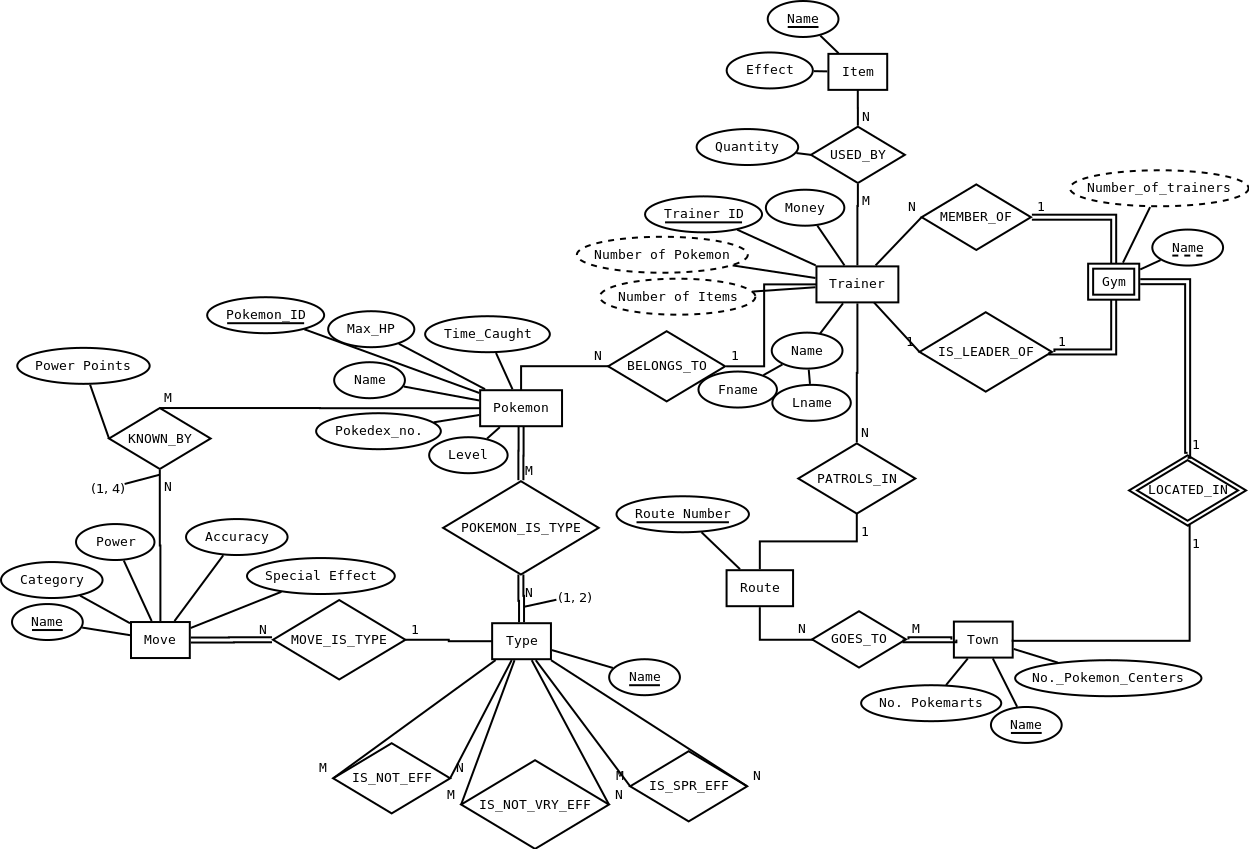
## ***COSC344 Assignment 2***

Team **1**

Leader **Sam Fleury**

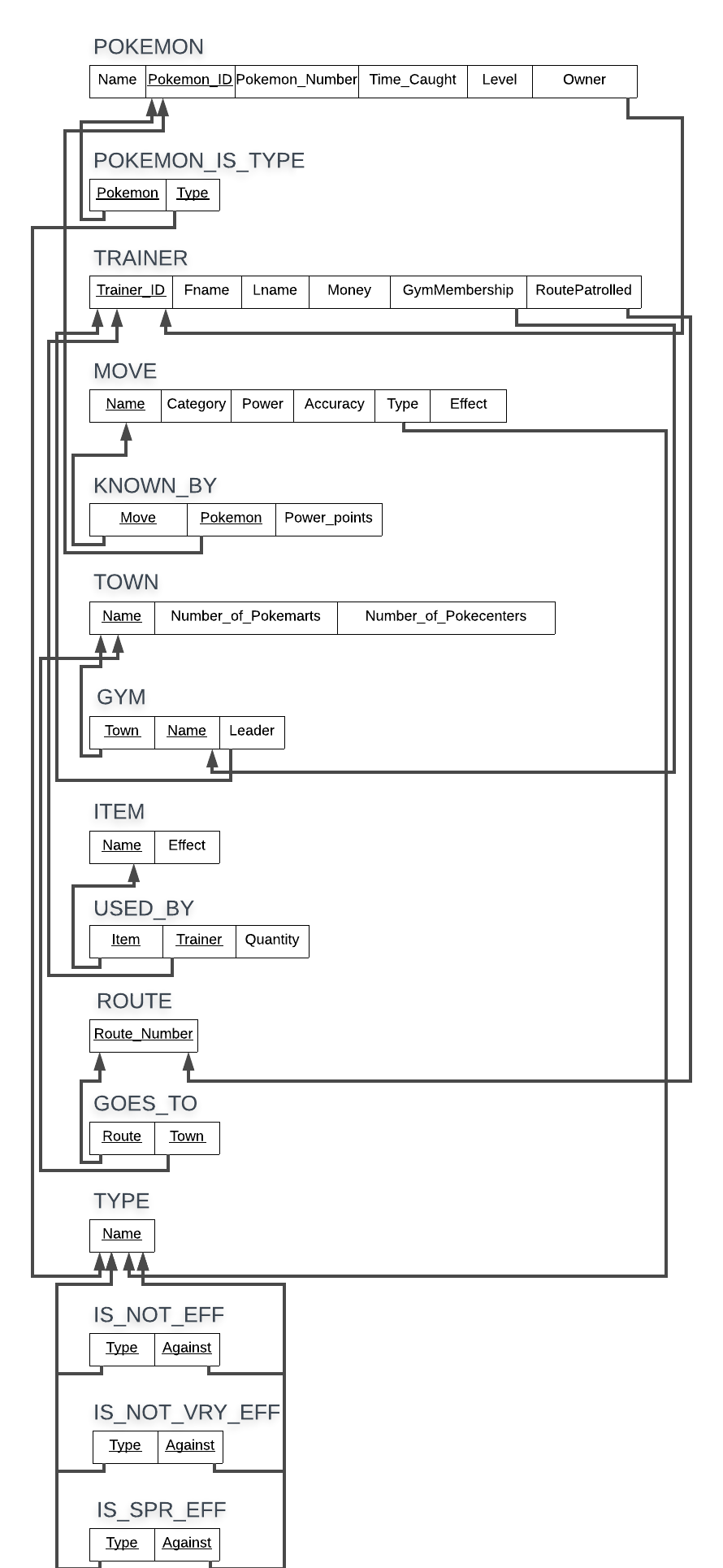
Members: **Sam Fleury**, **Tomofumi Kimura**, **Daniel Davidson, Brock Fairweather**

