Pokémon MoveDex

By

Brandon Kline

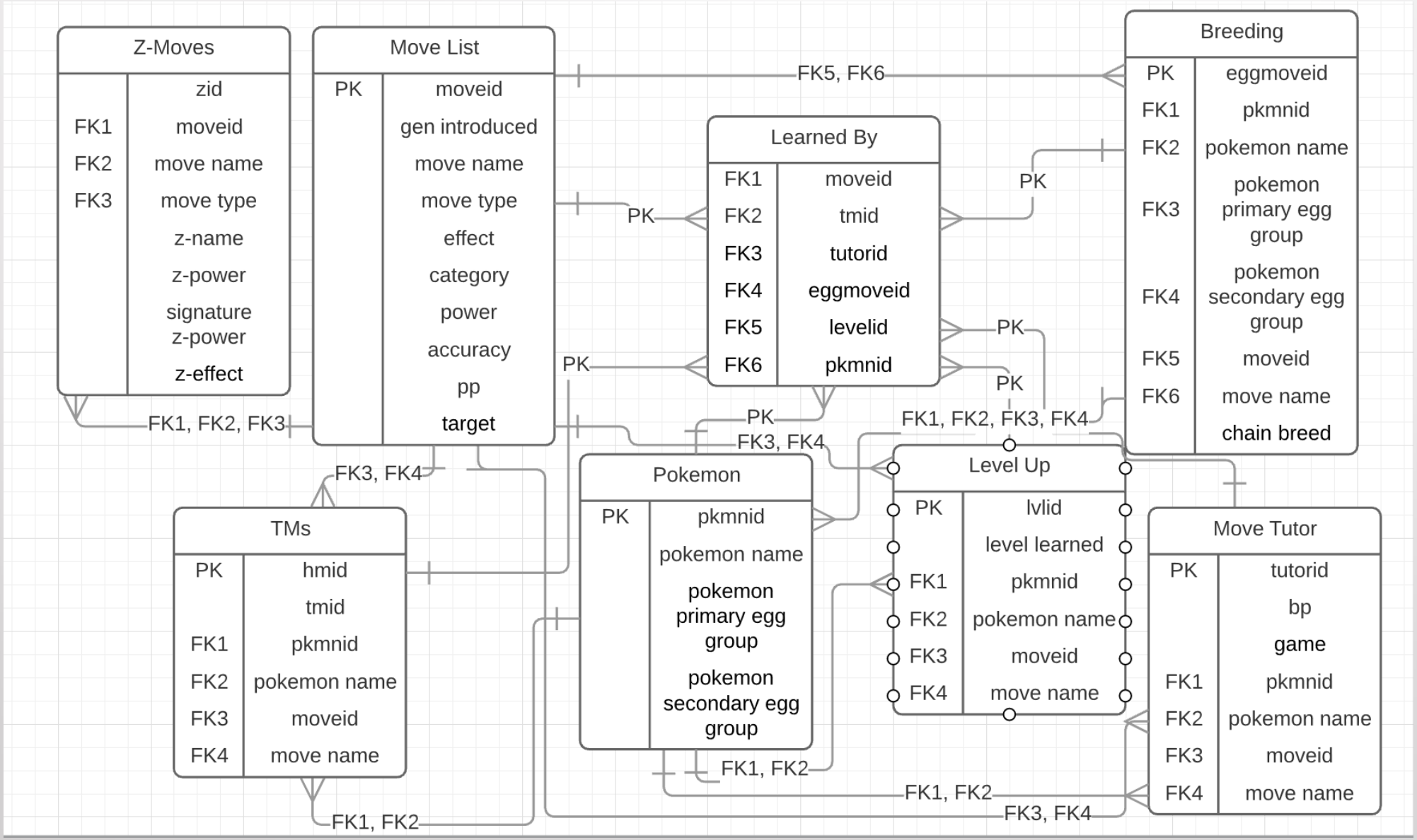
12/4/2017

Executive Summary

With the widespread popularity of the Pokémon video game series, it was inevitable that the games would evolve (no pun intended) into some competitive scene shortly after their release. While the games have always been very simplistic in design and accommodating to newcomers on the surface, there are a vast number of systems and mechanics in place that grow more complex with each new release and form the basis for competitive play. One such aspect that is crucial to both competitive play and the game itself is a proper understanding of moves. As the games use a turn-based battle system, the player must select a move to use against an opposing Pokémon or itself or allies to progress the battle. However, a deeper understanding of moves and their functions is all but required for competitive play.

