Pokémon Relational Database Project Plan

University of Manitoba
COMP 3380 - A02

Group #68 Members: Alyssa Gregorash, Henry Wong, Luc Miron
Part 1 Due: Oct. 16, 2022

COMP 3380 – Database Project part 1

Group Members: Alyssa Gregorash, Henry Wong, Luc Miron
Due Oct. 16, 2022

About the Dataset

The dataset used in this project contains information about Pokémon, monsters in the (primarily video game) franchise called Pokémon. All data in this set is made from or derived from the Pokémon main series games, including information from generations 1-8 inclusive. Generation 9 and side games' content is not included, because generation 9 is not released fully and side games may include contradictory information to the main series.

The dataset contains data for 905 Pokémon plus over a hundred forms and abilities, hundreds of moves and items and dozens of statuses.

The dataset itself is provided by Pokémon Essentials, a fan-made tool to allow anyone to make a Pokémon fan game. This dataset (and Pokémon Essentials itself) is not endorsed or supported by Nintendo and this data belongs to Nintendo and the Pokémon company. Some supplementary information was taken from Bulbapedia, a Pokémon wiki as Pokémon Essentials had a few definitions (such as some statuses) buried deep in its code.

There are some game-specific or engine-specific things in this dataset that are not included in our model, such as move flags and function codes.

The data is centered around defining a Pokémon and all its traits. It defines a species of Pokémon, rather than an instance. For example, Pikachu refers to every Pikachu, not specifically Gary's Pikachu.

General statement about primary keys in this set: Except for the Pokémon themselves which use form IDs and dex numbers, most entities use their name as a key. This is because this dataset came using internal names for most references. These names are enforced as unique.

Entity Definitions

Pokémon

Pokémon refers to a species of Pokémon. Every Pokémon has a national Pokedex number, name, hit points(hp), attack(atk), defense(def), special attack(satk), special defense(sdef), speed(sped), generation number, form ID and Pokedex entry (description).

The *Primary key* of a Pokémon is its national dex number and form ID. This is because forms can range from anything from a changed appearance to being effectively a separate Pokémon with its own stats and relations. A Pokémon always has a base form with form ID 0.

In relational model: Pokémon (pokedex_number, form_id, name, p_generation, p_hp, p_atk, p_def, p_satk, p_sdef, p_sped, p_description)

Abilities

Abilities are passive traits a Pokémon has. A Pokémon usually has between one and three abilities. A different form of a Pokémon might have a different ability pool than its base form. An ability has a name and a description. The name is unique and is the primary key.

In relational model: Abilities(a_name, a_description)

EggGroup

Every Pokémon is part of one or more egg groups, even if that egg group is 'undiscovered'. An egg group has a unique name, which is its primary key.

In relational model: EggGroup(e_name)

Type

Type is an elemental property. Its name is unique, and its primary key.

In relational model: Type(t_name)

Moves

Moves are actions a Pokémon can do, such as attack or apply status. Its name is unique and is the primary key.

In relational model: Moves(m_name, m_powerpoints, m_category, m_accuracy, m_range, m_description)

Items

Items are objects that can influence a Pokémon. Items can do many things, such as changing a status or cause a Pokémon to evolve. The primary key is its name, which is enforced as unique.

In relational model: Items(I_name, I_description)

Status

Status refers to conditions that might afflict a Pokémon, such as 'sleep', 'poison' or 'confusion'. It also includes states such as 'safeguarded' or 'aqua ring'. Status is identified by an internal name. The volatile property refers to whether the status disappears when a Pokémon switches in battle. Duration is a descriptive string, such as "2-3 turns" or "Until cured".

In relational model: Status(s_name, s_description, s_volatile, s_duration)

Relations and Participation

Pokémon_Type

Every Pokémon has one or two types. Different forms of a Pokémon might have different types. For example, 'Ninetales' has the fire type. Another form, 'Alolan Ninetales', has the 'ice' type and the 'fairy' type. A Pokémon must have at least one type, hence the total participation. The type itself only has partial participation in this relation, since not all types belong to a Pokémon. For example, there is/was a type '???' or 'Unknown' which did not belong to any Pokémon.

