University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: 2

Date: Oct 15 2024

Group Number: 98

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Matthew Wan	80099211	k2m5q	mattwan1@student.ubc.ca
Joshua Wu	61994414	x3e0l	joshuawu2004@gmail.com
albert chang	26234147	b4g1b	alb12345@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

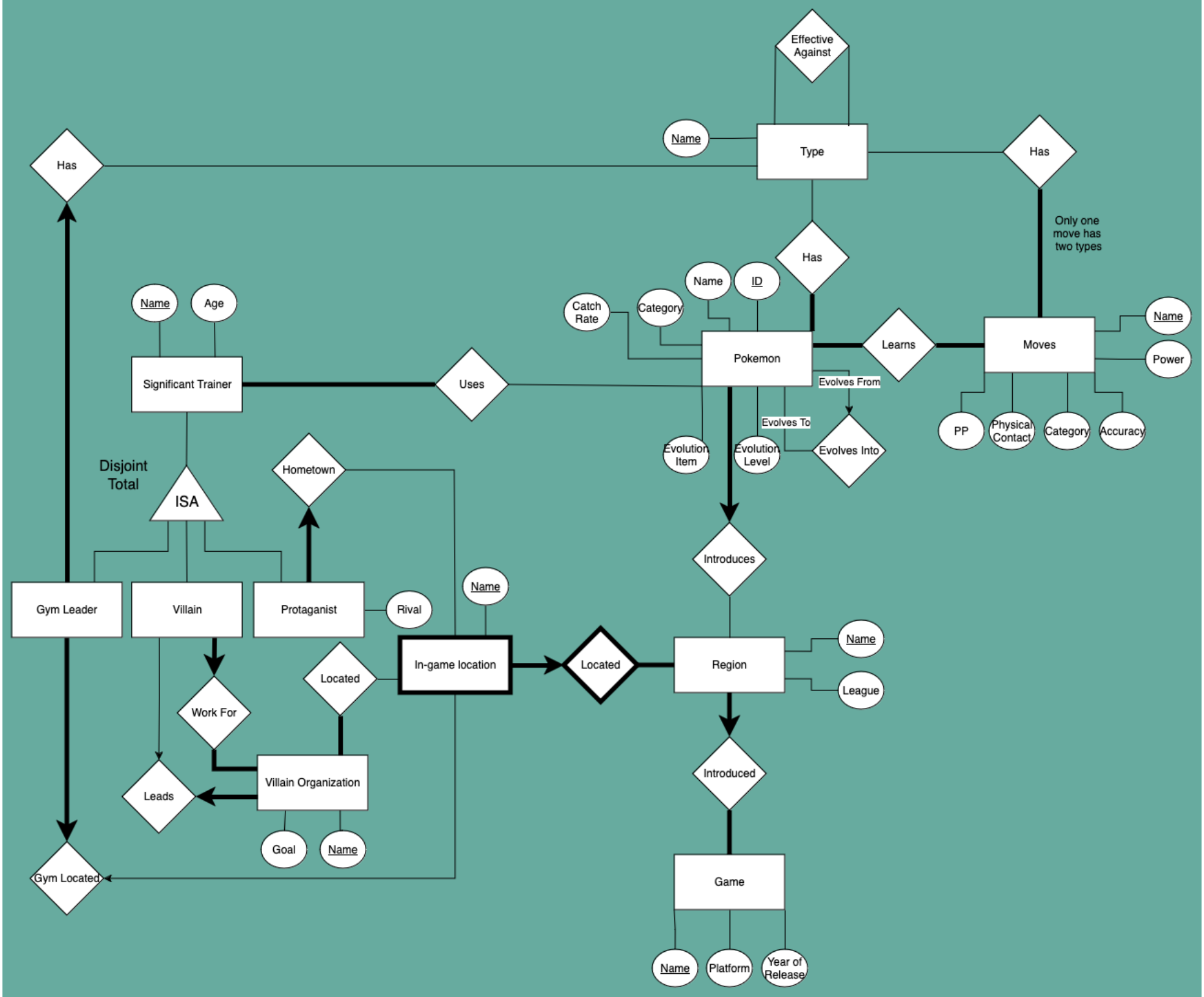
2. Project Summary:

Our project is a centralized hub of information that allows users to access and contribute detailed information about the Pokémon video game series. Information about the game's Pokémon, trainers, gym leaders, moves, and more are to be tracked so users are able to develop strategies and keep track of the vast amount of information available in the Pokemon series. Users should be able to find information such as which trainers possess which Pokémon, what are the weaknesses or strengths of a type, or which Pokémon can learn certain moves.

3. ER DIAGRAM (Available on the following page)

Our project has made some changes to our previous ER diagram. Here are a list of our changes:

- Changed name of Trainer from Trainer to Significant Trainer because it is a more descriptive naming of the types of trainers we want to track in our application.
- Changed name of Criminal Organization to Villain Organization because not every villain in the game is a criminal. Some villains in the game do not conduct illegal activities so we wanted to make the naming more representative of that
- Added goals and leaders to Villain Organizations since those are key reasons villain organizations operate and major characters to look out for when playing the game
- Added types to Gym Leaders as they each represent a type and there are counters to each type in the game making this a major area for strategy.
- Removed weight and height attributes from Pokemon, as these attributes are not consistent throughout the series and are not important to any game mechanics.
- Added evolution level and evolution item attributes to Pokemon as these are key determinants to which Pokemon they will evolve into.
- Added attribute "league" Region table, as the Pokemon league that the player will have to face later in the game is tied to the region the game is set in.