**1. Revised ER Diagram**



* Added (Min, Max) notation to relationships with limits.
* Move is now total participation in MOVE\_IS\_TYPE
* Time Caught is no longer a composite value

**2. Relational Schema**

****

**3. Normalization**

**Trainer:**

Step 1 - 1NF: TRAINER is already 1NF because all their attributes are atomic.

Step 2 - 2NF: TRAINER is already in 2NF because they are in 1NF and all their non-prime attributes are fully functionally dependent on the primary key.

Step 3 - 3NF: TRAINER is already in 3NF format because they are in the 2NF and do not have Transitive Dependency.

Step 4 - BCNF: TRAINER is already in BCNF format because it is in 3NF and Trainer ID is a super key.

**Gym**:

Step 1 - 1NF: GYM is already 1NF because its attribute are atomic.

Step 2 - 2NF: GYM is already in 2NF because they are in 1NF and all their non-prime attributes are fully functionally dependent on the primary key.

Step 3 - 3NF: GYM is already in 3NF format because they are in the 2NF and do not have Transitive Dependency.

Step 4 - BCNF: GYM is already in BCNF because they are in 3NF and the only dependencies that exist are dependent on super keys.

**Town:**

Step 1 - 1NF: TOWN is already in 1NF because all its values are atomic.

Step 2 - 2NF: TOWN is already 2NF because every attribute is fully functionally dependent on primary key Town Name there are no partial functional dependencies.

Step 3 - 3NF: TOWN is already 3NF because there are no transitive dependencies.

Step 4 - BCNF: TOWN is already BCNF because every other attribute depends on Town Name, which is a super key.

**Type:**

(Note: When it came time to implement this, we discovered that “TYPE” is a reserved word in Oracle, so we have implemented this as “PTYPE”.)

Step 1 - 1NF: Type is already 1NF because it has only one attribute, which is atomic.

Step 2 - 2NF: Type has only one attribute, its primary key, and therefore nothing that doesn't fully depend on it.

Step 3 - 3NF: Again, having only one attribute means that Type can not possibly have any transitive dependencies.

Step 4 - BCNF: Type is 3NF and its only key is its superkey, therefore it is BCNF.

**Pokemon:**

(Note: Max-HP is a derived value of Level and Pokedex\_no, so has been left off our relational schema. Also, Time\_caught has been collapsed down into just time caught, as the DATE type includes both date and time as an atomic value.)

Step 1 - 1NF: Pokemon is already in 1NF because all its values are atomic.

Step 2 - 2NF: Pokemon is already 2NF because every attribute is fully functionally dependent on primary key Pokemon\_ID; there are no partial functional dependencies.

Step 3 - 3NF: Pokemon is already 3NF because there are no transitive dependencies.

Step 4 - BCNF: Pokemon is already BCNF because every other attribute depends on Pokemon\_ID, which is a candidate key.

**Route:**

Step 1 - 1NF: Route is already 1NF because it has only one attribute, which is atomic.

Step 2 - 2NF: Route has only one attribute, its primary key Route\_Number, and therefore nothing that doesn't fully depend on it.

Step 3 - 3NF: Again, having only one attribute means that there can't be any transitive dependencies within Route.

Step 4 - BCNF: Route is 3NF and its only key is its superkey, so it is BCNF.

**Item:**

Step 1 - 1NF: Multi-valued attribute 'Effect' removed from ITEM and made into a new relation: ITEM\_EFFECT.

ITEM (name)

ITEM\_EFFECT(effect, item\_name)

Step 2 - 2NF: Already in 2NF because it is in 1NF and there are no non-prime attributes.

Step 3 - 3NF: Already in 3NF format because it is in 2NF and does not have Transitive Dependencies.

Step 4 - BCNF: Already in BCNF because it is in 3NF and there are no functional dependencies.

**Move:**

Step 1 - 1NF: Multi-valued attribute 'Special Effect' removed from MOVE and made into a new relation: MOVE\_EFFECT.

MOVE(name, category, power, accuracy, type)

MOVE\_EFFECT(effect, move\_name)

Step 2 - 2NF: Already in 2NF because it is in 1NF and all the non-prime attributes are fully functionally dependent on the primary key.

Step 3 - 3NF: Already in 3NF format because it is in 2NF and does not have Transitive Dependencies.

Step 4 - BCNF: Already in BCNF because it is in 3NF and there are no functional dependencies.

**Known\_By:**

Step 1 - 1NF: Already in 1NF because all attributes are atomic.

Step 2 - 2NF: Already in 2NF because it is in 1NF and all the non-prime attributes are fully functionally dependent on the primary key.

Step 3 - 3NF: Already in 3NF format because it is in 2NF and does not have Transitive Dependencies.

Step 4 - BCNF: already in BCNF because they are in 3NF and the only dependencies that exist are dependent on super keys.

**Used\_By:**

Step 1 - 1NF: Already in 1NF because all attributes are atomic.

Step 2 - 2NF: Already in 2NF because it is in 1NF and all the non-prime attributes are fully functionally dependent on the primary key.

Step 3 - 3NF: Already in 3NF format because it is in 2NF and does not have Transitive Dependencies.

Step 4 - BCNF: already in BCNF because they are in 3NF and the only dependencies that exist are dependent on super keys.

**Item\_Effect:**

Step 1 - 1NF: Already in 1NF because all attributes are atomic.

Step 2 - 2NF: Already in 2NF because it is in 1NF and there are no non-prime attributes.

Step 3 - 3NF: Already in 3NF because it is in 2NF and does not have Transitive Dependencies.

Step 4 - BCNF: Already in BCNF because it is in 3NF and there are no functional dependencies.

**Move\_Effect:**

Step 1 - 1NF: Already in 1NF because all attributes are atomic.

