University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: 2

Date: 2023-03-01

Group Number: 108

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Sahil Thind	92272954	l4c0t	thindsahil3@gamil.com
David Sopheap	24296634	g0g3b	david.sopheap@yahoo.ca
Zach Taylor	48297956	l1n5s	taylorzachary8@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

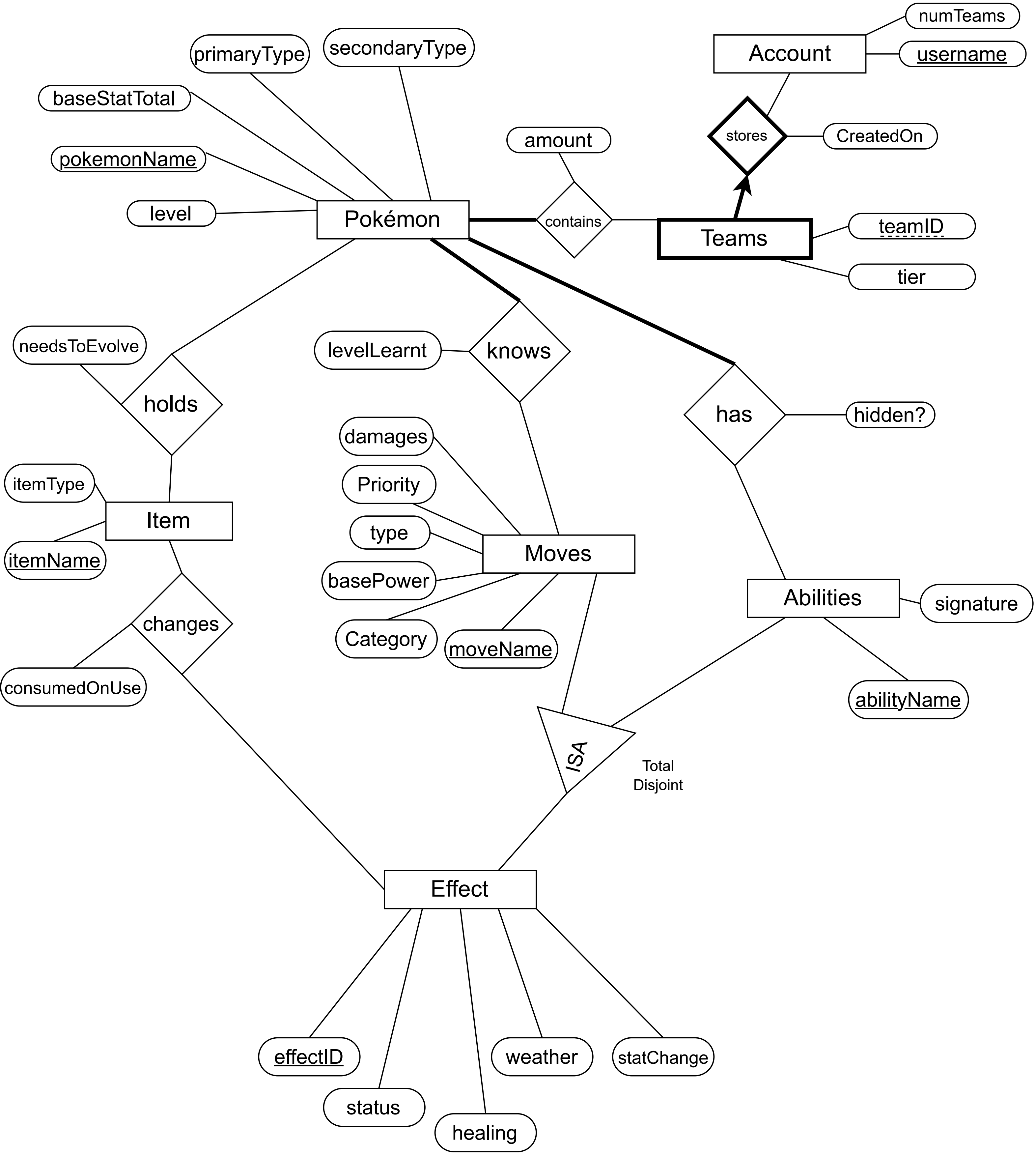
2.

