University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: 2

Date: October 21st, 2022

Group Number: 51

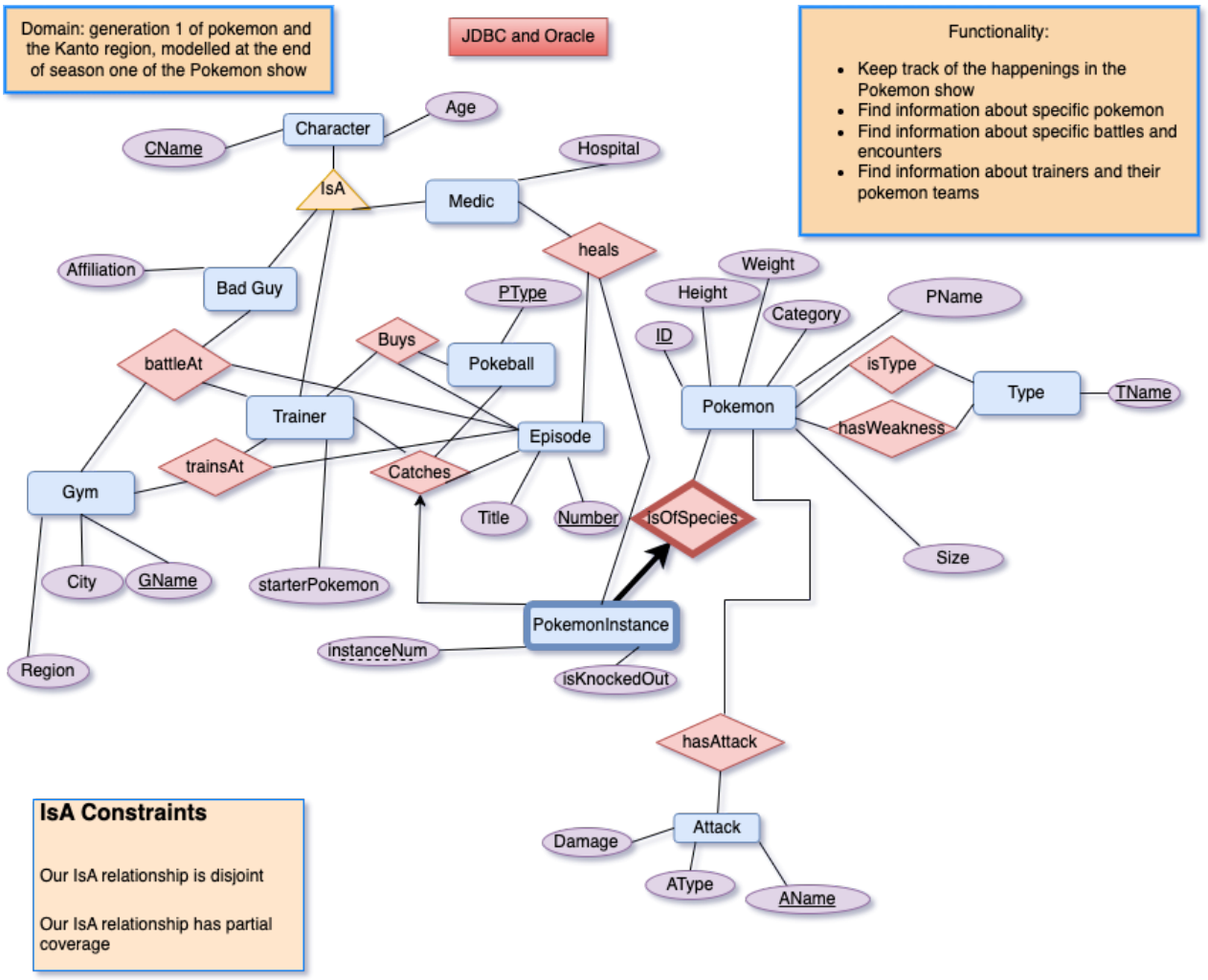
Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Mathias de Carle	21960349	h3z2b	mathiasdecarle@gmail.com
Henry Larsen	58398876	k4k3b	henrylarsen01@yahoo.ca
Will Oxtoby	24563199	k7j3b	will.oxtoby@gmail.com

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

Question 2.

