University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: 2

Date: 07/21/2024

Group Number: 28

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Julianna Loresco	21397633	z4u4m	jmariel@student.ubc.ca
Patricia Rae Villa	22680565	n8e1f	prvilla@student.ubc.ca
Chowdhury Zayn Ud-Din Shams	46176756	v7v0f	zaynchow@student.ubc.ca

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.)

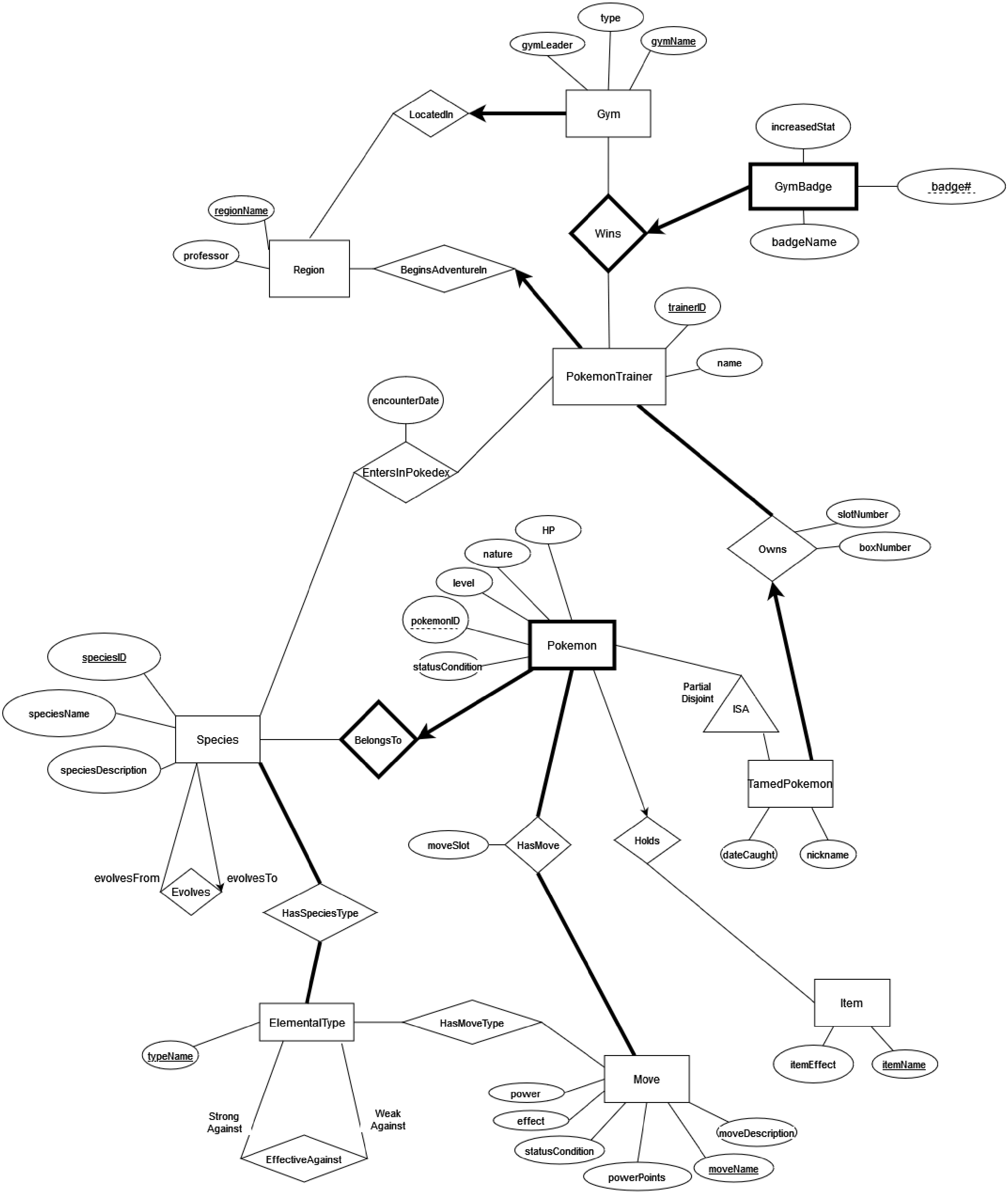
In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

Project Description

Our project concerns the domain of documenting and managing Pokemon from the popular game franchise Pokemon. Our application will allow users such as Pokemon fans to track their own Pokemon collections using a Pokédex and PC.