Effective Against
Name
Type
Has
Has
Only one move has two types
Has
Name
ID
Catch Rate
Category
Pokemon
Learns
Moves
Name
Power
Name
Age
Significant Trainer
Uses
Evolves From
Evolves To
Evolution Item
Evolution Level
Evolves Into
PP
Physical Contact
Category
Accuracy
Disjoint Total
ISA
Hometown
Introduces
Gym Leader
Villain
Protaganist
Rival
Name
In-game location
Located
Region
Name
League
Located
Work For
Introduced
Leads
Villain Organization
Goal
Name
Gym Located
Game
Name
Platform
Year of Release

4/5. Schema and FDs derived from the above ER Diagram

Please note, there are some relationship tables that cannot be enforced without using assertions. We have made notes about it such as the relationship “SignificantTrainer” below and provided reasoning as to why we are unable to completely map the relationship based on the slides presented to us during lecture.

We structured our work to be similar to what was done in class, below is a guide:

- Underscored variables - Key (or composite key)
- Bolded variables - Foreign key
- Foreign Key = (FK)

The following schema is NOT in proper DDL. This is similar to DDL so that we are able to determine and create tables more easily after doing our normalization.

SignificantTrainer(
name: VARCHAR,
age: INT,
)
** Can’t perfectly enforce that a SignificantTrainer name is only allowed to be associated with only one of Villain, Gym Leader, or Protagonist without assertions

SignificantTrainer FD:

- name -> age

Villain(
name: VARCHAR, (FK: SignificantTrainer(name))
villain_organizaton_name: VARCHAR, (FK: VillainOrganization(organization_name))
)
Villain FD:

- trainer_name-> villain_organization_name

GymLeader(
name: VARCHAR, (FK: SignificantTrainer(name))
gym_location: VARCHAR, (FK: InGameLocation(in_game_location_name))
gym_region: VARCHAR, (FK: Region(region_name))
type: VARCHAR, (FK: Type(type_name))
)
GymLeader FD:

- name -> gym_location, gym_region, type

Protagonist(
name: VARCHAR, (FK: SignificantTrainer(name))
home_location: VARCHAR, (FK: InGameLocation(in_game_location_name))
home_region: VARCHAR, (FK: Region(region_name))
rival: VARCHAR,
)
Protagonist FD:

- trainer_name -> rival, home_location, home_region

VillainOrganization(
organization_name: VARCHAR,
goal: VARCHAR,
leader: VARCHAR, (FK: Villain(name))
)
VillainOrganization FD:

- organization_name -> goal, leader
- leader -> organization_name, goal

VillainBases(
organization_name: VARCHAR, (FK: VillainOrganization(organization_name)),
base_region: VARCHAR, (FK: Region(region_name)),
base_location: VARCHAR, (FK: InGameLocation(in_game_location_name))
)

PokemonRoster(
name: VARCHAR, (FK: SignificantTrainer(name))
id: int, (FK: Pokemon(id))
)

Pokemon(
id: int,
pokemon_name: VARCHAR,
category: VARCHAR,
evolution_item: VARCHAR,
evolution_level: int,
catch_rate: int,
region_name: VARCHAR (FK: Region(region_name)),
from_id: int (FK: Pokemon(id))
)
Pokemon FD:

- id -> pokemon_name, category, catch_rate, region_name, from_id
- pokemon_name -> id, category, catch_rate, region_name, from_id
- evolution_item, evolution_level, from_id -> id, pokemon_name

Uses(
name: VARCHAR, (FK: SignificantTrainer(Name))
pokemon_id: int, (FK: Pokemon(id))
)
** Can't perfectly enforce 1-M (Pokemon) and 0-M (Type) relationships without assertions.

Type(
type_name: VARCHAR
)

Effective Against(
strong-name: VARCHAR (FK: Type(type_name)),
weak-name: VARCHAR (FK: Type(type_name))
)

PokemonHasType(
id: int, (FK: Pokemon(id))
type_name: VARCHAR (FK: Type(type_name))
)
** Can't perfectly enforce 1-M (Pokemon) and 0-M (Type) relationships without assertions where pokemon needs to have at least 1 type.

MoveHasType(
move_name: VARCHAR, (FK: Move(move_name))
type_name: VARCHAR (FK: Type(type_name))
)
** Can't perfectly enforce 1-M (Moves) and 0-M (Type) relationships where moves can have more than one type, but need at least one.

Move(
move_name: VARCHAR,
power: int,
pp: int,
physical_contact: boolean,
category: VARCHAR,
accuracy: int
)
Move:
- name -> power, pp, physical_contact, category, accuracy

Learns(
move_name: VARCHAR (FK: Move(move_name)),
id: int (FK: Pokemon(id))
)
** Can't perfectly enforce 1-M (Pokemon) and 1-M (Move) relationships without assertions.

Game(
game_name: VARCHAR,
platform: VARCHAR,
release_year: YEAR
)
Game:
- game_name -> platform, release_year

Region(
region_name: VARCHAR,
introduced_by_game: VARCHAR (FK: Game(game_name))
league: VARCHAR
)
Region:
- region_name -> introduced_by_game, league

InGameLocation(
in_game_location_name: VARCHAR,
region_name: VARCHAR (FK: Region(name))
)

InGameLocation:

- in_game_location_name, region_name -> gym_name, protagonist
- gym_name -> in_game_location_name, region_name
- protagonist_name -> in_game_location_name, region_name

VillainOrganizationLocatedIn(
organization_name: VARCHAR (FK: VillainOrganization(name)),
location_name: VARCHAR (FK: InGameLocation(in_game_location_name)),
region_name: VARCHAR (FK: Region(region_name)),
)

** Can't perfectly enforce 1-M (Organization) and 0-M (InGameLocation) relationships without assertions.

6. Normalization

Villain Relationship:
VillainOrganization(organization_name, goal, leader)

VillainOrganization FD:

- organization_name -> goal, leader
- leader -> organization_name, goal

Keys are name and leader:
organization_name+ = {organization_name, leader, goal}
leader+ = {leader, organization_name, goal}

3NF:
Step 1: Finding minimal cover:

organization_name -> goal
organization_name -> leader
leader -> goal
leader -> organization_name

No LHS to reduce

Removing organization_name->goal does not affect either of the closures so organization_name->goal is redundant.

