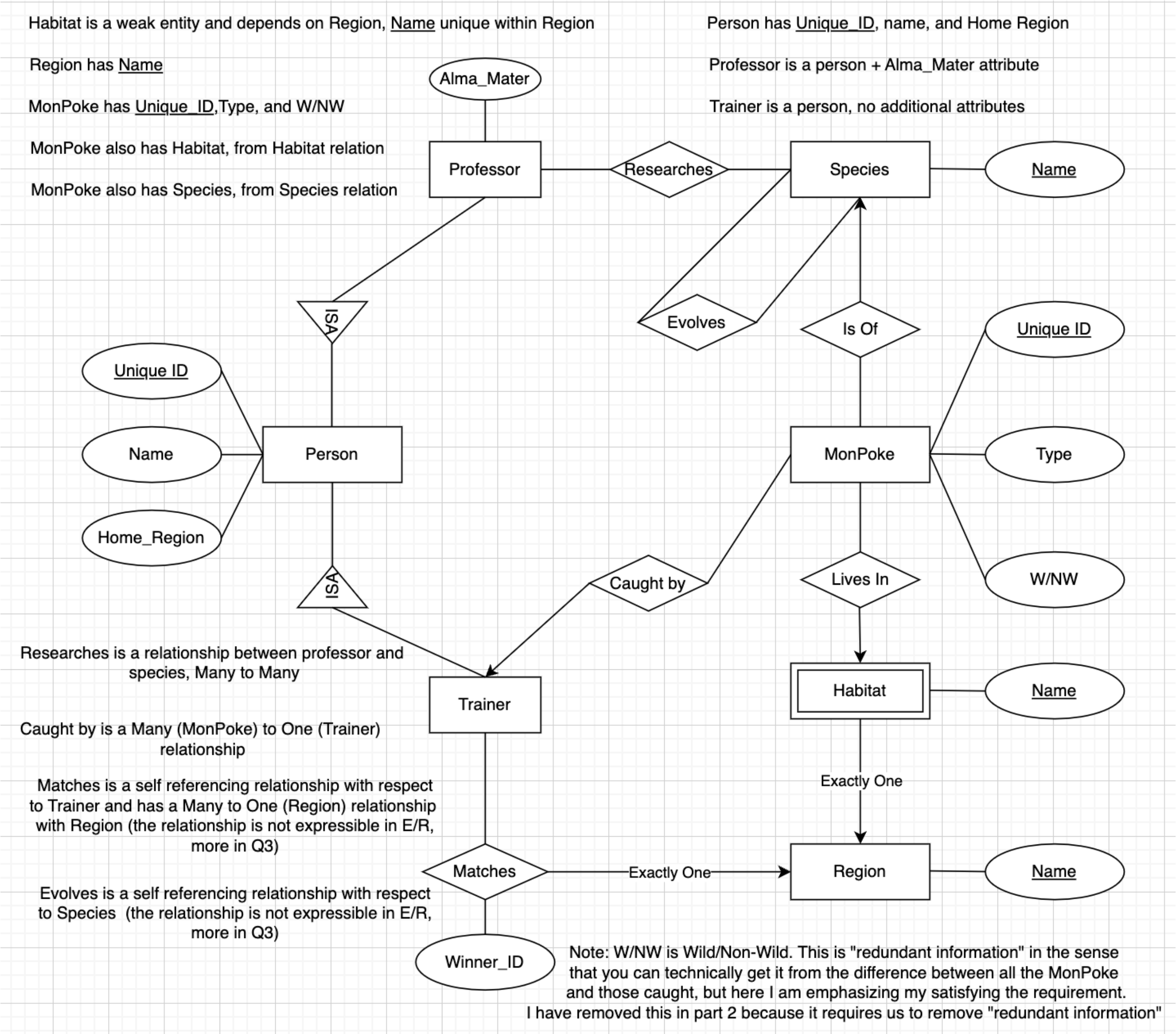

Habitat is a weak entity and depends on Region, Name unique within Region

Region has Name

MonPoke has Unique_ID,Type, and W/NW

MonPoke also has Habitat, from Habitat relation

MonPoke also has Species, from Species relation

Person has Unique_ID, name, and Home Region

Professor is a person + Alma_Mater attribute

Trainer is a person, no additional attributes

Researches is a relationship between professor and species, Many to Many

Caught by is a Many (MonPoke) to One (Trainer) relationship

Matches is a self referencing relationship with respect to Trainer and has a Many to One (Region) relationship with Region (the relationship is not expressible in E/R, more in Q3)

Evolves is a self referencing relationship with respect to Species (the relationship is not expressible in E/R, more in Q3)

Note: W/NW is Wild/Non-Wild. This is "redundant information" in the sense that you can technically get it from the difference between all the MonPoke and those caught, but here I am emphasizing my satisfying the requirement. I have removed this in part 2 because it requires us to remove "redundant information"

Professor(Unique_ID, Name, Home_Region, Alma_Mater)

Trainer(Unique_ID, Name, Home_Region)

Habitat(Habitat_Name, Region.Region_Name)

Region(Region_Name)

MonPoke(Unique_ID, Type, Region.Region_Name, Habitat.Habitat_Name, Species.Evolved_Species)

Species(Evolved_Species, Initial_Species)

Matches(Trainer.Trainer_ID1, Trainer.Trainer_ID2, Winner_ID, Region.Region_Name)

Researches(Professor.Unique_ID, Species.Evolved_Species)

Caught_By(MonPoke.Unique_ID, Trainer.Unique_ID)

Note: If a species cannot evolve, its initial species is null. All species are listed once and can be chained.

Logical Constraint 1: Each pair of trainers can only battle once

This means that “matches” is a relationship that is self-referencing with respect to the Trainer entity. This, in of itself can be represented in an E/R diagram. However, there is no way to represent the uniqueness of a PAIR of trainers. None of the relationships (Many to Many, Many One, and One One) captures that. Thus, we defer this constraint to DBMS, setting the pair of unique IDs as the primary key of the relationship.

Logical Constraint 2: Each species has 0, 1, or more choices for evolution, but each evolved species has only one origin species

This means that “evolution” is a relationship that is self-referencing with respect to the Species entity. Again, this can be represented in an E/R diagram. However, this is simultaneously a One One relationship with respect to the evolved species, and a Many One relationship with respect to the origin species. Thus, it is impossible to express this constraint. Thus, we make evolved species the primary key.