ER Diagram Notes

- We have changed Pokemon to be a weak entity of Species and removed our previous ISA of WildPokemon.
- We have modified some of the key constraints to more accurately reflect our intended database application, namely between Item and Holds and between Species and EntersInPokedex.
- We have removed some redundant attributes that were previously PKs, namely moveID and itemID, and changed their respective entities' PKs to moveName and itemName.
- We decided to remove the “StoresInPC” relationship between Pokemon and PokemonTrainer, replacing it with the one-to-many “Owns” relationship with boxNumber and slotNumber. Pokemon in boxNumber 0 with slotNumber 1-6 will be interpreted as being part of the trainer's current party.
- We have removed the “IsNativeTo” relationship between Species and Region to better reflect the domain of the Pokemon games.


type
gymLeader
gymName
LocatedIn
Gym
increasedStat
GymBadge
badge#
Wins
badgeName
regionName
professor
Region
BeginsAdventureIn
trainerID
name
PokemonTrainer
encounterDate
EntersInPokedex
slotNumber
Owns
boxNumber
HP
nature
level
pokemonID
Pokemon
statusCondition
speciesID
speciesName
Species
BelongsTo
Partial
Disjoint
ISA
speciesDescription
TamedPokemon
moveSlot
HasMove
Holds
dateCaught
nickname
evolvesFrom
Evolves
evolvesTo
HasSpeciesType
Item
ElementalType
HasMoveType
typeName
itemEffect
itemName
Move
power
effect
Strong
Against
Weak
Against
moveDescription
EffectiveAgainst
statusCondition
powerPoints
moveName

Schemas

Gym(gymName: varchar, gymLeader: varchar, type: varchar, **locatedIn_regionName**: varchar)

- gymLeader, type and locatedIn_regionName must not be null
- gymLeader must be unique

GymBadge(**gymName**: varchar, **trainerID**: int, badgeID: int, badgeName: varchar, increasedStat: varchar)

- increasedStat and badgeName must not be null

PokemonTrainer(trainerID: int, name: varchar, **startingIn_regionName**: varchar)

- name and startingIn_regionName must not be null

Region(regionName: varchar, professor: varchar)

- professor must not be null
- professor must be unique

Species(speciesID: int, speciesName: varchar, speciesDescription: varchar, **evolvesFrom_speciesID**: int)

- speciesName is a candidate key
- speciesName and speciesDescription must not be null

Pokemon(pokemonID: int, **speciesID**: int, level: int, nature: varchar, HP: int, statusCondition: varchar, **holding_itemName**: varchar)

- level is 1 at default
- nature, and HP must not be null

TamedPokemon(**pokemonID**: int, **speciesID**, dateCaught: date, nickname: varchar, **ownedBy_trainerID**: int, boxNumber: int, slotNumber: int)

- dateCaught, boxNumber, and slotNumber must not be null
- (ownedBy_trainerID, boxNumber, slotNumber) must be unique

ElementalType(typeName: varchar)

Item(itemName: varchar, itemEffect: varchar)

- itemEffect must not be null

Move(moveName: varchar, moveDescription: varchar, powerPoints: int, statusCondition: varchar, power: int, effect: varchar)

- moveDescription and powerPoints must not be null

EntersInPokedex(**speciesID**: int, **trainerID**: int**,** encounterDate: date)

- encounterDate must not be null

HasSpeciesType(**speciesID**: int, **typeName**: varchar)

HasMove(**speciesID**: int**,** **pokemonID**: int, **moveName**: varchar**,** moveSlot: int)

- moveSlot must not be null
- (speciesID, pokemonID, moveSlot) must be unique

HasMoveType(**typeName**: varchar, **moveName**: varchar)

EffectiveAgainst(**strong_typeName**: varchar, **weak_typeName**: varchar)

Functional Dependencies

Gym(gymName: varchar, gymLeader: varchar, type: varchar, **locatedIn_regionName**: varchar)

- gymName → gymLeader, type, locatedIn_regionName

GymBadge(**gymName**: varchar, **trainerID**: int, badgeID: int, badgeName: varchar, increasedStat: varchar)

- gymName, trainerID, badgeID → badgeName, increasedStat
- gymName → badgeName, increasedStat
- badgeName → increasedStat

PokemonTrainer(trainerID: int, name: varchar, **startingIn_regionName**: varchar)

- trainerID → name, startingIn_regionName

Region(regionName: varchar, professor: varchar)