The following documentation will include an entity-relationship diagram displaying every table and their respective attributes within the database, as well as SQL create statements and functional dependencies for table creation should the database need to be replicated elsewhere. Additionally, included will be a collection of sample data from query testing, as well as view definitions, stored procedures, reports and security for the database. Finally, a set of implementation notes and a list of known problems and possible future enhancements will be included for reference.

E/R Diagram


Z-Moves
zid
FK1 moveid
FK2 move name
FK3 move type
z-name
z-power
signature z-power
z-effect
Move List
PK moveid
gen introduced
move name
move type
effect
category
power
accuracy
pp
target
Breeding
PK eggmoveid
FK1 pkmnid
FK2 pokemon name
FK3 pokemon primary egg group
FK4 pokemon secondary egg group
FK5 moveid
FK6 move name
chain breed
Learned By
FK1 moveid
FK2 tmid
FK3 tutorid
FK4 eggmoveid
FK5 levelid
FK6 pkmnid
FK5, FK6
PK
FK1, FK2, FK3
FK3, FK4
TMs
PK hmid
tmid
FK1 pkmnid
FK2 pokemon name
FK3 moveid
FK4 move name
Pokemon
PK pkmnid
pokemon name
pokemon primary egg group
pokemon secondary egg group
Level Up
PK lvlid
level learned
FK1 pkmnid
FK2 pokemon name
FK3 moveid
FK4 move name
Move Tutor
PK tutorid
bp
game
FK1 pkmnid
FK2 pokemon name
FK3 moveid
FK4 move name
FK1, FK2
FK3, FK4

Move List

This table acts as a master list for every move in the database, as well as all attributes associated with each move.

CREATE TABLE moveList (

moveid INT NOT NULL UNIQUE,

genIntroduced TEXT NOT NULL CHECK (genIntroduced = ‘I’ OR genIntroduced = ‘II’ OR GenIntroduced = ‘III” OR genIntroduced = ‘IV’ OR genIntroduced = ‘V’ OR genIntroduced = ‘VI’ OR genIntroduced = ‘VII’),

moveName TEXT NOT NULL UNIQUE,

moveType TEXT NOT NULL,

effect TEXT NOT NULL,

category TEXT NOT NULL CHECK (category = ‘Physical’ OR category = ‘Special’ OR category = ‘Status’),

power INT,

accuracy INT,

pp INT NOT NULL,

target TEXT NOT NULL,

PRIMARY KEY (moveid));

Functional Dependencies:

moveid → {moveName, genIntroduced, moveType, category, power, accuracy, pp, target}