Changes made : IsA Constraints added after milestone 1



Sources:

- https://pokemondb.net/pokedex/spearow
- https://www.pokemon.com/us/pokedex/
- https://bulbapedia.bulbagarden.net/wiki/User:Force_Fire/List_of_anime_Pok%C3%A9mon_by_capture

Question 3.

Legend:

primaryKey
foreignKey
candidateKey

Schemas:

Pokemon (id, height, weight, category, PName, size)

isType (**id**, TName)

hasWeakness (**id**, TName)

pokemonIsOfInstance (**id**, instanceNum, isKnockedOut)

Catches (**instanceNum**, **CName**, **EpisodeNumber**, **PType**)

ASSUMPTION: no pokemon gets caught more than once

Episode (*title*, EpisodeNumber)

Gym (Gname, location, region)

Character (Cname, age)

ASSUMPTION: everyone's name is unique

Trainer (**Cname**, starterPokemon)

BadGuy(**Cname**, Affiliation)

Medic(**Cname**, Hospital)

Pokeball(PType)

Heals (**CName**, **instanceNum**, **EpisodeNumber**)

Buys(**PType**, **EpisodeNumber**, **Cname**)

TrainsAt(**Cname**, **EpisodeNumber**, **Gname**)

BattleAt(**TrainerName**, **BadGuyName**, **Gname**, **EpisodeNumber**)

HasAttack(**AttackName**, **id**)

Attack(AttackName, Type, Damage)

Question 4.

Key for Abstraction:

Pokemon
- Id = A
- Height = B
- Weight = C
- Category = D
- Pname = E
- Size = W

Type
- TName = F

Attack
- AName = G
- AType = H
- Damage = I

PokemonInstance
- isKnockedOut = J
- instanceNum = K

Episode
- Number = L
- Trainer Name = Y
- starterPokemon = S
- Title = M

Pokeball
- PType = N

Character
- Age = P
- CName = Q

Medic
- Hospital = O

Bad Guy
- Affiliation = R
- BadGuyName = Z

Trainer
- StarterPokemon = S
- TrainerName = Y

Gym
- Region = T
- City = V
- GName = U

Notes:

INTERPRETATION: Candidate Keys (CK's) will be reflected in the relation where they are a CK of. In other relations, we found it redundant to rewrite everything with the CK and PK (Relevant to Episode).
ASSUMPTION: all city names (V) are unique.

Pokemon (A, B, C, D, E, W)
- A → BCDEW
- BC → W

isType (**A**, F)
- AF → AF

hasWeakness (**A**, F)
- AF → AF

pokemonIsOfInstance (**A**, K, J)
- AK → J

Catches (**K**, Q, L, N)
- K → KQLN

Episode (*M*, L)

M → L

L → M

Gym (U, T, V)

U → UTV

T → V

Character (Q, P)

Q → QP

Trainer (**Q**, S)

Q → S

Bad Guy(**Q**, R)

Q → R

Medic(**Q**, O)

Q → O

Pokeball(N)

N → N

Heals (**Q**, **K**, **L**)

QKL → QKL

QKM → QKM

Buys (**N**, **L**, **Q**)

NLQ → NLQ

TrainsAt(**Q**, **L**, **U**)

QLU → QLU

BattleAt(**Y**, **Z**, **U**, **L**)

YZUL → YZUL

HasAttack(**G**, **A**)

GA → GA

Attack(G, H, I)

G → HI

Question 5.

Pokemon Relation

Pokemon (A, B, C, D, E, W)

A → BCDEW

BC → W

Step 1: Find Minimal Cover

Step 1.a: Put all functional dependencies in standard form

A → B

A → C

A → D

A → E

A → W

BC → W

Step 1.b: Minimize left hand side of functional dependencies

All functional dependencies are already minimized.

Step 1.c: Delete redundant functional dependencies

**** This is our minimal cover ****

Step 2/3: For each functional dependency X→b, add relation Xb to the decomposition for R.