- regionName → professor

Species(speciesID: int, speciesName: varchar, speciesDescription: varchar, **evolvesFrom_speciesID**: int)

- speciesID → speciesName, speciesDescription, evolvesFrom_speciesID
- speciesName → speciesDescription, speciesID, evolvesFrom_speciesID

Pokemon(pokemonID: int, **speciesID**: int, level: int, nature: varchar, HP: int, statusCondition: varchar, **holding_itemName**: varchar)

- pokemonID, speciesID → level, nature, HP, statusCondition, holding_itemName
- speciesID, level → HP
    - For the purposes of this project, we are not including individual Pokemon stats in our application. As a result we have specified that level and species are sufficient to determine HP.

TamedPokemon(**pokemonID**: int, **speciesID**, dateCaught: date, nickname: varchar, **ownedBy_trainerID**: int, boxNumber: int, slotNumber: int)

- pokemonID, speciesID → dateCaught, nickname, ownedBy_trainerID, boxNumber, slotNumber
- ownedBy_trainerID, boxNumber, slotNumber → pokemonID, speciesID, dateCaught, nickname
    - We felt that you should be able to view a given Pokemon by viewing it in a PC given a trainer's ID and box and slot number.

Item(itemName: varchar, itemEffect: varchar)

- itemName → itemEffect

Move(moveName: varchar, moveDescription: varchar, powerPoints: int, power: int, statusCondition: varchar, effect: varchar)

- moveName → moveDescription, powerPoints, statusCondition, power, effect

EntersInPokedex(**speciesID**: int, **trainerID**: int**,** encounterDate: date)

- speciesID, trainerID → encounterDate

HasMove(**speciesID**: int**, pokemonID**: int, **moveName**: varchar**,** moveSlot: int)

- speciesID, pokemonID, moveName → moveSlot
- speciesID, pokemonID, moveSlot → moveName

Normalization

Pokemon(pokemonID: int, **speciesID**: int, level: int, nature: varchar, HP: int, statusCondition: varchar, **holding_itemName**: varchar)

- Decompose on speciesID, level → HP
- Pokemon1(**speciesID**: int, **level**: int, HP: int)
- Pokemon2(pokemonID: int, **speciesID**: int, level: int, nature: varchar, statusCondition: varchar, **holding_itemName**: varchar)

- Renamed to:
- PokemonHP(**speciesID**: int, **level**: int, HP: int)
- Pokemon(pokemonID: int, **speciesID**: int, level: int, nature: varchar, statusCondition: varchar, **holding_itemName**: varchar)

GymBadge(**gymName**: varchar, **trainerID**: int, badgeID: int, badgeName: varchar, increasedStat: varchar)

- Decompose on gymName → badgeName, increasedStat
- GymBadge1(**gymName**: varchar, badgeName: varchar, increasedStat: varchar)
- GymBadge2(**gymName**: varchar, **trainerID**: int, badgeID: int)

- Decompose on badgeName → increasedStat
- GymBadge3(**gymName**: varchar, badgeName: varchar)
- GymBadge4(badgeName: varchar, increasedStat: varchar)

- Final Decomposition:
- GymBadge2(**gymName**: varchar, **trainerID**: int, badgeID: int)
- GymBadge3(**gymName**: varchar, badgeName: varchar)
- GymBadge4(**badgeName**: varchar, increasedStat: varchar)

- Renamed to:
- GymBadge(**gymName**: varchar, **trainerID**: int, badgeID: int)
- GymBadgeName(**gymName**: varchar, badgeName: varchar)
- GymBadgeDamage(**badgeName**: varchar, increasedStat: varchar)