	Moveid [PK] integer	GenIntroduced text	MoveName text	MoveType text	Effect text	Category text	Accuracy text	PP integer	Target text	Power text
1	1	I	Tackle	Normal	Deals damage and has no seconda...	Physical	100%	35	Any Adjacent Pokemon	40
2	2	I	Hyper Beam	Normal	The user must recharge next turn.	Special	90%	5	Any Adjacent Pokemon	150
3	3	I	Swift	Normal	Does not check accuracy.	Special	[null]	20	All Adjacent Foes	60
4	4	II	Belly Drum	Normal	Cuts HP by 50% to max out the user...	Status	[null]	10	Self	[null]
5	5	IV	Brave Bird	Flying	Deals 33% recoil damage.	Physical	100%	15	Any Pokemon	120
6	6	IV	Ice Shard	Ice	Has +1 priority.	Special	100%	30	Any Adjacent Pokemon	40
7	7	I	Night Shade	Ghost	Deals damage equal to the user's l...	Special	[null]	15	Any Adjacent Pokemon	Varies
8	8	I	Toxic	Poison	Badly poisons the target.	Status	90%	10	Any Adjacent Pokemon	[null]
9	9	II	Crunch	Dark	Has a 20% chance of lowering the t...	Physical	100%	15	Any Adjacent Pokemon	80
10	10	II	Dynamic Punch	Fighting	Has a 100% chance to confuse the t...	Physical	50%	5	Any adjacent Pokemon	100
11	11	III	Muddy Water	Water	Has a 30% chance of lowering the t...	Special	85%	10	All Adjacent Foes	90
12	12	VI	Draining Kiss	Fairy	Restores the user's HP by 75% of d...	Special	100%	10	Any Adjacent Pokemon	50
13	13	I	Earthquake	Ground	Deals damage and has no addidtio...	Physical	100%	10	All Pokemon	100
14	14	IV	Charge Beam	Electric	Has a 70% change to raise the user...	Special	90	10	Any Adjacent Pokemon	50
15	15	III	Calm Mind	Psychic	Raises the user's Special Attack an...	Status	[null]	20	Self	[null]

Pokémon

This table acts as a master list for every Pokémon in the database, as well as their types and egg groups for breeding purposes.

CREATE TABLE pokémon (

pkmnid INT NOT NULL UNIQUE,

pokemonName TEXT NOT NULL UNIQUE,

pokémonPrimaryType TEXT NOT NULL,

pokemonSecondaryType TEXT,

pokémonPrimaryEggGroup TEXT NOT NULL,

pokémonSecondaryEggGroup TEXT,

PRIMARY KEY (pkmnid));

Functional Dependencies:

pkmnid → {pokemonName, pokemonPrimaryType, pokemonSecondaryType, pokemonPrimaryEggGroup, pokemonSecondaryEggGroup}

pkmnid, pokemonName → {pokemonPrimaryType, pokemonSecondaryType, pokemonPrimaryEggGroup, pokemonSecondaryEggGroup}

	Pkmnid [PK] integer	PokemonName text	PokemonPrimaryType text	PokemonSecondaryType text	PokemonPrimaryEggGroup text	PokemonSecondaryEggGroup text
1	000	Alan	Electric	Steel	Field	Mineral
2	001	Bulbasaur	Grass	Poison	Monster	Grass
3	145	Zapdos	Electric	Flying	Undiscovered	[null]
4	260	Swampert	Water	Ground	Monster	Water 1
5	399	Bidoof	Normal	[null]	Water 1	Field
6	461	Weavile	Dark	Ice	Field	[null]
7	606	Beheeyem	Psychic	[null]	Human-Like	[null]
8	132	Ditto	Normal	[null]	Ditto	[null]
9	373	Salamance	Dragon	Flying	Dragon	[null]
10	742	Cutiefly	Bug	Fairy	Bug	Fairy
11	112	Rhydon	Rock	Ground	Monster	Field
12	607	Litwick	Ghost	Fire	Amorphous	[null]
13	286	Breloom	Grass	Fighting	Fairy	Grass
14	245	Suicune	Water	[null]	Undiscovered	[null]
15	687	Malamar	Dark	Psychic	Water 1	Water 2

Z- Moves

This table acts as a master list for every Z-Move in the database, as well as their attributes and base move counterparts.

CREATE TABLE z-moves (

zid INT NOT NULL UNIQUE,

moveid INT NOT NULL UNIQUE REFERENCES moveList (moveid),

moveName TEXT NOT NULL, REFERENCES moveList (moveName),

moveType TEXT NOT NULL,

z-name TEXT NOT NULL,

z-power INT,

signatureZ-power INT,

z-effect TEXT,

PRIMARY KEY (zid));

