PokmonDB

Dead Tree Edition

Addison “Maddie” Morrison
Devon Schimming
Jacob Hoggatt
Romesh Jayathilake

11 April 2019

Contents

1 Phase 1

1.1 Purpose

The purpose of this database is to make information about different pokemon accross multiple titles easily avaliable. This also includes media such as fan-art and in game assets. The user should be able to query a specific pokemon's attributes in a certain game, and on its designated platform.

1.2 Entity Descriptions

1.2.1 Platforms

The Platforms entity serves to convey which Nintendo platform the Pokmon game appeared on.

- This Entity's only attribute is the name of the platform.

1.2.2 Games

The Games entity describes the publication of the game.

- The Name attribute is the title of the game
- The ReleaseDate attribute is the game's initial release date
- The Platform attribute refers to the platform for which the game was initially developed.

1.2.3 PokemonApperance

The PokemonAppearance entity contains the data belonging to the Pokmon appearing in a certain game. Each game can contain different Pokmon, and Pokmon can appear in more than one game.

- The ID attribute is the Pokmon's designated number
- The BaseHp attribute is the base health points stat the Pokmon begins with
- The BaseSpeed attribute is the Pokmon's initial speed stat
- The BaseAttack attribute is the Pokmon's initial attack stat

- The BaseDefence attribute is the Pokmon's initial defense stat
- The BaseSpAttack attribute is the Pokmon's initial special attack stat
- The BaseSpDefence attribute is the Pokmon's initial special defense stat
- The Pokemon attribute is a package containing Pokemon, it is the many end of the relation
- The Game attribute is a package containing Game, it is the many end of the relation

1.2.4 PokemonCanLearn

The PokemonCanLearn entity coveys which abilities, at a certain level, a Pokmon is capable of learning.

- The AtLevel attribute states the level at which an attack can be learned
- The Appearance attribute is a package containing PokemonAppearance, it is the many end of the relation
- The Attack attribute is a package containing Attacks, it is the many end of the relation

1.2.5 Attacks

The Attacks entity describes the attack abilities which Pokmon can use.

- The Name attribute is the name of the attack
- The Priority attribute value decides who attacks first
- The Accuracy attribute is the likelihood of it hitting the opponent
- The TMNumber attribute is an attack's id when on a technical machine in the game
- The PP attribute refers to power points, which are the cost to use the attack
- The Power attribute is the damage the attack can do.

1.2.6 AttackTypes

The AttackTypes entity contains the details of what Pokmon type an attack is considered.

- The Attack attribute is a package containing Attacks, it is the many end of the relation
- The Type attribute is a package containing Types, it is the many end of the relation

1.2.7 Types

The Types entity contains the name of the type. Currently in Pokmon there are normal, fighting, flying, poison, ground, rock, bug, ghost, steel, fire, water, grass, electric, psychic, ice, dragon, dark, and fairy types. These types describe what are considered to be "elemental" properties.

- The only attribute is Name which states the name of the type

1.2.8 TypeMatchups

The TypeMatchups entity describes which types are strong against, or are weaker to other types. This entity's attributes are the effectiveness multiplier given to an attack when there is a type advantage, the self-attribute which indicates if the damage is against the player's Pokmon, and the Other attribute which indicates if the damage is against the opponent.

- The EffectivenessMultiplier attribute contains the value that the effectiveness is modified by
- The self attribute contains whether the advantage is with or against the player
- The Other attribute contains whether the advantage is with or against the opponent

1.2.9 Pokmon

The Pokmon entity details an individual Pokmon's characteristics.

- The PokedexID attribute is the Pokmon's ID given by the in game Pokdex tool
- The Name attribute is the Pokmon's name
- The GenderDistribution is the gender of the Pokmon
- The Legendary attribute states whether the Pokmon is considered to be of a legendary variety

1.2.10 PokmonEvolutions

The PokemonEvolutions entity contains the evolution information for a Pokmon.

- The Base attribute is the Pokmon's current form
- The Evolution attribute is the form the Pokmon can become.

