Pokemon Database System

Designed by Katharine Craven

Table of Contents

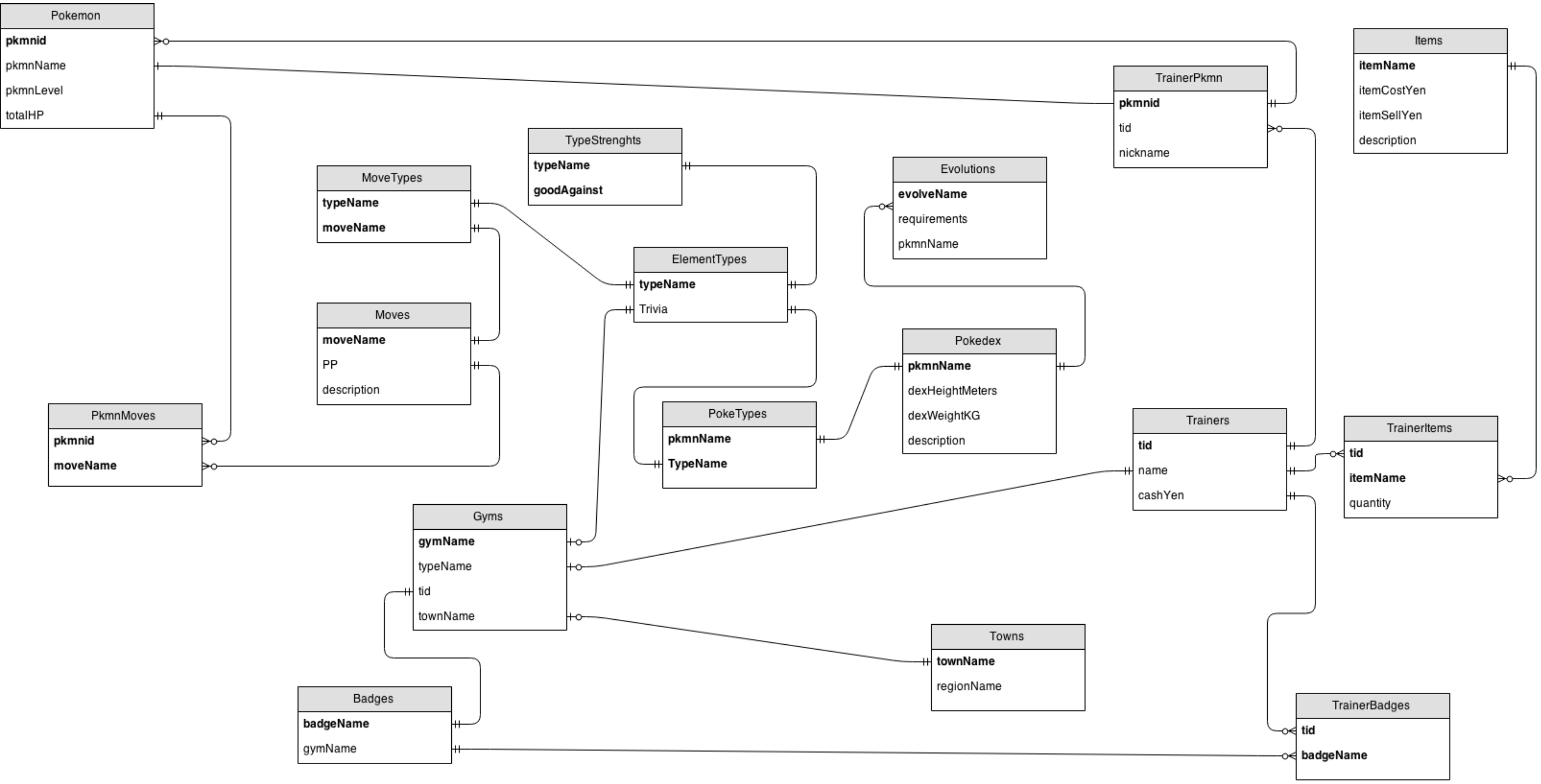
Executive Summary

Hello, and welcome to the world of Pokemon! There is a lot of things to keep track of in a world like this, so the Scientists of the Pokemon world decided it was time to work on a big database to help trainers starting on their journeys. It used to be that someone would wait in Gyms all day to tell you what kind of Pokemon element the Gym Leader would use and what types of Pokemon elements were strong against that. With the help of technology, the scientists decided to give that hardworking fellow a vacation and make it easier for everyone to get access to that information at the same time.

There is a lot of bulky or hard to remember information for Professors as well. What was the name of that Pokemon trainer again? What did Pikachu evolve into? What type of move was 'Tackle' again? What was the name of the Celadon City Gym Leader again? As a professor, you never want to seem at a loss for knowledge, and everything will be right at your fingertips. It also used to be a problem that one professor's findings would never reach other regions. With a database you can update, that won't be a problem.

The overall objective of this database was to make the world of Pokemon a little more accessable to everyone who wanted to partake in it.

ER Diagram

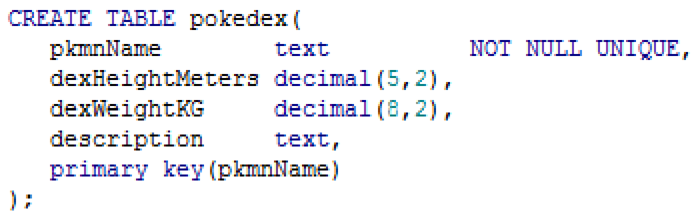

Pokemon
pkmnid
pkmnName
pkmnLevel
totalHP
TrainerPkmn
pkmnid
tid
nickname
Items
itemName
itemCostYen
itemSellYen
description
TypeStrenghts
typeName
goodAgainst
MoveTypes
typeName
moveName
Evolutions
evolveName
requirements
pkmnName
ElementTypes
typeName
Trivia
Moves
moveName
PP
description
Pokedex
pkmnName
dexHeightMeters
dexWeightKG
description
PkmnMoves
pkmnid
moveName
PokeTypes
pkmnName
TypeName
Trainers
tid
name
cashYen
TrainerItems
tid
itemName
quantity
Gyms
gymName
typeName
tid
townName
Towns
townName
regionName
Badges
badgeName
gymName
TrainerBadges
tid
badgeName

Pokedex Table

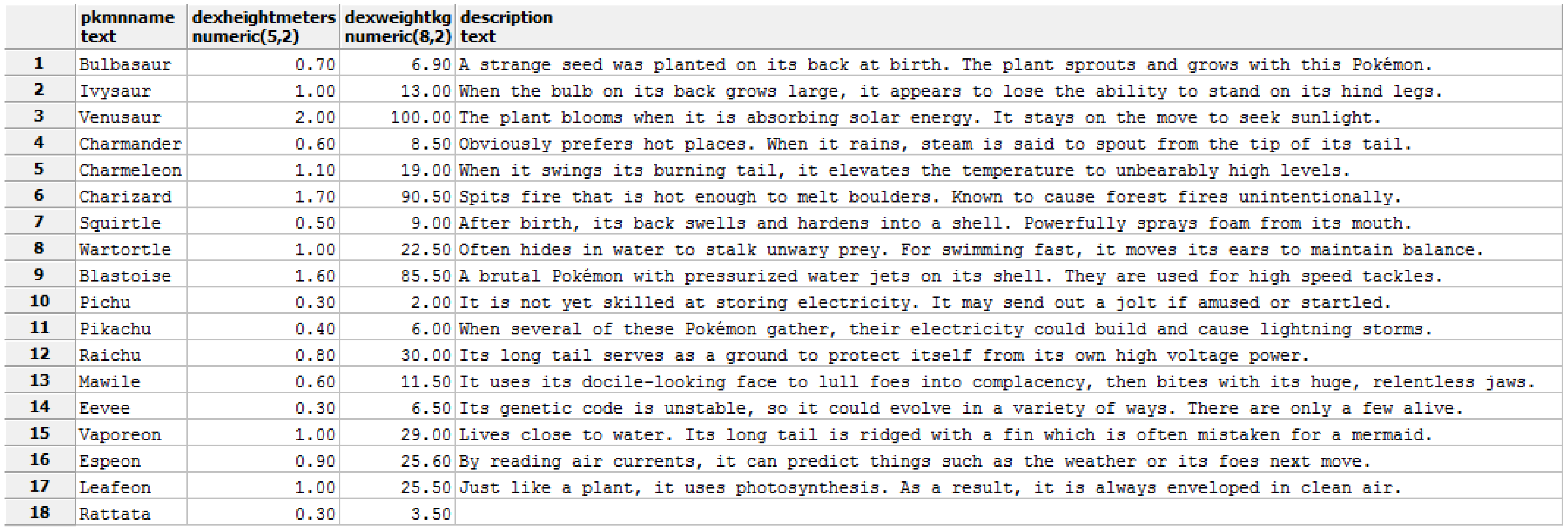
The Pokedex table stores information about different Pokemon species. The text includes the name of the Pokemon, its height in meters, its weight in meters, and a description of the Pokemon.