Pokemon1 (A, B)

Pokemon2 (A, C)

Pokemon3 (A, D)

Pokemon4 (A, E)

Pokemon5 (A, W)

Pokemon6 (B, C, W)

Primary Key: A

**** Pokemon converted to 3NF ****

Gym Relation

Gym (U, T, V)

U → TV

T → V

Step 1: Find Minimal Cover

Step 1.a: Put all functional dependencies in standard form

U → T

T → V

U → V

Step 1.b: Minimize left hand side of functional dependencies

All functional dependencies are already minimized.

Step 1.c: Delete redundant functional dependencies.

** This is our minimal cover **

Step 2/3: For each functional dependency X→b, add relation Xb to the decomposition for R.

Gym1 (U, T)
Gym2 (U, V)
Gym3 (T, V)

Primary Key: U

****Gym converted to 3NF****

All other tables are already in 3NF and listed below

isType (**A**, F)
AF → AF

hasWeakness (**A**, F)
AF → AF

pokemonIsOfInstance (**A**, K, J)
AK → J

Catches (**K**, Q, L, N)
K → KQLN

Episode (*M*, L)
M → L
L → M

Character (Q, P)
Q → QP

Trainer (**Q**, S)
Q → S

Bad Guy(**Q**, R)
Q → R

Medic(**Q**, O)
Q → O

Pokeball(N)
N → N

Heals (**Q**, **K**, **L**)
QKL → QKL
QKM → QKM

Buys (**N**, **L**, **Q**)
NLQ → NLQ

TrainsAt(**Q**, **L**, **U**)
QLU → QLU

BattleAt(**Y**, **Z**, **U**, **L**)
YZUL → YZUL

HasAttack(**G**, **A**)
GA → GA

Attack(G, H, I)
G → HI

Question 6.

```
CREATE TABLE Pokemon
       (id           INTEGER(3),
       height        INTEGER(10),
       weight        INTEGER(10),
       category      CHARACTER(20),
       PName         CHARACTER(20),
       size          CHARACTER(10),
       PRIMARY KEY (id),
       UNIQUE(id, PName));

CREATE TABLE isType
       (id           INTEGER(3),
       TName         CHARACTER(50),
       PRIMARY KEY (id, TName),
       FOREIGN KEY (id)
         REFERENCES Pokemon
       ON DELETE CASCADE
       ON UPDATE CASCADE);

CREATE TABLE hasWeakness
       (id           INTEGER(3),
       TName         CHARACTER(50),
       PRIMARY KEY (id, TName),
       FOREIGN KEY (id)
         REFERENCES Pokemon
       ON DELETE CASCADE
       ON UPDATE CASCADE);

CREATE TABLE PokemonIsOfInstance
       (id           INTEGER(3),
       instanceNum   INTEGER(3),
       isKnockedOut  CHARACTER(5),
       PRIMARY KEY (ID, instanceNum),
       UNIQUE(instanceNum),
       FOREIGN KEY (id)
         REFERENCES Pokemon
       ON DELETE CASCADE
       ON UPDATE CASCADE,
       UNIQUE(instanceNum));

CREATE TABLE Catches
       (instanceNum   INTEGER(3),
       CName          CHARACTER(20),
       EpisodeNumber INTEGER(3),
       PType          CHARACTER(20),
       PRIMARY KEY (instanceNum, CName,
Number, PType),
       FOREIGN KEY (instanceNum)
         REFERENCES PokemonIsOfInstance
       ON DELETE CASCADE
       ON UPDATE CASCADE,
       FOREIGN KEY (CName)
         REFERENCES Character
       ON DELETE CASCADE
```

```
CREATE TABLE Trainer
       (Cname        CHARACTER(20),
       age      INTEGER(3),
       PRIMARY KEY (Cname),
       FOREIGN KEY (Cname)
         REFERENCES Character
       ON DELETE CASCADE
       ON UPDATE CASCADE);

CREATE TABLE BadGuy
       (Cname       CHARACTER(20),
       Affiliation INTEGER(8),
       PRIMARY KEY (Cname),
       FOREIGN KEY (Cname)
         REFERENCES Character
       ON DELETE CASCADE
       ON UPDATE CASCADE);

CREATE TABLE Medic
       (Cname      CHARACTER(20),
       Hospital    CHARACTER(20),
       PRIMARY KEY (Cname),
       FOREIGN KEY (Cname)
       REFERENCES Character
       ON DELETE CASCADE
       ON UPDATE CASCADE);

CREATE TABLE Pokeball
       (PType      CHARACTER(20),
       PRIMARY KEY (PType));

CREATE TABLE Heals
