Project Step 2 Draft Version: ERD & Schema

By: Kyle Yoneshige, Stuart Kyawmyint
CS 340 - Introduction to Databases

Project Outline Updates

No feedback on project outline was provided this week. No updates were made to the below project outline.

Project Outline

We will be making a database that represents the world of Pokémon. Pokémon, or Pocket Monsters, is a world in which humans capture and train Pokémon to battle another Pokémon. The ultimate goal of the trainer is to have powerful rare Pokémon to become the Pokémon champion. This entails catching as many Pokémon as possible, raising Pokémon's levels and creating a well-balanced team to beat five of the best trainers in the region.

Pokémon was created by Game Freak and Nintendo in 1995. It was originally released in Japan with two games for the Game Boy: Pokémon Red and Pokémon Green. The Pokémon franchise has grown into one of the most recognizable icons and multibillion-dollar franchise with consistent game releases, anime, trading cards, and an assortment of toys and knick knacks for people to own.

Due to the vast nature of trainers and a variety of Pokémon to capture, this franchise is a great candidate for the database project. There are a lot of entities, attributes, and relationships to observe in the wild world of Pokémon.

Database Outline, in Words

The ***entities*** in the database are as follows:

- **Trainer** -- The Trainer entity is the most important entity in this database as it encompasses most of the other entities in this database.
  - **id:** This number is automatically assigned to each Trainer when they are added and recorded to the database. The primary key is an auto-incrementing number.
  - **first_name:** This is the first name of the trainer which is a string of maximum 50 characters. It cannot be blank and there is no default.
  - **last_name:** This is the last name of the trainer which is a string of maximum 50 characters. It cannot be blank and there is no default.

    - **age:** This is the age of a trainer that will be an integer value from 10 to 100. It cannot be blank and the default value is 10.
    - **gender:** This is the gender of the trainer, which is a string of maximum 6 characters and it can either be one of two values: “Male” or “Female”. It cannot be blank and the default gender is “Male”.
    - **hometown**: This is the hometown that the trainer is originally from. It cannot be blank since the trainer will always have a hometown associated with it. This will contain the id of the Town to which the trainer is home to. A character cannot belong to a town which is not in the database.
- **Pokemon** -- These are the different pokemon that are in the world that a trainer can catch. It is recorded in their “Pokedex” using the Pokemon entity.
    - **pokemon_number:** This is the integer number that is assigned to each pokemon. The number corresponds to the Pokedex number that is associated with the pokemon. The number cannot be negative or zero and the maximum is 807. There cannot be any duplicate numbers. This is the primary key that uses the candidate key.
    - **pokemon_name:** This is the name of the pokemon which is a string of maximum 50 characters. It cannot be blank and there is no default. There cannot be any duplicates of pokemon_name in the database.
    - **main_type:** This is the main type that is associated with the pokemon. This will contain the id of the Type to which the pokemon has as a property. It cannot be blank since the pokemon will always have a main type associated to it. There is no default type.
    - **secondary_type:** This is the secondary type that is associated with the pokemon. It can be blank since the pokemon may not have a secondary type. Also, the secondary type cannot be the same as the main type. This will contain the id of a Type. Default is blank.
- **Type** -- These are the different elemental properties for pokemon using the Type entity. There are currently 18 different elemental types in the show/game.
    - **id:** This number is automatically assigned to each Type when they are added and recorded to the database. The primary key is an auto-incrementing number.
    - **type_name:** This is the name of the element type which is a string of maximum 50 characters. It cannot be blank and there is no default. There cannot be any duplicates of type_name.
- **Badge --** These are the different badges that a trainer can own (by defeating a gym leader) in the world of Pokémon using the Badge entity.
    - **id:** This number is automatically assigned to each Badge when they are added and recorded to the database. The primary key is an auto-incrementing number.

