

**ER Model Report:**

**Trainer:** As specified, the trainer would be the main entity in this ERD, with the given attributes & would have everything relating to it. Team and Level have their own separate entities and relationships with the trainer as they would have unique attributes and requirements of their own. Every level has a required XP, which would need to be stored separately. Also, although Pokémon Go currently just has three teams in their database system, if any additional teams do need to be added, changes would just need to be made to one entity and not others. Team has an optional one to optional many relationship with trainer, as a team can contain 0 or more trainers, and a trainer can belong to 0 or 1 team (depends on the level they are at). Level would have a mandatory one to optional many relationship with trainer, as a level has 0 or more trainers (specially the higher levels in the game) and a trainer can have just one level at a particular time/XP level.

**Item & Egg:** I have created an associative entity between trainer and item to resolve the many to many relationship between them. A trainer can own 1 or more items (trainer has an infinite egg incubator right at the start of the game) and a particular item can be owned by 0 or more trainers. From this associative entity, as it contains the common attributes of a TrainerItem, I have created an exclusive superset-subset relationship to separately identify each of the type of items. An egg is an item with its own required travel distance, and an incubator has multiple types of its own with completely different functionalities, and has a particular relationship to an egg. An egg needs to be placed in an incubator for it to hatch, which gives a mandatory one to optional many relationship from incubator to egg. An incubator can hatch 0 or more eggs (infinite incubator can hatch many / an incubator may not be used to hatch any egg).

**Pokémon & Evolution:** Pokémon as an entity, with its attributes as specified in the assignment is related to a trainer with an optional one to many relationship, as a trainer can own 0 or more Pokémon, & a particular instance of a Pokémon can be owned by one trainer or none (Wild Pokémon). Even so, species can include one or more Pokémon, but a Pokémon can only belong to one species. Then a Pokémon species itself can have one or two types, due to which I’ve created an associative entity in order to resolve the mandatory many to many relationship between both entities. Moreover, I’ve modelled evolution with a unary relationship from species to itself, without having to create a separate entity, as that only duplicated data. One species evolves into another, and the optional one to many indicates that not all species evolve, and one species can evolve into one or more species.

**Encounter:** As a Pokémon can be encountered by 0 or multiple trainers & a trainer can encounter 0 or multiple Pokémon, encounter has been made a separate entity with its own attributes & acts as an associative entity as well. In addition, I haven’t modelled the different outcomes and gains as rewards, as they’re only functional requirements.

**Location / PokeStop & Gym / Battle:** As a trainer can have one or more locations, and a location can have 0 or more trainers present at that place, this gives us a many to many relationship, creating an associative entity. This entity, TrainerLocation includes the GPS coordinates, the trainerID, as well as the time visited, so that the location for every second is recorded. Location is then made a superset with the fixed location type, PokeStop & Gym as subsets, to create an exclusive relationship as a fixed location can be either one or the other. Gym is then related to Pokémon, with GymOccupant as an associative entity with its start and end time, to indicate when the gym is empty or occupied and the teams occupying at that time. Battle then again acts as an associative entity as well.