Our project attempts to allow users to look-up information on all ~1000 Pokémon, and most game-relevant data points pertinent to them, namely, their names, stats, moves, items and abilities. These data points will be categorized based on data relevant to *them,* namely power, type, and effect. It will also be possible to categorize them into teams that users can then save and load in the future.

3.

We made the following changes to the ER diagram:

- Combined 'Ability ISA Effect' and 'Move ISA Effect' into 1 ISA.
- Added total-disjoint constraint to ISA, which was missing last time.
- Changed primary key names to differentiate keys that had the same name in different.
- Added new attributes to existing entities.
    - secondaryType
    - Damages
    - itemType
    - signature
    - numTeams


numTeams
Account
username
primaryType
secondaryType
baseStatTotal
stores
CreatedOn
amount
pokemonName
level
Pokémon
contains
Teams
teamID
tier
needsToEvolve
levelLearnt
knows
holds
has
hidden?
damages
itemType
Priority
Item
type
Moves
itemName
basePower
Abilities
signature
changes
Category
moveName
consumedOnUse
abilityName
ISA
Total
Disjoint
Effect
effectID
weather
statChange
status
healing

4.

Pokémon(pokémonName, level, primaryType, secondaryType, baseStateTotal)
Primary Key: name
Candidate Keys: name
Foreign Keys:
Constraints: NOT NULL on all attributes except level and secondaryType

Moves(effectID, movesName, basePower, type, category, damages)
Primary Key: effectID
Candidate Keys: movesName, effectID
Foreign Keys:
Constraints: effectID REFERENCES Effect, UNIQUE on movesName, NOT NULL on all attributes

Knows(pokémonName, movesName, levelLearnt)
Primary Key: (pokémonName, movesName)
Candidate Keys: (pokémonName, movesName)
Foreign Keys: pokémonName, movesName
Constraints: pokémonName REFERENCES Pokémon, movesName REFERENCES Moves, NOT NULL on levelLearnt

Abilities(effectID, abilityName, signature)
Primary Key: effectID
Candidate Keys: effectID, abilityName
Foreign Keys:
Constraints: effectID REFERENCES Effect, UNIQUE on abilityName, NOT NULL on abilityName, NOT NULL on signature

Effect(effectID, status, healing, weather, statChange)
Primary Key: effectID
Candidate Keys: effectID
Foreign Keys:
Constraints: NOT NULL on healing

Account(username, numTeams)
Primary Key: username
Candidate Keys: username
Foreign Keys:
Constraints: NOT NULL on numTeams

Item(itemName, itemType)
Primary Key: itemName
Candidate Keys: itemName
Foreign Keys:
Constraints: NOT NULL on itemType

StoresTeams(teamID, tier, username)
Primary Key: (teamID, username)
Candidate Keys: (teamID, username)
Foreign Keys: username
Constraints: NOT NULL on username

Contains(pokemonName, teamID, amount)
Primary Key: (pokemonName, teamID)
Candidate Keys: (pokemonName, teamID)
Foreign Keys: pokemonName REFERENCES Pokemon, teamID REFERENCES Teams
Constraints: NOT NULL on amount

Holds(pokemonName, itemName, needsToEvolve)
Primary Key: (pokemonName, itemName)
Candidate Keys: (pokemonName, itemName)
Foreign Keys: pokemonName REFERENCES Pokemon, itemName REFERENCES Items
Constraints:

Changes(itemName, effectID, consumedToActivate)
Primary Key: (itemName, effectID)
Candidate Keys: (itemName, effectID)
Foreign Keys: itemNameREFERENCES Item, effectID REFERENCES Effect
Constraints:

Has(pokemonName, abilityName, hidden)
Primary Key: (pokemonName, abilityName)
Candidate Keys: (pokemonName, abilityName)
Foreign Keys: pokemonName REFERENCES Pokemon, abilityName REFERENCES Abilities
Constraints:

5.