In relational model: Pokemon_Type(pokedex_number, form_id, t_name)

Pokémon_Egg

Every Pokémon is in one or more egg groups, with 'Undiscovered' still being registered as its own egg group. Many Pokémon are part of one egg group, and egg groups have many Pokémon. A Pokémon's form does not change what egg group it is in.

In relational model: Pokemon_Egg(pokedex_number, e_name)

Pokémon_Evolution

Some Pokémon evolve into others. One Pokémon can potentially evolve into many different Pokémon, such as 'Eevee', which can evolve into any of eight other Pokémon. Form must be passed along with this relation, since an 'Alolan Sandshrew' needs to know whether to evolve into 'Sandslash' or 'Alolan Sandslash', both of which have the same national dex number. Some Pokémon do not evolve.

In relational model: Pokemon_Evolution(pre_evo, form_id, post_evo)

Pokémon_Item_Evolution

One item can cause many Pokémon to evolve.

In relational model: Pokemon_Item_Evolution(pre_pokedex_number, form_id, I_name, post_pokedex_number)

Pokémon_Form

Some Pokémon change between their forms. One Pokémon might turn into many forms, such as 'Kyurem' turning into 'Kyurem-white' or 'Kyurem-black'. Unlike evolution, the pokedex number stays the same before and after. Most form changes are reversible, whereas evolution is not. Not all Pokémon change form.

In relational model: Pokemon_Form(pokedex_number, pre_form_id, post_form_id)

Pokémon_Item_Form_Change

Some Pokémon change into another form when exposed with an item. One Pokémon might turn into any of a few forms. Example: a 'DNA splicer' can be used to turn 'Kyurem' into 'Kyurem-white' or 'Kyurem-black'. Unlike evolution, the pokedex number stays the same before and after.

In relational model: Pokemon_Item_Form_Change(pokedex_number, pre_form_id, post_form_id, I_name)

Pokémon_Ability

All Pokémon need at least one ability (total participation). Some unused abilities do not belong to any Pokémon (partial participation). Pokémon can have many potential abilities, and abilities are not (usually) unique to one Pokémon.

In relational model: Pokemon_Abilities(pokedex_number, form_id, a_name)

Pokémon_Moves

All Pokémon learn moves. Almost all moves are learned by Pokémon, but this isn't the case for some unused moves like 'Light of Ruin', hence partial participation. The moves a Pokémon learns can vary by form.

In relational model: Pokemon_Moves(pokedex_number, form_id, m_name)

Item_Status_Effects

Some items affect status, such as a 'burn heal' removing 'burn' and a 'toxic orb' applying 'poisoned'.

In relational model: Item_Status_Effects(I_name, s_name)

Item_Move_Teach

One item can teach one move. A move is taught by one item. Not all moves are teachable by items, and not all items teach moves.

In relational model: Item_Move_Teach(m_name, I_name)

Move_Type

A move has one type. A type has many moves. Not all moves have a type though, as there is an exception in the move 'Struggle', which does not have a type.

In relational model: Move_Type(m_name, t_name)

Move_Status_Effect

A move may have a chance to inflict one or more status effects. For example, the move 'thunder fang' has a chance to inflict 'flinch' and a chance to inflict 'paralysis'.

In relational model: Move_Status_Effects(m_name, s_name, chance)

Type_Resist

One type may resist many other types.

In relational model: Type_Resist (resist_defender, resist_attacker)

Type_Immune

One type may be immune to many other types.

In relational model: Type_Immunity (immune_defender, immune_attacker)

Type_Weakness

One type may be weak to many other types.

In relational model: Type_Weakness (weakness_defender, weakness_attacker)

Type_Status_Immunity

One type may be immune to multiple status conditions. For example, 'Fire' type cannot be affected by 'burn'. 'Ghost' type cannot be affected by 'bind' or 'arena trapped'. Not all types are immune to a status condition, such as the 'water' type.

