CPSC 304 Project Cover Page

Milestone #: __2____

Date: Oct 25, 2021

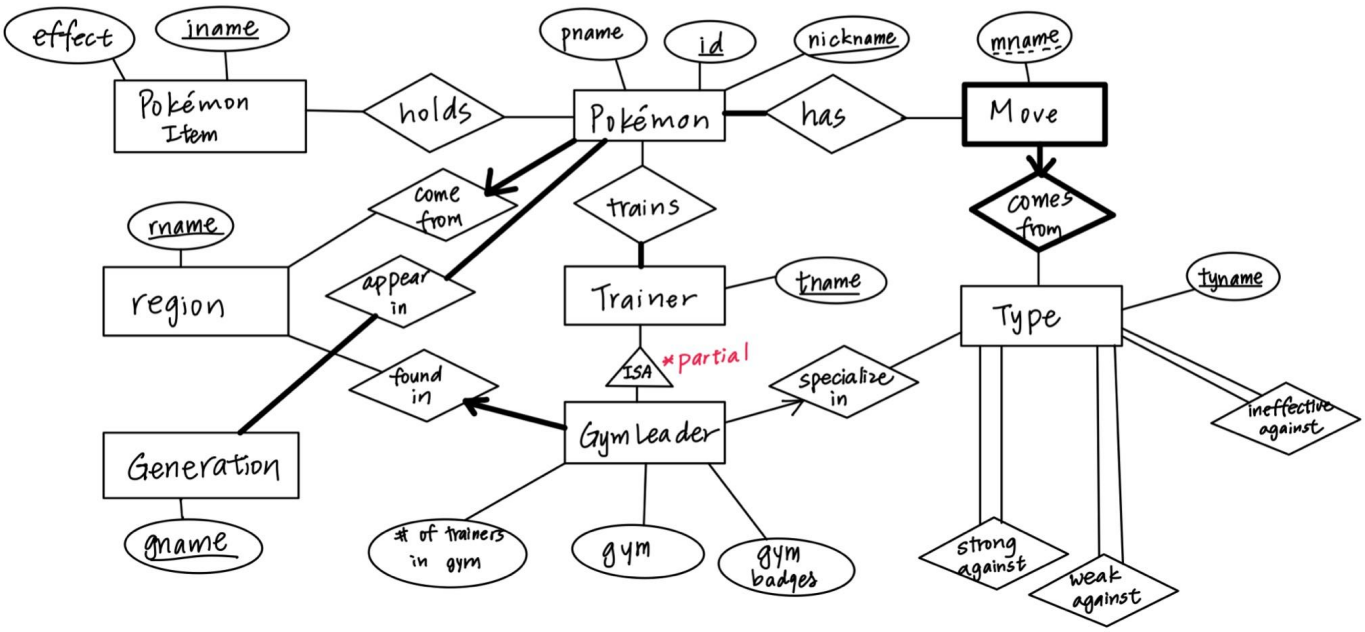
Group Number: _1____

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Tianne Lee	68834183	o6j3b	tianneubc@gmail.com
Terry Lin	65178329	j3j3b	terry95193@gmail.com
Andrew Lance Chan	75597444	n8q2b	andrewlancechan@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

2. ER Diagram



Changes to the ER diagram:

1. Our gym leader will inherit all the traits of a trainer, so gym would just be a regular attribute and glname is removed since it is redundant.
2. A total contribution constraint is added between generation and Pokémon so that every Pokémon has to appear in at least one generation.
3. Flipped the thick line between trainer and Pokémon so that every trainer has to have at least one Pokémon as opposed to every Pokémon having a trainer since there are wild Pokémon.
4. Move's attribute mname uses a dotted line instead of a solid line since Move is a weak entity.
5. Remove owner since Pokémon has a trainer.
6. Underline nickname so we can distinguish between pokémon of the same species.
7. Removed region attribute from trainer since they travel, unlike gym leaders.
8. Added two attributes for the FDs that do not have PK or CK: gym badge and gym trainer numbers (number of trainers you have to beat before getting to the gym leader)

3. Schema (<u>key is underlined</u>, **foreign keys are bolded**)

- Pokémon (*pname*, <u>id</u>, <u>nickname</u>, **region**)
    - CK: - {pname, nickname}
      {id, nickname}