       (Cname         CHARACTER(20),
       instanceNum    INTEGER(3),
       EpisodeNumber INTEGER(3),
       PRIMARY KEY (Cname,instanceNum,
EpisodeNumber),
       FOREIGN KEY (Cname)
         REFERENCES Character
       ON DELETE CASCADE
       ON UPDATE CASCADE,
       FOREIGN KEY (InstanceNum)
         REFERENCES PokemonIsOfInstance
       ON DELETE CASCADE
       ON UPDATE CASCADE,
       FOREIGN KEY (EpisodeNumber)
         REFERENCES Episode
       ON DELETE CASCADE
       ON UPDATE CASCADE);

CREATE TABLE Buys
       (PType         CHARACTER(20),
       EpisodeNumber INTEGER(3),
       Cname          CHARACTER(20),
```

```
        ON UPDATE CASCADE,
        FOREIGN KEY (EpisodeNumber)
          REFERENCES Episode
        ON DELETE CASCADE
        ON UPDATE CASCADE,
        FOREIGN KEY (PType)
          REFERENCES Pokeball
        ON DELETE CASCADE
        ON UPDATE CASCADE);

CREATE TABLE Episode
        (title CHARACTER(20),
        EpisodeNumber INTEGER(3),
        PRIMARY KEY (EpisodeNumber),
        UNIQUE(EpisodeNumber));

CREATE TABLE Gym
        (Gname        CHARACTER(20),
        location      CHARACTER(20),
        region        CHARACTER(20),
        PRIMARY KEY (Gname),
        UNIQUE(Gname));

CREATE TABLE Character
        (CName        CHARACTER(20),
        age         INTEGER(3),
        PRIMARY KEY (CName),
        UNIQUE(CName));

CREATE TABLE TrainsAt
        (Cname        CHARACTER(20),
        EpisodeNumber INTEGER(3),
        Gname         CHARACTER(20),
        PRIMARY KEY (Cname,EpisodeNumber,
Gname),
        FOREIGN KEY (Cname)
          REFERENCES Character
        ON DELETE CASCADE
        ON UPDATE CASCADE,
        FOREIGN KEY (EpisodeNumber)
          REFERENCES Episode
        ON DELETE CASCADE
        ON UPDATE CASCADE,
        FOREIGN KEY (Gname)
          REFERENCES Gym
        ON DELETE CASCADE
        ON UPDATE CASCADE);

CREATE TABLE HasAttack
        (AttackName        CHARACTER(20),
        Id                 INTEGER(3),
        PRIMARY KEY (AttackName, id),
        FOREIGN KEY (AttackName)
          REFERENCES Attack
        PRIMARY KEY (Cname,PType,
EpisodeNumber),
        FOREIGN KEY (Cname)
          REFERENCES Character
        ON DELETE CASCADE
        ON UPDATE CASCADE,
        FOREIGN KEY (PType)
          REFERENCES Pokeball
        ON DELETE CASCADE
        ON UPDATE CASCADE,
        FOREIGN KEY (EpisodeNumber)
          REFERENCES Episode
        ON DELETE CASCADE
        ON UPDATE CASCADE);

CREATE TABLE BattleAt
        (Cname        CHARACTER(20),
        Cname         CHARACTER(20),
        Gname         CHARACTER(20),
        EpisodeNumber INTEGER(3),
        PRIMARY KEY (Cname,Cname, Gname,
EpisodeNumber),
        FOREIGN KEY (Cname)
          REFERENCES Trainer
        ON DELETE CASCADE
        ON UPDATE CASCADE,
        FOREIGN KEY (Cname)
          REFERENCES BadGuy
        ON DELETE CASCADE
        ON UPDATE CASCADE,
        FOREIGN KEY (Gname)
          REFERENCES Gym
        ON DELETE CASCADE
        ON UPDATE CASCADE,
        FOREIGN KEY (EpisodeNumber)
          REFERENCES Episode
        ON DELETE CASCADE
        ON UPDATE CASCADE);

CREATE TABLE Attack
        (AttackName        CHARACTER(20),
        Type               CHARACTER(20),
        Damage             INTEGER(5),
        PRIMARY KEY (AttackName),
        UNIQUE(AttackName));
```

```
        ON DELETE CASCADE
        ON UPDATE CASCADE,
        FOREIGN KEY (id)
          REFERENCES Pokemon
        ON DELETE CASCADE
        ON UPDATE CASCADE);
```