PK/CK Related FDs:

pokemonName → name, level, baseStatTotal, primaryType, secondaryType
pokemonName, moveName → levelLearnt
pokemonName, itemName → needsToEvolve
moveName → type, basePower, category, priority, damages, effectID, status, healing, weather, statChange
abilityName → effectID, status, healing, weather, statChange, signature
pokemonName, username, teamID → amount
effectID → status, healing, weather, statChange, type, basePower, category, priority, damages, abilityName, moveName, signature
itemName → itemType
username, teamID → createdOn
username, teamID → tier
pokemonName, itemName → holds
itemName, effectID → consumedOnUse
pokemonName, abilityName → hidden

Non-PK/CK Related FDs:

category -> damages
priority, basePower, category -> name

6.

We normalize all tables to BCNF.

New Tables:

Moves needs to be normalized because that table has FDs:

- moveName -> type, basePower, category, priority, damages, effectID
- category -> damages
- priority, basePower, category -> moveName
- effectID -> type, basePower, category, priority, damages, moveName

The second relation violates BCNF. Removing the second one first gives us:

1. Moves(effectID, movesName, category, basePower, type)
2. MovesCategoryDamage(category, damages)

So altogether we have :

Moves(effectID, movesName, category, basePower, type)
Primary Key: effectID
Candidate Keys: movesName, effectID
Foreign Keys:

MovesCategoryDamage(category, damages)
Primary Key: category
Candidate Keys: category
Foreign Keys: category

Old Tables:

Pokémon(pokémonName, level, primaryType, secondaryType, baseStateTotal)
Primary Key: name
Candidate Keys: name
Foreign Keys:

Knows(pokémonName, movesName, levelLearnt)
Primary Key: (pokémonName, movesName)
Candidate Keys: (pokémonName, movesName)
Foreign Keys: pokémonName, movesName

Abilities(effectID, abilityName, signature)
Primary Key: effectID
Candidate Keys: effectID, abilityName
Foreign Keys:

Effect(effectID, status, healing, weather, statChange)
Primary Key: effectID
Candidate Keys: effectID
Foreign Keys:

Item(itemName, itemType)
Primary Key: itemName
Candidate Keys: itemName
Foreign Keys:

StoresTeams(teamID, tier, username)
Primary Key: (teamID, username)
Candidate Keys: (teamID, username)
Foreign Keys: username

Account(username, numTeams)
Primary Key: username
Candidate Keys: username
Foreign Keys:

Contains(pokemonName, teamID, amount)
Primary Key: (pokemonName, teamID)
Candidate Keys: (pokemonName, teamID)
Foreign Keys: pokemonName REFERENCES Pokemon, teamID REFERENCES Teams

Holds(pokemonName, itemName, needsToEvolve)
Primary Key: (pokemonName, itemName)
Candidate Keys: (pokemonName, itemName)
Foreign Keys: pokemonName REFERENCES Pokemon, itemName REFERENCES Items

Changes(itemName, effectID, consumedToActivate)
Primary Key: (itemName, effectID)
Candidate Keys: (itemName, effectID)
Foreign Keys: itemNameREFERENCES Item, effectID REFERENCES Effect

Has(pokemonName, abilityName, hidden)
Primary Key: (pokemonName, abilityName)
Candidate Keys: (pokemonName, abilityName)
Foreign Keys: pokemonName REFERENCES Pokemon, abilityName REFERENCES Abilities