Functional Dependencies:

zid → {moveid, moveName, z-name, z-power, signatureZ-power, z-effect}

	Zid [PK] integer	Moveid integer	MoveName text	MoveType text	Z-Power text	Signature Z-Power text	Z-Effect text	Z-Name text
1	1	1	Tackle	Normal	100	[null]	Deals damage and has no additional effect.	Breakneck Blitz
2	2	2	Giga Impact	Normal	200	210	Deals damage and has no additional effect.	Pulverizing Pancake
3	3	3	Swift	Normal	120	[null]	Deals damage and has no additional effect.	Breakneck Blitz
4	4	4	Belly Drum	Normal	[null]	[null]	Fully restores the user's HP.	Z-Belly Drum
5	5	5	Brave Bird	Flying	190	[null]	Deals damage and has no additional effect.	Supersonic Skystrike
6	6	6	Ice Shard	Ice	100	[null]	Deals damage and has no additional effect.	Subzero Slammer
7	7	7	Night Shade	Ghost	100	[null]	Deals damage and has no additional effect.	Never-Ending Nightmare
8	8	8	Toxic	Poison	[null]	[null]	Raises the user's Defense stat by one stage.	Z-Toxic
9	9	9	Crunch	Dark	160	[null]	Deals damage and has no additional effect.	Black Hole Eclipse
10	10	10	Dynamic Punch	Fighting	180	[null]	Deals damage and has no additional effect.	All-Out Pummeling
11	11	11	Muddy Water	Water	175	[null]	Deals damage and has no additional effect.	Hydro Vortex
12	12	12	Draining Kiss	Fairy	100	[null]	Deals damage and has no additional effect.	Twinkle Tackle
13	13	13	Earthquake	Ground	180	[null]	Deals damage and has no additional effect.	Tectonic Rage
14	14	14	Thunderbolt	Electric	175	195	Raises the user's critical hit ratio by two stages.	10,000,000 Volt Thunderbolt
15	15	15	Calm Mind	Psychic	[null]	[null]	Resets all of the user's lowered stats.	Z-Calm Mind

TMs

This table acts as a master list for every Technical Machine in the database, as well as each Pokémon that can learn them.

CREATE TABLE tms (

hmid INT NOT NULL UNIQUE,

tmid INT NOT NULL UNIQUE,

pkmnid INT NOT NULL UNIQUE REFERENCES pokemon (pkmnid),

pokemonName TEXT NOT NULL UNIQUE REFERENCES pokemon (pokemonName),

moveid INT NOT NULL UNIQUE REFERENCES moveList (moveid),

moveName TEXT NOT NULL UNIQUE REFERENCES moveList (moveName),

PRIMARY KEY (hmid));

Functional Dependencies:

hmid → {tmid, pkmnid, pokemonName, moveid, moveName}

	HMid [PK] integer	Pkmnid integer	PokemonName text	Tmid integer	Moveid integer	MoveName text
20	20	607	Litwick	06	8	Toxic
21	21	286	Breloom	06	8	Toxic
22	22	245	Suicune	06	8	Toxic
23	23	687	Malamar	06	8	Toxic
24	24	260	Swampert	26	13	Earthquake
25	25	373	Salamance	26	13	Earthquake
26	26	112	Rhydon	26	13	Earthquake
27	27	000	Alan	24	14	Thunderbolt
28	28	145	Zapdos	24	14	Thunderbolt
29	29	399	Bidoof	24	14	Thunderbolt
30	30	606	Beheeyem	24	14	Thunderbolt
31	31	112	Rhydon	24	14	Thunderbolt
32	32	687	Malamar	24	14	Thunderbolt
33	33	000	Alan	03	15	Calm Mind
34	34	461	Weavile	03	15	Calm Mind
35	35	606	Beheeyem	03	15	Calm Mind
36	36	742	Cutiefly	03	15	Calm Mind

Level Up

This table contains every move that a Pokémon in the database can learn by leveling up and at what level they learn them.