The minimal cover is then thus:
organization_name -> leader
leader -> goal
leader -> organization_name

Step 2:
Using synthesis:

We create relationships for all the minimal cover FDs:
VillainOrganization(<u>organization_name</u>, leader)
OrganizationGoal(<u>leader</u>, goal)

There are no redundant FDs, no reduction is necessary and the above is our final answer

Pokemon Relationship
PokemonFD:

- id -> pokemon_name, category, catch_rate, from_id, region_name
- pokemon_name -> id, category, catch_rate, from_id, region_name
- evolution_item, evolution_level, from_id -> id, pokemon_name

Pokemon(id, name, category, evolution_item, evolution_level, catch rate, region_name, from_id)

Pokemon(
id: int,
pokemon_name: VARCHAR,
category: VARCHAR,
evolution_item: VARCHAR,
evolution_level: int,
catch_rate: int,
region_name: VARCHAR (FK: Region(region_name)),
from_id: int (FK: Pokemon(id))
)

LHS: evolution_item, evolution_level
MID: id, pokemon_name, from_id
RHS: category, catch rate, from_id, region_name

Keys are:

evolution_item, evolution_level, id
evolution_item, evolution_level, pokemon_name
evolution_item, evolution_level, from_id

Finding minimal cover:

Put FDs in standard form:

- id -> pokemon_name
- id -> category,
- id -> catch_rate
- id -> from_id
- id -> region_name

- pokemon_name -> id
- pokemon_name -> category
- pokemon_name -> catch_rate
- pokemon_name -> from_id
- pokemon_name -> region_name

- evolution_item, evolution_level, from_id -> id
- evolution_item, evolution_level, from_id -> pokemon_name

We cannot remove anything from LHS

id+ = {id, pokemon_name, category, catch_rate, from_id, region_name}

pokemon_name+ = {pokemon_name, id, category, catch_rate, from_id, region_name}

evolution_item, evolution_level, from_id+ = {id, category, catch rate, category, catch rate, from_id, region_name}

We can remove
- id -> category,
- id -> catch_rate
- id -> from_id
- id -> region_name

Because we can still find the closures using: id -> pokemon_name
We can also remove:
- evolution_item, evolution_level, from_id -> pokemon_name

Because we can find pokemon name using evolution_item, evolution_level, from_id -> id

Our final minimal cover is:

- pokemon_name -> id
- pokemon_name -> category
- pokemon_name -> catch_rate
- pokemon_name -> from_id
- pokemon_name -> region_name

- id -> pokemon_name

- evolution_item, evolution_level, from_id -> id

Decompose into BCNF using lossless join:

Original relationship:
Pokemon(id, pokemon_name, category, catch_rate, from_id, region_name, evolution_item, evolution_level)

Decompose using: pokemon_name->id
Pokemon1(pokemon_name, id)
Pokemon2(pokemon_name, category, catch_rate, from_id, region_name, evolution_item, evolution_level)

Decompose using: pokemon_name->category
Pokemon1(id, pokemon_name)
Pokemon2(pokemon_name, category)
Pokemon3(pokemon_name, catch_rate, from_id, region_name, evolution_item, evolution_level)

Decompose using: pokemon_name -> catch_rate
Pokemon1(id, pokemon_name)
Pokemon2(pokemon_name, category)
Pokemon3(pokemon_name, catch rate)

Pokemon4(pokemon_name, from_id, region_name, evolution_item, evolution_level)

Decompose using pokemon_name -> from_id
Pokemon1(id, pokemon_name)
Pokemon2(pokemon_name, category)
Pokemon3(pokemon_name, catch_rate)
Pokemon4(pokemon_name, from_id)
Pokemon5(pokemon_name, region_name, evolution_item, evolution_level)

Decompose using pokemon_name -> region_name
Pokemon1(id, pokemon_name)
Pokemon2(pokemon_name, category)
Pokemon3(pokemon_name, catch_rate)
Pokemon4(pokemon_name, from_id)
Pokemon5(pokemon_name, region_name)
Pokemon6(pokemon_name, evolution_item, evolution_level)

Add back in any removed FDs:
Pokemon7(evolution_item, evolution_level, from_id, id)

Our final decomposition results in:

Pokemon1(id, pokemon_name)
Pokemon2(pokemon_name, category)
Pokemon3(pokemon_name, catch_rate)
Pokemon4(pokemon_name, from_id)
Pokemon5(pokemon_name, region_name)
Pokemon6(pokemon_name, evolution_item, evolution_level)
Pokemon7(evolution_item, evolution_level, from_id, id)

We can remove Pokemon4 and Pokemon6 because Pokemon7 encompasses all the details of this FD for us since we can find the pokemon_name from id. Pokemon7 contains more relevant information for us to keep.
And we rename the relationships to be more descriptive:

PokemonNameId(id, pokemon_name)
PokemonCategory(pokemon_name, category)
PokemonCatchRate(pokemon_name, catch_rate)
PokemonIntroducedIn(pokemon_name, region_name)
PokemonEvolveConditions(evolution_item, evolution_level, from_id, id)

7. SQL DDL

```
CREATE TABLE SignificantTrainer (

   name VARCHAR(255),

   age INT,

   PRIMARY KEY (name)

);



CREATE TABLE Villain (

   name VARCHAR(255),

   villain_organization_name VARCHAR(255),

   PRIMARY KEY (name),

   CONSTRAINT fk_significant_trainer FOREIGN KEY (name)

       REFERENCES SignificantTrainer(name) ON DELETE CASCADE ON UPDATE CASCADE,

   CONSTRAINT fk_villain_organization FOREIGN KEY (villain_organization_name)

       REFERENCES VillainOrganization(organization_name) ON DELETE SET NULL ON UPDATE
CASCADE

);



CREATE TABLE GymLeader (

   name VARCHAR(255),
```

```
    gym_location VARCHAR(255),
    gym_region VARCHAR(255),
    type VARCHAR(50),
    PRIMARY KEY (name),
    CONSTRAINT fk_significant_trainer FOREIGN KEY (name)
        REFERENCES SignificantTrainer(name) ON DELETE CASCADE ON UPDATE CASCADE,
    CONSTRAINT fk_in_game_location FOREIGN KEY (gym_location, gym_region)
        REFERENCES InGameLocation(in_game_location_name, region_name) ON DELETE SET
NULL ON UPDATE CASCADE,
    CONSTRAINT fk_type FOREIGN KEY (type)
        REFERENCES Type(type_name) ON DELETE SET NULL ON UPDATE CASCADE
);


CREATE TABLE Protagonist (
    name VARCHAR(255),
    hometown_location VARCHAR(255),
    hometown_region VARCHAR(255),
    rival VARCHAR(255),
    PRIMARY KEY (name),
    CONSTRAINT fk_significant_trainer FOREIGN KEY (name)
```

```
      REFERENCES SignificantTrainer(name) ON DELETE CASCADE ON UPDATE CASCADE,

   CONSTRAINT fk_hometown FOREIGN KEY (hometown_location, hometown_region)

      REFERENCES InGameLocation(in_game_location_name, region_name) ON DELETE SET
NULL ON UPDATE CASCADE

);



CREATE TABLE VillainOrganization (

   organization_name VARCHAR(255),

   leader VARCHAR(255) NOT NULL,

   PRIMARY KEY (organization_name),

   CONSTRAINT fk_villain FOREIGN KEY (leader)

      REFERENCES Villain(name) ON DELETE CASCADE ON UPDATE CASCADE

);



CREATE TABLE OrganizationGoal (

   leader VARCHAR(255),

   goal VARCHAR(255),

   PRIMARY KEY (leader),

   CONSTRAINT fk_villain FOREIGN KEY (leader)

      REFERENCES Villain(name) ON DELETE CASCADE ON UPDATE CASCADE
```

```
);



CREATE TABLE VillainBases (

   organization_name VARCHAR(255),

   base_location VARCHAR(255),

   base_region VARCHAR(255),

   PRIMARY KEY (organization_name, base_location, base_region),

   CONSTRAINT fk_villain_organization FOREIGN KEY (organization_name)

       REFERENCES VillainOrganization(organization_name) ON DELETE CASCADE ON UPDATE
CASCADE,

   CONSTRAINT fk_base_location FOREIGN KEY (base_location, base_region)

       REFERENCES InGameLocation(in_game_location_name, region_name) ON DELETE CASCADE
ON UPDATE CASCADE

);



CREATE TABLE PokemonRoster (

   name VARCHAR(255),

   id INT,

   PRIMARY KEY (name, id)

   CONSTRAINT fk_significant_trainer FOREIGN KEY (name)

       REFERENCES SignificantTrainer(name) ON DELETE CASCADE ON UPDATE CASCADE
```

```
    CONSTRAINT fk_pokemon_id FOREIGN KEY (id)
        REFERENCES PokemonNameId(id) ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE PokemonNameId (
    id INT,
    pokemon_name VARCHAR(255) NOT NULL UNIQUE,
    PRIMARY KEY (id)
);

CREATE TABLE PokemonCategory (
    pokemon_name VARCHAR(255),
    category VARCHAR(255) NOT NULL,
    PRIMARY KEY (pokemon_name),
    CONSTRAINT fk_pokemon_name FOREIGN KEY (pokemon_name)
        REFERENCES PokemonNameId(pokemon_name) ON DELETE CASCADE
);

CREATE TABLE PokemonCatchRate (
    pokemon_name VARCHAR(255),
```

```
    catch_rate INT NOT NULL,
    PRIMARY KEY (pokemon_name),
    CONSTRAINT fk_pokemon_name FOREIGN KEY (pokemon_name)
        REFERENCES PokemonNameId(pokemon_name) ON DELETE CASCADE
);


CREATE TABLE PokemonIntroducedIn (
    pokemon_name VARCHAR(255),
    region_name VARCHAR(255) NOT NULL,
    PRIMARY KEY (pokemon_name),
    CONSTRAINT fk_pokemon_name FOREIGN KEY (pokemon_name)
        REFERENCES PokemonNameId(pokemon_name) ON DELETE CASCADE
);


CREATE TABLE PokemonEvolveConditions (
    evolution_item VARCHAR(255),
    evolution_level INT,
    from_id INT,
    id INT NOT NULL,
    PRIMARY KEY (evolution_item, evolution_level, from_id),
```

```
    CONSTRAINT fk_from_id FOREIGN KEY (from_id)
        REFERENCES PokemonNameId(id) ON DELETE CASCADE
    CONSTRAINT fk_pokemon_id FOREIGN KEY (id)
        REFERENCES PokemonNameId(id) ON DELETE CASCADE
);


CREATE TABLE Type (
    type_name VARCHAR(50),
    PRIMARY KEY (type_name)
);


CREATE TABLE EffectiveAgainst (
    strong_name: VARCHAR(50),
    weak_name: VARCHAR(50),
    PRIMARY KEY (strong_name, weak_name),
    CONSTRAINT fk_strong_name FOREIGN KEY (strong_name)
        REFERENCES Type(type_name) ON DELETE CASCADE ON UPDATE CASCADE,
    CONSTRAINT fk_weak_name FOREIGN KEY (weak_name)
        REFERENCES Type(type_name) ON DELETE CASCADE ON UPDATE CASCADE
);
```

```sql
CREATE TABLE PokemonHasType (
   id INT,
   type_name VARCHAR(50),
   PRIMARY KEY (id, type_name),
   CONSTRAINT fk_pokemon_id FOREIGN KEY (id)
       REFERENCES PokemonNameId(id) ON DELETE CASCADE ON UPDATE CASCADE,
   CONSTRAINT fk_type_name FOREIGN KEY (type_name)
       REFERENCES Type(type_name) ON DELETE CASCADE ON UPDATE CASCADE
);


CREATE TABLE MoveHasType (
   move_name VARCHAR(255),
   type_name VARCHAR(50),
   PRIMARY KEY (move_name, type_name),
   CONSTRAINT fk_move_name FOREIGN KEY (move_name)
       REFERENCES Move(move_name) ON DELETE CASCADE ON UPDATE CASCADE,
   CONSTRAINT fk_type_name FOREIGN KEY (type_name)
       REFERENCES Type(type_name) ON DELETE CASCADE ON UPDATE CASCADE
);
```

```sql
CREATE TABLE Move (
   move_name VARCHAR(255),
   power INT NOT NULL,
   pp INT NOT NULL,
   physical_contact BOOLEAN NOT NULL,
   category VARCHAR(50) NOT NULL,
   accuracy INT NOT NULL,
   PRIMARY KEY (move_name)
);


CREATE TABLE Learns (
   move_name VARCHAR(255),
   id INT,
   PRIMARY KEY (move_name, id),
   CONSTRAINT fk_move_name FOREIGN KEY (move_name)
       REFERENCES Move(move_name) ON DELETE CASCADE ON UPDATE CASCADE,
   CONSTRAINT fk_pokemon_id FOREIGN KEY (id)
       REFERENCES PokemonNameId(id) ON DELETE CASCADE ON UPDATE CASCADE
);
```

```sql
CREATE TABLE Game (
   game_name VARCHAR(255),
   platform VARCHAR(255),
   release_year YEAR,
   PRIMARY KEY (game_name)
);

CREATE TABLE Region (
   region_name VARCHAR(255),
   introduced_by_game VARCHAR(255),
   league VARCHAR(255),
   PRIMARY KEY (region_name),
   CONSTRAINT fk_game_name FOREIGN KEY (introduced_by_game)
       REFERENCES Game(game_name) ON DELETE SET NULL ON UPDATE CASCADE
);

CREATE TABLE InGameLocation (
   in_game_location_name VARCHAR(255),
   region_name VARCHAR(255),
```

```
    PRIMARY KEY (in_game_location_name, region_name),

    CONSTRAINT fk_region_name FOREIGN KEY (region_name)

        REFERENCES Region(region_name) ON DELETE CASCADE ON UPDATE CASCADE

);
```