    - **badge_name:** This is the name of the badge which is a string of maximum 50 characters. It cannot be blank and there is no default. There cannot be any duplicates of badge_name.
    - **gym_name:** This is the name of the gym that the badge comes from, which is a string of maximum 50 characters. It cannot be blank and there is no default. There cannot be any duplicates of gym_name.
    - **badge_town**: This is the town at which the badge can be obtained from. It cannot be blank since the badge will always have a town associated with it. This will contain the id of the Town to which the badge can be acquired from. A badge cannot belong to a town which is not in our database.
    - **badge_type:** This is the type that is associated with the badge. This will contain the id of the Type to which the badge belongs to. It cannot be blank since the badge will always have a type associated to it.
- **Town --** These are the different towns that are in the world of Pokémon using the Town entity.
    - **id:** This number is automatically assigned to each Town when they are added and recorded to the database. The primary key is an auto-incrementing number.
    - **town_name:** This is the name of the town which is a string of maximum 50 characters. It cannot be blank and there is no default. There cannot be any duplicate town names.
    - **region_name:** This is the name of a region where the town is in, which is a string of maximum 50 characters. It cannot be blank and there is no default.

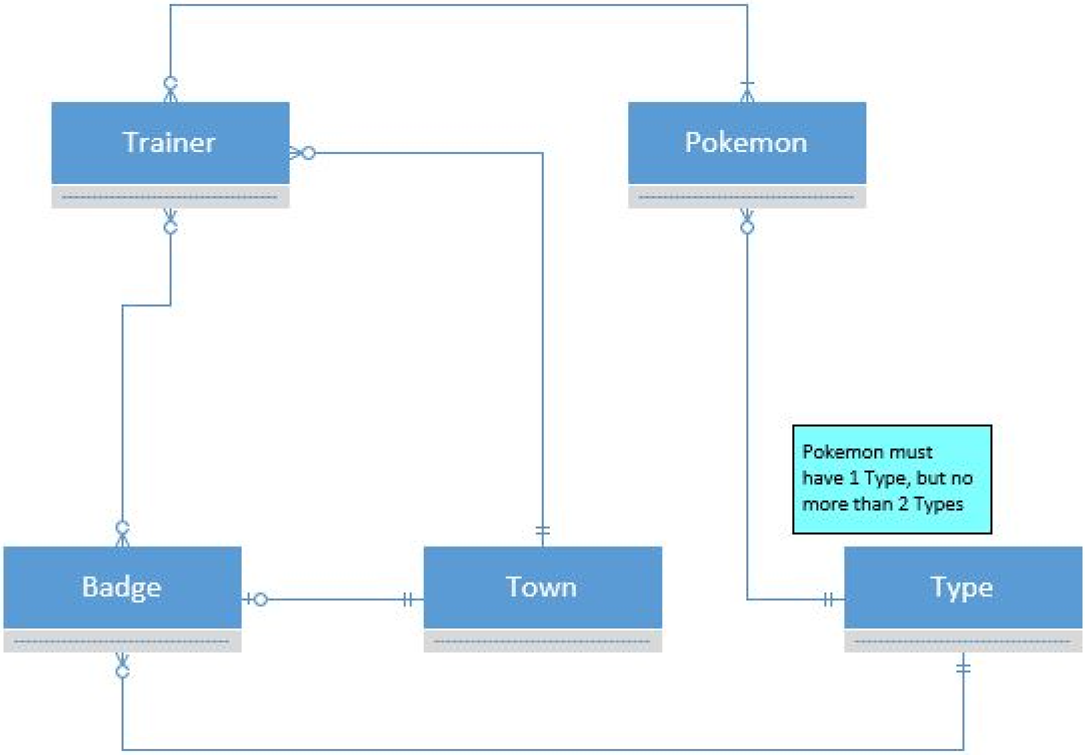
The ***relationships*** in the database are as follows:

- **Trainers are from towns** - A trainer can only be from one town, but a town can be the hometown to many trainers. Thus, the Trainer and Town entities are in a *one-to-many relationship*.
- **Trainers have pokemon** - A trainer can have many pokemon and many trainers can have the same pokemon (using the idea of a Pokedex from the series). Thus, the Trainer and Pokemon entities are in a *many-to-many relationship*.
- **Trainers have badges** - A trainer can have many badges and many trainers can have the same badges. The badges are much like certifications. Thus, the Trainer and Badge entities are in a *many-to-many relationship*.
- **Pokemon have elemental types** - A pokemon can only have one or two elemental types, but a type can be a property of many pokemon. Thus, the Pokemon and Type entities are in a *one-to-many relationship.*
- **Badges are acquired from towns** - A badge can only be acquired from one town and a town can only have one badge that can be acquired from it. However, it is

possible that a town may not have a badge that can be acquired. Thus, the Badge and Town entities are in a “special case” *one-to-one relationship*.