CREATE TABLE levelUp (

lvlid INT NOT NULL UNIQUE,

levelLearned INT NOT NULL,

pkmnid INT NOT NULL UNIQUE REFERENCES pokemon (pkmnid),

pokemonName TEXT NOT NULL UNIQUE REFERENCES pokemon (pokemonName),

moveid INT NOT NULL UNIQUE REFERENCES moveList (moveid),

moveName TEXT NOT NULL UNIQUE REFERENCES moveList (moveName),

PRIMARY KEY (lvlid));

Functional Dependencies:

lvlid → {levelLearned, pkmnid, pokemonName, moveid, moveName}

	Lvlid [PK] integer	LevelLearned integer	Pkmnid integer	PokemonName text	Moveid integer	MoveName text
3	3	1	286	Breloom	001	Tackle
4	4	1	399	Bidoof	001	Tackle
5	5	1	687	Malamar	001	Tackle
6	6	13	606	Litwick	007	Night Shade
7	7	25	373	Salamance	009	Crunch
8	8	25	399	Bidoof	009	Crunch
9	10	50	286	Breloom	010	Dynamic Punch
10	11	39	260	Swampert	011	Muddy Water
11	12	16	742	Cutiefly	012	Draining Kiss
12	13	48	112	Rhydon	013	Earthquake
13	14	51	260	Swampert	013	Earthquake
14	15	42	000	Alan	014	Thunderbolt
15	16	23	000	Alan	007	Night Shade
16	17	78	245	Suicune	015	Calm Mind
17	18	45	606	Beheeyem	015	Calm Mind

Breeding

This table contains every move that a Pokémon in the database can learn by breeding and if the move requires chain breeding to be learned.

```
CREATE TABLE breeding (

eggMoveid INT NOT NULL UNIQUE,

pkmnid INT NOT NULL UNIQUE REFERENCES pokemon (pkmnid),

pokemonName TEXT NOT NULL UNIQUE REFERENCES pokemon (pokemonName),

moveid INT NOT NULL UNIQUE REFERENCES moveList (moveid),

moveName TEXT NOT NULL UNIQUE REFERENCES moveList (moveName),

chainBreed TEXT NOT NULL,

PRIMARY KEY (eggMoveid) );
```

Functional Dependencies:

eggMoveid → {pkmnid, pokemonName, moveid, moveName, chainBreed}

	eggmoveid [PK] integer	pkmnid integer	pokemonName text	moveid integer	moveName text	chainBreed text
1	1	461	Weavile	6	Ice Shard	No
2	2	112	Rhydon	9	Crunch	No
3	3	260	Swampert	16	Barrier	Yes
4	4	606	Beheeyem	16	Barrier	No
5	5	373	Salamance	17	Dragon Rush	No
6	6	0	Alan	16	Barrier	No

Move Tutors

This table contains every move that a Pokémon in the database can learn via Move Tutors and the Battle Points each move costs as well as the game each move can be learned in.

CREATE TABLE moveTutors (

tutorid INT NOT NULL UNIQUE,

bp INT NOT NULL,

game TEXT NOT NULL,

pkmnid INT NOT NULL UNIQUE REFERENCES pokemon (pkmnid),

pokemonName TEXT NOT NULL UNIQUE REFERENCES pokemon (pokemonName),

moveid INT NOT NULL UNIQUE REFERENCES moveList (moveid),

moveName TEXT NOT NULL UNIQUE REFERENCES moveList (moveName),

PRIMARY KEY (tutorid));

Functional Dependencies:

tutorid→ {pkmnid, pokemonName, bp, moveid, moveName, game}

	tutorid [PK] integer	pkmnid integer	pokemonName text	moveid integer	moveName text	BP integer	Game text
1	1	0	Alan	18	Shock Wave	4	Ultra Sun and Ultra Moon
2	2	112	Rhydon	18	Shock Wave	4	Ultra Sun and Ultra Moon
3	3	145	Zapdos	18	Shock Wave	4	Ultra Sun and Ultra Moon
4	4	399	Bidoof	18	Shock Wave	4	Ultra Sun and Ultra Moon
5	5	606	Beheeyem	18	Shock Wave	4	Ultra Sun and Ultra Moon
6	6	607	Litwick	18	Shock Wave	4	Ultra Sun and Ultra Moon
7	7	373	Salamance	19	Draco Meteor	[null]	All Games
8	8	0	Alan	20	Telekenesis	8	Ultra Sun and Ultra Moon
9	9	606	Beheeyem	20	Telekenesis	8	Ultra Sun and Ultra Moon
10	10	607	Litwick	20	Telekenesis	8	Ultra Sun and Ultra Moon
11	11	687	Malamar	20	Telekenesis	8	Ultra Sun and Ultra Moon
12	12	742	Cutiefly	20	Telekenesis	8	Ultra Sun and Ultra Moon