1.2.11 Media

The Media entity details the media in which a Pokmon is featured.

- The IntegerId attribute contains the Pokmon Id the media is about
- The FileName attribute contains the name of the file in which the media is stored
- The BLOBdata attribute is the media data stored in the database
- The Pokmon attribute is a package containing Pokemon, it is the many end of the relation
- The MIMEType attribute is a package containing MIMETypes, it is the many end of the relation

1.2.12 MIMETypes

The MIMETypes entity contains the Multipurpose Internet Mail Extension for the database which indicates the format of documents, files, or assorted bytes.

- The MIMEString attribute contains strings used in the format
- The FriendlyName attribute contains a property allowing code to be easier to read
- The Extension attribute contains the extensions

1.3 Relation Descriptions

1.3.1 PlayedOn

The PlayedOn relation is the relation between the Platform and Games entities. Each Platform can play many games, however the original game can only be played on one platform.

- GameName and PlatformName would be the two keys for this relation

1.3.2 GameAppearance

The GameAppearance relation is the relation between the Games and PokemonAppearance entities. Each game can have only one pokemon appearance, however pokemon appear in multiple games.

- GameName and AppearanceId would be the two keys for this relation

1.3.3 AppearanceCanLearn

The AppearanceCanLearn relation is the relation between the PokemonAppearance and PokemonCanLearn entities. Each appearance can have only one minimum level at which it can learn moves, however there are multiple appearances that each have their own minimum levels.

- AppearanceId and LearnAtLevel would be the two keys for this relation

1.3.4 CanLearnAttacks

The CanLearnAttacks relation is the relation between the PokemonCanLearn and Attacks entities. Each level can have multiple attacks available at it, however individual attacks always are learnable at the same level.

- LearnAtLevel and AttackName would be the two keys for this relation

1.3.5 TypeOfAttack

The TypeOfAttack relation is the relation between the Attacks and AttackTypes entities. Each attack has only one type, however multiple attacks are classified as each type.

- AttackName would be the key for the relation as AttackTypes references a foreign key

1.3.6 TypeOf

The TypeOf relation is the relation between the Types and AttackTypes entities. Each type has only one attack type as it corresponds directly with the entity, however attack types can have multiple types.

- TypeName would be the key for the relation as AttackTypes references a foreign key

1.3.7 TypeMatch

The Typematch relation is the relation between the Types and TypeMatchups entities. Each type has two matchups, either against itself, or against another. The matchup however can work against multiple different types.

- TypeEffectivenessMultiplier and Type Name are the two keys for this relation

1.3.8 PokemonApp

The PokemonApp relation is the relation between the Pokemon and PokemonAppearance entities. Each Pokemon has one appearance per game, however each appearance can have multiple Pokemon.

- AppearanceID and PokemonID are the two keys for this relation

1.3.9 EvolvesTo

The EvolvesTo relation is the relation between the Pokemon and PokemonEvolutions entities. Each Pokemon has two states, its current form and its evolved form. Many Pokemon have evolutions.

- PokemonID would be the key for the relation as PokemonEvolutions references a foreign key.

1.3.10 PokemonHasMedia

The PokemonHasMedia relation is the relation between the Pokemon and Media entities. Each Pokemon would have only one media file, while there are multiple Pokemon to have media for.

- MediaID and PokemonId would be the two keys for this relation

1.3.11 FileType

The FileType relation is the relation between the Media and MIMEType entities. Each Media file would contain multiple different MIME formats, and MIME types would be used by multiple different media files.