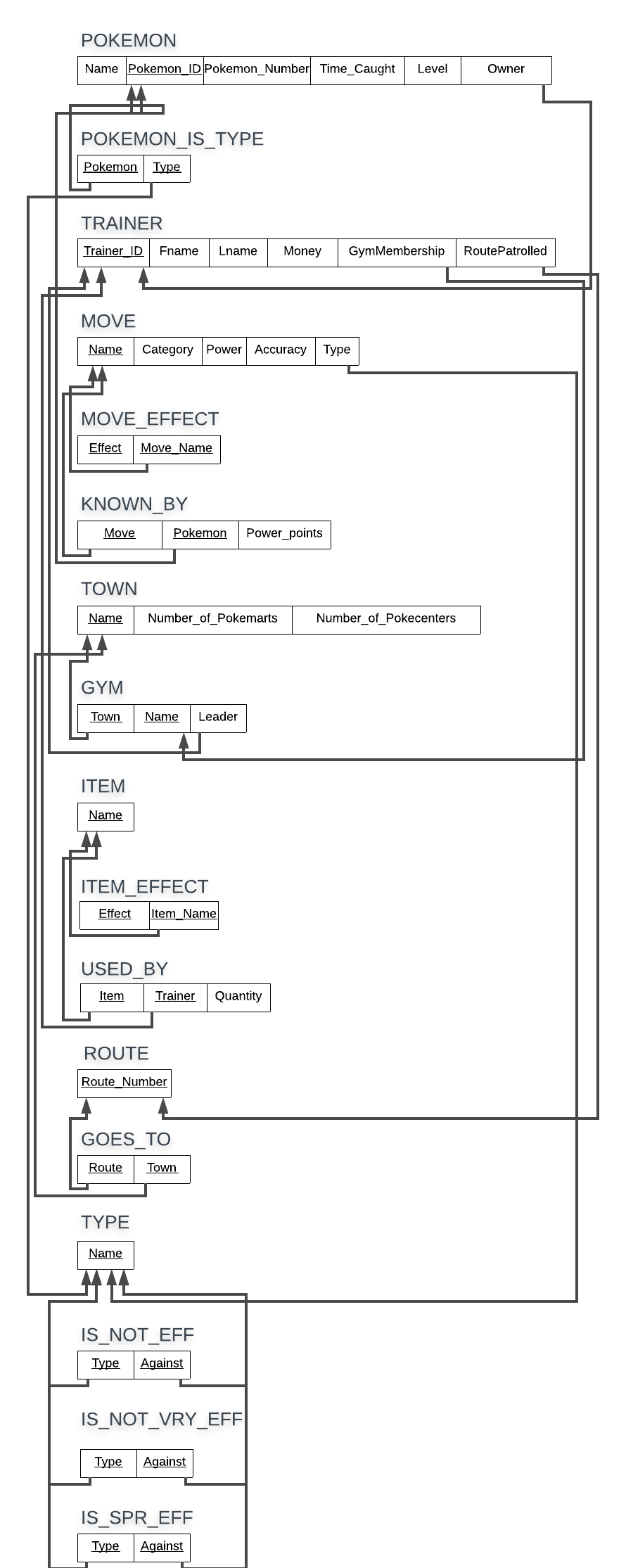
Step 2 - 2NF: Already in 2NF because it is in 1NF and there are no non-prime attributes.

Step 3 - 3NF: Already in 3NF because it is in 2NF and does not have Transitive Dependencies.

Step 4 - BCNF: Already in BCNF because it is in 3NF and there are no functional dependencies.

Pokemon\_Is\_Type, Goes\_To, Is\_Not\_Eff, Is\_Not\_Vry\_Eff, and Is\_Spr\_Eff each have no attributes which aren't foreign keys, so they do not need normalised.

**Relational Schema Post-Normalisation:**

****

**4. Implement Database (Load.sql)**

DROP TABLE Route CASCADE CONSTRAINTS;

DROP TABLE Town CASCADE CONSTRAINTS;

DROP TABLE Trainer CASCADE CONSTRAINTS;

DROP TABLE GOES\_TO CASCADE CONSTRAINTS;

DROP TABLE PType CASCADE CONSTRAINTS;

DROP TABLE Gym CASCADE CONSTRAINTS;

DROP TABLE Move CASCADE CONSTRAINTS;

DROP TABLE Move\_Effect CASCADE CONSTRAINTS;

DROP TABLE Pokemon CASCADE CONSTRAINTS;

DROP TABLE Known\_By CASCADE CONSTRAINTS;

DROP TABLE Item CASCADE CONSTRAINTS;

DROP TABLE Item\_Effect CASCADE CONSTRAINTS;

DROP TABLE Used\_By CASCADE CONSTRAINTS;

DROP TABLE IS\_NOT\_EFF CASCADE CONSTRAINTS;

DROP TABLE IS\_NOT\_VRY\_EFF CASCADE CONSTRAINTS;

DROP TABLE IS\_SPR\_EFF CASCADE CONSTRAINTS;

DROP TABLE Pokemon\_is\_type CASCADE CONSTRAINTS;

CREATE TABLE Route(

route\_number INT NOT NULL,

PRIMARY KEY(route\_number));

INSERT INTO Route VALUES(1);

INSERT INTO Route VALUES(2);

INSERT INTO Route VALUES(3);

INSERT INTO Route VALUES(4);

INSERT INTO Route VALUES(5);

INSERT INTO Route VALUES(6);

INSERT INTO Route VALUES(7);

INSERT INTO Route VALUES(8);

INSERT INTO Route VALUES(9);

INSERT INTO Route VALUES(10);

CREATE TABLE Town (

town\_name VARCHAR2(20) NOT NULL,

poke\_marts INT NOT NULL,

poke\_centres INT NOT NULL,

PRIMARY KEY(town\_name));

INSERT INTO Town VALUES('Pallet Town', 0, 0);

INSERT INTO Town VALUES('Viridian City', 1, 1);

INSERT INTO Town VALUES('Pewter City', 1, 1);

INSERT INTO Town VALUES('Cerulean City', 1, 1);

INSERT INTO Town VALUES('Vermilion City', 1, 1);

INSERT INTO Town VALUES('Lavender Town', 1, 1);

INSERT INTO Town VALUES('Celadon City', 5, 1);

INSERT INTO Town VALUES('Fuschia City', 1, 1);

INSERT INTO Town VALUES('Cinnabar Island', 1, 1);

INSERT INTO Town VALUES('Saffron City', 1, 1);

CREATE TABLE GOES\_TO(

route INT CONSTRAINT goestoroute REFERENCES Route(route\_number) DISABLE,

Town VARCHAR2(20) CONSTRAINT goestotown REFERENCES TOWN(town\_name) DISABLE);

INSERT INTO goes\_to VALUES (1, 'Pallet Town');