Learned By

This table connects all the ids from each table as well as Pokémon names and move names, and acts as a intermediary table for all tables except Z-Moves, as Z-Moves are not “learned”.

```
CREATE TABLE learnedBy (

moveid INT NOT NULL UNIQUE REFERENCES moveList (moveid),

moveName TEXT NOT NULL UNIQUE REFERENCES moveList (moveName),

pkmnid INT NOT NULL UNIQUE REFERENCES pokemon (pkmnid),

pokemonName TEXT NOT NULL UNIQUE REFERENCES pokemon (pokemonName),

eggMoveid INT NOT NULL UNIQUE REFERENCES breeding (eggMoveid),

hmid INT NOT NULL UNIQUE REFERENCES tms (hmid),

tutorid INT NOT NULL UNIQUE REFERENCES moveTutors (tutorid),

lvlid INT NOT NULL UNIQUE REFERENCES levelUp (lvlid),

);
```

Breeding Reference

This view combines the Breeding table with the Pokémon table to display every Pokémon that can learn a move via breeding, the moves in question, and the Egg Groups of those Pokémon.

```
CREATE VIEW Breedref AS

SELECT breeding.pkmnid,

    breeding."pokemonName",

    pokemon."pokemonPrimaryEggGroup",

    pokemon."pokemonSecondaryEggGroup",

    breeding."moveName",

    breeding."chainBreed"

  FROM breeding

  JOIN pokemon ON breeding.pkmnid = pokemon.pkmnid;
```

	pkmnid integer	pokemonName text	pokemonPrimaryEggGroup text	pokemonSecondaryEggGroup text	moveName text	chainBreed text
1	112	Rhydon	Monster	Field	Crunch	No
2	461	Weavile	Field	[null]	Ice Shard	No
3	260	Swampert	Monster	Water 1	Barrier	Yes
4	606	Beheeyem	Human-Like	[null]	Barrier	No
5	373	Salamance	Dragon	[null]	Dragon Rush	No
6	0	Alan	Field	Mineral	Barrier	No

Explain Move Tutors

This view expands the Move Tutors table to include additional move information present in the master Move List table.

```
CREATE OR REPLACE VIEW explainMoveTutors

SELECT pkmnid, moveTutors."pokemonName", moveTutors."moveName", bp,
moveList."moveType", effect, category, accuracy, pp, target, power

FROM moveTutors

RIGHT OUTER JOIN moveList

ON moveTutors.tutorid = moveList.moveid

WHERE pkmnid IS NOT NULL

ORDER BY pkmnid ASC;
```

	pkmnid integer	pokemonName text	moveName text	bp integer	moveType text	effect text	category text	accuracy text	pp integer	target text	power text
1	0	Alan	Shock Wave	4	Normal	Deals ...	Physical	100%	35	Any Adj...	40
2	0	Alan	Telekenesis	8	Poison	Badly p...	Status	90%	10	Any Adj...	[null]
3	112	Rhydon	Shock Wave	4	Normal	The us...	Physical	90%	5	Any Adj...	150
4	145	Zapdos	Shock Wave	4	Normal	Does n...	Special	[null]	20	All Adja...	60
5	373	Salamance	Draco Meteor	[null]	Ghost	Deals ...	Special	[null]	15	Any Adj...	Varies
6	399	Bidoof	Shock Wave	4	Normal	Cuts H...	Status	[null]	10	Self	[null]
7	606	Beheeyem	Telekenesis	8	Dark	Has a ...	Physical	100%	15	Any Adj...	80
8	606	Beheeyem	Shock Wave	4	Flying	Deals ...	Physical	100%	15	Any Po...	120
9	607	Litwick	Shock Wave	4	Ice	Has +1...	Special	100%	30	Any Adj...	40
10	607	Litwick	Telekenesis	8	Fighting	Has a ...	Physical	50%	5	Any adj...	100
11	687	Malamar	Telekenesis	8	Water	Has a ...	Special	85%	10	All Adja...	90
12	742	Cutiefly	Telekenesis	8	Fairy	Restor...	Special	100%	10	Any Adj...	50