7.

```
CREATE TABLE Moves {
        effectID        char(255),
        movesName       char(255) NOT NULL UNIQUE,
        basePower       int NOT NULL,
        category        bit NOT NULL,
        type            char(255) NOT NULL,
        PRIMARY KEY (effectID)
        FOREIGN KEY (effectID) REFERENCES Effects(effectID) ON DELETE CASCADE
}

CREATE TABLE MovesCategoryDamage {
        category        char(255),
        damage          int NOT NULL,
        PRIMARY KEY (category)
        FOREIGN KEY (category ) REFERENCES Moves(category)}

CREATE TABLE Pokemon {
        pokemonName char(255),
        level           int,
        primaryType char(255) NOT NULL,
        secondaryType char(255),
        baseStatTotal NOT NULL,
        PRIMARY KEY (pokemonName)
}

CREATE TABLE Knows {
        pokemonName char(255),
        moveName        char(255),
        levelLearnt     int NOT NULL,
        PRIMARY KEY (pokemonName, moveName)
        FOREIGN KEY (pokemonName) REFERENCES Pokemon(pokemonName)
        ON DELETE CASCADE
        FOREIGN KEY (moveName) REFERENCES MoveName(moveName)
        ON DELETE CASCADE
}

CREATE TABLE Abilities {
        effectID        char(255),
        abilityName     char(255) NOT NULL UNIQUE,
        signature       bit NOT NULL,
        PRIMARY KEY (effectID) REFERENCES Effects(effectID) ON DELETE CASCADE
}

CREATE TABLE Effect {
        effectID        char(255),
        status          char(255) ,
        healing         bit NOT NULL,
        weather         char(255),
        statChange      char(255),
        PRIMARY KEY (effectID)
}

CREATE TABLE Account {
        username        char(255),
        numTeams        char(255),
        PRIMARY KEY (username)

}

CREATE TABLE StoresTeams {
        teamID          int,
        tier            char(255),
        username        char(255),
        PRIMARY KEY(username, teamID),
        FOREIGN KEY username REFERENCES Account ON DELETE CASCADE
}

CREATE TABLE Item {
        itemName        char(255),
        itemType        char(255) NOT NULL,
        PRIMARY KEY (itemName)
}

CREATE TABLE Contains {
        pokemonName char(255),
        teamID          int,
        amount          int NOT NULL,
        PRIMARY KEY (pokemonName, teamID),
        FOREIGN KEY (pokemonName) REFERENCES Pokemon ON DELETE CASCADE,
        FOREIGN KEY (teamID) REFERENCES Teams ON DELETE CASCADE
}

CREATE TABLE Holds {
        pokemonName char(255),
        itemName     char(255),
        needsToEvolve bit,
        PRIMARY KEY (pokemonName, itemName),
        FOREIGN KEY (pokemonName) REFERENCES Pokemon ON DELETE CASCADE,
        FOREIGN KEY (itemName) REFERENCES Item ON DELETE CASCADE
}

CREATE TABLE Changes {
        itemName     char(255),
        effectID     char(255),
        consumedToActivate bit,
        PRIMARY KEY (itemName, effectID),
        FOREIGN KEY (itemName) REFERENCES Item ON DELETE CASCADE,
        FOREIGN KEY (effectID) REFERENCES Effect ON DELETE CASCADE
}

CREATE TABLE Has {
        pokemonName char(255),
        abilityName  char(255),
        hidden       bit,
        PRIMARY KEY (pokemonName, abilityName),
        FOREIGN KEY (pokemonName) REFERENCES Pokemon ON DELETE CASCADE,
        FOREIGN KEY (abilityName) REFERENCES Abilities ON DELETE CASCADE
}
```

8.

Moves:

```
INSERT INTO Moves (effectID, movesName, category, basePower, type)
VALUES (‘TACKLE’, ‘Tackle’, ‘physical’, 40, ‘Normal’)
```

```
INSERT INTO Moves (effectID, movesName, category, basePower, type)
VALUES (‘ANCIENT_POWER’, ‘Ancient Power’, ‘special’, 60, ‘Rock’)
```

```
INSERT INTO Moves (effectID, movesName, category, basePower, type)
VALUES (‘SCRATCH’, ‘Scratch’, ‘physical’, ‘40’, ‘Normal’)
```

```
INSERT INTO Moves (effectID, movesName, category, basePower, type)
VALUES (‘LEECH_LIFE’, ‘Leech Life’, ‘physical’, ‘80’, ‘Bug’)
```

```
INSERT INTO Moves (effectID, movesName, category, basePower, type)
VALUES (‘AQUA_JET’, ‘Aqua Jet’, ‘physical’, ‘40’, ‘Normal’)
```

MovesCategoryDamage:

Note that there are only 3 possible categories, and since this table represents category -> damage, we only have 3 tuples in this table (until Gamefreak adds more move categories to Pokemon).

```
INSERT INTO MovesCategoryDamage (category, damage)
VALUES (‘physical’, 1)
```

```
INSERT INTO MovesCategoryDamage (category, damage)
VALUES (‘special’, 1)
```

```
INSERT INTO MovesCategoryDamage (category, damage)
VALUES (‘status’, 0)
```

Pokemon:

```
INSERT INTO Pokemon(pokémonName, level, primaryType, secondaryType, baseStateTotal)
VALUES (‘Electrode’, NULL, ‘Electric’, NULL, 490)
```

```
INSERT INTO Pokemon(pokémonName, level, primaryType, secondaryType, baseStateTotal)
VALUES (‘Diglett’, NULL, Ground, NULL, 265)
```

```
INSERT INTO Pokemon(pokémonName, level, primaryType, secondaryType, baseStateTotal)
VALUES (‘Nidoran♂’, 100, ‘Poison’, NULL, 273)

INSERT INTO Pokemon(pokémonName, level, primaryType, secondaryType, baseStateTotal)
VALUES (‘Mankey’, NULL, ‘Fighting’, NULL, 305)

INSERT INTO Pokemon(pokémonName, level, primaryType, secondaryType, baseStateTotal)
VALUES (‘Ivysaur’, NULL, ‘Grass’, ‘Poison’, 405)
```

Knows:

```
INSERT INTO Knows(pokemonName, moveName, levelLearnt)
VALUES (‘Ivysaur’, ‘Growl’, 1)

INSERT INTO Knows(pokemonName, moveName, levelLearnt)
VALUES (‘Ivysaur’, ‘Growth’, 1)

INSERT INTO Knows(pokemonName, moveName, levelLearnt)
VALUES (‘Ivysaur’, ‘Tackle’, 1)

INSERT INTO Knows(pokemonName, moveName, levelLearnt)
VALUES (‘Ivysaur’, ‘Vine Whip’, 1)

INSERT INTO Knows(pokemonName, moveName, levelLearnt)
VALUES (‘Ivysaur’, ‘Leech Seed’, 9)
```

Abilities:

```
INSERT INTO Abilities(effectID, abilityName)
VALUES (‘OVERGROW’, ‘Overgrow’)

INSERT INTO Abilities(effectID, abilityName)
VALUES (‘BLAZE’, ‘Blaze’)

INSERT INTO Abilities(effectID, abilityName)
VALUES (‘TORRENT’, ’Torrent’)

INSERT INTO Abilities(effectID, abilityName)
VALUES (‘SWARM’, ‘Swarm’)
```

```
INSERT INTO Abilities(effectID, abilityName)
VALUES (‘LEVITATE’, ‘Levitate’)
```

Effects:

```
INSERT INTO Effects(effectID, status, healing, weather, statChange)
VALUES (‘LEVITATE’, NULL, 0, NULL, NULL)

INSERT INTO Effects(effectID, status, healing, weather, statChange)
VALUES (‘SWIFT SWIM’, NULL, 0, ‘Rain’, ‘Speed’)

INSERT INTO Effects(effectID, status, healing, weather, statChange)
VALUES (‘TOXIC BOOST’, ‘Poison’, 0, NULL, ‘Attack’)

INSERT INTO Effects(effectID, status, healing, weather, statChange)
VALUES (‘ABSORB’, NULL, 1, NULL, NULL)

INSERT INTO Effects(effectID, status, healing, weather, statChange)
VALUES (‘GROWTH’, NULL, 1, 1, NULL)
```

Account:

```
INSERT INTO Account(username, numTeams)
VALUES (‘bigJohn55’, 99)

INSERT INTO Account(username, numTeams)
VALUES (‘lilRichard69’, 0)

INSERT INTO Account(username, numTeams)
VALUES (‘bigDipper99’, 1)

INSERT INTO Account(username, numTeams)
VALUES (‘a283ka’, 3)

INSERT INTO Account(username, numTeams)
VALUES (‘aklsdjsl;f’, 3)
```