- Trainer (tname)
- Gym Leader (**tname**, gym,gym badges, gym number of trainers, **tyname, rname**)
- Pokémon item (iname, effect)
- Generation (gname)
- MoveComesFrom (**mname**, **tyname**)
- Type (tyname)
- Trains (**tname, id, nickname**)
- Holds (**id, nickname, iname**)
- region (rname)
- AppearIn (**id, nickname, gname)**
- hasMoves (**id, nickname, mname**)
- ineffectiveAgainst (**tyname, tyname**)
- weakAgainst (**tyname, tyname**)
- St rongAgainst (**tyname, tyname**)

4. Functional Dependencies
(I obtained this from the arrows, and the entities' attributes)

Pokémon item -> Pokémon item effect

Pokémon ID, Pokémon nickname -> Pokémon name, Pokémon region

Trainer name -> Gym, Gym leader region, Gym leader type, Gym badge, Gym number of trainers

Gym -> Gym badge, Gym number of trainers

5. 3NF Normalization

To normalize a relation to 3NF, choose the tables such that we have tables for each FD and a table for the key. In class we talked about how the tables with two attributes are in BCNF, so I looked at those with more than three attributes to see if they need to be broken into smaller tables.

First take closure of all FDs:

{Pokémon item}+ = Pokémon item effect

{Pokémon ID, Pokémon nickname}+ = Pokémon name, Pokémon region

{Trainer name}+ = Gym, Gym leader region, Gym leader type, Gym badge, Gym number of trainers

{Gym}+ = Gym badge, Gym number of trainers

- Pokemon(id, pname, nickname, **rname**)

FDs that hold:
Pokémon ID, Pokémon nickname -> Pokémon name, Pokémon region

- Trainer(tname)
- Trains(**id, nickname, tname**)

FDs that hold:
None

- Pokemon_item(iname, effect)
- Holds(**id, nickname, iname**)

FDs that hold:
None

Gym_leader(tname, gym, **rname, tyname,** gym_badge, gym_number_of_trainers)
FDs that hold:
Trainer name -> Gym, Gym leader region, Gym leader type, Gym badge, Gym number of trainers
Gym -> Gym badge, Gym number of trainers
This can be decomposed. We create the minimal cover by splitting the FDs to have only one attribute on the right side, then we can remove attributes on the left-hand side (here we do nothing), then remove FD's that don't affect the closure:
Trainer name -> Gym
Trainer name -> Gym badge (remove this)
Trainer name -> Gym number of trainers (remove this)
Gym -> Gym badge
Gym -> Gym number of trainers
Now we create the following two relational tables

- Gym_leader(tname, **rname, tyname,** gym) (this contains the key for the original gym_leader)
- Gym(gym, gym_badge, gym_number_of_trainers)

FDs that hold:
Gym -> Gym badge, Gym number of trainers

- Region(rname)
- Generation(gname)
- Appear_in_generation(**id, nickname, gname**)

FDs that hold:
None

- Move(mname, **tyname**)
- HasMove**(id, nickname, mname, tyname**)

FDs that hold:
None

- Type(tyname)
- Strong_against(**tyname_self, tyname_other**)
- Weak_against(**tyname_self, tyname_other**)
- Ineffective_against(**tyname_self, tyname_other**)