Find Pokemon

This function acts as a quick search to return all the basic information about a Pokémon by simply typing in it's name.

```
CREATE OR REPLACE FUNCTION findPokemon (TEXT)

RETURNS TABLE (pkmnid INT, pokemonName TEXT, pokemonPrimaryType TEXT, pokemonSecondaryType TEXT,
pokemonPrimaryEggGroup TEXT, pokemonSecondaryEggGroup TEXT) AS

$$

DECLARE

desiredPokemon TEXT := 1;

BEGIN

RETURN QUERY

SELECT pokemon.pkmnid, pokemon.pokemonName, pokemon.pokemonPrimaryType,
pokemon.pokemonSecondaryType, pokemon.pokemonPrimaryEggGroup, pokemon.pokemonSecondaryEggGroup

FROM pokemon

WHERE pokemon.pokemonName = desiredPokemon

ORDER BY pkmnid ASC;

END;

$$ LANGUAGE plpgsql;
```

	pkmnid integer	pokemonName text	pokemonPrimaryType text	pokemonSecondaryType text	pokemonPrimaryEggGroup text	pokemonSecondaryEggGroup text
1	0	Alan	Electric	Steel	Field	Mineral

TM Search

This function allows the user to search any TM by number and returns all relevant information for the move.

```
CREATE OR REPLACE FUNCTION tmsearch (INT)

RETURNS TABLE (tmid INT, moveName TEXT, moveType TEXT, effect TEXT, category TEXT, accuracy TEXT, pp TEXT,
target TEXT, power TEXT) AS

$$

DECLARE

tmnum INT := 1;

BEGIN

RETURN QUERY

SELECT tms.tmid, tms.moveName, moveList."moveType", moveList.effect, moveList.category, moveList.accuracy,
moveList.pp, moveList.target, moveList.power

FROM tms, moveList

WHERE tms.tmid = tmnum AND moveList.moveid = tms.moveid

ORDER BY tmid ASC;

END;

$$ LANGUAGE plpgsql;
```

	tmid integer	moveName text	moveType text	effect text	category text	accuracy text	pp integer	target text	power text
1	68	Giga Impact	Normal	The us...	Physical	90%	5	Any Adj...	150
2	68	Giga Impact	Normal	The us...	Physical	90%	5	Any Adj...	150
3	68	Giga Impact	Normal	The us...	Physical	90%	5	Any Adj...	150
4	68	Giga Impact	Normal	The us...	Physical	90%	5	Any Adj...	150
5	68	Giga Impact	Normal	The us...	Physical	90%	5	Any Adj...	150
6	68	Giga Impact	Normal	The us...	Physical	90%	5	Any Adj...	150
7	68	Giga Impact	Normal	The us...	Physical	90%	5	Any Adj...	150
8	68	Giga Impact	Normal	The us...	Physical	90%	5	Any Adj...	150
9	68	Giga Impact	Normal	The us...	Physical	90%	5	Any Adj...	150

Reports

This query shows all the moves a given Pokémon can learn via leveling up and additional information about the moves themselves.

```
SELECT levelUp."levelLearned", pkmnid, levelUp."pokemonName", moveList."moveName",
moveList."genIntroduced", moveList."moveType", effect, category, power, accuracy, pp, target

FROM levelUp

RIGHT OUTER JOIN moveList

ON levelUp."moveName" = moveList."moveName"

WHERE levelUp."pokemonName" = 'Breloom';
```

	levelLearned integer	pkmnid integer	pokemonName text	moveName text	genIntroduced text	moveType text	effect text	category text	power text	accuracy text	pp integer	target text
1	1	286	Breloom	Tackle	I	Normal	Deals ...	Physical	40	100%	35	Any Adj...
2	50	286	Breloom	Dynamic Punch	II	Fighting	Has a ...	Physical	100	50%	5	Any adj...