Question 7.

```
INSERT INTO Pokemon
VALUES (001, 0070,6900, 'seed', 'Bulbasaur', 'small'),
(002,0100, 13000, 'seed', 'Ivysaur', 'medium'),
(003,0200, 100000, 'seed', 'Venusaur', 'large'),
(004,0060, 8500, 'lizard', 'Charmander', 'small'),
(005,0110, 19000, 'lizard', 'Charmeleon', 'medium');
```

Pokemon

id	Height (cm)	Weight (g)	category	PName	size
001	0070	6900	seed	Bulbasoar	small
002	0100	13000	seed	Ivysoar	medium
003	0200	100000	seed	Venusoar	large
004	0060	8500	lizard	Charmander	small
005	0110	19000	lizard	charmeleon	medium

```
INSERT INTO isType
VALUES (004, 'fire'),
(007, 'water'),
(010, 'bug'),
(012, 'bug,flying'),
(013, 'bug,poison');
```

isType

id	TName
004	fire

007	water
010	bug
012	bug, flying
013	bug, poison

```
INSERT INTO hasWeakness
VALUES (004, 'water, ground, rock'),
(007, 'water, grass, dragon'),
(010, 'fire, flying, rock'),
(012, 'fire, flying, electric, ice, rock'),
(013, 'fire, psychic, flying, rock');
```

hasWeakness

id	TName
004	water, ground, rock
007	water, grass, dragon
010	fire, flying, rock
012	fire, flying, electric, ice, rock
013	fire, psychic, flying, rock

```
INSERT INTO PokemonIsOfInstance
VALUES (001, 001,'yes'),
(002, 001,'yes'),
(001, 002,'no')),
(100, 001,'no'),
(074, 001,'no');
```

PokemonIsOfInstance

id	InstanceNum	isKnockedOut
001	001	yes
002	001	yes
001	002	no
100	001	no
074	001	no

```
INSERT INTO catches
VALUES (001, 'Josh',012, 'quick'),
(002, 'Ash',002, 'heal'),
(003, 'Bill',004, 'ultra'),
(004, 'John',012, 'great'),
(005, 'James',098, 'great');
```

Catches

instanceNum	CName	EpisodeNumber	PType
001	Josh	012	quick
002	Ash	002	heal
003	Bill	004	ultra
004	John	012	great
005	James	098	great

```
INSERT INTO Episode
VALUES ('The great adventure', 001),
('New pokemon', 002),
('Big battles', 003),
('Exciting events', 004),
('Final battle', 005);
```

Episode

title	EpisodeNumber
The great adventure	001
New pokemon	002
Big battles	003
Exciting events	004
Final battle	005

```
INSERT INTO Gym
VALUES (001, 'Pewter','kanto'),
(002, 'Cerulean','kanto'),
(003, 'Vermilion','kanto'),
```

```
(004, 'Celadon','kanto'),
(005, 'Saffron','kanto');
```

Gym

Gname	location	region
001	Pewter	kanto
002	Cerulean	kanto
003	Vermilion	kanto
004	Celadon	kanto
005	Saffron	kanto

```
INSERT INTO Character
VALUES ('Ella', 12),
('Jenna', 13),
('Gabriella', 14),
('Taylor', 15),
('Rihanna', 16);
```