6. SQL DDL for creating tables & 7. Populate table - See .sql file in link: pokedex_normalized.sql

For reference, the code is also copied below.

```
CREATE TABLE Pokemon(
   id INT,
   nickname CHAR(50),
   pname CHAR(30) NOT NULL,
   rname CHAR(30) NOT NULL,
   PRIMARY KEY(id, nickname),
   FOREIGN KEY (rname) REFERENCES Region ON DELETE CASCADE,
);

INSERT INTO 'Pokemon' ('id', 'pname', 'nickname', 'rname') VALUES
('1','Bulbasaur', 'Bulby', 'Kanto'),
('1', 'Bulbasaur', 'Blubber', 'Kanto'),
('25', 'Pikachu', 'Pika', 'Kanto'),
('25', 'Pikachu', 'Chu', 'Kanto'),
('38', 'Alolan Ninetales', 'Kurama', 'Alola'),
('141', 'Snorlax', 'Tiannes snorlax', 'Kanto'),
('79', 'Slowpoke', 'Procrastination', 'Kanto');

CREATE TABLE Region(
   rname CHAR(30) PRIMARY KEY,
);

INSERT INTO 'Region' ('rname') VALUES
('Kanto'),
('Alola'),
('Johto'),
```

```
('Hoenn'),
('Sinnoh');

CREATE TABLE Trainer(
   tname CHAR(30) PRIMARY KEY,
);

INSERT INTO 'Trainer' ('tname') VALUES
('Ash'),
('Misty'),
('Brock'),
('Iris'),
('Clemon'),
('Aaron'),
('Terry'),
('Andrew'),
('Tianne');

CREATE TABLE Trains(
   id INT,
   nickname CHAR(50),
   tname CHAR(30),
   PRIMARY KEY(id, nickname, tname),
   FOREIGN KEY(id, nickname) REFERENCES Pokemon ON DELETE CASCADE,
   FOREIGN KEY (tname) REFERENCES Trainer ON DELETE CASCADE,
);

INSERT INTO 'Trains' ('id', 'nickname', 'tname') VALUES
('1', 'Bulby', 'Ash'),
('1', 'Potato', 'Brock'),
('25', 'Pika', 'Ash'),
('25', 'Chu', 'Brock'),
('38', ' Kurama', 'Terry'),
('141', 'Tiannes snorlax', 'Tianne'),
```

```
('79', 'Procrastination', 'Andrew');

CREATE TABLE Pokemon_item(
   iname CHAR(30) PRIMARY KEY,
   effect CHAR(100) NOT NULL,
);

INSERT INTO 'Pokemon_item' ('iname', 'effect') VALUES
('Bug Gem', 'Increases the power of a Bug-type move only once.'),
('Charcoal', 'Increases the power of Fire-type moves.'),
('Focus Band', 'An item to be held by a Pokémon. The holder may endure a
potential KO attack, leaving it with just 1 HP.'),
('Grass Stone', 'Increases the power of Grass-type moves.'),
('Leftovers', 'An item to be held by a Pokémon. The holders HP is gradually
restored during battle.');

CREATE TABLE holds(
   id INT,
   nickname CHAR(50),
   iname CHAR(30),
   PRIMARY KEY(id, nickname, iname),
   FOREIGN KEY(id, nickname) REFERENCES Pokemon ON DELETE CASCADE,
   FOREIGN KEY (iname) REFERENCES Pokemon_Item ON DELETE CASCADE,
);

INSERT INTO 'holds' ('id', 'nickname', 'iname') VALUES
('1', 'Bulby', 'Bug Gem'),
('1', 'Potato', 'Charcoal'),
('25', 'Pika', 'Focus Band'),
('25', 'Chu', 'Grass Stone'),
('38', ' Kurama', 'Leftovers'),
('141', 'Tiannes snorlax', 'Leftovers');

CREATE TABLE Gym_leader(
```

```
   tname CHAR(30) PRIMARY KEY,
   gym CHAR(20) NOT NULL,
   rname CHAR(30) NOT NULL,
   tyname CHAR(30),
   FOREIGN KEY (tyname) REFERENCES Type ON DELETE CASCADE,
   FOREIGN KEY (rname) REFERENCES Region ON DELETE CASCADE,
);

INSERT INTO 'Gym_leader' ('tname', 'gym', 'rname', 'tyname') VALUES
('Misty', 'Cerulean', 'Kanto', 'Water'),
('Roxanne', 'Rustboro', 'Hoenn', 'Rock'),
('Cilan', 'Striaton', 'Unova', 'Grass'),
('Chili', 'Striaton', 'Unova', 'Fire'),
('Blue', 'Viridian', 'Kanto', NULL);

CREATE TABLE Gym(
   gym CHAR(30) NOT NULL,
   gym_badge CHAR(30),
   gym_number_of_trainers INT,
);

INSERT INTO 'Gym' ('gym', 'gym_badge', 'gym_number_of_trainers') VALUES
('Cerulean', 'Cascade', '10'),
('Rustboro', 'Stone', '5'),
('Striaton', 'Unova', '9'),
('Striaton', 'Unova', '2'),