Functional Dependencies:

pkmnName -> dexHeightMeters
pkmnName -> dexWeightKG
pkmnName -> description



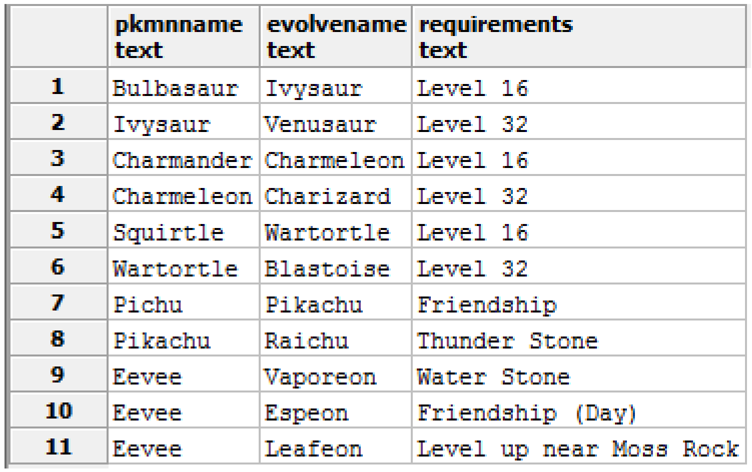
```
CREATE TABLE pokedex(
   pkmnName        text         NOT NULL UNIQUE,
   dexHeightMeters decimal(5,2),
   dexWeightKG     decimal(8,2),
   description     text,
   primary key(pkmnName)
);
```



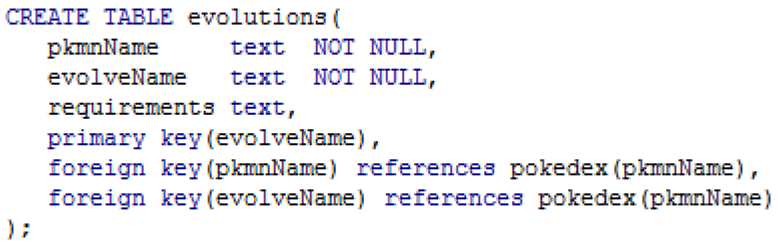
	pkmnname text	dexheightmeters numeric(5,2)	dexweightkg numeric(8,2)	description text
1	Bulbasaur	0.70	6.90	A strange seed was planted on its back at birth. The plant sprouts and grows with this Pokémon.
2	Ivysaur	1.00	13.00	When the bulb on its back grows large, it appears to lose the ability to stand on its hind legs.
3	Venusaur	2.00	100.00	The plant blooms when it is absorbing solar energy. It stays on the move to seek sunlight.
4	Charmander	0.60	8.50	Obviously prefers hot places. When it rains, steam is said to spout from the tip of its tail.
5	Charmeleon	1.10	19.00	When it swings its burning tail, it elevates the temperature to unbearably high levels.
6	Charizard	1.70	90.50	Spits fire that is hot enough to melt boulders. Known to cause forest fires unintentionally.
7	Squirtle	0.50	9.00	After birth, its back swells and hardens into a shell. Powerfully sprays foam from its mouth.
8	Wartortle	1.00	22.50	Often hides in water to stalk unwary prey. For swimming fast, it moves its ears to maintain balance.
9	Blastoise	1.60	85.50	A brutal Pokémon with pressurized water jets on its shell. They are used for high speed tackles.
10	Pichu	0.30	2.00	It is not yet skilled at storing electricity. It may send out a jolt if amused or startled.
11	Pikachu	0.40	6.00	When several of these Pokémon gather, their electricity could build and cause lightning storms.
12	Raichu	0.80	30.00	Its long tail serves as a ground to protect itself from its own high voltage power.
13	Mawile	0.60	11.50	It uses its docile-looking face to lull foes into complacency, then bites with its huge, relentless jaws.
14	Eevee	0.30	6.50	Its genetic code is unstable, so it could evolve in a variety of ways. There are only a few alive.
15	Vaporeon	1.00	29.00	Lives close to water. Its long tail is ridged with a fin which is often mistaken for a mermaid.
16	Espeon	0.90	25.60	By reading air currents, it can predict things such as the weather or its foes next move.
17	Leafeon	1.00	25.50	Just like a plant, it uses photosynthesis. As a result, it is always enveloped in clean air.
18	Rattata	0.30	3.50	

Evolutions Table

The Evolutions table contains the various evolutions a Pokemon may have. It includes the evolution name, the name of the Pokemon it is evolved from, and the requirements for needed for the Pokemon to evolve.



	pkmnname text	evolvename text	requirements text
1	Bulbasaur	Ivysaur	Level 16
2	Ivysaur	Venusaur	Level 32
3	Charmander	Charmeleon	Level 16
4	Charmeleon	Charizard	Level 32
5	Squirtle	Wartortle	Level 16
6	Wartortle	Blastoise	Level 32
7	Pichu	Pikachu	Friendship
8	Pikachu	Raichu	Thunder Stone
9	Eevee	Vaporeon	Water Stone
10	Eevee	Espeon	Friendship (Day)
11	Eevee	Leafeon	Level up near Moss Rock



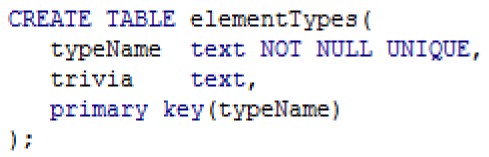
```
CREATE TABLE evolutions(
   pkmnName     text  NOT NULL,
   evolveName   text  NOT NULL,
   requirements text,
   primary key(evolveName),
   foreign key(pkmnName) references pokedex(pkmnName),
   foreign key(evolveName) references pokedex(pkmnName)
);
```

Functional Dependencies:

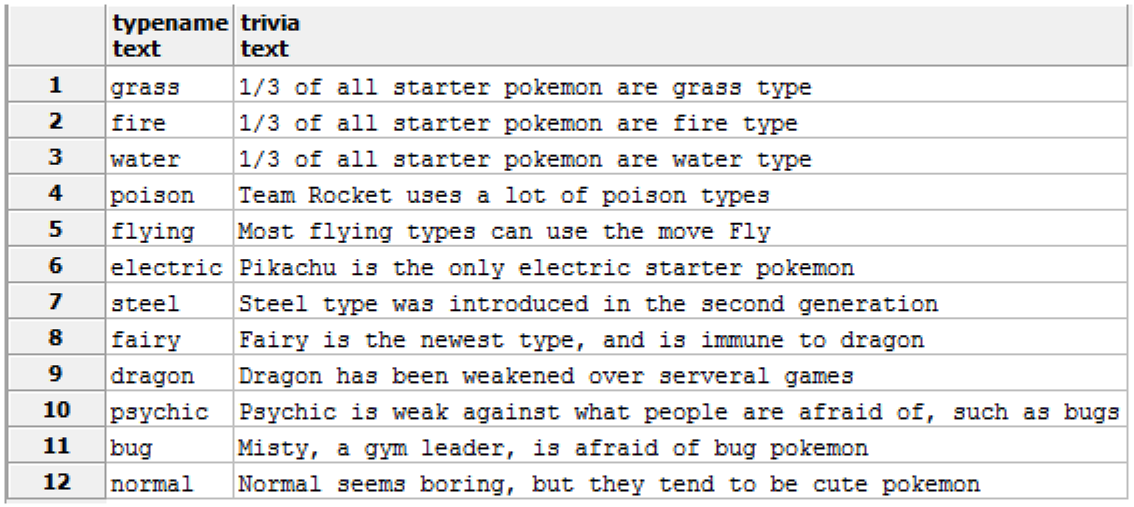
evolveName-> pkmnName
evolveName -> requirements

Element Types Table

The Element Types table contains element attributes and some trivia about different elements to go along with it. These types are associated both with Pokemon and Pokemon moves.



```
CREATE TABLE elementTypes(
   typeName  text NOT NULL UNIQUE,
   trivia    text,
   primary key(typeName)
);
```

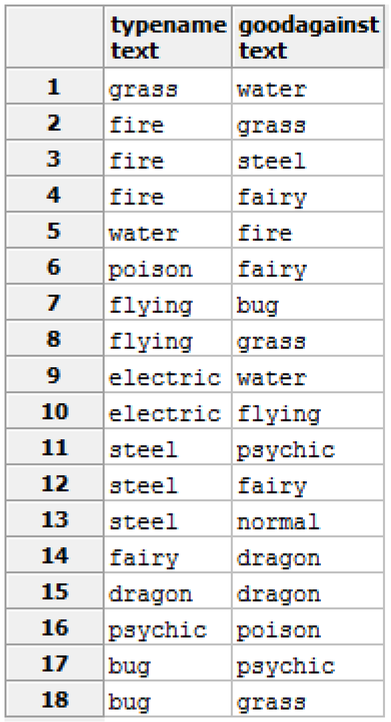


	typename text	trivia text
1	grass	1/3 of all starter pokemon are grass type
2	fire	1/3 of all starter pokemon are fire type
3	water	1/3 of all starter pokemon are water type
4	poison	Team Rocket uses a lot of poison types
5	flying	Most flying types can use the move Fly
6	electric	Pikachu is the only electric starter pokemon
7	steel	Steel type was introduced in the second generation
8	fairy	Fairy is the newest type, and is immune to dragon
9	dragon	Dragon has been weakened over serveral games
10	psychic	Psychic is weak against what people are afraid of, such as bugs
11	bug	Misty, a gym leader, is afraid of bug pokemon
12	normal	Normal seems boring, but they tend to be cute pokemon