SQL DDL Statements

```
CREATE TABLE Gym
        (gymName                VARCHAR     PRIMARY KEY,
        gymLeader               VARCHAR     UNIQUE NOT NULL,
        type                    VARCHAR     NOT NULL,
        locatedIn_regionName VARCHAR        NOT NULL,
        FOREIGN KEY (locatedIn_regionName) REFERENCES
                Region(regionName)
                ON UPDATE CASCADE
        )

CREATE TABLE GymBadge
        (gymName        VARCHAR,
        trainerID       VARCHAR,
        badgeID         INTEGER,
        PRIMARY KEY (gymName, trainerID, badgeID),
        FOREIGN KEY (gymName) REFERENCES Gym(gymName)
                ON DELETE CASCADE
                ON UPDATE CASCADE,
        FOREIGN KEY (trainerID) REFERENCES PokemonTrainer(trainerID)
                ON DELETE CASCADE
                ON UPDATE CASCADE
        )

CREATE TABLE GymBadgeName
        (gymName        VARCHAR     PRIMARY KEY,
        badgeName       VARCHAR     NOT NULL,
        FOREIGN KEY (gymName) REFERENCES Gym(gymName)
                ON DELETE CASCADE
                ON UPDATE CASCADE,
        FOREIGN KEY (badgeName) REFERENCES GymBadgeName(badgeName)
                ON DELETE CASCADE
        )

CREATE TABLE GymBadgeDamage
        (badgeName              VARCHAR     PRIMARY KEY,
        increasedStat           VARCHAR     NOT NULL)

CREATE TABLE PokemonTrainer
        (trainerID                      INTEGER     PRIMARY KEY,
        name                            VARCHAR     NOT NULL,
        startingIn_regionName           VARCHAR     NOT NULL,
        FOREIGN KEY (startingIn_regionName) REFERENCES
                Region(regionName)
                ON UPDATE CASCADE
        )

CREATE TABLE Region
        (regionName     VARCHAR         PRIMARY KEY,
        professor       VARCHAR         UNIQUE
        )

CREATE TABLE Species
        (speciesID              INTEGER         PRIMARY KEY,
        speciesName             VARCHAR         NOT NULL,
        speciesDescription      VARCHAR         NOT NULL,
        evolvesFrom_speciesID INTEGER,
        FOREIGN KEY (evolvesFrom_speciesID) REFERENCES
                Species(speciesID)
                ON UPDATE CASCADE
                ON DELETE SET NULL
        )

CREATE TABLE Pokemon
        (pokemonID              INTEGER,
        speciesID               INTEGER,
        level                   INTEGER         DEFAULT 1,
        nature                  VARCHAR         NOT NULL,
        statusCondition         VARCHAR,
        holding_itemName        VARCHAR,
        PRIMARY KEY (pokemonID, speciesID),
        FOREIGN KEY (speciesID) REFERENCES
                Species(speciesID)
                ON UPDATE CASCADE,
        FOREIGN KEY (holding_itemName) REFERENCES
                Item(itemName)
                ON UPDATE CASCADE
                ON DELETE SET NULL
        )

CREATE TABLE PokemonHP
        (speciesID              INTEGER,
        level                   INTEGER         DEFAULT 1,
        HP                      INTEGER         NOT NULL,
        PRIMARY KEY (speciesID, level),
        FOREIGN KEY (speciesID) REFERENCES
                Species(speciesID)
                ON DELETE CASCADE
                ON UPDATE CASCADE,
        FOREIGN KEY (level) REFERENCES
                Pokemon(level)
                ON DELETE CASCADE
                ON UPDATE CASCADE
        )

CREATE TABLE TamedPokemon
        (pokemonID              INTEGER,
        speciesID               INTEGER,
        dateCaught              DATE            NOT NULL,
        nickname                VARCHAR,
        ownedBy_trainerID       VARCHAR         NOT NULL,
        boxNumber               INTEGER         NOT NULL,
        slotNumber              INTEGER         NOT NULL,
        CONSTRAINT Box_Location UNIQUE (ownedBy_trainerID, boxNumber, slotNumber),
        PRIMARY KEY (pokemonID, speciesID),
        FOREIGN KEY (pokemonID) REFERENCES
                Pokemon(pokemonID)
                ON UPDATE CASCADE,
        FOREIGN KEY (speciesID) REFERENCES
                Species(speciesID)
                ON UPDATE CASCADE
        )


CREATE TABLE ElementalType
        (typeName       VARCHAR         PRIMARY KEY
        )

CREATE TABLE Item
        (itemName       VARCHAR         PRIMARY KEY,
        itemEffect      VARCHAR         NOT NULL
        )

CREATE TABLE Move
        (moveName               VARCHAR         PRIMARY KEY,
        moveDescription         VARCHAR         NOT NULL,
        powerPoints             INTEGER         NOT NULL,
        power                   INTEGER         DEFAULT 0,
        statusCondition         VARCHAR,
        effect                  VARCHAR
        )
```

```
CREATE TABLE EntersInPokedex
        (speciesID              INTEGER,
        trainerID               INTEGER,
        encounterDate           DATE            NOT NULL,
        PRIMARY KEY (speciesID, trainerID),
        FOREIGN KEY (speciesID) REFERENCES
                Species(speciesID)
                ON DELETE CASCADE
                ON UPDATE CASCADE,
        FOREIGN KEY (trainerID) REFERENCES
                PokemonTrainer(trainerID)
                ON DELETE CASCADE
                ON UPDATE CASCADE
        )

CREATE TABLE HasSpeciesType
        (speciesID              INTEGER,
        typeName                INTEGER,
        PRIMARY KEY (speciesID, typeName),
        FOREIGN KEY (speciesID) REFERENCES
                Species(speciesID)
                ON UPDATE CASCADE
                ON DELETE CASCADE,
        FOREIGN KEY (typeName) REFERENCES
                ElementalType(typeName)
                ON UPDATE CASCADE
                ON DELETE CASCADE
        )

CREATE TABLE HasMove
        (speciesID              INTEGER,
        pokemonID               INTEGER,
        moveName                VARCHAR,
        moveSlot                INTEGER         NOT NULL,
        PRIMARY KEY (speciesID, pokemonID, moveName),
        FOREIGN KEY (speciesID) REFERENCES
                Species(speciesID)
                ON UPDATE CASCADE
                ON DELETE CASCADE,
        FOREIGN KEY (pokemonID) REFERENCES
                Pokemon(pokemonID)
                ON UPDATE CASCADE
                ON DELETE CASCADE,
        FOREIGN KEY (moveName) REFERENCES
```

```
                Move(moveName)
                ON UPDATE CASCADE
                ON DELETE CASCADE,
        )

CREATE TABLE HasMoveType
        (moveName               VARCHAR,
        typeName                INTEGER,
        PRIMARY KEY (moveName, typeName),
        FOREIGN KEY (moveName) REFERENCES
                Move(moveName)
                ON UPDATE CASCADE
                ON DELETE CASCADE,
        FOREIGN KEY (typeName) REFERENCES
                ElementalType(typeName)
                ON UPDATE CASCADE
                ON DELETE CASCADE
        )

CREATE TABLE EffectiveAgainst
        (strong_typeName                VARCHAR,
        weak_typeName                   VARCHAR,
        PRIMARY KEY (strong_typeName, weak_typeName),
        FOREIGN KEY (strong_typeName) REFERENCES
                ElementalType(typeName)
                ON UPDATE CASCADE
                ON DELETE CASCADE,
        FOREIGN KEY (weak_typeName) REFERENCES
                ElementalType(typeName)
                ON UPDATE CASCADE
                ON DELETE CASCADE,
        )
```