('Viridian', 'Kanto', '0');

CREATE TABLE Generation(
   gname CHAR(50) PRIMARY KEY,
);

INSERT INTO 'Generation' ('gname') VALUES
('First generation'),
```

```
('Second generation'),
('Third generation'),
('Fourth generation'),
('Seventh generation');

CREATE TABLE Appear_in_Generation(
   id INT,
   nickname CHAR(50),
   gname CHAR(30),
   PRIMARY KEY(id, nickname, gname),
   FOREIGN KEY(id, nickname) REFERENCES Pokemon ON DELETE CASCADE,
   FOREIGN KEY (gname) REFERENCES Generation ON DELETE CASCADE,
);

INSERT INTO 'Appear_in_Generation' ('id', 'nickname', 'gname') VALUES
('1', 'Blubber', 'First generation'),
('201', 'Wazoo', 'Second generation'),
('281', 'Kirlazy', 'Third generation'),
('474', 'Porygon-Z', 'Fourth generation'),
('38', 'Kurama', 'Seventh generation');

CREATE TABLE Move(
   mname CHAR(30),
   tyname CHAR(30),
   PRIMARY KEY(mname,tyname),
   FOREIGN KEY (tyname) REFERENCES Type ON DELETE CASCADE,
);

INSERT INTO 'Move' ('mname', 'tyname') VALUES
('Dragon Meteor', 'Dragon'),
('Hyper Beam', 'Normal'),
('Hydro Pump', 'Water'),
('Megahorn', 'Bug'),
('Hurricane', 'Flying'),
```

```
('Twinkle Tackle', 'Fairy');

CREATE TABLE HasMove(
   id INT,
   nickname CHAR(50),
   mname CHAR(30),
   tyname CHAR(30),
   PRIMARY KEY(id, nickname, mname, tyname),
   FOREIGN KEY(id, nickname) REFERENCES Pokemon ON DELETE CASCADE,
   FOREIGN KEY(mname, tyname) REFERENCES Move_comes_from ON DELETE CASCADE,
);

INSERT INTO 'HasMove' ('id', 'nickname', 'mname', 'tyname') VALUES
('25', 'Pika', 'Charge Beam', 'Electric'),
('25', 'Chu', 'Spark', 'Electric'),
('201', 'Wazoo', 'Hidden Power', 'Normal'),
('281', 'Kirlazy', 'Charm', 'Fairy'),
('474', 'Porygone', 'Endure', 'Normal');

CREATE TABLE Type(
   tyname CHAR(30) PRIMARY KEY,
);

INSERT INTO 'Type' ('tyname') VALUES
('Normal'),
('Water'),
('Fire'),
('Electric'),
('Grass');

CREATE TABLE Strong_against(
   tyname_self CHAR(30),
   tyname_other CHAR(30),
   PRIMARY KEY (tyname_self, tyname_other),
```

```
   FOREIGN KEY (tyname_self) REFERENCES Type(tyname) ON DELETE SET NULL,
   FOREIGN KEY (tyname_other) REFERENCES Type(tyname) ON DELETE SET NULL,
);

INSERT INTO 'Strong_against' ('tyname_self', 'tyname_other') VALUES
('Electric', 'Water'),
('Electric', 'Flying'),
('Fire', 'Grass'),
('Fire', 'Bug'),
('Grass', 'Rock');

CREATE TABLE Weak_against(
   tyname_self CHAR(30),
   tyname_other CHAR(30),
   PRIMARY KEY (tyname_self, tyname_other),
   FOREIGN KEY (tyname_self) REFERENCES Type(tyname) ON DELETE SET NULL,
   FOREIGN KEY (tyname_other) REFERENCES Type(tyname) ON DELETE SET NULL,
);

INSERT INTO 'Weak_against' ('tyname_self', 'tyname_other') VALUES
('Electric', 'Electric'),
('Electric', 'Dragon'),
('Fire', 'Water'),
('Fire', 'Rock'),
('Grass', 'Steel');

CREATE TABLE Ineffective_against(
   tyname_self CHAR(30),
   tyname_other CHAR(30),
   PRIMARY KEY (tyname_self, tyname_other),
   FOREIGN KEY (tyname_self) REFERENCES Type(tyname) ON DELETE SET NULL,
   FOREIGN KEY (tyname_other) REFERENCES Type(tyname) ON DELETE SET NULL,
);
```

```
INSERT INTO 'Ineffective_against' ('tyname_self', 'tyname_other') VALUES
('Normal', 'Ghost'),
('Electric', 'Ground'),
('Ground', 'Flying'),
('Psychic', 'Dragon'),
('Dragon', 'Fairy');
```

8. Example queries:
We met a rival's pokémon Bulby the Bulbasaur. We can insert him into the Pokémon table. (Bulbasaur has pokédex index 1)

Gym leader Misty is retiring. Delete Misty from Gym_leader table.