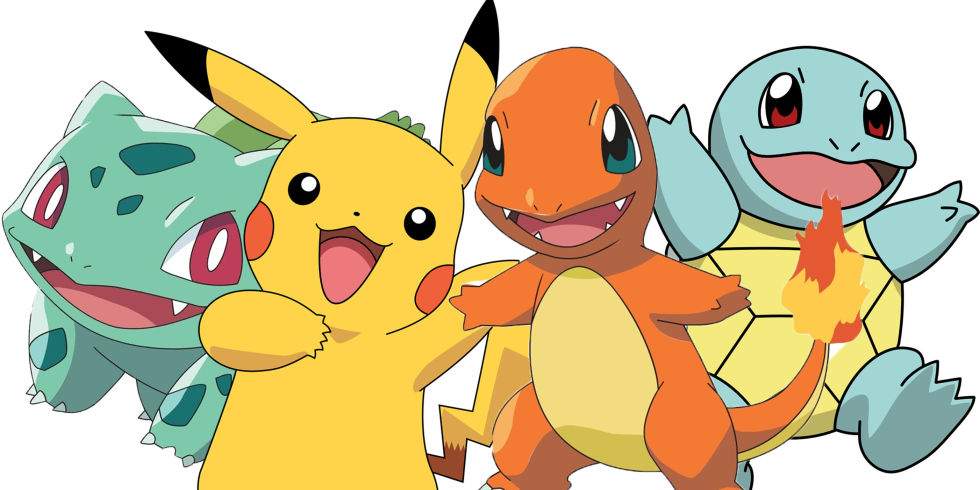
Darwin Huang, [huang7@cooper.edu](mailto:huang7@cooper.edu)

Eric Mendoza-Conner [mendoz@cooper.edu](mailto:mendoz@cooper.edu)

Sebastien Charles [charle5@cooper.edu](mailto:charle5@cooper.edu)

Professor Carl Sable ECE464: Databases

Games/TV Database

**What are Pokemon?**

Pokemon are fictional creatures featured in the eponymous game series and animated TV series. Pokemon species are characterized by a set of base stats (hp, attack, defense, special attack, special defense and speed), by a type combination and by a set of abilities. Pokemon also have a set of moves they can learn (either from a technical machine (TM), a move tutor or from reaching the appropriate in-game level). Pokemon moves are characterized by their type, their strength, their accuracy, and their category (they can be either a physical move, a special move or a status move). There are currently 18 different types. Pokemon can have either one or two types while moves have exactly one type. When a damaging move is used on a pokemon a modifier is applied on the on amount of damage dealt depending on the relation between the damaging move type and the type combination of the attacked pokemon.