INSERT INTO goes\_to VALUES (1, 'Viridian City');

INSERT INTO goes\_to VALUES (2, 'Viridian City');

INSERT INTO goes\_to VALUES (2, 'Pewter City');

INSERT INTO goes\_to VALUES (3, 'Pewter City');

INSERT INTO goes\_to VALUES (3, 'Cerulean City');

INSERT INTO goes\_to VALUES (3, 'Vermilion City');

CREATE TABLE PType

(name VARCHAR(20) NOT NULL,

PRIMARY KEY(name));

INSERT INTO PType VALUES ('Normal');

INSERT INTO PType VALUES ('Fire');

INSERT INTO PType VALUES ('Fighting');

INSERT INTO PType VALUES ('Water');

INSERT INTO PType VALUES ('Flying');

INSERT INTO PType VALUES ('Grass');

INSERT INTO PType VALUES ('Poison');

INSERT INTO PType VALUES ('Electric');

INSERT INTO PType VALUES ('Ground');

INSERT INTO PType VALUES ('Psychic');

INSERT INTO PType VALUES ('Rock');

INSERT INTO PType VALUES ('Ice');

INSERT INTO PType VALUES ('Bug');

INSERT INTO PType VALUES ('Dragon');

INSERT INTO PType VALUES ('Ghost');

INSERT INTO PType VALUES ('Dark');

INSERT INTO PType VALUES ('Steel');

INSERT INTO PType VALUES ('Fairy');

CREATE TABLE TRAINER

( trainerID int PRIMARY KEY,

fname varchar2(20) NOT NULL,

lname varchar2(20),

money INT NOT NULL);

INSERT INTO trainer VALUES (001, 'Ash', 'Ketchum', 2);

INSERT INTO trainer VALUES (002, 'Gary', 'Oak', 168);

INSERT INTO trainer VALUES (003, 'Jesse', 'Rocket', 50);

INSERT INTO trainer VALUES (004, 'Giovanni', NULL, 1000000);

INSERT INTO trainer VALUES (011, 'Brock', NULL, 150);

INSERT INTO trainer VALUES (012, 'Misty', NULL, 200);

INSERT INTO trainer VALUES (013, 'Lieutenant', 'Surge', 250);

INSERT INTO trainer VALUES (014, 'Erika', NULL, 300);

INSERT INTO trainer VALUES (015, 'Koga', NULL, 350);

INSERT INTO trainer VALUES (016, 'Sabrina', NULL, 400);

INSERT INTO trainer VALUES (017, 'Blaine', NULL, 450);

CREATE TABLE Gym

(gym\_name VARCHAR(20) NOT NULL,

town VARCHAR(20) NOT NULL CONSTRAINT gymtown\_fk REFERENCES town(town\_name) DISABLE,

leader int NOT NULL CONSTRAINT gymleader\_fk REFERENCES trainer(trainerID) DISABLE,

PRIMARY KEY (gym\_name, town));

INSERT INTO Gym VALUES ('Pewter Gym', 'Pewter City', 11);

INSERT INTO Gym VALUES ('Cerulean Gym', 'Cerulean City', 12);

INSERT INTO Gym VALUES ('Vermilion Gym', 'Vermilion City', 13);

INSERT INTO Gym VALUES ('Celadon Gym', 'Celadon City', 14);

INSERT INTO Gym VALUES ('Fuschia Gym', 'Fuschia City', 15);

INSERT INTO Gym VALUES ('Saffron Gym', 'Saffron City', 16);

INSERT INTO Gym VALUES ('Cinnabar Gym', 'Cinnabar Island',17);

INSERT INTO Gym VALUES ('Viridian Gym', 'Viridian City', 004);

CREATE TABLE Move

(name VARCHAR2(20) NOT NULL,

category VARCHAR2(20) NOT NULL,

power INT,

accuracy INT,

type VARCHAR(20) NOT NULL CONSTRAINT movetype\_fk REFERENCES PType(name) DISABLE,

PRIMARY KEY(name),

CONSTRAINT category\_check CHECK (category IN ('physical', 'special', 'status')));

INSERT INTO Move VALUES ('Bind', 'physical', 15, 85, 'Normal');

INSERT INTO Move VALUES ('Ember', 'special', 40, 100, 'Fire');

INSERT INTO Move VALUES ('Counter', 'physical', NULL, 100, 'Fighting');

INSERT INTO Move VALUES ('Bubble Beam', 'special', 65, 100, 'Water');

INSERT INTO Move VALUES ('Drill Peck', 'physical', 80, 100, 'Flying');

INSERT INTO Move VALUES ('Absorb', 'special', 20, 100, 'Grass');

INSERT INTO Move VALUES ('Acid', 'special', 40, 100, 'Poison');

INSERT INTO Move VALUES ('Thunder Punch', 'physical', 75, 100, 'Electric');

INSERT INTO Move VALUES ('Bone Club', 'physical', 65, 85, 'Ground');

INSERT INTO Move VALUES ('Amnesia', 'status', NULL, NULL, 'Psychic');

INSERT INTO Move VALUES ('Rock Slide', 'physical', 75, 90, 'Rock');

INSERT INTO Move VALUES ('Ice Beam', 'special', 90, 100, 'Ice');

INSERT INTO Move VALUES ('Pin Missile', 'physical', 25, 95, 'Bug');

INSERT INTO Move VALUES ('Dragon Rage', 'special', NULL, 100, 'Dragon');

INSERT INTO Move VALUES ('Lick', 'physical', 30, 100, 'Ghost');

INSERT INTO Move VALUES ('Bite', 'physical', 60, 100, 'Dark');

INSERT INTO Move VALUES ('Clamp', 'physical', 35, 85, 'Water');

CREATE TABLE Move\_Effect

(effect VARCHAR2(100) NOT NULL,

move\_name VARCHAR(20) NOT NULL CONSTRAINT moveeffectmove\_fk REFERENCES Move(name) DISABLE,

PRIMARY KEY(effect, move\_name));

INSERT INTO Move\_Effect VALUES('Multi turn attack', 'Bind');

INSERT INTO Move\_Effect VALUES('Chance of burning the target', 'Ember');

INSERT INTO Move\_Effect VALUES('Decreased priority', 'Counter');

INSERT INTO Move\_Effect VALUES('Doubled damage if user has been attacked', 'Counter');

INSERT INTO Move\_Effect VALUES('Chance of lowering targets speed', 'Bubble Beam');

INSERT INTO Move\_Effect VALUES('Damage done is restored to the user', 'Absorb');

INSERT INTO Move\_Effect VALUES('Chance of lowering the targets defence', 'Acid');