8. INSERT Statements

SignificantTrainer:
- name, age

INSERT INTO SignificantTrainer (name, age)
VALUES (“Red”, 11)
INSERT INTO SignificantTrainer (name, age)
VALUES (“Cyrus”, 27)
INSERT INTO SignificantTrainer (name, age)
VALUES (“Brock”, 15)
INSERT INTO SignificantTrainer (name, age)
VALUES (“Giovanni”, NULL)
INSERT INTO SignificantTrainer (name, age)
VALUES (“Ash”, 10)

name	age
Red	11
Cyrus	27
Brock	15
Giovanni	NULL
Ash	10

Villain:
- name, villain_organization_name

INSERT INTO Villain (name, villain_organization_name)
VALUES (“Ariana”, “Team Rocket”)
INSERT INTO Villain (name, villain_organization_name)
VALUES (“Archer”, “Team Rocket”)
INSERT INTO Villain (name, villain_organization_name)
VALUES (“Tabitha”, “Team Magma”)
INSERT INTO Villain (name, villain_organization_name)
VALUES (“Shelly”, “Team Aqua”)
INSERT INTO Villain (name, villain_organization_name)
VALUES (“Saturn”, “Team Galactic”)

name	villain_organization_name
Ariana	Team Rocket
Archer	Team Rocket
Tabitha	Team Magma

Shelly	Team Aqua
Saturn	Team Galactic

GymLeader:
- name, gym_location, gym_region, type

INSERT INTO GymLeader (name, gym_location, gym_region, type)
VALUES (“Brock”, “Pewter City”, “Kanto”, “Rock”)
INSERT INTO GymLeader (name, gym_location, gym_region, type)
VALUES (“Misty”, “Cerulean City”, “Kanto”, “Water”)
INSERT INTO GymLeader (name, gym_location, gym_region, type)
VALUES (“Lt. Surge”, “Vermilion City”, “Kanto”, “Electric”)
INSERT INTO GymLeader (name, gym_location, gym_region, type)
VALUES (“Erika”, “Celadon City”, “Kanto”, “Grass”)
INSERT INTO GymLeader (name, gym_location, gym_region, type)
VALUES (“Koga”, “Fuchsia City”, “Kanto”, “Poison”)

name	gym_location	gym_region	type
Brock	Pewter City	Kanto	Rock
Misty	Cerulean City	Kanto	Water
Lt. Surge	Vermilion City	Kanto	Electric
Erika	Celadon City	Kanto	Grass
Koga	Fuchsia City	Kanto	Poison

