Tree of Evil');
```

```
INSERT
INTO      DungeonMinLevelToDifficulty(minLevel, difficulty)
VALUES    (5, 1),
          (40, 4),
          (30, 3),
          (90, 8),
          (20, 2);
```

```
INSERT
INTO      DungeonMinLevel(boss, difficulty, minLevel)
VALUES    ('The Bull', 1, 5),
          ('The Frost Dragon', 4, 40),
```

(‘Skulls of Fear’, 3, 30),
(‘The Prison Guard’, 3, 30),
(‘The Tree of Evil’, 2, 20);

INSERT
INTO DungeonRegion(boss, difficulty, region)
VALUES (‘The Bull’, 1, ‘Lowlands’),
(‘The Frost Dragon’, 4, ‘Mount Veritas’),
(‘Skulls of Fear’, 3, ‘Beach’),
(‘The Prison Guard’, 3, ‘Beach’),
(‘The Tree of Evil’, 2, ‘Wicked Forest’);

INSERT
INTO NPC(name, title, village)
VALUES (‘Gerald’, ‘Leader’, ‘Tutorial Town’),
(‘Archibald’, ‘Forest Elf’, ‘Aria Falls’),
(‘Petra’, ‘Farmer’, ‘Farms’),
(‘Maurelle’, ‘Shiphand’, ‘Shipton’),
(‘Sampson’, ‘Adventurer’, ‘Frostford’);

INSERT
INTO Interacts(NPC, playableCharacter)
VALUES (‘Gerald’, ‘Jay’),
(‘Archibald’, ‘Jay’),
(‘Petra’, ‘Jay’),
(‘Maurelle’, ‘Jay’),
(‘Sampson’, ‘Jay’);

INSERT
INTO Quest(title, difficulty, reward, length, minLevel, startNPC)
VALUES (‘First Steps’, 1, 500, 2, 1, ‘Gerald’),
(‘Too Many Weeds’, 2, ‘600’, 2, 5, ‘Petra’),
(‘Collect Clams’, 4, 1500, 3, 30, ‘Maurelle’),
(‘Forest Restoration’, 3, 800, 3, 20, ‘Archibald’),
(‘Slippery Hike’, 5, 2000, 4, 40, ‘Sampson’);

```
INSERT
INTO WorksOn(quest, NPC, playableCharacter)
VALUES ('First Steps', 'Gerald', 'Jay'),
       ('Too Many Weeds', 'Petra', 'Simon'),
       ('Collect Clams', 'Maurelle', 'Jay'),
       ('Forest Restoration', 'Archibald', 'Elle'),
       ('Slippery Hike', 'Sampson', 'Alice');
```

```
INSERT
INTO PetOwner(name, owner)
VALUES ('Bob', 'Jay'),
       ('Larry', 'Elle'),
       ('Tabby', 'Simon'),
       ('Edwin', 'Alice'),
       ('Stacy', 'Steven');
```

```
INSERT
INTO PetLevel(species, abilityCooldown, pLevel)
VALUES ('Cat', 8, 3),
       ('Dog', 3, 8),
       ('Cat', 6, 5),
       ('Bird', 10, 6),
       ('Dog', 5, 6);
```

```
INSERT
INTO Monster(name, type, monsLevel, health, attack, defense, defends)
VALUES ('Timid Zombie', 'Undead', 1, 50, 5, 0, NULL),
       ('Clam', 'Water', 20, 800, 30, 10, NULL),
       ('Tree Sprite', 'Earth', 5, 300, 10, 2, NULL),
       ('Mountain Goat', 'Ice', 30, 1200, 50, 30, NULL),
       ('Clay Dummy', 'Earth', 10, 700, 20, 5, NULL),
       ('The Bull', 'Normal', 10, 700, 20, 5, 'The Abandoned Farm'),
       ('Skulls of Fear', 'Undead', 1, 50, 5, 0, 'The Raging Catacombs'),
       ('The Prison Guard', 'Undead', 1, 50, 5, 0, 'The Eternal Cells'),
       ('The Tree of Evil', 'Earth', 25, 1000, 75, 100, 'The Bleak Tunnels'),
       ('The Frost Dragon', 'Ice', 75, 5000, 250, 400, 'Lair of the Perished');
```

Mountain'),
('Sleepy Spider', 'Earth', 10, 700, 20, 5, NULL),
('Beach Crab', 'Water', 30, 1200, 50, 30, NULL),
('Mountain Bear', 'Ice', 30, 1200, 50, 30, NULL);

INSERT
INTO Fights(playableCharacter, monster)
VALUES ('Jay', 'Timid Zombie'),
('Simon', 'Clam'),
('Elle', 'Tree Sprite'),
('Alice', 'Mountain Goat'),
('Stacy', 'Clay Dummy');

INSERT
INTO Boss(name, ability)
VALUES ('The Bull', 'Unstoppable Charge'),
('Clam', 'Water Gun'),
('Skulls of Fear', 'Death Bite'),
('The Prison Guard', 'Bat Strike'),
('The Tree of Evil', 'Roots of Doom'),
('The Frost Dragon', 'Frozen Roar'),
('Tree Sprite', 'Vine Whip');

INSERT
INTO Neutral(name, triggeredBy)
VALUES ('Sleepy Spider', 'Loud noise'),
('Timid Zombie', 'Proximity'),
('Clay Dummy', 'Proximity'),
('Beach Crab', 'Damage'),
('Mountain Bear', 'Damage');