In relational model: Type_Status_immunity(t_name, s_name)

Relational Model Summary

Pokemon(pokedex_number, form_id, name, p_generation, p_hp, p_atk, p_def, p_satk, p_sdef, p_sped, p_description)

Abilities(a_name, a_description)

EggGroup(e_name)

Type(t_name)

Moves(m_name, m_powerpoints, m_category, m_accuracy, m_range, m_description)

Items(I_name, I_description)

Status(s_name, s_description, s_volatile, s_duration)

Pokemon_Type(pokedex_number, form_id, t_name)
'pokedex_number' and 'form_id' references 'Pokemon'
't_name' references 'Type'

Pokemon_Evolution(pre_evo, form_id, post_evo)
'pre_evo', 'post_evo', and 'form_id' references 'Pokemon'

Pokemon_Egg(pokedex_number, e_name)
'pokedex_number' references 'Pokemon'
'e_name' references 'Egg_Group'

Pokemon_Abilities(pokedex_number, form_id, a_name)
'pokedex_number' and 'form_id' references 'Pokemon'
'a_name' references 'Ability'

Pokemon_Moves(pokedex_number, form_id, m_name)
'pokedex_number' and 'form_id' references 'Pokemon'
'm_name' references 'Moves'

Pokemon_Form(pokedex_number, pre_form_id, post_form_id)
'pokedex_number', 'pre_form_id', and 'post_form_id' references 'Pokemon'

Pokemon_Item_Form_Change(pokedex_number, pre_form_id, post_form_id, I_name)
'pokedex_number', 'pre_form_id', and 'post_form_id' references 'Pokemon'
'i_name' references 'Items'

Pokemon_Item_Evolution(pre_pokedex_number, form_id, I_name, post_pokedex_number)
'pre_pokedex_number', 'form_id', and 'post_pokedex_number' references 'Pokemon'
'i_name' references 'Item'

Item_Status_Effects(I_name, s_name)
‘i_name’ references ‘Items’
‘s_name’ references ‘Status’

Item_Move_Teach(m_name, I_name)
‘m_name’ references ‘Moves’
‘i_name’ references ‘Items’

Move_Type(m_name, t_name)
‘m_name’ references ‘Moves’
‘t_name’ references ‘Type’

Move_Status_Effects(m_name, s_name, chance)
‘m_name’ references ‘Moves’
‘s_name’ references ‘Status’