Protagonist:
- name, hometown_location, hometown_region, rival

INSERT INTO Protagonist (name, hometown_location, hometown_region, rival)
VALUES (“Red”, “Pallet Town”, “Kanto”, “Blue”)
INSERT INTO Protagonist (name, hometown_location, hometown_region, rival)
VALUES (“Ethan”, “New Bark Town”, “Johto”, “Silver”)
INSERT INTO Protagonist (name, hometown_location, hometown_region, rival)
VALUES (“May”, “Littleroot Town”, “Hoenn”, “Wally”)
INSERT INTO Protagonist (name, hometown_location, hometown_region, rival)
VALUES (“Dawn”, “Twinleaf Town”, “Sinnoh”, “Barry”)
INSERT INTO Protagonist (name, hometown_location, hometown_region, rival)
VALUES (“Hilbert”, “Nuvema Town”, “Unova”, “Natural Harmonia Gropius”)

name	hometown_location	hometown_region	rival
Red	Pallet Town	Kanto	Blue
Ethan	New Bark Town	Johto	Silver

May	Littleroot Town	Hoenn	Wally
Dawn	Twinleaf Town	Sinnoh	Barry
Hilbert	Nuvema Town	Unova	Natural Harmonia Gropius

VillainOrganization:
- organization_name, leader

INSERT INTO VillainOrganization (organization_name, leader)
VALUES (“Team Rocket”, “Giovanni”)
INSERT INTO VillainOrganization (organization_name, leader)
VALUES (“Team Magma”, “Maxie”)
INSERT INTO VillainOrganization (organization_name, leader)
VALUES (“Team Aqua”, “Archie”)
INSERT INTO VillainOrganization (organization_name, leader)
VALUES (“Team Galactic”, “Cyrus”)
INSERT INTO VillainOrganization (organization_name, leader)
VALUES (“Team Plasma”, “Ghetsis”)

organization_name	leader
Team Rocket	Giovanni
Team Magma	Maxie
Team Aqua	Archie
Team Galactic	Cyrus
Team Plasma	Ghetsis

OrganizationGoal:
- leader, goal

INSERT INTO OrganizationGoal (leader, goal)
VALUES (“Giovanni”, “Steal and exploit pokemon for profit, world domination”)
INSERT INTO OrganizationGoal (leader, goal)
VALUES (“Maxie”, “Expand the amount of landmass in the world”)
INSERT INTO OrganizationGoal (leader, goal)
VALUES (“Archie”, “Expand the amount of water in the world”)
INSERT INTO OrganizationGoal (leader, goal)
VALUES (“Cyrus”, “Recreation of entire pokemon universe”)
INSERT INTO OrganizationGoal (leader, goal)
VALUES (“Ghetsis”, “Liberate all pokemon from trainers”)

leader	goal
Giovanni	Steal and exploit pokemon for profit, world

	domination
Maxie	Expand the amount of landmass in the world
Archie	Expand the amount of water in the world
Cyrus	Recreation of entire pokemon universe
Ghetsis	Liberate all pokemon from trainers

VillainBases:

- organization_name, base_location, base_region

INSERT INTO VillainBases (organization_name, base_location, base_region)
VALUES (“Team Rocket”, “Mahogany Town”, “Kanto”)
INSERT INTO VillainBases (organization_name, base_location, base_region)
VALUES (“Team Magma”, “Lilycove City”, “Hoenn”)
INSERT INTO VillainBases (organization_name, base_location, base_region)
VALUES (“Team Aqua”, “Lilycove City”, “Hoenn”)
INSERT INTO VillainBases (organization_name, base_location, base_region)
VALUES (“Team Galactic”, “Veilstone City”, “Sinnoh”)
INSERT INTO VillainBases (organization_name, base_location, base_region)
VALUES (“Team Plasma”, “Castelia City”, “Unova”)

organization_name	base_location	base_region
Team Rocket	Mahogany Town	Kanto
Team Magma	Lilycove City	Hoenn
Team Aqua	Lilycove City	Hoenn
Team Galactic	Veilstone City	Sinnoh
Team Plasma	Castelia City	Unova

PokemonRoster:

- name, id

INSERT INTO PokemonRoster (name, id)
VALUES (“Red”, 21)
INSERT INTO PokemonRoster (name, id)
VALUES (“Ethan”, 157)
INSERT INTO PokemonRoster (name, id)
VALUES (“May”, 471)
INSERT INTO PokemonRoster (name, id)
VALUES (“Dawn”, 393)
INSERT INTO PokemonRoster (name, id)
VALUES (“Hilbert”, 644)

name	id
Red	21
Ethan	157
May	471
Dawn	393
Hilbert	644

PokemonNameId:
- id, pokemon_name

INSERT INTO PokemonNameId (id, pokemon_name)
VALUES (1, “Bulbasaur”)
INSERT INTO PokemonNameId (id, pokemon_name)
VALUES (12, “Butterfree”)
INSERT INTO PokemonNameId (id, pokemon_name)
VALUES (20, “Raticate”)
INSERT INTO PokemonNameId (pokemon_name, category)
VALUES (134, “Vaporeon”)
INSERT INTO PokemonNameId (pokemon_name, category)
VALUES (292, “Shedinja”)