INSERT INTO Move\_Effect VALUES('Chance of paralyzing the target', 'Thunder Punch');

INSERT INTO Move\_Effect VALUES('Chance of causing target to flinch', 'Bone Club');

INSERT INTO Move\_Effect VALUES('Increases users Special by two stages', 'Amnesia');

INSERT INTO Move\_Effect VALUES('Chance of freezing the target', 'Ice Beam');

INSERT INTO Move\_Effect VALUES('Hits the target 2-5 times', 'Pin Missile');

INSERT INTO Move\_Effect VALUES('Always inflicts exactly 40 damage', 'Dragon Rage');

INSERT INTO Move\_Effect VALUES('Chance of paralyzing the target', 'Lick');

INSERT INTO Move\_Effect VALUES('Multi turn attack', 'Clamp');

INSERT INTO Move\_Effect VALUES('Traps the target', 'Clamp');

CREATE TABLE Item

(name VARCHAR(20) NOT NULL,

PRIMARY KEY(name));

INSERT INTO Item VALUES ('Antidote');

INSERT INTO Item VALUES ('Elixir');

INSERT INTO Item VALUES ('Full Heal');

INSERT INTO Item VALUES ('Poke Ball');

INSERT INTO Item VALUES ('Revive');

INSERT INTO Item VALUES ('Potion');

CREATE TABLE Item\_Effect

(item\_name VARCHAR2(20) NOT NULL CONSTRAINT itemeffitem REFERENCES Item(name) DISABLE,

effect VARCHAR2(100) NOT NULL,

PRIMARY KEY(effect, item\_name));

INSERT INTO Item\_Effect VALUES ('Antidote', 'Cures Poison Status');

INSERT INTO Item\_Effect VALUES ('Elixir', 'Restores 10PP');

INSERT INTO Item\_Effect VALUES ('Full Heal', 'Fully restore Pokemon HP');

INSERT INTO Item\_Effect VALUES ('Poke Ball', 'Used to catch Pokemon');

INSERT INTO Item\_Effect VALUES ('Revive', 'Recover fainted Pokemon');

INSERT INTO Item\_Effect VALUES ('Potion', 'Restore portion of HP');

CREATE TABLE Used\_By

(item\_name VARCHAR(20) NOT NULL CONSTRAINT usedbyitem REFERENCES Item(name) DISABLE,

trainer\_id INT NOT NULL CONSTRAINT usedbytrainer REFERENCES Trainer(trainerid) DISABLE,

quantity INT NOT NULL,

PRIMARY KEY(item\_name, trainer\_id));

INSERT INTO used\_by VALUES('Antidote', 003, 2);

INSERT INTO used\_by VALUES('Poke Ball', 003, 3);

INSERT INTO used\_by VALUES('Revive', 003, 1);

INSERT INTO used\_by VALUES('Antidote', 001, 1);

INSERT INTO used\_by VALUES('Potion', 001, 4);

INSERT INTO used\_by VALUES('Revive', 001, 2);

INSERT INTO used\_by VALUES('Elixir', 002, 2);

INSERT INTO used\_by VALUES('Antidote', 004, 10);

CREATE TABLE Pokemon

(name VARCHAR(12) NOT NULL,

pokemon\_ID INT PRIMARY KEY,

Pokemon\_no INT NOT NULL,

time\_caught date NOT NULL,

plevel INT NOT NULL,

Owner int NOT NULL CONSTRAINT pokemonowner REFERENCES TRAINER(trainerID) DISABLE);

INSERT INTO Pokemon VALUES ('Bulbasaur', 001, 097176, TO\_DATE('20180101 09:41:43', 'YYYYMMDD HH24:MI:SS'), 5, 001);

INSERT INTO Pokemon VALUES ('Charmander', 004, 002226, TO\_DATE('20180102 16:41:43', 'YYYYMMDD HH24:MI:SS'), 10, 001);

INSERT INTO Pokemon VALUES ('Squirtle', 007, 000337, TO\_DATE('20180105 18:35:00', 'YYYYMMDD HH24:MI:SS'), 23, 001);

INSERT INTO Pokemon VALUES ('Caterpie', 010, 038227, TO\_DATE('20180107 01:00:43', 'YYYYMMDD HH24:MI:SS'), 75, 001);

INSERT INTO Pokemon VALUES ('Pidgey', 016, 066729, TO\_DATE('20180204 10:41:59', 'YYYYMMDD HH24:MI:SS'), 100, 002);

INSERT INTO Pokemon VALUES ('Rattata', 019, 087433, TO\_DATE('20180505 11:07:43', 'YYYYMMDD HH24:MI:SS'), 56, 003);

INSERT INTO Pokemon VALUES ('Ekans', 023, 028189, TO\_DATE('20180901 12:01:40', 'YYYYMMDD HH24:MI:SS'), 78, 003);

INSERT INTO Pokemon VALUES ('Pikachu', 025, 034440, TO\_DATE('20181101 19:19:19', 'YYYYMMDD HH24:MI:SS'), 79, 001);

INSERT INTO Pokemon VALUES ('Sandshrew', 027, 005559, TO\_DATE('20181201 08:00:10', 'YYYYMMDD HH24:MI:SS'), 45, 011);

INSERT INTO Pokemon VALUES ('Clefairy', 035, 028548, TO\_DATE('20180516 23:17:12', 'YYYYMMDD HH24:MI:SS'), 34, 016);

INSERT INTO Pokemon VALUES ('Oddish', 043, 058703, TO\_DATE('20180704 13:34:43', 'YYYYMMDD HH24:MI:SS'), 23, 015);

INSERT INTO Pokemon VALUES ('Mankey', 056, 098976, TO\_DATE('20180819 09:12:21', 'YYYYMMDD HH24:MI:SS'), 46, 013);

INSERT INTO Pokemon VALUES ('Abra', 063, 090279, TO\_DATE('20181018 10:10:10', 'YYYYMMDD HH24:MI:SS'), 23, 014);

INSERT INTO Pokemon VALUES ('Magnemite', 081, 075832, TO\_DATE('20180614 23:56:56', 'YYYYMMDD HH24:MI:SS'), 89, 013);

INSERT INTO Pokemon VALUES ('Haunter', 093, 009237, TO\_DATE('20180813 18:41:43', 'YYYYMMDD HH24:MI:SS'), 21, 017);

INSERT INTO Pokemon VALUES ('Onix', 095, 094582, TO\_DATE('20180426 18:41:50', 'YYYYMMDD HH24:MI:SS'), 54, 011);

INSERT INTO Pokemon VALUES ('Lapras', 131, 050809, TO\_DATE('20180910 19:19:59', 'YYYYMMDD HH24:MI:SS'), 33, 012);