Functional Dependencies:

typeName-> trivia

Type Strengths Table



	typename text	goodagainst text
1	grass	water
2	fire	grass
3	fire	steel
4	fire	fairy
5	water	fire
6	poison	fairy
7	flying	bug
8	flying	grass
9	electric	water
10	electric	flying
11	steel	psychic
12	steel	fairy
13	steel	normal
14	fairy	dragon
15	dragon	dragon
16	psychic	poison
17	bug	psychic
18	bug	grass

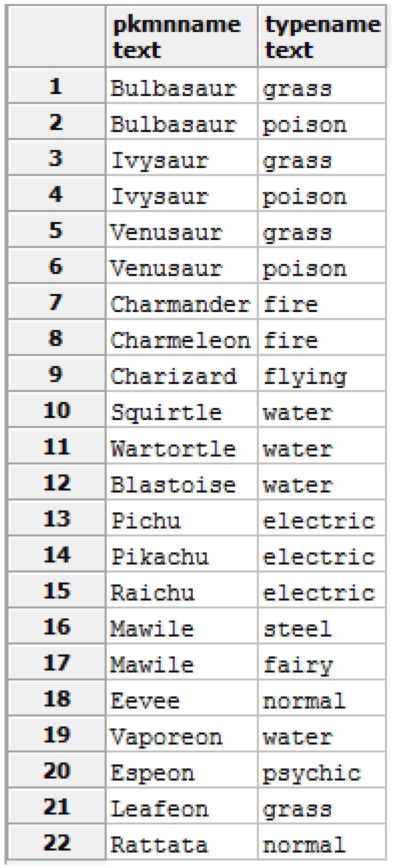
The Type Strengths table organizes what each of the elements is effective fighting against.

```
CREATE TABLE typeStrengths(
   typeName    text NOT NULL,
   goodAgainst text NOT NULL,
   primary key(typeName, goodAgainst),
   foreign key(typeName) references elementTypes(typeName)
);
```

<u>Functional Dependencies:</u>

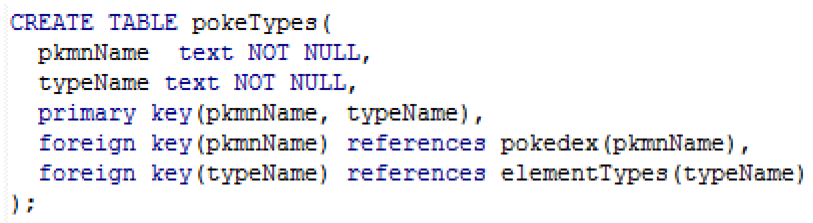
typeName-> goodAgainst

PokeTypes Table



	pkmnname text	typename text
1	Bulbasaur	grass
2	Bulbasaur	poison
3	Ivysaur	grass
4	Ivysaur	poison
5	Venusaur	grass
6	Venusaur	poison
7	Charmander	fire
8	Charmeleon	fire
9	Charizard	flying
10	Squirtle	water
11	Wartortle	water
12	Blastoise	water
13	Pichu	electric
14	Pikachu	electric
15	Raichu	electric
16	Mawile	steel
17	Mawile	fairy
18	Eevee	normal
19	Vaporeon	water
20	Espeon	psychic
21	Leafeon	grass
22	Rattata	normal

The PokeTypes table determines what kind of elemental type a Pokemon is. A Pokemon can have more than one elemental type.



```
CREATE TABLE pokeTypes(
  pkmnName  text NOT NULL,
  typeName text NOT NULL,
  primary key(pkmnName, typeName),
  foreign key(pkmnName) references pokedex(pkmnName),
  foreign key(typeName) references elementTypes(typeName)
);
```

Functional Dependencies:

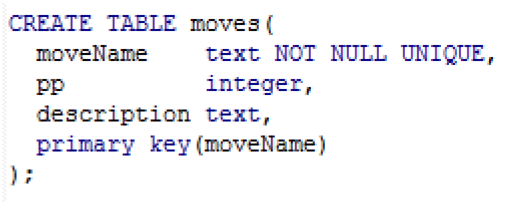
{pkmnName typeName} ->

Moves Table

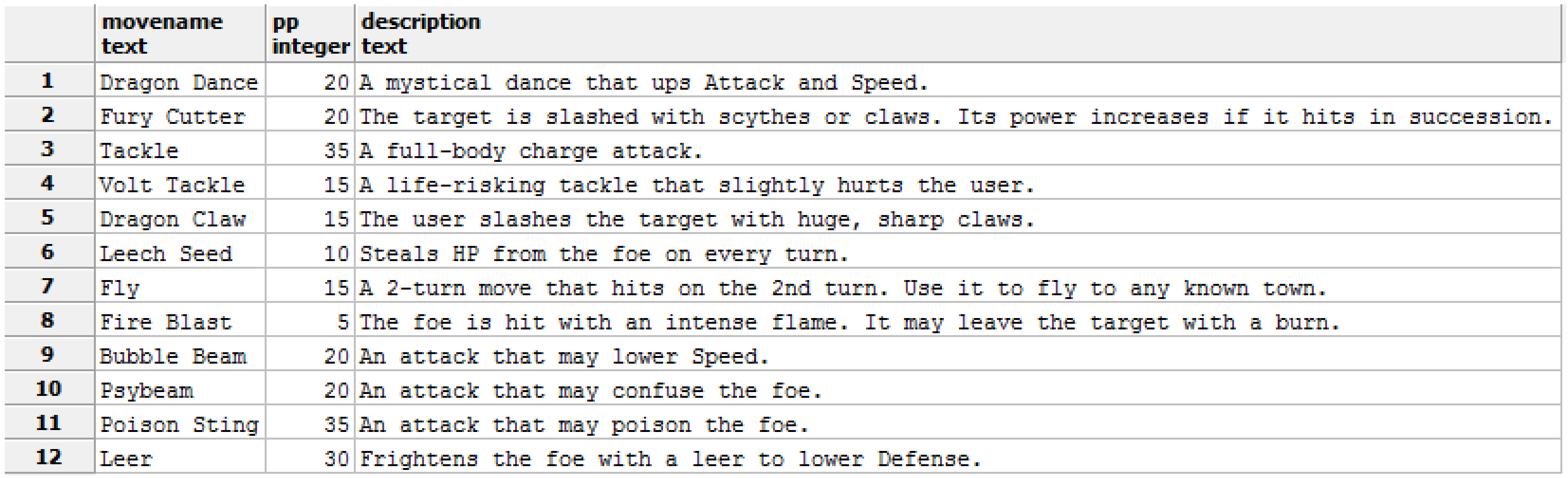
The Moves table contains information on the different kinds of attacks a Pokemon can use. The table includes the attack name, the PP (the number of times it can be used in battle), and a description of the move.

Functional Dependencies:

moveName -> PP
moveName -> description



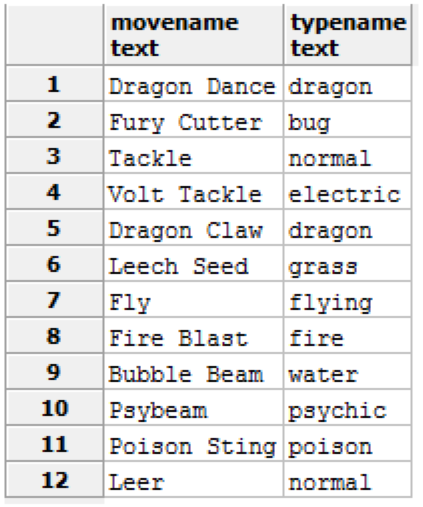
```
CREATE TABLE moves(
  moveName    text NOT NULL UNIQUE,
  pp          integer,
  description text,
  primary key(moveName)
);
```



	movename text	pp integer	description text
1	Dragon Dance	20	A mystical dance that ups Attack and Speed.
2	Fury Cutter	20	The target is slashed with scythes or claws. Its power increases if it hits in succession.
3	Tackle	35	A full-body charge attack.
4	Volt Tackle	15	A life-risking tackle that slightly hurts the user.
5	Dragon Claw	15	The user slashes the target with huge, sharp claws.
6	Leech Seed	10	Steals HP from the foe on every turn.
7	Fly	15	A 2-turn move that hits on the 2nd turn. Use it to fly to any known town.
8	Fire Blast	5	The foe is hit with an intense flame. It may leave the target with a burn.
9	Bubble Beam	20	An attack that may lower Speed.
10	Psybeam	20	An attack that may confuse the foe.
11	Poison Sting	35	An attack that may poison the foe.
12	Leer	30	Frightens the foe with a leer to lower Defense.