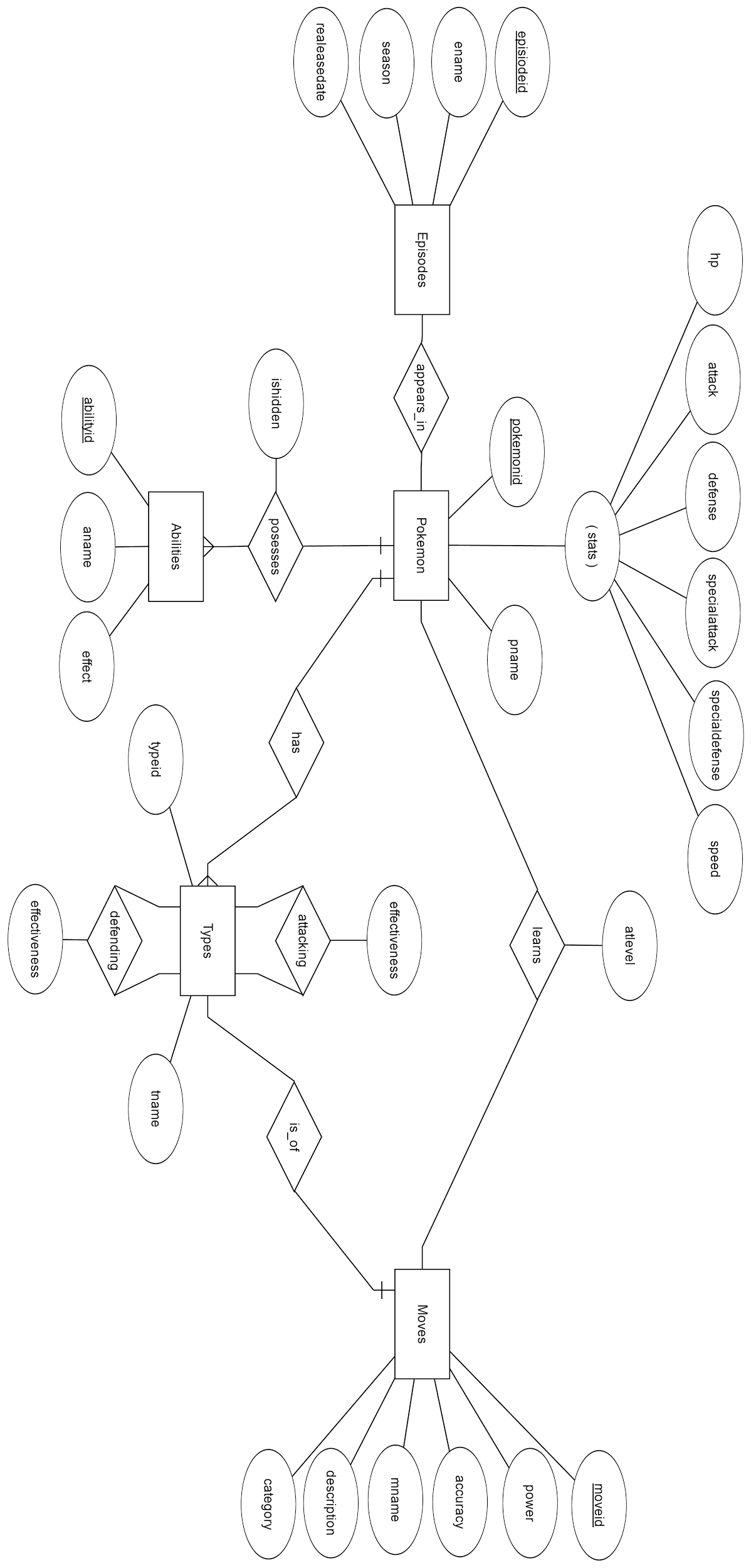
**Changes from previous ER Diagrams**

* Capitalized entities names, “\_” between relation names
* Made entities plural
* Standardized names
* Switched the side on which the “many” symbol appeared
* Added some new attributes to the moves entity set
* Made it so that entity sets only have one primary key

**Legend for the ER diagram**

|  |  |
| --- | --- |
|  | Indicates that the entity can have one or more of this relationship |
|  | Indicates a total participation constraint |
|  | Represents an entity set |
|  | Represents an attribute |
|  | Represents a relationship |
|  | Represents a key attribute |
| composite.png | Composite attribute (a set of attributes) |

**ER Diagrams**



**SCHEMAS**

**entities:**

Pokemon(pokemonid: integer, pname: string, hp: integer, attack: integer,

defense: integer, specialattack: integer, specialdefense: integer,

speed: integer)

Episodes(episodeid: integer, ename: string, season: integer, releasedate:

date)

Types(typeid: integer, tname: string)

Moves(moveid: integer, power: integer, accuracy: integer, mname: string, description: string, category: integer)

Abilities(abilityid: integer, aname: string, effect: string)

**Relations:**

appears\_in(pokemonid: integer, episodeid: integer)

learns(pokemonid: integer, moveid: integer, atlevel: integer)

possesses(pokemonid: integer, abilityid: integer)

has(pokemonid: integer, typeid: integer)

is\_of(moveid: integer, typeid: integer)

attacking(typeid1: integer, typeid2: integer, effectiveness: float)

defending(typeid1: integer, typeid2: integer, effectiveness: float)

**SQL COMMANDS**

**Entities:**

CREATE TABLE Pokemon (

pokemonid INTEGER,

pname CHAR(20),

hp INTEGER,

attack INTEGER,

defense INTEGER,

specialattack INTEGER,

specialdefense INTEGER,

speed INTEGER,

PRIMARY KEY(pname)

);

CREATE TABLE Episodes (

episodeid INTEGER,

ename CHAR(150),

season INTEGER,

releasedate DATE,

PRIMARY KEY(episodeid)

);

CREATE TABLE Types (

typeid INTEGER,

tname CHAR(10),

PRIMARY KEY(typeid)

);

CREATE TABLE Moves (

moveid INTEGER,

mname CHAR(20),

description CHAR(200),

category INTEGER,

power INTEGER,

accuracy INTEGER,

PRIMARY KEY (moveid)

);

CREATE TABLE Abilities (

abilityid INTEGER,

aname CHAR(20),

effect CHAR(200),

PRIMARY KEY (abilityid)

);

**Relations:**

CREATE TABLE appears\_in (

pokemonid INTEGER,

episodeid INTEGER,

PRIMARY KEY(pokemonid, episodeid)

FOREIGN KEY (pokemonid) REFERENCES Pokemon,

FOREIGN KEY (episodeid) REFERENCES Episodes

);

CREATE TABLE learns (

pokemonid INTEGER,

moveid INTEGER,

atlevel INTEGER,

PRIMARY KEY (pokemonid, moveid)

FOREIGN KEY (pokemonid) REFERENCES Pokemon,

FOREIGN KEY (moveid) REFERENCES Moves

);

CREATE TABLE possesses (

pokemonid INTEGER,

abilityid INTEGER,

PRIMARY KEY (pokemonid, abilityid)

FOREIGN KEY (pokemonid) REFERENCES Pokemon,

FOREIGN KEY (abilityid) REFERENCES Moves

);

CREATE TABLE has (

pokemonid INTEGER,

typeid INTEGER,

PRIMARY KEY(pokemonid, typeid)

FOREIGN KEY (pokemonid) REFERENCES Pokemon,

FOREIGN KEY (typeid) REFERENCES Types

);

CREATE TABLE is\_of (

moveid INTEGER,

typeid INTEGER,

effectiveness REAL,

PRIMARY KEY(moveid, typeid),

FOREIGN KEY (moveid) REFERENCES Moves,

FOREIGN KEY (typeid) REFERENCES Types

);

CREATE TABLE attacking (

typeid1 INTEGER,

typeid2 INTEGER,

PRIMARY KEY(typeid1, typeid2)

FOREIGN KEY (typeid1) REFERENCES Types(typeid),

FOREIGN KEY (typeid2) REFERENCES Types(typeid)

);

CREATE TABLE defending (

typeid1 INTEGER,

typeid2 INTEGER,

effectiveness REAL,

PRIMARY KEY(typeid1, typeid2),

FOREIGN KEY (typeid1) REFERENCES Types(typeid),

FOREIGN KEY (typeid2) REFERENCES Types(typeid)

);