INSERT Statements

Gym

```
INSERT INTO Gym(gymName, gymLeader, type, locatedIn_regionName)
        VALUES('Pewter City Gym', 'Brock', 'Rock', 'Kanto')
INSERT INTO Gym(gymName, gymLeader, type, locatedIn_regionName)
        VALUES('Cerulean City Gym', 'Misty', 'Water', 'Kanto')
INSERT INTO Gym(gymName, gymLeader, type, locatedIn_regionName)
        VALUES('Vermillion City Gym', 'Lt. Surge', 'Electric', 'Kanto')
INSERT INTO Gym(gymName, gymLeader, type, locatedIn_regionName)
        VALUES('Celadon City Gym', 'Erika', 'Grass', 'Kanto')
INSERT INTO Gym(gymName, gymLeader, type, locatedIn_regionName)
        VALUES('Fuchsia City Gym', 'Koga', 'Poison', 'Kanto')
```

GymBadge

```
INSERT INTO GymBadge(gymName, trainerID, badgeID)
        VALUES('Pewter City Gym', 001301, 000001)
INSERT INTO GymBadge(gymName, trainerID, badgeID)
        VALUES('Pewter City Gym', 006214, 000002)
INSERT INTO GymBadge(gymName, trainerID, badgeID)
        VALUES('Cerulean City Gym', 006845, 000001)
INSERT INTO GymBadge(gymName, trainerID, badgeID)
        VALUES('Cerulean City Gym', 006214, 000010)
INSERT INTO GymBadge(gymName, trainerID, badgeID)
        VALUES('Fuchsia City Gym', 006214, 000001)
```