Move Types Table

The MoveTypes table determines what kind of elemental power a move has. A move might have more than one elemental power on rare occasions.



	movename text	typename text
1	Dragon Dance	dragon
2	Fury Cutter	bug
3	Tackle	normal
4	Volt Tackle	electric
5	Dragon Claw	dragon
6	Leech Seed	grass
7	Fly	flying
8	Fire Blast	fire
9	Bubble Beam	water
10	Psybeam	psychic
11	Poison Sting	poison
12	Leer	normal

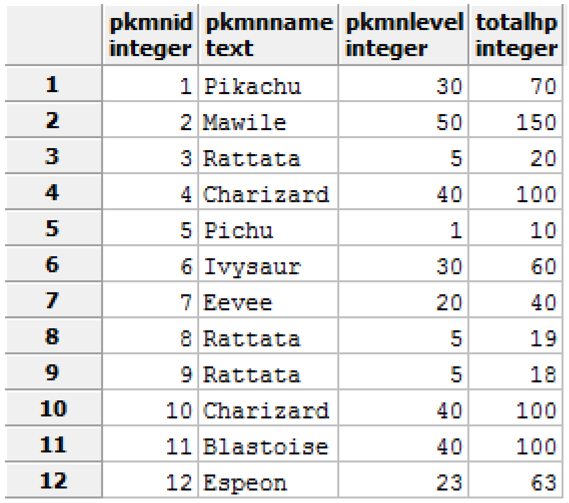
```
CREATE TABLE moveTypes(
  moveName text NOT NULL,
  typeName text NOT NULL,
  primary key (moveName, typeName),
  foreign key(typeName) references elementTypes(typeName),
  foreign key(moveName) references moves(moveName)
);
```

Functional Dependencies:

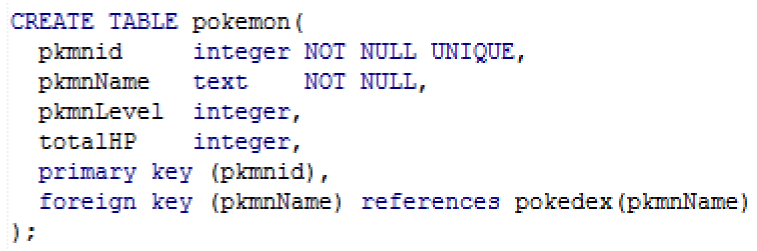
{moveName typeName} ->

Pokemon Table

The Pokemon table accounts for each individual Pokemon, identified by a pkmnid. The table includes the species name of the Pokemon, the pkmnLevel (the level the Pokemon has been raised to), and the totalHP (amount of damage a Pokemon can take before fainting in battle).



	pkmnid	pkmnname	pkmnlevel	totalhp
	integer	text	integer	integer
1	1	Pikachu	30	70
2	2	Mawile	50	150
3	3	Rattata	5	20
4	4	Charizard	40	100
5	5	Pichu	1	10
6	6	Ivysaur	30	60
7	7	Eevee	20	40
8	8	Rattata	5	19
9	9	Rattata	5	18
10	10	Charizard	40	100
11	11	Blastoise	40	100
12	12	Espeon	23	63

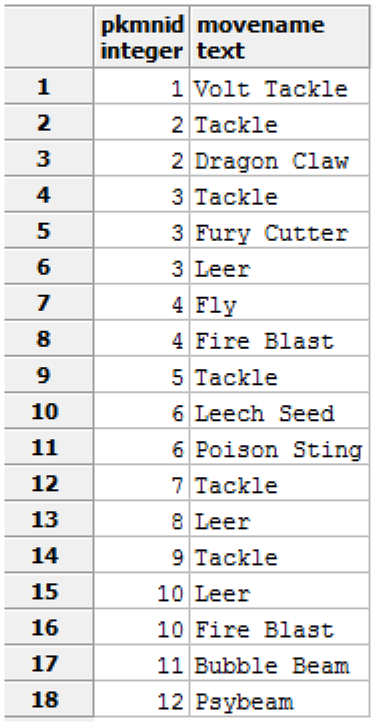


```
CREATE TABLE pokemon(
  pkmnid     integer NOT NULL UNIQUE,
  pkmnName   text    NOT NULL,
  pkmnLevel  integer,
  totalHP    integer,
  primary key (pkmnid),
  foreign key (pkmnName) references pokedex(pkmnName)
);
```

Functional Dependencies:

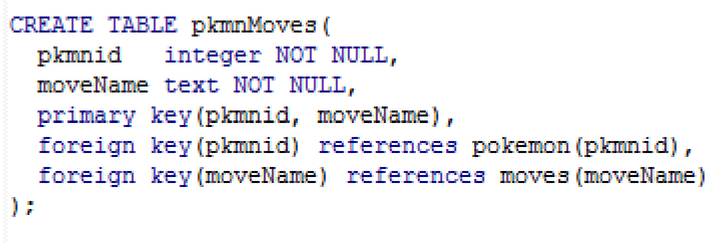
pkmnid -> pkmnName
pkmnid -> pkmnLevel
pkmnid -> totalHP

PkmnMoves Table



	pkmnid integer	movename text
1	1	Volt Tackle
2	2	Tackle
3	2	Dragon Claw
4	3	Tackle
5	3	Fury Cutter
6	3	Leer
7	4	Fly
8	4	Fire Blast
9	5	Tackle
10	6	Leech Seed
11	6	Poison Sting
12	7	Tackle
13	8	Leer
14	9	Tackle
15	10	Leer
16	10	Fire Blast
17	11	Bubble Beam
18	12	Psybeam

The PkmnMoves table records which individual Pokemon have which moves. A Pokemon can know more than one move.



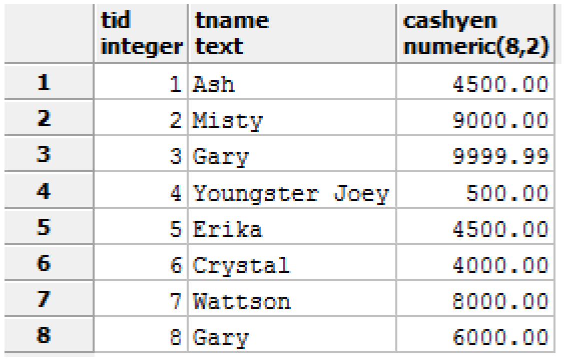
```
CREATE TABLE pkmnMoves(
  pkmnid   integer NOT NULL,
  moveName text NOT NULL,
  primary key(pkmnid, moveName),
  foreign key(pkmnid) references pokemon(pkmnid),
  foreign key(moveName) references moves(moveName)
);
```

Functional Dependencies:

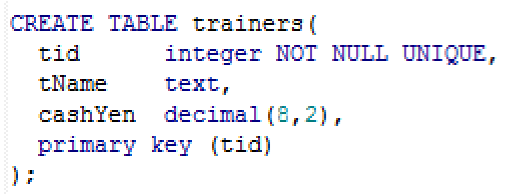
{pkmnid, moveName} ->

Trainers Table

The trainers table records information about pokemon trainers. The table includes a unique identifier (tid), the name of the trainer (only first names are commonly shared in Pokemon), and the amount of cash in terms of Yen the trainer has.



	tid integer	tname text	cashyen numeric(8,2)
1	1	Ash	4500.00
2	2	Misty	9000.00
3	3	Gary	9999.99
4	4	Youngster Joey	500.00
5	5	Erika	4500.00
6	6	Crystal	4000.00
7	7	Wattson	8000.00
8	8	Gary	6000.00



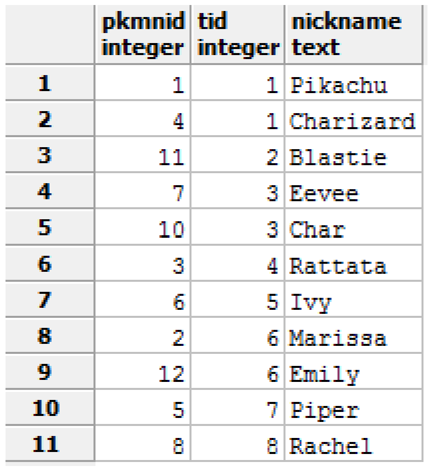
```
CREATE TABLE trainers(
  tid       integer NOT NULL UNIQUE,
  tName     text,
  cashYen   decimal(8,2),
  primary key (tid)
);
```

Functional Dependencies:

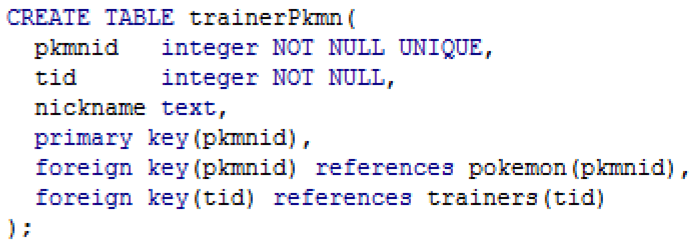
tid -> tName
tid -> cashYen

TrainerPkmn Table

This table records what Pokemon belongs to which trainer, and also the nickname that trainer gave to that Pokemon.



	pkmnid integer	tid integer	nickname text
1	1	1	Pikachu
2	4	1	Charizard
3	11	2	Blastie
4	7	3	Eevee
5	10	3	Char
6	3	4	Rattata
7	6	5	Ivy
8	2	6	Marissa
9	12	6	Emily
10	5	7	Piper
11	8	8	Rachel



```
CREATE TABLE trainerPkmn(
  pkmnid   integer NOT NULL UNIQUE,
  tid      integer NOT NULL,
  nickname text,
  primary key(pkmnid),
  foreign key(pkmnid) references pokemon(pkmnid),
  foreign key(tid) references trainers(tid)
);
```

Functional Dependencies:

{pkmnid, tid} -> nickname

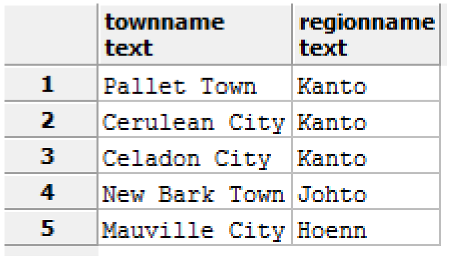
Towns Table

```
CREATE TABLE towns(
  townName    text NOT NULL UNIQUE,
  regionName text,
  primary key(townName)
);
```

This table records different towns in the Pokemon world, and what region of the Pokemon world that town is located in.