id	pokemon_name
1	Bulbasaur
12	Butterfree
20	Raticate
134	Vaporeon
292	Shedinja

PokemonCategory:
- pokemon_name, category

INSERT INTO PokemonCategory (pokemon_name, category)
VALUES (“Bulbasaur”, “Seed pokemon”)
INSERT INTO PokemonCategory (pokemon_name, category)
VALUES (“Butterfree”, “Butterfly pokemon”)
INSERT INTO PokemonCategory (pokemon_name, category)
VALUES (“Raticate”, “Mouse pokemon”)
INSERT INTO PokemonCategory (pokemon_name, category)
VALUES (“Vaporeon”, “Bubble Jet pokemon”)

INSERT INTO PokemonCategory (pokemon_name, category)
VALUES (“Shedinja”, “Shed pokemon”)

pokemon_name	category
Bulbasaur	Seed pokemon
Butterfree	Butterfly pokemon
Raticate	Mouse pokemon
Vaporeon	Bubble Jet pokemon
Shedinja	Shed pokemon

PokemonCatchRate:

- pokemon_name, catch_rate

INSERT INTO PokemonCatchRate (pokemon_name, catch_rate)
VALUES (“Bulbasaur”, 45)
INSERT INTO PokemonCatchRate (pokemon_name, catch_rate)
VALUES (“Butterfree”, 45)
INSERT INTO PokemonCatchRate (pokemon_name, catch_rate)
VALUES (“Raticate”, 127)
INSERT INTO PokemonCatchRate (pokemon_name, catch_rate)
VALUES (“Vaporeon”, 45)
INSERT INTO PokemonCatchRate (pokemon_name, catch_rate)
VALUES (“Shedinja”, 45)

pokemon_name	catch_rate
Bulbasaur	45
Butterfree	45
Raticate	127
Vaporeon	45
Shedinja	45

PokemonIntroducedIn:

- pokemon_name, region_name

INSERT INTO PokemonIntroducedIn (pokemon_name, region_name)
VALUES (“Bulbasaur”, “Kanto”)
INSERT INTO PokemonIntroducedIn (pokemon_name, region_name)
VALUES (“Butterfree”, “Kanto”)
INSERT INTO PokemonIntroducedIn (pokemon_name, region_name)
VALUES (“Raticate”, “Kanto”)

INSERT INTO PokemonIntroducedIn (pokemon_name, region_name)
VALUES (“Vaporeon”, “Kanto”)
INSERT INTO PokemonIntroducedIn (pokemon_name, region_name)
VALUES (“Shedinja”, “Kanto”)

pokemon_name	region_name
Bulbasaur	Kanto
Butterfree	Kanto
Raticate	Kanto
Vaporeon	Kanto
Shedinja	Kanto

PokemonEvolveConditions:

- evolution_item, evolution_level, from_id, id

INSERT INTO PokemonEvolveConditions (evolution_item, evolution_level, from_id, id)
VALUES (“None”, NULL, 1, 2)
INSERT INTO PokemonEvolveConditions (evolution_item, evolution_level, from_id)
VALUES (“None”, 10, 11, 12)
INSERT INTO PokemonEvolveConditions (evolution_item, evolution_level, from_id)
VALUES (“None”, 20, 19, 20)
INSERT INTO PokemonEvolveConditions (evolution_item, evolution_level, from_id)
VALUES (“Water stone”, NULL, 133, 134)
INSERT INTO PokemonEvolveConditions (evolution_item, evolution_level, from_id)
VALUES (“Pokeball”, 20, 290, 292)

evolution_item	evolution_level	from_id	id
None	16	1	2
None	10	11	12
None	20	19	20
Water stone	NULL	133	134
Pokeball	20	290	292

Type:

- type_name

INSERT INTO Type (type_name)
VALUES (“Normal”)
INSERT INTO Type (type_name)
VALUES (“Fighting”)

INSERT INTO Type (type_name)
VALUES (“Ghost”)
INSERT INTO Type (type_name)
VALUES (“Fairy”)
INSERT INTO Type (type_name)
VALUES (“Ice”)

type_name
Normal
Fighting
Ghost
Fairy
Ice

Effective Against:

- strong_name, weak_name

INSERT INTO EffectiveAgainst (strong_name, weak_name)
VALUES (“Fighting”, “Normal”)
INSERT INTO EffectiveAgainst (strong_name, weak_name)
VALUES (“Water”, “Fire”)
INSERT INTO EffectiveAgainst (strong_name, weak_name)
VALUES (“Fire”, “Ice”)
INSERT INTO EffectiveAgainst (strong_name, weak_name)
VALUES (“Fire”, “Grass”)
INSERT INTO EffectiveAgainst (strong_name, weak_name)
VALUES (“Electric”, “Flying”)

strong_name	weak_name
Fighting	Normal
Water	Fire
Fire	Ice
Fire	Grass
Electric	Flying

PokemonHasType:

- id, type_name

INSERT INTO PokemonHasType (id, type_name)
VALUES (10, “Bug”)

INSERT INTO PokemonHasType (id, type_name)
VALUES (4, “Fire”)
INSERT INTO PokemonHasType (id, type_name)
VALUES (147, “Dragon”)
INSERT INTO PokemonHasType (id, type_name)
VALUES (821, “Flying”)
INSERT INTO PokemonHasType (id, type_name)
VALUES (60, “Water”)

id	type_name
10	Bug
4	Fire
147	Dragon
821	Flying
60	Water

MoveHasType:

- move_name, type_name

INSERT INTO MoveHasType (move_name, type_name)
VALUES (“After You”, “Normal”)
INSERT INTO MoveHasType (move_name, type_name)
VALUES (“Acid Spray”, “Poison”)
INSERT INTO MoveHasType (move_name, type_name)
VALUES (“Sheer Cold”, “Ice”)
INSERT INTO MoveHasType (move_name, type_name)
VALUES (“Hyper Beam”, “Normal”)
INSERT INTO MoveHasType (move_name, type_name)
VALUES (“Sand Attack”, “Ground”)

move_name	type_name
After You	Normal
Acid Spray	Poison
Sheer Cold	Ice
Hyper Beam	Normal
Sand Attack	Ground

Move:

- name, power, pp, physical_contact, category, accuracy

INSERT INTO Move (name, power, pp, physical_contact, category, accuracy)
VALUES (“Absorb”, 20, 25, “No”, “Special”, 100)
INSERT INTO Move (name, power, pp, physical_contact, category, accuracy)
VALUES (“Amnesia”, 0, 20, “No”, “Status”, 0)
INSERT INTO Move (name, power, pp, physical_contact, category, accuracy)
VALUES (“Blizzard”, 110, 5, “No”, “Special”, 70)
INSERT INTO Move (name, power, pp, physical_contact, category, accuracy)
VALUES (“Seismic Toss”, 0, 20, “Yes”, “Physical”, 100)
INSERT INTO Move (name, power, pp, physical_contact, category, accuracy)
VALUES (“Splash”, 0, 40, “No”, “Status”, 0)

name	power	pp	physical_contact	category	accuracy
Absorb	20	25	No	Special	100
Amnesia	0	20	No	Status	0
Blizzard	110	5	No	Special	70
Seismic Toss	0	20	Yes	Physical	100
Splash	0	40	No	Status	0

Learns:

- move_name, id

INSERT INTO Learns (move_name, id)
VALUES (“Earthquake”, 27)
INSERT INTO Learns (move_name, id)
VALUES (“Aerial Ace”, 214)
INSERT INTO Learns (move_name, id)
VALUES (“Tail Whip”, 8)
INSERT INTO Learns (move_name, id)
VALUES (“Slash”, 257)
INSERT INTO Learns (move_name, id)
VALUES (“Body Slam”, 218)

move_name	id
Earthquake	27
Aerial Ace	214
Tail Whip	8
Slash	257
Body Slam	218

Game:

- game_name, platform, release_year

INSERT INTO Game (game_name, platform, release_year)
VALUES (“Pokemon Red”, “Gameboy”, 1996)
INSERT INTO Game (game_name, platform, release_year)
VALUES (“Pokemon Gold”, “Game Boy Colour”, 1999)
INSERT INTO Game (game_name, platform, release_year)
VALUES (“Pokemon SoulSilver”, “Nintendo DS”, 2009)
INSERT INTO Game (game_name, platform, release_year)
VALUES (“Pokemon Shining Pearl”, “Nintendo Switch”, 2021)
INSERT INTO Game (game_name, platform, release_year)
VALUES (“Pokemon Violet”, “Nintendo Switch”, 2022)
INSERT INTO Game (game_name, platform, release_year)
VALUES (“Omega Ruby”, NULL, 2014)

game_name	platform	release_year
Pokemon Red	Gameboy	1996
Pokemon Gold	Game Boy Colour	1999
Pokemon SoulSilver	Nintendo DS	2009
Pokemon Shining Pearl	Nintendo Switch	2021
Pokemon Violet	Nintendo Switch	2022
Omega Ruby	NULL	2014

Region:

- region_name, introduced_by_game, league

INSERT INTO Region (region_name, introduced_by_game, league)
VALUES (“Kanto”, “Pokemon Red”, “Indigo League”)
INSERT INTO Region (region_name, introduced_by_game, league)
VALUES (“Johto”, “Pokemon Gold”, “Johto League”)
INSERT INTO Region (region_name, introduced_by_game, league)
VALUES (“Hoenn”, “Pokemon Ruby”, “Hoenn League”)
INSERT INTO Region (region_name, introduced_by_game, league)
VALUES (“Sinnoh”, “Pokemon Diamond”, “Sinnoh League”)
INSERT INTO Region (region_name, introduced_by_game, league)
VALUES (“Unova”, “Pokemon White”, “Unova League”)

region_name	introduced_by_game	league
Kanto	Pokemon Red	Indigo League
Johto	Pokemon Gold	Johto League
Hoenn	Pokemon Ruby	Hoenn League

Sinnoh	Pokemon Diamond	Sinnoh League
Unova	Pokemon White	Unova League

InGameLocation:
- in_game_location_name, region_name

INSERT INTO InGameLocation (in_game_location_name, region_name)
VALUES (“Viridian City”, “Kanto”)
INSERT INTO InGameLocation (in_game_location_name, region_name)
VALUES (“Azalea Town”, “Johto”)
INSERT INTO InGameLocation (in_game_location_name, region_name)
VALUES (“Petalburg City”, “Hoenn”)
INSERT INTO InGameLocation (in_game_location_name, region_name)
VALUES (“Oreburgh City”, “Sinnoh”)
INSERT INTO InGameLocation (in_game_location_name, region_name)
VALUES (“Nacrene City”, “Unova”)

in_game_location_name	region_name
Viridian City	Kanto
Azalea Town	Johto
Petalburg City	Hoenn
Oreburgh City	Sinnoh
Nacrene City	Unova