GymBadgeName

```
INSERT INTO GymBadgeName(gymName, badgeName)
        VALUES('Pewter City Gym', 'Boulder Badge')
INSERT INTO GymBadgeName(gymName, badgeName)
        VALUES('Cerulean City Gym', 'Cascade Badge')
INSERT INTO GymBadgeName(gymName, badgeName)
        VALUES('Vermillion City Gym', 'Thunder Badge')
INSERT INTO GymBadgeName(gymName, badgeName)
        VALUES('Celadon City Gym', 'Rainbow Badge')
INSERT INTO GymBadgeName(gymName, badgeName)
        VALUES('Fuchsia City Gym', 'Soul Badge')
```

GymBadgeDamage

```
INSERT INTO GymBadgeDamage(badgeName, increasedStat)
        VALUES('Boulder Badge', 'Raises Attack')
INSERT INTO GymBadgeDamage(badgeName, increasedStat)
        VALUES('Cascade Badge', 'Increases Obeying Pokemon up to Lv. 30')
INSERT INTO GymBadgeDamage(badgeName, increasedStat)
```

```
        VALUES('Thunder Badge', 'Raises Defense')
INSERT INTO GymBadgeDamage(badgeName, increasedStat)
        VALUES('Rainbow Badge', 'Increases Obeying Pokemon up to Lv. 50')
INSERT INTO GymBadgeDamage(badgeName, increasedStat)
        VALUES('Soul Badge', 'Raises Speed')
```

PokemonTrainer

```
INSERT INTO PokemonTrainer(trainerID, name, startingIn_regionName)
        VALUES(001301, 'Ash', 'Kanto')
INSERT INTO PokemonTrainer(trainerID, name, startingIn_regionName)
        VALUES(006845, 'Brock', 'Kanto')
INSERT INTO PokemonTrainer(trainerID, name, startingIn_regionName)
        VALUES(006214, 'Jessie', 'Kanto')
INSERT INTO PokemonTrainer(trainerID, name, startingIn_regionName)
        VALUES(220601, 'Dawn', 'Sinnoh')
INSERT INTO PokemonTrainer(trainerID, name, startingIn_regionName)
        VALUES(220602, 'Zoey', 'Sinnoh')
```

Region

```
INSERT INTO Region(regionName, professor)
        VALUES('Kanto', 'Professor Oak')
INSERT INTO Region(regionName, professor)
        VALUES('Johto', 'Professor Elm')
INSERT INTO Region(regionName, professor)
        VALUES('Hoenn', 'Professor Birch')
INSERT INTO Region(regionName, professor)
        VALUES('Sinnoh', 'Professor Rowan')
INSERT INTO Region(regionName, professor)
        VALUES('Alola', 'Professor Kukui')
```

Species

```
INSERT INTO Species(speciesID, speciesName, speciesDescription, evolvesFrom_speciesID)
      VALUES(0025, 'Pikachu', 'When it is angered, it immediately discharges the energy stored
      in the pouches in its cheeks.', 0172)
INSERT INTO Species(speciesID, speciesName, speciesDescription, evolvesFrom_speciesID)
      VALUES(0026, 'Raichu', 'Its tail discharges electricity into the ground, protecting it from
      getting shocked.', 0025)
INSERT INTO Species(speciesID, speciesName, speciesDescription, evolvesFrom_speciesID)
      VALUES(0133, 'Eevee', 'Its ability to evolve into many forms allows it to adapt smoothly
      and perfectly to any environment.', NULL)
INSERT INTO Species(speciesID, speciesName, speciesDescription, evolvesFrom_speciesID)
      VALUES(0197, 'Umbreon', 'When exposed to the moon's aura, the rings on its body glow
      faintly and it gains a mysterious power.', 0133)
INSERT INTO Species(speciesID, speciesName, speciesDescription, evolvesFrom_speciesID)
```

```
        VALUES(0052, 'Meowth', 'All it does is sleep during the daytime. At night, it patrols its
        territory with its eyes aglow.', NULL)
INSERT INTO Species(speciesID, speciesName, speciesDescription, evolvesFrom_speciesID)
        VALUES(0185, 'Sudowoodo', 'Although it always pretends to be a tree, its composition
        appears more similar to rock than to vegetation.', 0438)
INSERT INTO Species(speciesID, speciesName, speciesDescription, evolvesFrom_speciesID)
        VALUES(0393, 'Piplup', 'A poor walker, it often falls down. However, its strong pride
        makes it puff up its chest without a care.', NULL)
```