Functional Dependencies:

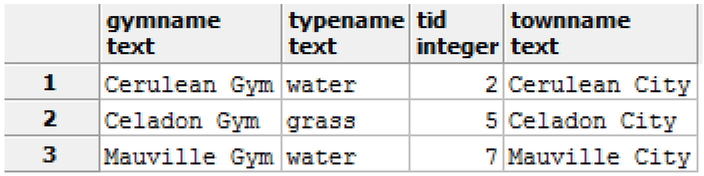
townName -> regionName



	townname text	regionname text
1	Pallet Town	Kanto
2	Cerulean City	Kanto
3	Celadon City	Kanto
4	New Bark Town	Johto
5	Mauville City	Hoenn

Gyms Table

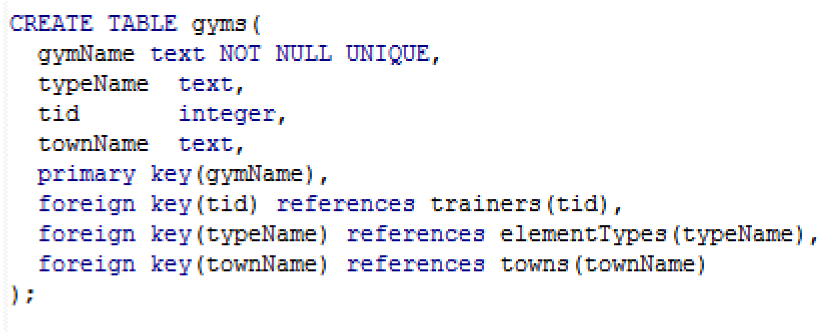
This table records information on different Pokemon gyms, which are places you go to battle elite trainers called gym leaders. The table has the name of the gym, the type of element the gym exclusively trains in, the trainer id of the gym leader, and the town the gym is located in.



	gymname text	typename text	tid integer	townname text
1	Cerulean Gym	water	2	Cerulean City
2	Celadon Gym	grass	5	Celadon City
3	Mauville Gym	water	7	Mauville City

Functional Dependencies:

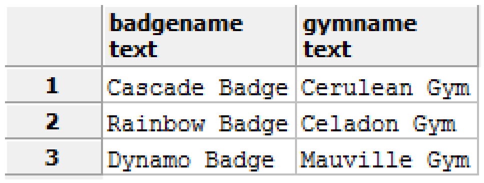
gymName -> typeName
gymName -> tid
gymName -> townName



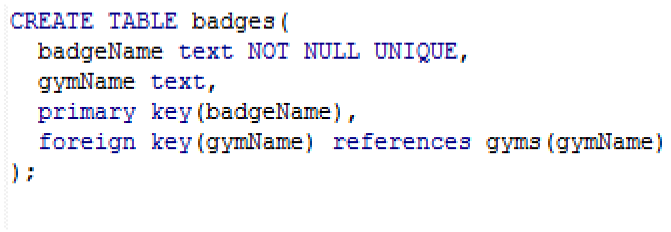
```
CREATE TABLE gyms(
  gymName text NOT NULL UNIQUE,
  typeName  text,
  tid       integer,
  townName  text,
  primary key(gymName),
  foreign key(tid) references trainers(tid),
  foreign key(typeName) references elementTypes(typeName),
  foreign key(townName) references towns(townName)
);
```

Badges Table

This table records information on the different gym badges. There is a badge name and the gym it belongs to. Badges are given to trainers as proof that they have beaten a gym leader.



	badgename text	gymname text
1	Cascade Badge	Cerulean Gym
2	Rainbow Badge	Celadon Gym
3	Dynamo Badge	Mauville Gym



```
CREATE TABLE badges(
  badgeName text NOT NULL UNIQUE,
  gymName text,
  primary key(badgeName),
  foreign key(gymName) references gyms(gymName)
);
```

Functional Dependencies:

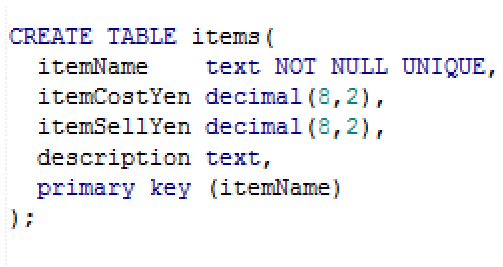
badgeName -> gymName

Items Table

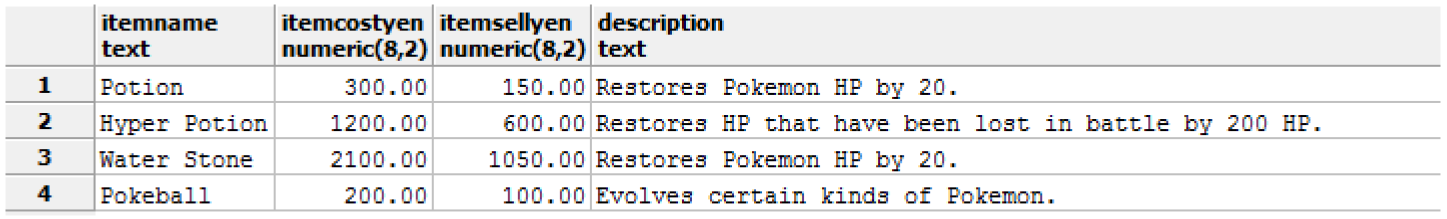
This table has information on various items. It has the itemName, how much the item costs to buy in yen, how much the item sells for in yen, and a description of the item.

Functional Dependencies:

itemName -> itemCostYen
itemName -> itemSellYen
itemName ->description



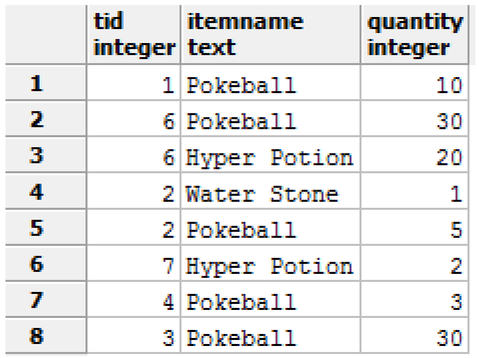
```
CREATE TABLE items(
  itemName    text NOT NULL UNIQUE,
  itemCostYen decimal(8,2),
  itemSellYen decimal(8,2),
  description text,
  primary key (itemName)
);
```



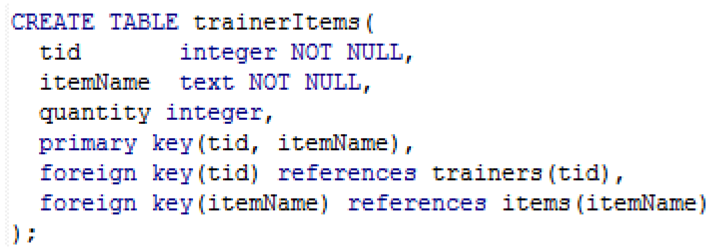
	itemname text	itemcostyen numeric(8,2)	itemsellyen numeric(8,2)	description text
1	Potion	300.00	150.00	Restores Pokemon HP by 20.
2	Hyper Potion	1200.00	600.00	Restores HP that have been lost in battle by 200 HP.
3	Water Stone	2100.00	1050.00	Restores Pokemon HP by 20.
4	Pokeball	200.00	100.00	Evolves certain kinds of Pokemon.