- **Badges have a type -** A badge can only have one type associated to it, but a type can be a property of many different badges. Thus, the Badge and Type entities are in a *one-to-many relationship.*

Entity-Relationship Diagram (ERD)

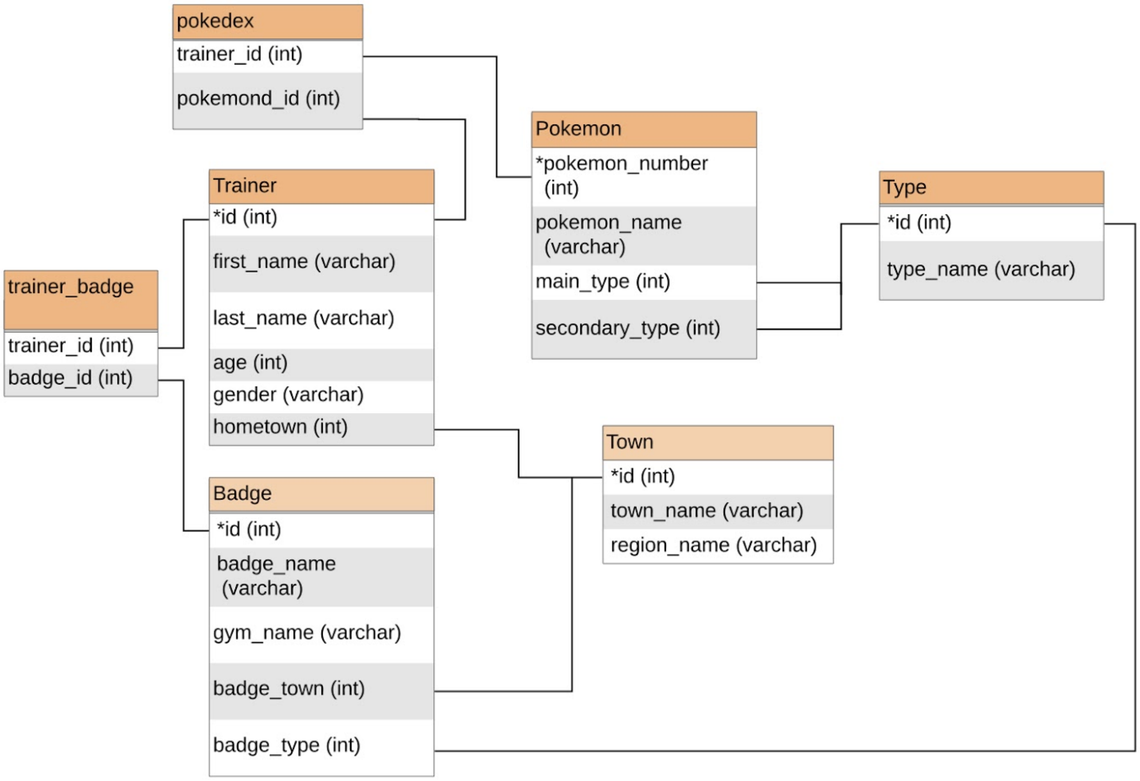


ENTITY RELATIONSHIP DIAGRAM

Schema

Project Step 2: Pokemon Schema

Group 29: Kyle Yoneshige & Stuart Kyawmyint | April 20, 2019



pokedex
trainer_id (int)
pokemond_id (int)

Pokemon
*pokemon_number (int)
pokemon_name (varchar)
main_type (int)
secondary_type (int)

Type
*id (int)
type_name (varchar)

Trainer
*id (int)
first_name (varchar)
last_name (varchar)
age (int)
gender (varchar)
hometown (int)

trainer_badge
trainer_id (int)
badge_id (int)

Town
*id (int)
town_name (varchar)
region_name (varchar)

Badge
*id (int)
badge_name (varchar)
gym_name (varchar)
badge_town (int)
badge_type (int)