Pokemon

```
INSERT INTO Pokemon(pokemonID, speciesID, level, nature, statusCondition,
        holding_itemName)
        VALUES(025555, 0025, 43, 'Brave', NULL, 'Everstone')
INSERT INTO Pokemon(pokemonID, speciesID, level, nature, statusCondition,
        holding_itemName)
        VALUES(101010, 0052, 1, 'Naughty', 'PAR', NULL)
INSERT INTO Pokemon(pokemonID, speciesID, level, nature, statusCondition,
        holding_itemName)
        VALUES(123123, 0054, 99, 'Quirky', NULL, NULL)
INSERT INTO Pokemon(pokemonID, speciesID, level, nature, statusCondition,
        holding_itemName)
        VALUES(456456, 0185, 83, 'Brave', NULL, 'Exp. Share')
INSERT INTO Pokemon(pokemonID, speciesID, level, nature, statusCondition,
        holding_itemName)
        VALUES(789789, 0393, 20, 'Sassy', NULL, 'Chesto Berry')
```

PokemonHP

```
INSERT INTO PokemonHP(speciesID, level, HP)
          VALUES(0025, 43, 83)
INSERT INTO PokemonHP(speciesID, level, HP)
          VALUES(0185, 83, 209)
INSERT INTO PokemonHP(speciesID, level, HP)
          VALUES(0052, 1, 11)
INSERT INTO PokemonHP(speciesID, level, HP)
          VALUES(0054, 99, 208)
INSERT INTO PokemonHP(speciesID, level, HP)
          VALUES(0197, 15, 53)
```

TamedPokemon

```
INSERT INTO TamedPokemon(pokemonID, speciesID, dateCaught, nickname,
        ownedBy_trainerID, boxNumber, slotNumber)
        VALUES(025555, 0025, '1996-02-27', NULL, 001301, 0, 1)
INSERT INTO TamedPokemon(pokemonID, speciesID, dateCaught, nickname,
        ownedBy_trainerID, boxNumber, slotNumber)
```

```
    VALUES(123123, 0054, '1996-02-29', 'Sir Duckington', 001301, 1, 3)
INSERT INTO TamedPokemon(pokemonID, speciesID, dateCaught, nickname,
    ownedBy_trainerID, boxNumber, slotNumber)
    VALUES(101010, 0052, '1996-02-28', NULL, 006214, 0, 1)
INSERT INTO TamedPokemon(pokemonID, speciesID, dateCaught, nickname,
    ownedBy_trainerID, boxNumber, slotNumber)
    VALUES(456456, 0185, '2024-07-19', NULL, 006845, 0, 1)
INSERT INTO TamedPokemon(pokemonID, speciesID, dateCaught, nickname,
    ownedBy_trainerID, boxNumber, slotNumber)
    VALUES(789789, 0393, '2024-04-20', NULL, 220601, 0, 1)
```

ElementalType

```
INSERT INTO ElementalType(typeName) VALUES ('Rock')
INSERT INTO ElementalType(typeName) VALUES ('Water')
INSERT INTO ElementalType(typeName) VALUES ('Electric')
INSERT INTO ElementalType(typeName) VALUES ('Grass')
INSERT INTO ElementalType(typeName) VALUES ('Poison')
INSERT INTO ElementalType(typeName) VALUES ('Normal')
INSERT INTO ElementalType(typeName) VALUES ('Dark')
INSERT INTO ElementalType(typeName) VALUES ('Fairy')
INSERT INTO ElementalType(typeName) VALUES ('Psychic')
```

Item

```
INSERT INTO Item(itemName, itemEffect)
    VALUES('Exp. Share', 'The holder gets a share of a battle's Exp. Points without battling.')
INSERT INTO Item(itemName, itemEffect)
    VALUES('Potion', 'Restores 20 HP.')
INSERT INTO Item(itemName, itemEffect)
    VALUES('Everstone', 'The Pokemon holding this peculiar stone is prevented from
    evolving.')
INSERT INTO Item(itemName, itemEffect)
    VALUES('Chesto Berry', 'If held by a Pokemon, it recovers from sleep.')
INSERT INTO Item(itemName, itemEffect)
    VALUES('Rock Gem', 'Increases the power of a Rock-type move only once.')
```