Trainer Items Table

This table shows what trainer has which items, and how many of each item (the quantity) that trainer has.



	tid integer	itemname text	quantity integer
1	1	Pokeball	10
2	6	Pokeball	30
3	6	Hyper Potion	20
4	2	Water Stone	1
5	2	Pokeball	5
6	7	Hyper Potion	2
7	4	Pokeball	3
8	3	Pokeball	30



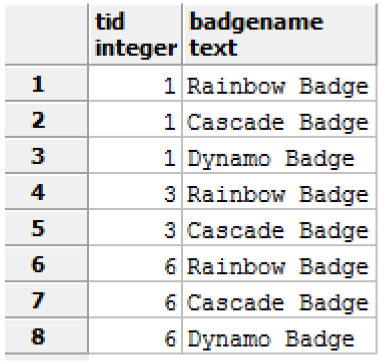
```
CREATE TABLE trainerItems(
  tid       integer NOT NULL,
  itemName  text NOT NULL,
  quantity integer,
  primary key(tid, itemName),
  foreign key(tid) references trainers(tid),
  foreign key(itemName) references items(itemName)
);
```

Functional Dependencies:

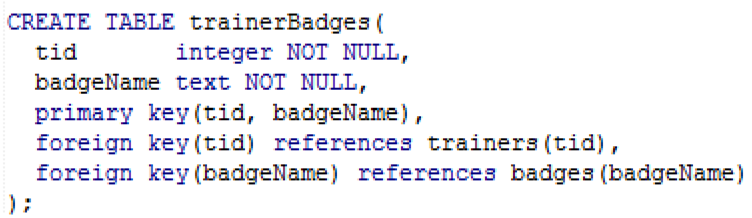
{tid, itemName} -> quantity

Trainer Badges Table

This table shows which trainers have earned which badges from beating gym leaders. This is very important to trainers, as it is a mark of their progress.



	tid integer	badgename text
1	1	Rainbow Badge
2	1	Cascade Badge
3	1	Dynamo Badge
4	3	Rainbow Badge
5	3	Cascade Badge
6	6	Rainbow Badge
7	6	Cascade Badge
8	6	Dynamo Badge



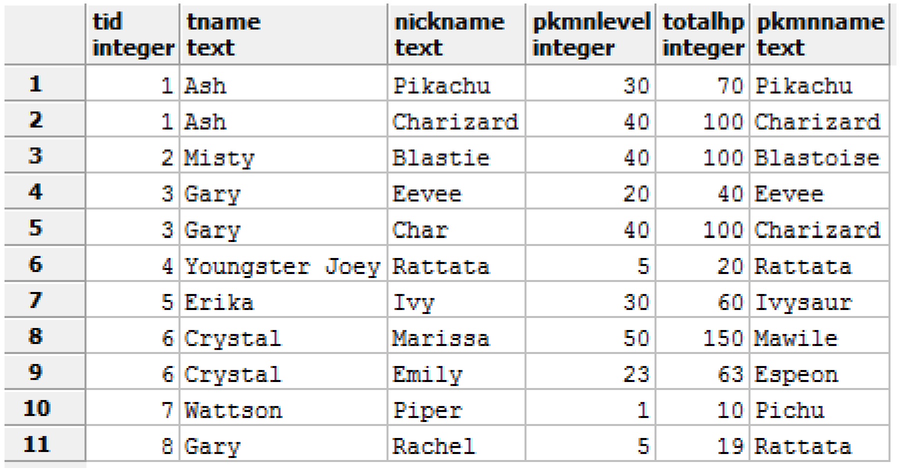
```
CREATE TABLE trainerBadges(
  tid       integer NOT NULL,
  badgeName text NOT NULL,
  primary key(tid, badgeName),
  foreign key(tid) references trainers(tid),
  foreign key(badgeName) references badges(badgeName)
);
```

Functional Dependencies:

{tid, badgeName} ->

View Trainer Pokemon Names

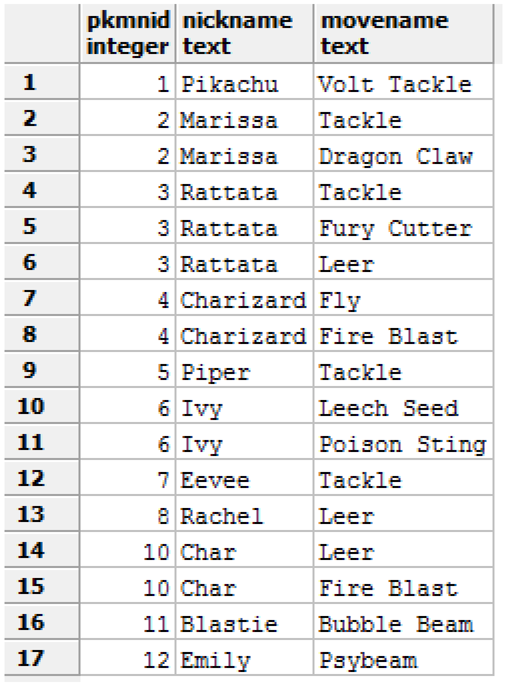
TrainerPokemonNames shows the all of a Trainer's Pokemon, including that Pokemon's name, nickname, level, and total HP. The name of the trainer is shown next to the tid. This view is to give trainers a rundown on their opponents and their teams.



	tid integer	tname text	nickname text	pkmnlevel integer	totalhp integer	pkmnname text
1	1	Ash	Pikachu	30	70	Pikachu
2	1	Ash	Charizard	40	100	Charizard
3	2	Misty	Blastie	40	100	Blastoise
4	3	Gary	Eevee	20	40	Eevee
5	3	Gary	Char	40	100	Charizard
6	4	Youngster Joey	Rattata	5	20	Rattata
7	5	Erika	Ivy	30	60	Ivysaur
8	6	Crystal	Marissa	50	150	Mawile
9	6	Crystal	Emily	23	63	Espeon
10	7	Wattson	Piper	1	10	Pichu
11	8	Gary	Rachel	5	19	Rattata

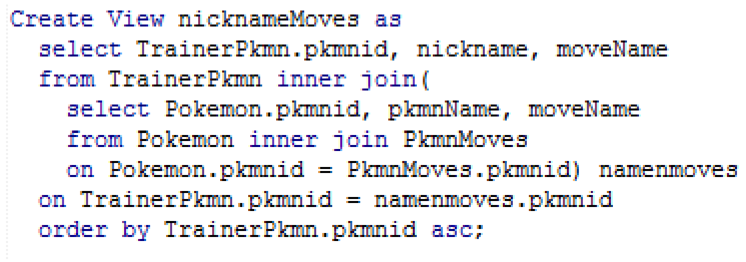
```
Create View TrainerPokemonNames as
  select Trainers.tid, tName, nickname, pkmnLevel, totalHP, pkmnName
  from Trainers inner join(
    select tid, nickname, pkmnName, pkmnLevel, totalHP
    from TrainerPkmn inner join Pokemon
    on TrainerPkmn.pkmnid = Pokemon.pkmnid) tpkmn
  on tpkmn.tid = Trainers.tid
  order by Trainers.tid asc;
```

View Nickname Moves



	pkmnid integer	nickname text	movename text
1	1	Pikachu	Volt Tackle
2	2	Marissa	Tackle
3	2	Marissa	Dragon Claw
4	3	Rattata	Tackle
5	3	Rattata	Fury Cutter
6	3	Rattata	Leer
7	4	Charizard	Fly
8	4	Charizard	Fire Blast
9	5	Piper	Tackle
10	6	Ivy	Leech Seed
11	6	Ivy	Poison Sting
12	7	Eevee	Tackle
13	8	Rachel	Leer
14	10	Char	Leer
15	10	Char	Fire Blast
16	11	Blastie	Bubble Beam
17	12	Emily	Psybeam

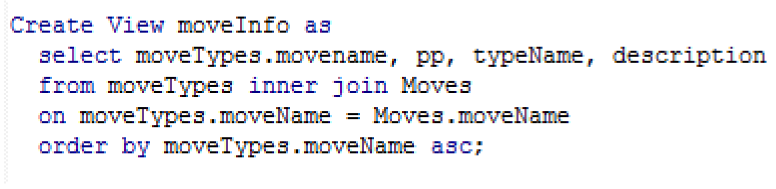
The view NickNameMoves displays all of a Pokemon’s moves. The Pokemon’s nickname is shown next to the pkmnid. This view is so Trainers can see the moves their beloved Pokemon would have. They would probably recognize their nickname for the Pokemon first, but just in case, there is the pkmnid number to be clear.