INSERT INTO Pokemon VALUES ('Dragonite', 149, 091346, TO\_DATE('20180209 15:50:39', 'YYYYMMDD HH24:MI:SS'), 67, 017);

CREATE TABLE POKEMON\_IS\_TYPE

(Pokemon int NOT NULL CONSTRAINT istypepokemon REFERENCES POKEMON(pokemon\_ID) DISABLE ,

Type VARCHAR2(12) NOT NULL CONSTRAINT istypetype REFERENCES PTYPE(name) DISABLE);

INSERT INTO Pokemon\_Is\_Type VALUES (001, 'Grass');

INSERT INTO Pokemon\_Is\_Type VALUES (004, 'Fire');

INSERT INTO Pokemon\_Is\_Type VALUES (007, 'Water');

INSERT INTO Pokemon\_Is\_Type VALUES (010, 'Bug');

INSERT INTO Pokemon\_Is\_Type VALUES (016, 'Flying');

INSERT INTO Pokemon\_Is\_Type VALUES (019, 'Normal');

INSERT INTO Pokemon\_Is\_Type VALUES (023, 'Poison');

INSERT INTO Pokemon\_Is\_Type VALUES (025, 'Electric');

INSERT INTO Pokemon\_Is\_Type VALUES (027, 'Ground');

INSERT INTO Pokemon\_Is\_Type VALUES (035, 'Fairy');

INSERT INTO Pokemon\_Is\_Type VALUES (043, 'Grass');

INSERT INTO Pokemon\_Is\_Type VALUES (056, 'Fighting');

INSERT INTO Pokemon\_Is\_Type VALUES (063, 'Psychic');

INSERT INTO Pokemon\_Is\_Type VALUES (081, 'Steel');

INSERT INTO Pokemon\_Is\_Type VALUES (093, 'Ghost');

INSERT INTO Pokemon\_Is\_Type VALUES (095, 'Rock');

INSERT INTO Pokemon\_Is\_Type VALUES (131, 'Ice');

INSERT INTO Pokemon\_Is\_Type VALUES (149, 'Dragon');

CREATE TABLE Known\_By

(move\_name VARCHAR(20) NOT NULL CONSTRAINT knownbymovename REFERENCES Move(name) DISABLE,

pokemon\_ID INT NOT NULL CONSTRAINT knownbypokemonid REFERENCES Pokemon(pokemon\_ID) DISABLE,

power\_points INT NOT NULL,

PRIMARY KEY(move\_name, pokemon\_ID));

INSERT INTO Known\_By VALUES ('Ember', 004, 25);

INSERT INTO Known\_By VALUES ('Counter', 004, 15);

INSERT INTO Known\_By VALUES ('Lick', 004, 35);

INSERT INTO Known\_By VALUES ('Absorb', 001, 35);

INSERT INTO Known\_By VALUES ('Bind', 001, 20);

INSERT INTO Known\_By VALUES ('Acid', 001, 35);

INSERT INTO Known\_By VALUES ('Bubble Beam', 007, 20);

INSERT INTO Known\_By VALUES ('Ice Beam', 007, 15);

INSERT INTO Known\_By VALUES ('Lick', 007, 35);

INSERT INTO Known\_By VALUES ('Pin Missile', 010, 20);

INSERT INTO Known\_By VALUES ('Bite', 010, 20);

INSERT INTO Known\_By VALUES ('Drill Peck', 016, 15);

INSERT INTO Known\_By VALUES ('Counter', 016, 20);

INSERT INTO Known\_By VALUES ('Counter', 019, 25);

INSERT INTO Known\_By VALUES ('Bite', 019, 20);

INSERT INTO Known\_By VALUES ('Lick', 019, 30);

INSERT INTO Known\_By VALUES ('Bind', 023, 20);

INSERT INTO Known\_By VALUES ('Bite', 023, 20);

INSERT INTO Known\_By VALUES ('Thunder Punch', 025, 15);

INSERT INTO Known\_By VALUES ('Bite', 025, 20);

INSERT INTO Known\_By VALUES ('Counter', 025, 15);

INSERT INTO Known\_By VALUES ('Bite', 027, 20);

INSERT INTO Known\_By VALUES ('Ember', 027, 25);

INSERT INTO Known\_By VALUES ('Bone Club', 027, 15);

INSERT INTO Known\_By VALUES ('Bite', 035, 20);

INSERT INTO Known\_By VALUES ('Amnesia', 035, 15);

INSERT INTO Known\_By VALUES ('Ice Beam', 035, 15);

INSERT INTO Known\_By VALUES ('Absorb', 043, 25);

INSERT INTO Known\_By VALUES ('Bind', 043, 15);

INSERT INTO Known\_By VALUES ('Counter', 056, 20);

INSERT INTO Known\_By VALUES ('Thunder Punch', 056, 15);

INSERT INTO Known\_By VALUES ('Amnesia', 063, 20);

INSERT INTO Known\_By VALUES ('Lick', 063, 35);

INSERT INTO Known\_By VALUES ('Counter', 081, 20);

INSERT INTO Known\_By VALUES ('Thunder Punch', 081, 15);

INSERT INTO Known\_By VALUES ('Lick', 093, 35);

INSERT INTO Known\_By VALUES ('Amnesia', 093, 20);

INSERT INTO Known\_By VALUES ('Rock Slide', 095, 15);

INSERT INTO Known\_By VALUES ('Bite', 095, 20);

INSERT INTO Known\_By VALUES ('Ice Beam', 131, 20);

INSERT INTO Known\_By VALUES ('Bubble Beam', 131, 15);

INSERT INTO Known\_By VALUES ('Dragon Rage', 149, 15);

INSERT INTO Known\_By VALUES ('Counter', 149, 20);

CREATE TABLE IS\_NOT\_EFF

(Type VARCHAR2(12) REFERENCES PTYPE(name) NOT NULL,

Against VARCHAR2(12) REFERENCES PTYPE(name) NOT NULL);

INSERT INTO Is\_Not\_Eff VALUES ('Normal', 'Ghost');

INSERT INTO Is\_Not\_Eff VALUES ('Electric', 'Ground');

INSERT INTO Is\_Not\_Eff VALUES ('Fighting', 'Ghost');

INSERT INTO Is\_Not\_Eff VALUES ('Poison', 'Steel');

INSERT INTO Is\_Not\_Eff VALUES ('Ground', 'Flying');

INSERT INTO Is\_Not\_Eff VALUES ('Psychic', 'Dark');

INSERT INTO Is\_Not\_Eff VALUES ('Ghost', 'Normal');