- MIMEText and Media ID would be the two keys for this relation

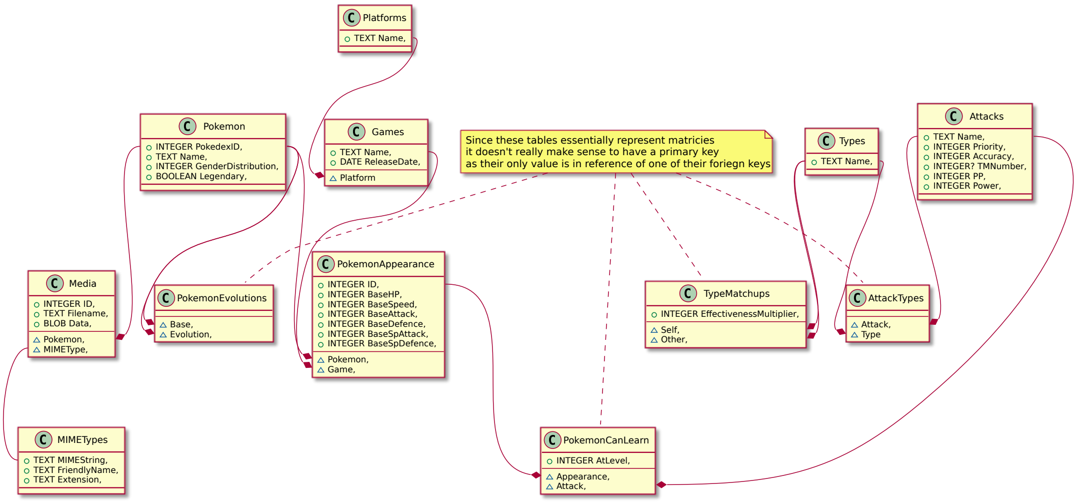
1.4 Information That Could Be Extracted

We expect the user to be able to extract any relevant information about the Pokmon available on a platform. For example, the user could find out the number of Pokmon games available on a console, or what Pokmon are available on a console. The user could quire a Pokmon on the Gameboy and find out what moves it can use, and at which level it can learn them. They could also extract what that Pokmon's base stats are. Using the Media entity, the user could also see if any media files are available for said Pokmon. The user could go so far as to find which move types are advantageous against others. The user could also find the differences in a Pokmon's evolution between titles

1.5 Updates to the Database

A new game would cause many additions to the database, with nearly every entity receiving new entries. This is because Pokmon from previous titles are often brought over into new titles. This would mean that all of the entities would receive those duplicate Pokmon, with possible updates to their stats and moves. The Media and MIMEType entities are less likely to see an immediate update as they require people to create media for the new Pokmon. The Media and MIMEType could also be modified but changes to these entities would not create changes in the rest of the database.

1.6 UML Diagram



Since these tables essentially represent matricies
it doesn't really make sense to have a primary key
as their only value is in reference of one of their foriegn keys

1.7 Schemas

- Platforms(**Name**:TEXT)
- Games(**GameName**:TEXT, *PlatformName*:TEXT, ReleaseDate:DATE)
- PokemonAppearance(**ID**:INTEGER, *GameName*:TEXT, *PokedexID*:INTEGER, BaseHP:INTEGER, BaseSpeed:INTEGER, BaseAttack:INTEGER, BaseDefence:INTEGER, BaseSpAttack:INTEGER, BaseSpDefence:INTEGER)
- Pokemon(**PokedexID**:INTEGER, Name:TEXT, GenderDistribution:INTEGER, Legendary:BOOLEAN)
- Media(**ID**:INTEGER, *PokedexID*:INTEGER, *MIMEString*:TEXT, Filename:TEXT, Data:BLOB)
- MIMETypes(**MIMEString**:TEXT, FriendlyName:TEXT, Extension:TEXT)
- Types(**Name**:TEXT)
- Attacks(**Name**:TEXT, Priority:INTEGER, Accuracy:INTEGER, TMNumber:INTEGER?, PP:INTEGER, Power:INTEGER)

1.7.1 Relationship-Only Tables

- AttackTypes(*AttackName*:TEXT, *TypeName*:TEXT)
- PokemonEvolutions(*PokemonBaseID*:INTEGER, *PokemonEvolutionID*:INTEGER)

- TypesMatchups(*SelfName*:TEXT, *OtherName*:TEXT , EffectivenessMultiplier:INTEGER)

- PokemonCanLearn(*PokemonID*:INTEGER, *AttackName*:TEXT, AtLevel:INTEGER)