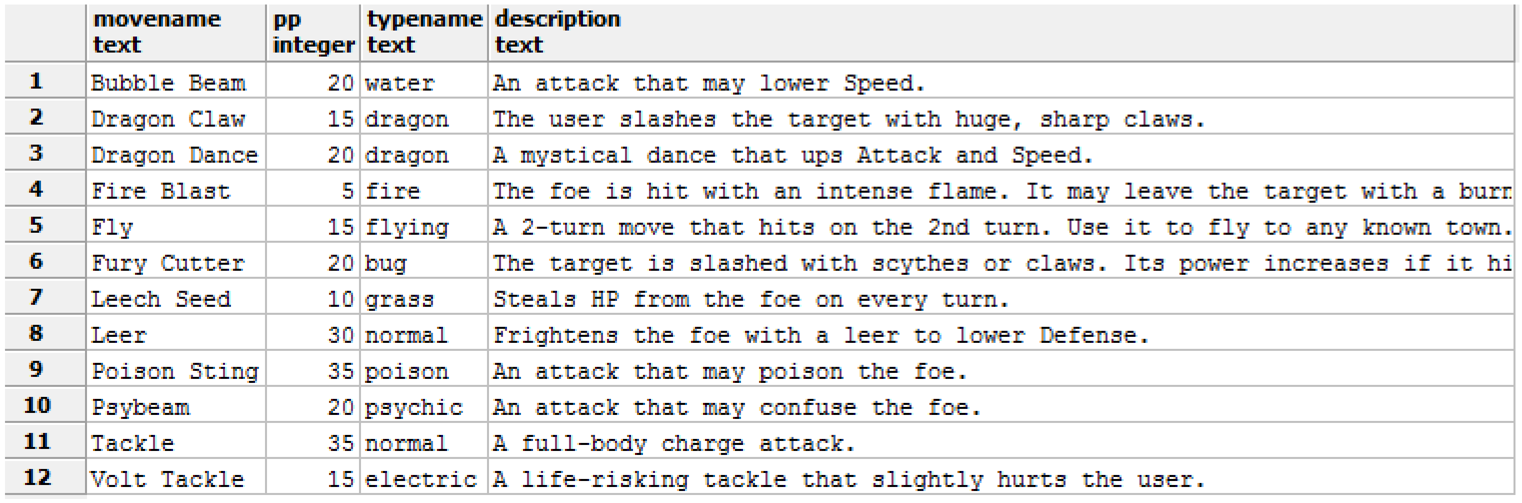
```
Create View nicknameMoves as
  select TrainerPkmn.pkmnid, nickname, moveName
  from TrainerPkmn inner join(
    select Pokemon.pkmnid, pkmnName, moveName
    from Pokemon inner join PkmnMoves
    on Pokemon.pkmnid = PkmnMoves.pkmnid) namenmoves
  on TrainerPkmn.pkmnid = namenmoves.pkmnid
  order by TrainerPkmn.pkmnid asc;
```

View Move Info

The view moveInfo gives all information about a Pokemon move, including the PP, type, and move description.



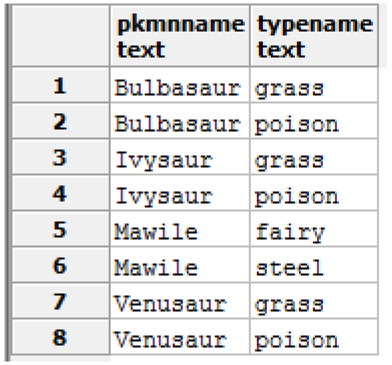
```
Create View moveInfo as
  select moveTypes.movename, pp, typeName, description
  from moveTypes inner join Moves
  on moveTypes.moveName = Moves.moveName
  order by moveTypes.moveName asc;
```



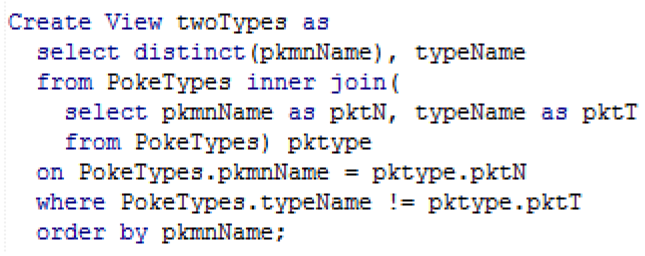
	movename text	pp integer	typename text	description text
1	Bubble Beam	20	water	An attack that may lower Speed.
2	Dragon Claw	15	dragon	The user slashes the target with huge, sharp claws.
3	Dragon Dance	20	dragon	A mystical dance that ups Attack and Speed.
4	Fire Blast	5	fire	The foe is hit with an intense flame. It may leave the target with a burn
5	Fly	15	flying	A 2-turn move that hits on the 2nd turn. Use it to fly to any known town.
6	Fury Cutter	20	bug	The target is slashed with scythes or claws. Its power increases if it hi
7	Leech Seed	10	grass	Steals HP from the foe on every turn.
8	Leer	30	normal	Frightens the foe with a leer to lower Defense.
9	Poison Sting	35	poison	An attack that may poison the foe.
10	Psybeam	20	psychic	An attack that may confuse the foe.
11	Tackle	35	normal	A full-body charge attack.
12	Volt Tackle	15	electric	A life-risking tackle that slightly hurts the user.

View Two Types

This view is to show users all Pokemon that have more than one type assigned to them and to see what those types are.



	pkmnname text	typename text
1	Bulbasaur	grass
2	Bulbasaur	poison
3	Ivysaur	grass
4	Ivysaur	poison
5	Mawile	fairy
6	Mawile	steel
7	Venusaur	grass
8	Venusaur	poison

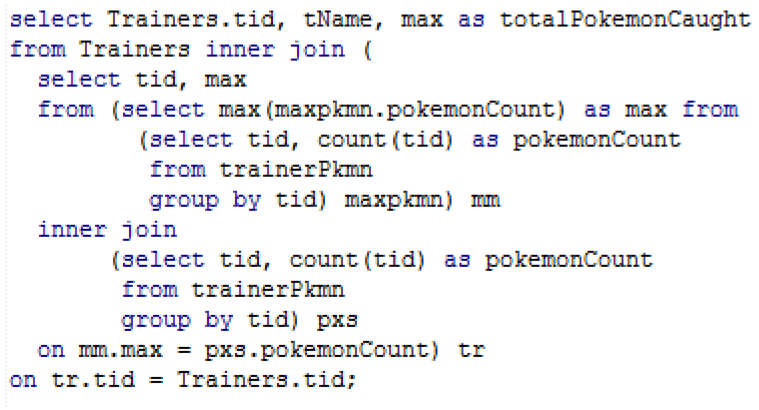


```
Create View twoTypes as
  select distinct(pkmnName), typeName
  from PokeTypes inner join(
    select pkmnName as pktN, typeName as pktT
    from PokeTypes) pktype
  on PokeTypes.pkmnName = pktype.pktN
  where PokeTypes.typeName != pktype.pktT
  order by pkmnName;
```

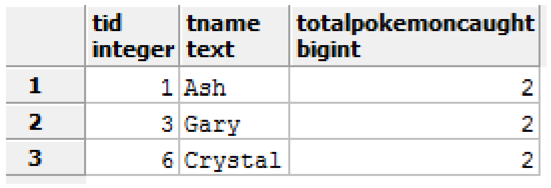
Report #1

Which Trainers (tid and name) have the most Pokemon?

The goal is to catch them all- at least, that's what the slogan says. But who is actually catching the most? This report allows you to see the trainers who owns the most Pokemon, and how many Pokemon they caught.



```
select Trainers.tid, tName, max as totalPokemonCaught
from Trainers inner join (
  select tid, max
  from (select max(maxpkmn.pokemonCount) as max from
         (select tid, count(tid) as pokemonCount
          from trainerPkmn
          group by tid) maxpkmn) mm
  inner join
       (select tid, count(tid) as pokemonCount
        from trainerPkmn
        group by tid) pxs
  on mm.max = pxs.pokemonCount) tr
on tr.tid = Trainers.tid;
```

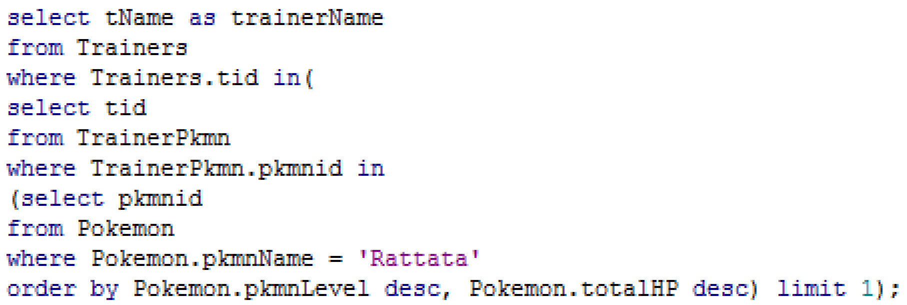


	tid integer	tname text	totalpokemoncaught bigint
1	1	Ash	2
2	3	Gary	2
3	6	Crystal	2

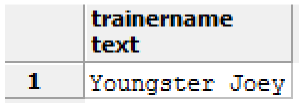
Report #2

Who has the Rattata that is in the top percentage of Rattata?

Youngster Joey will tell anybody that his Rattata is in the Top Percentage of Rattata? But if we take into account levels and HP, is he really all that?