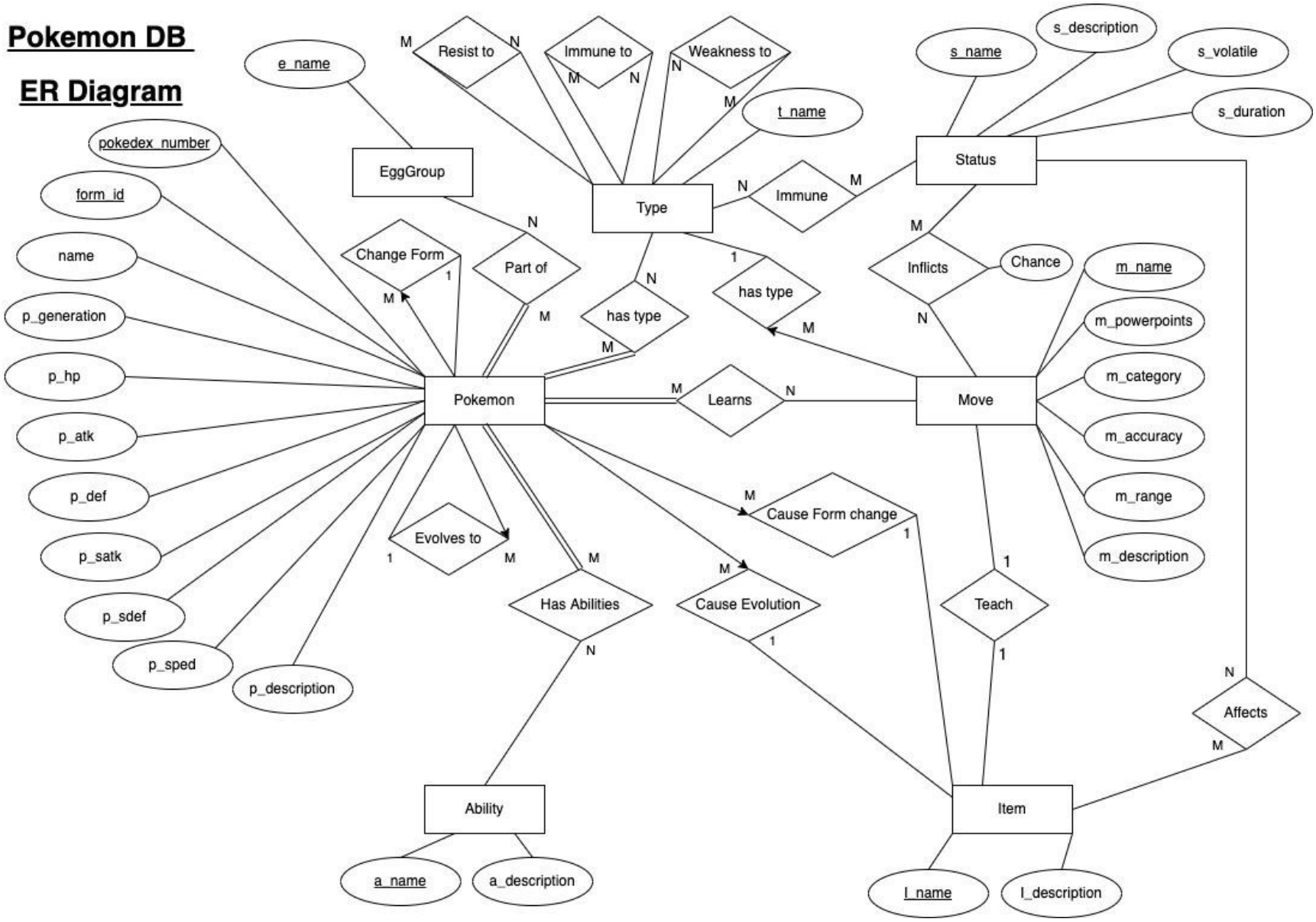
Type_Resist(resist_defender, resist_attacker)
‘resist_defender’ and ‘resist_attacker’ references ‘Type’

Type_Immunity(immune_defender, immune_attacker)
‘immune_defender’ and ‘immune_attacker’ references ‘Type’

Type_Weakness(weakness_defender, weakness_attacker)
‘weakness_defender’ and ‘weakness_attacker’ references ‘Type’

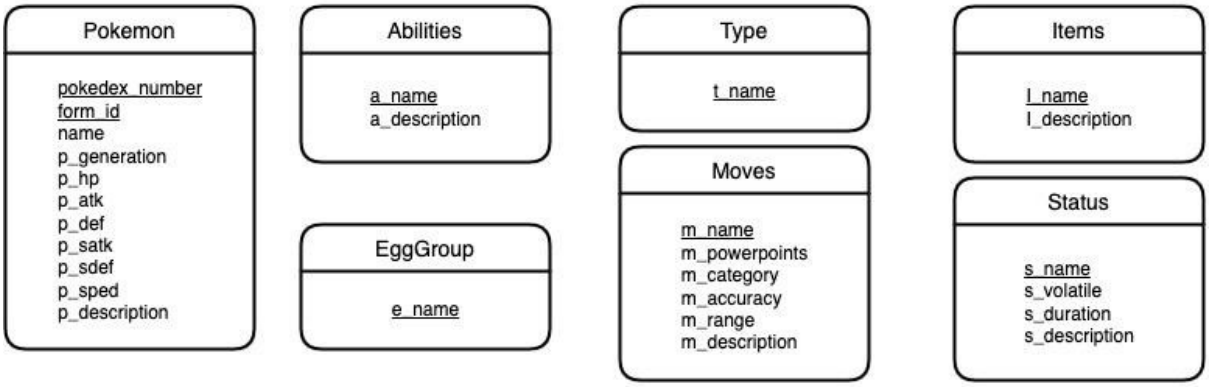
Type_Status_Immunity(t_name, s_name)
‘t_name’ references ‘Type’
‘s_name’ references ‘Status’

Annex A (ER Diagram)



Annex B (ER Tables)

Entities



Relations