Character

Cname	age
Ella	12
Jenna	13
Gabriella	14
Taylor	15
Rihanna	16

```
INSERT INTO Trainer
VALUES ('Tyrone', 12),
('Tyreese', 13),
('Daquan', 14),
('Charlise', 15),
('Abneet', 16);
```

Trainer

Cname	age
Tyrone	12
Tyreese	13

Daquan	14
Charlise	15
Abneet	16

```
INSERT INTO BadGuy
VALUES ('Katie', Rocket),
('Danae', Aqua),
('Holly', Galactic),
('Erin', Rocket),
('Amanda', Plasma);
```

BadGuy

Cname	Affiliation
Katie	Rocket
Danae	Aqua
Holly	Galactic
Erin	Rocket
Amanda	Plasma

```
INSERT INTO Medic
VALUES ('Chris', 'Hospital1'),
('Matt', 'Hospital2'),
('Jeffrey', 'Hospital3'),
('Alice', 'Hospital4'),
('Jenkins', 'Hospital5');
```

Medic

Cname	Hospital
Chris	Hospital1
Matt	Hospital2
Jeffrey	Hospital3
Alice	Hospital4
Jenkins	Hospital5

```
INSERT INTO Pokeball
```

```
VALUES ('quick'),
('heal'),
('ultra'),
('great'),
('great');
```

Pokeball

PType
quick
heal
ultra
great
great

```
INSERT INTO Heals
VALUES ('Chris', 001, 001),
('Matt', 002, 003),
('Jeffrey', 203, 024),
('Alice', 034, 012),
('Jenkins', 045, 011);
```

Heals

Cname	instanceNum	EpisodeNumber
Chris	001	001
Matt	002	003
Jeffrey	203	024
Alice	034	012
Jenkins	045	011

```
INSERT INTO Buys
VALUES ('quick', 001, 'Tyrone'),
('heal', 003, 'Tyreese'),
('ultra', 024, 'Daquan'),
('great', 012, 'Charlise'),
('great', 011, 'Abneet');
```

Buys

PType	EpisodeNumber	Cname

quick	001	Tyrone
heal	003	Tyreese
ultra	024	Daquan
great	012	Charlise
great	011	Abneet

```
INSERT INTO TrainsAt
VALUES ('Tyrone', 001, 001),
('Tyreese', 003, 002),
('Daquan', 024, 003),
('Charlise', 012, 004),
('Abneet', 011, 005);
```

TrainsAt

Cname	EpisodeNumber	Gname
Tyrone	001	001
Tyreese	003	002
Daquan	024	003
Charlise	012	004
Abneet	011	005

```
INSERT INTO BattleAt
VALUES ('Tyrone', 'Katie', 001, 001),
('Tyreese', 'Danae', 002, 003),
('Daquan', 'Holly', 003, 024),
('Charlise', 'Erin', 004, 012),
('Abneet', 'Amanda', 005, 011);
```

BattleAt

Cname	Cname	Gname	EpisodeNumber
Tyrone	Katie	001	001
Tyreese	Danae	002	003
Daquan	Holly	003	024
Charlise	Erin	004	012

Abneet	Amanda	005	011

```
INSERT INTO HasAttack
VALUES (001, 0070,6900, 'seed', 'Bulbasaur', 'small'),
(002,0100, 13000, 'seed', 'Ivysaur', 'medium'),
(003,0200, 100000, 'seed', 'Venusaur', 'large'),
(004,0060, 8500, 'lizard', 'Charmander', 'small'),
(005,0110, 19000, 'lizard', 'Charmeleon', 'medium');
```

HasAttack

AttackName	id
Pound	035
Thunder Punch	026
Fly	142
Vine Whip	001
Poison Sting	028

```
INSERT INTO Attack
VALUES ('Pound', 'Normal',40),
('Thunder Punch', 'Electric',15),
('Fly', 'Flying',90),
('Vine Whip', 'Grass',45),
('Poison Sting', 'Poison',15);
```

Attack

AttackName	Type	Damage
Pound	Normal	40
Thunder Punch	Electric	15
Fly	Flying	90
Vine Whip	Grass	45
Poison Sting	Poison	15