Move

```
INSERT INTO Move(moveName, moveDescription, powerPoints, power, statusCondition, effect)
    VALUES('Nuzzle', 'The user attacks by nuzzling its electrified cheeks against the target.
    This also leaves the target with paralysis.', 20, 20, 'Paralysis', NULL)
INSERT INTO Move(moveName, moveDescription, powerPoints, power, statusCondition, effect)
    VALUES('Scratch', 'Scratches the foe with sharp claws.', 35, 40, NULL, NULL)
INSERT INTO Move(moveName, moveDescription, powerPoints, power, statusCondition, effect)
    VALUES('Fake Tears', ' Feigns crying to sharply lower the foe's Sp. Def.', 20, NULL,
    NULL, 'Lowers the target's Special Defense stat by two stages.')
```

```
INSERT INTO Move(moveName, moveDescription, powerPoints, power, statusCondition, effect)
      VALUES('Bubble Beam', 'Forcefully sprays bubbles that may lower Speed.', 20, 65, NULL,
      'Has a 33.2% chance of lowering the target's Speed stat by one stage.')
INSERT INTO Move(moveName, moveDescription, powerPoints, power, statusCondition, effect)
      VALUES('Confusion', 'A psychic attack that may cause confusion.', 25, 50, 'Confusion',
      NULL)
INSERT INTO Move(moveName, moveDescription, powerPoints, power, statusCondition, effect)
      VALUES('Harden', 'Stiffens the body's muscles to raise Defense.', 30, 0, NULL, 'Increases
      the user's Defense stat by one stage.')
```

EntersInPokedex

```
INSERT INTO EntersInPokedex(speciesID, trainerID, encounterDate)
        VALUES(0025, 001301, '1996-02-27')
INSERT INTO EntersInPokedex(speciesID, trainerID, encounterDate)
        VALUES(0054, 001301, '1996-05-22')
INSERT INTO EntersInPokedex(speciesID, trainerID, encounterDate)
        VALUES(0025, 006214, '1996-02-29')
INSERT INTO EntersInPokedex(speciesID, trainerID, encounterDate)
        VALUES(0185, 006845, '2024-07-21')
INSERT INTO EntersInPokedex(speciesID, trainerID, encounterDate)
        VALUES(0197, 220602, '2024-07-14')
```

HasSpeciesType

```
INSERT INTO HasSpeciesType(speciesID, typeName)
        VALUES(0025, 'Electric')
INSERT INTO HasSpeciesType(speciesID, typeName)
        VALUES(0006, 'Fire')
INSERT INTO HasSpeciesType(speciesID, typeName)
        VALUES(0006, 'Flying')
INSERT INTO HasSpeciesType(speciesID, typeName)
        VALUES(0393, 'Water')
INSERT INTO HasSpeciesType(speciesID, typeName)
        VALUES(0133, 'Normal')
```

HasMove

```
INSERT INTO HasMove(speciesID, pokemonID, moveName, moveSlot)
        VALUES(0025, 025555, 'Nuzzle', 1)
INSERT INTO HasMove(speciesID, pokemonID, moveName, moveSlot)
        VALUES(0025, 025555, 'Bite', 2)
INSERT INTO HasMove(speciesID, pokemonID, moveName, moveSlot)
        VALUES(0185, 456456, 'Fake Tears', 1)
INSERT INTO HasMove(speciesID, pokemonID, moveName, moveSlot)
        VALUES(0185, 456456, 'Harden', 2)
INSERT INTO HasMove(speciesID, pokemonID, moveName, moveSlot)
```

```
        VALUES(0393, 789789, 'Bubble Beam', 3)
```

HasMoveType

```
INSERT INTO HasMoveType(moveName, typeName)
        VALUES('Nuzzle', 'Electric')
INSERT INTO HasMoveType(moveName, typeName)
        VALUES('Bubble Beam', 'Water')
INSERT INTO HasMoveType(moveName, typeName)
        VALUES('Bite', 'Dark')
INSERT INTO HasMoveType(moveName, typeName)
        VALUES('Confusion', 'Psychic')
INSERT INTO HasMoveType(moveName, typeName)
        VALUES('Harden', 'Normal')
```

EffectiveAgainst

```
INSERT INTO EffectiveAgainst(strong_typeName, weak_typeName)
        VALUES('Water', 'Fire')
INSERT INTO EffectiveAgainst(strong_typeName, weak_typeName)
        VALUES('Fire', 'Grass')
INSERT INTO EffectiveAgainst(strong_typeName, weak_typeName)
        VALUES('Grass', 'Water')
INSERT INTO EffectiveAgainst(strong_typeName, weak_typeName)
        VALUES('Fairy', 'Dark')
INSERT INTO EffectiveAgainst(strong_typeName, weak_typeName)
        VALUES('Dark', 'Psychic')
```