INSERT INTO Is\_Not\_Eff VALUES ('Dragon', 'Fairy');

CREATE TABLE IS\_NOT\_VRY\_EFF

(Type VARCHAR2(12) REFERENCES PTYPE(Name) NOT NULL,

Against VARCHAR2(12) REFERENCES PTYPE(Name) NOT NULL);

INSERT INTO Is\_Not\_Vry\_Eff VALUES ('Normal', 'Rock');

INSERT INTO Is\_Not\_Vry\_Eff VALUES ('Normal', 'Steel');

INSERT INTO Is\_Not\_Vry\_Eff VALUES ('Fire', 'Fire');

INSERT INTO Is\_Not\_Vry\_Eff VALUES ('Fire', 'Water');

INSERT INTO Is\_Not\_Vry\_Eff VALUES ('Fire', 'Rock');

INSERT INTO Is\_Not\_Vry\_Eff VALUES ('Fire', 'Dragon');

INSERT INTO Is\_Not\_Vry\_Eff VALUES ('Water', 'Fire');

INSERT INTO Is\_Not\_Vry\_Eff VALUES ('Water', 'Grass');

INSERT INTO Is\_Not\_Vry\_Eff VALUES ('Water', 'Dragon');

INSERT INTO Is\_Not\_Vry\_Eff VALUES ('Electric', 'Electric');

INSERT INTO Is\_Not\_Vry\_Eff VALUES ('Electric', 'Grass');

INSERT INTO Is\_Not\_Vry\_Eff VALUES ('Electric', 'Dragon');

INSERT INTO Is\_Not\_Vry\_Eff VALUES ('Grass', 'Fire');

INSERT INTO Is\_Not\_Vry\_Eff VALUES ('Grass', 'Grass');

INSERT INTO Is\_Not\_Vry\_Eff VALUES ('Grass', 'Poison');

INSERT INTO Is\_Not\_Vry\_Eff VALUES ('Grass', 'Flying');

INSERT INTO Is\_Not\_Vry\_Eff VALUES ('Grass', 'Bug');

INSERT INTO Is\_Not\_Vry\_Eff VALUES ('Grass', 'Dragon');

INSERT INTO Is\_Not\_Vry\_Eff VALUES ('Grass', 'Steel');

INSERT INTO Is\_Not\_Vry\_Eff VALUES ('Ice', 'Fire');

INSERT INTO Is\_Not\_Vry\_Eff VALUES ('Ice', 'Water');

INSERT INTO Is\_Not\_Vry\_Eff VALUES ('Ice', 'Ice');

INSERT INTO Is\_Not\_Vry\_Eff VALUES ('Ice', 'Steel');

INSERT INTO Is\_Not\_Vry\_Eff VALUES ('Fighting', 'Poison');

INSERT INTO Is\_Not\_Vry\_Eff VALUES ('Fighting', 'Flying');

INSERT INTO Is\_Not\_Vry\_Eff VALUES ('Fighting', 'Psychic');

INSERT INTO Is\_Not\_Vry\_Eff VALUES ('Fighting', 'Bug');

INSERT INTO Is\_Not\_Vry\_Eff VALUES ('Fighting', 'Fairy');

INSERT INTO Is\_Not\_Vry\_Eff VALUES ('Poison', 'Poison');

INSERT INTO Is\_Not\_Vry\_Eff VALUES ('Poison', 'Ground');

INSERT INTO Is\_Not\_Vry\_Eff VALUES ('Poison', 'Rock');

INSERT INTO Is\_Not\_Vry\_Eff VALUES ('Poison', 'Ghost');

INSERT INTO Is\_Not\_Vry\_Eff VALUES ('Ground', 'Grass');

INSERT INTO Is\_Not\_Vry\_Eff VALUES ('Ground', 'Bug');

INSERT INTO Is\_Not\_Vry\_Eff VALUES ('Flying', 'Electric');

INSERT INTO Is\_Not\_Vry\_Eff VALUES ('Flying', 'Rock');

INSERT INTO Is\_Not\_Vry\_Eff VALUES ('Flying', 'Steel');

INSERT INTO Is\_Not\_Vry\_Eff VALUES ('Psychic', 'Psychic');

INSERT INTO Is\_Not\_Vry\_Eff VALUES ('Psychic', 'Steel');

INSERT INTO Is\_Not\_Vry\_Eff VALUES ('Bug', 'Fire');

INSERT INTO Is\_Not\_Vry\_Eff VALUES ('Bug', 'Fighting');

INSERT INTO Is\_Not\_Vry\_Eff VALUES ('Bug', 'Poison');

INSERT INTO Is\_Not\_Vry\_Eff VALUES ('Bug', 'Flying');

INSERT INTO Is\_Not\_Vry\_Eff VALUES ('Bug', 'Ghost');

INSERT INTO Is\_Not\_Vry\_Eff VALUES ('Bug', 'Steel');

INSERT INTO Is\_Not\_Vry\_Eff VALUES ('Bug', 'Fairy');

INSERT INTO Is\_Not\_Vry\_Eff VALUES ('Rock', 'Fighting');

INSERT INTO Is\_Not\_Vry\_Eff VALUES ('Rock', 'Ground');

INSERT INTO Is\_Not\_Vry\_Eff VALUES ('Rock', 'Steel');

INSERT INTO Is\_Not\_Vry\_Eff VALUES ('Ghost', 'Dark');

INSERT INTO Is\_Not\_Vry\_Eff VALUES ('Dragon', 'Steel');

INSERT INTO Is\_Not\_Vry\_Eff VALUES ('Steel', 'Fire');

INSERT INTO Is\_Not\_Vry\_Eff VALUES ('Steel', 'Water');

INSERT INTO Is\_Not\_Vry\_Eff VALUES ('Steel', 'Electric');

INSERT INTO Is\_Not\_Vry\_Eff VALUES ('Steel', 'Steel');

INSERT INTO Is\_Not\_Vry\_Eff VALUES ('Fairy', 'Fire');

INSERT INTO Is\_Not\_Vry\_Eff VALUES ('Fairy', 'Fighting');

INSERT INTO Is\_Not\_Vry\_Eff VALUES ('Fairy', 'Steel');

CREATE TABLE IS\_SPR\_EFF

(Type VARCHAR2(12) REFERENCES PTYPE(Name) NOT NULL,

Against VARCHAR2(12) REFERENCES PTYPE(Name) NOT NULL);

INSERT INTO Is\_Spr\_Eff VALUES ('Fire', 'Grass');

INSERT INTO Is\_Spr\_Eff VALUES ('Fire', 'Ice');

INSERT INTO Is\_Spr\_Eff VALUES ('Fire', 'Bug');