Reports

This query shows all the moves in the breeding table that can be upgraded into Z-Moves, the Pokémon they can be bred onto, and their attributes.

```
SELECT z-power."moveName", z-moves."moveType", z-power, z-moves."signatureZ-power", z-effect, z-name, pkmnid, breeding."pokemonName", breeding."chainBreed"

FROM z-moves

LEFT OUTER JOIN breeding

ON z-moves."moveid" = breeding."moveid"

WHERE z-moves."moveName" = breeding."moveName";
```

	moveName text	moveType text	zpower text	signatureZPower text	zeffect text	zname text	pkmnid integer	pokemonName text	chainBreed text
1	Crunch	Dark	160	[null]	Deals d...	Black H...	112	Rhydon	No
2	Ice Shard	Ice	100	[null]	Deals d...	Subzero...	461	Weavile	No
3	Barrier	Psychic	[null]	[null]	Resets a...	Z-Barrier	260	Swampert	Yes
4	Barrier	Psychic	[null]	[null]	Resets a...	Z-Barrier	606	Beheeyem	No
5	Dragon Rush	Dragon	180	[null]	Deals d...	Devasta...	373	Salamance	No
6	Barrier	Psychic	[null]	[null]	Resets a...	Z-Barrier	0	Alan	No

Roles

The database currently has two roles: Admin and Trainer. Admins have full permissions across the database, while Trainers are able to update and insert into the Breeding, Move Tutors, Level Up, and TMs tables at their leisure, but may only select from the master Move List and Pokémon tables and may not view the Learned By table whatsoever.

```
CREATE ROLE Admin;

GRANT ALL ON ALL TABLES IN SCHEMA public TO Admin;


CREATE ROLE Trainer;

REVOKE ALL ON ALL TABLES IN SCHEMA public FROM trainer;

GRANT SELECT ON ALL TABLES IN SCHEMA public TO trainer;

GRANT INSERT ON breeding, levelup, moveTutors, tms, z-moves TO Trainer;

GRANT UPDATE ON breeding, levelup, moveTutors, tms, z-moves TO Trainer;
```

Implementation Notes

- This database is only accurate to Pokémon Sun, Moon, Ultra Sun, and Ultra Moon. It does not currently support previous games in the series. However, given some work, it could be modified to do so.

- In the case of a move that could be upgraded into a Signature Z-Move for a specific Pokémon, such as 10,000,000 Volt Thunderbolt or Pulverizing Pancake, that value and name are listed as opposed to the normal Z-Move for that attack, (Gigavolt Havoc and Breakneck Blitz, respectively). In it's current state, implementing a workaround for this would have been both time consuming for an admin and potentially confusing for users.

- Earlier builds of this database contained a "Parent" column in the Breeding table detailing exactly which Pokémon could pass down what Egg Moves to their children, (Example: A male Bidoof could breed with a female Rhydon and pass down the move Crunch to a the resulting Rhyhorn because the two Pokemon are in the Field Egg Group) but this feature was scrapped as a potentially huge number of Pokemon are all capable of passing down a single egg move to another Pokémon, obliterating the Law of First Normal Form.

Known Problems

- There is currently no way to support multiple Pokémon with the same Pokedex number (pkmnid) such as Pokémon with multiple formes or regional variants.

- There is currently no way to document Pokémon that learn normally unobtainable moves through special circumstances. i.e. real-world events, giveaways.

Future Enhancements

- Support can be added for older Pokémon games.
- A table on in-game locations for specific Pokémon could be included.
- Support for high-level play could be added, including columns and tables for Effort Values, Individual Values, Abilities, and Damage Calculation functions.