Items:

```
INSERT INTO Item(itemName, itemType)
VALUES(‘Antidote’, ‘Medicine’)

INSERT INTO Item(itemName, itemType)
VALUES(‘Zinc’, ‘Medicine’)

INSERT INTO Item(itemName, itemType)
VALUES(‘Steel Gem’, Gem Items’)

INSERT INTO Item(itemName, itemType)
VALUES(‘Choice Scarf’, ‘Choice Items’)

INSERT INTO Item(itemName, itemType)
VALUES(‘Wiki Berry’, ‘Berries’)
```

StoresTeams:

```
INSERT INTO Teams(teamID, tier, username)
VALUES(1341, ‘NULL’, ‘coolguy55’)

INSERT INTO Teams(teamID, tier, username)
VALUES(3531, “Ubers”, ‘swaglord4’)

INSERT INTO Teams(teamID, tier, username)
VALUES(1111, “Overused”, ‘11111’)

INSERT INTO Teams(teamID, tier, username)
VALUES(0135, “UnderUsed”, ‘g0g3b’)

INSERT INTO Teams(teamID, tier, username)
VALUES(1457, “NULL”, ‘BOBAMA’)
```

Contains:

```
INSERT INTO Contains(pokemonName, teamID, amount)
VALUES(‘Pikachu’, 0135, 1)

INSERT INTO Contains(pokemonName, teamID, amount)
VALUES(‘Charizard’, 0135, 1)
```

```
INSERT INTO Contains(pokemonName, teamID, amount)
VALUES(‘Squirtle’, 0135, 1)

INSERT INTO Contains(pokemonName, teamID, amount)
VALUES(‘Ivysaur’, 2941, 1)

INSERT INTO Contains(pokemonName, teamID, amount)
VALUES(‘Pikachu’, 2941, 1)
```

Holds:

```
INSERT INTO Holds(pokemonName, itemName, needsToEvolve)
VALUES(‘Pikachu’, ‘Antidote’, 0)

INSERT INTO Holds(pokemonName, itemName, needsToEvolve)
VALUES(‘Charizard’, ‘Choice Scarf’, 0)

INSERT INTO Holds(pokemonName, itemName, needsToEvolve)
VALUES(‘Pikachu’, ‘Wiki Berry’, 0)

INSERT INTO Holds(pokemonName, itemName, needsToEvolve)
VALUES(‘Squirtle’, ‘Zinc’, 0)

INSERT INTO Holds(pokemonName, itemName, needsToEvolve)
VALUES(‘Eevee’, ‘Water Stone’, 1)
```

Changes:

```
INSERT INTO Changes(itemName, effectID, consumedToActivate)
VALUES(‘Antidote’, ‘BLAZE’, ‘0’)

INSERT INTO Changes(itemName, effectID, consumedToActivate)
VALUES(‘Antidote’, ‘TORRENT’, ‘0’)

INSERT INTO Changes(itemName, effectID, consumedToActivate)
VALUES(‘Water Stone’, ‘SWARM’, ‘0’)

INSERT INTO Changes(itemName, effectID, consumedToActivate)
VALUES(‘Scarf’, ‘LEVITATE’, ‘0’)
```

```
INSERT INTO Changes(itemName, effectID, consumedToActivate)
VALUES(‘Toxic Orb’, ‘POISON’, ‘0’)
```

Has:

```
INSERT INTO Has(pokemonName, abilityName, hidden)
VALUES(‘Magikarp’, ‘Swift Swim’, 0)

INSERT INTO Has(pokemonName, abilityName, hidden)
VALUES(‘Pikachu’, ‘Lightning Rod’, 1)

INSERT INTO Has(pokemonName, abilityName, hidden)
VALUES(‘Charizard’, ‘Solar Power’, 1)

INSERT INTO Has(pokemonName, abilityName, hidden)
VALUES(‘Charmander’, ‘Blaze’, 0)

INSERT INTO Has(pokemonName, abilityName, hidden)
VALUES(‘Pidgey’, ‘Keen Eye’, 0)
```