INSERT INTO Is\_Spr\_Eff VALUES ('Fire', 'Steel');

INSERT INTO Is\_Spr\_Eff VALUES ('Water', 'Fire');

INSERT INTO Is\_Spr\_Eff VALUES ('Water', 'Ground');

INSERT INTO Is\_Spr\_Eff VALUES ('Water', 'Rock');

INSERT INTO Is\_Spr\_Eff VALUES ('Electric', 'Water');

INSERT INTO Is\_Spr\_Eff VALUES ('Electric', 'Flying');

INSERT INTO Is\_Spr\_Eff VALUES ('Grass', 'Water');

INSERT INTO Is\_Spr\_Eff VALUES ('Grass', 'Ground');

INSERT INTO Is\_Spr\_Eff VALUES ('Grass', 'Rock');

INSERT INTO Is\_Spr\_Eff VALUES ('Ice', 'Grass');

INSERT INTO Is\_Spr\_Eff VALUES ('Ice', 'Ground');

INSERT INTO Is\_Spr\_Eff VALUES ('Ice', 'Flying');

INSERT INTO Is\_Spr\_Eff VALUES ('Ice', 'Dragon');

INSERT INTO Is\_Spr\_Eff VALUES ('Fighting', 'Normal');

INSERT INTO Is\_Spr\_Eff VALUES ('Fighting', 'Ice');

INSERT INTO Is\_Spr\_Eff VALUES ('Fighting', 'Rock');

INSERT INTO Is\_Spr\_Eff VALUES ('Fighting', 'Dark');

INSERT INTO Is\_Spr\_Eff VALUES ('Fighting', 'Steel');

INSERT INTO Is\_Spr\_Eff VALUES ('Poison', 'Grass');

INSERT INTO Is\_Spr\_Eff VALUES ('Poison', 'Fairy');

INSERT INTO Is\_Spr\_Eff VALUES ('Ground', 'Fire');

INSERT INTO Is\_Spr\_Eff VALUES ('Ground', 'Electric');

INSERT INTO Is\_Spr\_Eff VALUES ('Ground', 'Poison');

INSERT INTO Is\_Spr\_Eff VALUES ('Ground', 'Rock');

INSERT INTO Is\_Spr\_Eff VALUES ('Ground', 'Steel');

INSERT INTO Is\_Spr\_Eff VALUES ('Flying', 'Grass');

INSERT INTO Is\_Spr\_Eff VALUES ('Flying', 'Fighting');

INSERT INTO Is\_Spr\_Eff VALUES ('Flying', 'Bug');

INSERT INTO Is\_Spr\_Eff VALUES ('Psychic', 'Fighting');

INSERT INTO Is\_Spr\_Eff VALUES ('Psychic', 'Poison');

INSERT INTO Is\_Spr\_Eff VALUES ('Bug', 'Grass');

INSERT INTO Is\_Spr\_Eff VALUES ('Bug', 'Psychic');

INSERT INTO Is\_Spr\_Eff VALUES ('Bug', 'Dark');

INSERT INTO Is\_Spr\_Eff VALUES ('Rock', 'Fire');

INSERT INTO Is\_Spr\_Eff VALUES ('Rock', 'Ice');

INSERT INTO Is\_Spr\_Eff VALUES ('Rock', 'Flying');

INSERT INTO Is\_Spr\_Eff VALUES ('Rock', 'Bug');

INSERT INTO Is\_Spr\_Eff VALUES ('Ghost', 'Psychic');

INSERT INTO Is\_Spr\_Eff VALUES ('Ghost', 'Ghost');

INSERT INTO Is\_Spr\_Eff VALUES ('Dragon', 'Dragon');

INSERT INTO Is\_Spr\_Eff VALUES ('Dark', 'Psychic');

INSERT INTO Is\_Spr\_Eff VALUES ('Dark', 'Ghost');

INSERT INTO Is\_Spr\_Eff VALUES ('Steel', 'Ice');

INSERT INTO Is\_Spr\_Eff VALUES ('Steel', 'Rock');

INSERT INTO Is\_Spr\_Eff VALUES ('Steel', 'Fairy');

INSERT INTO Is\_Spr\_Eff VALUES ('Fairy', 'Fighting');

INSERT INTO Is\_Spr\_Eff VALUES ('Fairy', 'Dragon');

INSERT INTO Is\_Spr\_Eff VALUES ('Fairy', 'Dark');

ALTER TABLE Gym ENABLE CONSTRAINT gymtown\_fk;

ALTER TABLE Gym ENABLE CONSTRAINT gymleader\_fk;

ALTER TABLE Move ENABLE CONSTRAINT movetype\_fk;

ALTER TABLE Known\_By ENABLE CONSTRAINT knownbymovename;

ALTER TABLE Known\_By ENABLE CONSTRAINT knownbypokemonid;

ALTER TABLE Move\_Effect ENABLE CONSTRAINT moveeffectmove\_fk;

ALTER TABLE Item\_Effect ENABLE CONSTRAINT itemeffitem;

ALTER TABLE Used\_By ENABLE CONSTRAINT usedbyitem;

ALTER TABLE Used\_By ENABLE CONSTRAINT usedbytrainer;

ALTER TABLE Pokemon ENABLE CONSTRAINT pokemonowner;

ALTER TABLE Goes\_To ENABLE CONSTRAINT goestoroute;

ALTER TABLE Goes\_To ENABLE CONSTRAINT goestotown;

ALTER TABLE Pokemon\_Is\_Type ENABLE CONSTRAINT istypepokemon;

ALTER TABLE Pokemon\_Is\_Type ENABLE CONSTRAINT istypetype;

**5. Teamwork Summary**

ERD:

Daniel made the necessary changes to our ERD, which everyone else signed off on.

Relational Schema and Normalisation:

We divided up our relations and normalisation similarly to how we devised them in the first assignment, frequently double-checking each others work.

Normalisation:

Pokemon Sam

Pokemon\_is\_type Sam

Trainer Tomo

Move Daniel

Move\_Effect Daniel

Known\_by Daniel

Town Brock

Gym Tomo

Item Daniel

Item\_Effect Daniel

Used\_by Daniel

Route Sam

Goes\_to Tomo

Type Sam

Is\_Not\_eff Sam

Is\_Not\_Vry\_eff Sam

Is\_Spr\_eff Brock

SQL Implementation:

This was done all together, with each of us creating tables and inserts one at a time until they were all done. Testing was also done collaboratively.

General:

Despite conflicting schedules, group work went smoothly on this project, and being able to spend some solid time all together in the lab at the end meant we were able to work together to iron out any errors in our implementation.