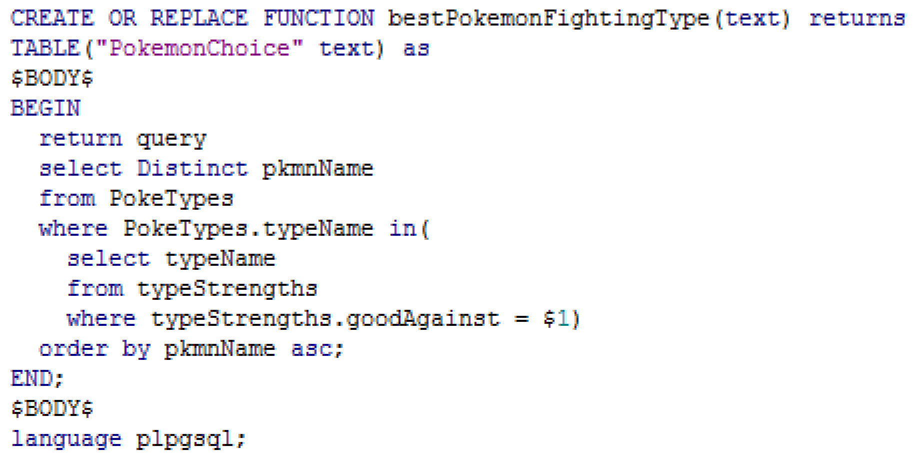
```sql
select tName as trainerName
from Trainers
where Trainers.tid in(
select tid
from TrainerPkmn
where TrainerPkmn.pkmnid in
(select pkmnid
from Pokemon
where Pokemon.pkmnName = 'Rattata'
order by Pokemon.pkmnLevel desc, Pokemon.totalHP desc) limit 1);
```



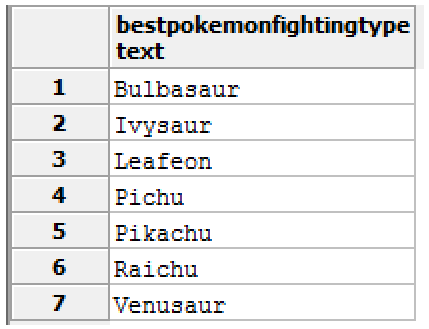
	trainername text
1	Youngster Joey

Stored Procedure bestPokemonFightingType()

If you want to do the best you can in a battle, you’ll need a type advantage. This stored procedure can tell you, if you which Pokemon are best to take on a certain type.



```
CREATE OR REPLACE FUNCTION bestPokemonFightingType(text) returns
TABLE("PokemonChoice" text) as
$BODY$
BEGIN
  return query
  select Distinct pkmnName
  from PokeTypes
  where PokeTypes.typeName in(
    select typeName
    from typeStrengths
    where typeStrengths.goodAgainst = $1)
  order by pkmnName asc;
END;
$BODY$
language plpgsql;
```

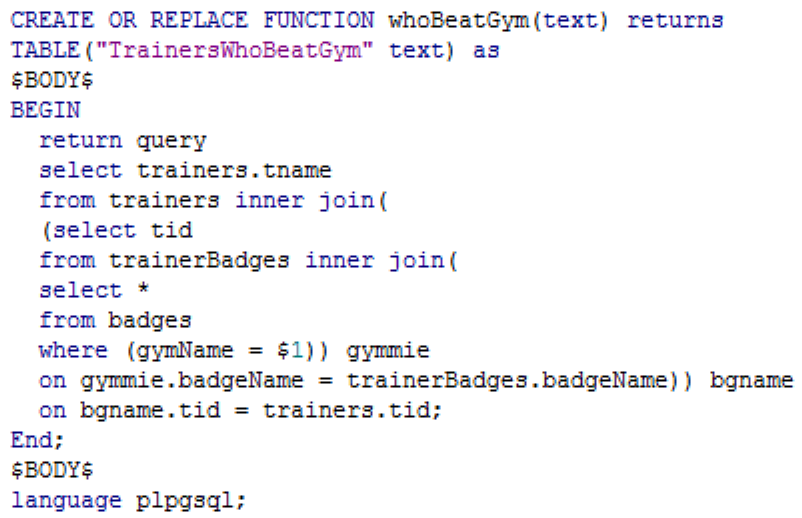


	bestpokemonfightingtype text
1	Bulbasaur
2	Ivysaur
3	Leafeon
4	Pichu
5	Pikachu
6	Raichu
7	Venusaur

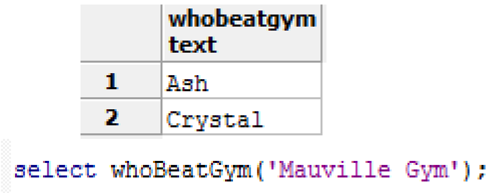
```
select bestPokemonFightingType('water');
```

Stored Procedure whoBeatGym()

This stored procedure shows the names of all trainers who were able to beat a particular gym.



```
CREATE OR REPLACE FUNCTION whoBeatGym(text) returns
TABLE("TrainersWhoBeatGym" text) as
$BODY$
BEGIN
  return query
  select trainers.tname
  from trainers inner join(
  (select tid
  from trainerBadges inner join(
  select *
  from badges
  where (gymName = $1)) gymmie
  on gymmie.badgeName = trainerBadges.badgeName)) bgname
  on bgname.tid = trainers.tid;
End;
$BODY$
language plpgsql;
```

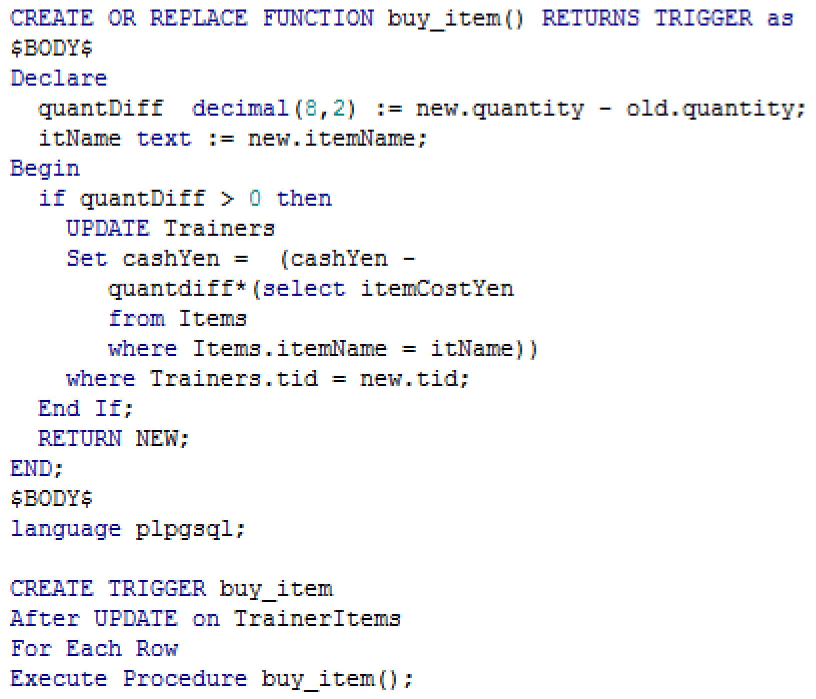


	whobeatgym text
1	Ash
2	Crystal

```
select whoBeatGym('Mauville Gym');
```

Trigger: buy_item()

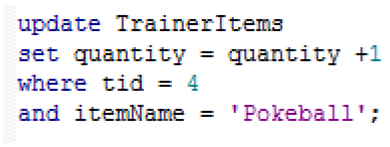
This trigger allows for cash to be subtracted from a Trainer whenever they buy an item from the store (the quantity of their items goes up).



```
CREATE OR REPLACE FUNCTION buy_item() RETURNS TRIGGER as
$BODY$
Declare
  quantDiff  decimal(8,2) := new.quantity - old.quantity;
  itName text := new.itemName;
Begin
  if quantDiff > 0 then
    UPDATE Trainers
    Set cashYen =  (cashYen -
       quantdiff*(select itemCostYen
       from Items
       where Items.itemName = itName))
    where Trainers.tid = new.tid;
  End If;
  RETURN NEW;
END;
$BODY$
language plpgsql;

CREATE TRIGGER buy_item
After UPDATE on TrainerItems
For Each Row
Execute Procedure buy_item();
```

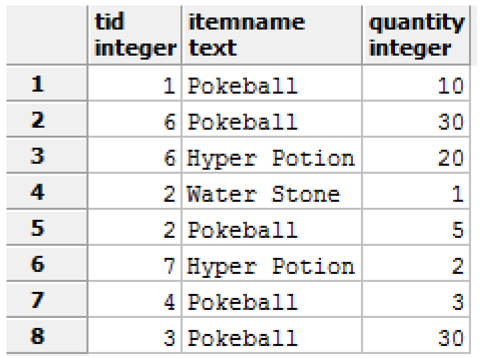
buy_item() Demo



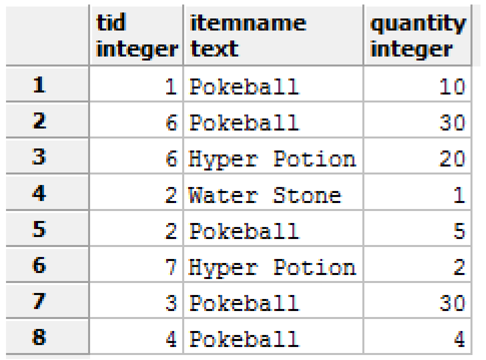
```
update TrainerItems
set quantity = quantity +1
where tid = 4
and itemName = 'Pokeball';
```

Before Update

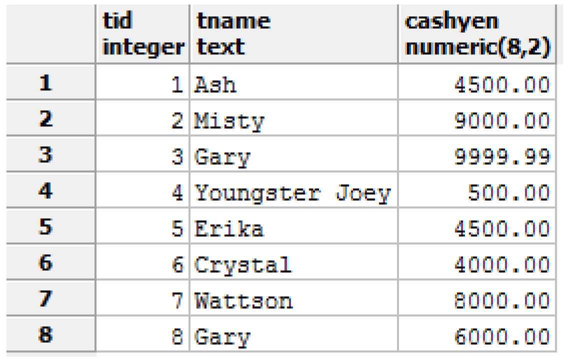
After Update



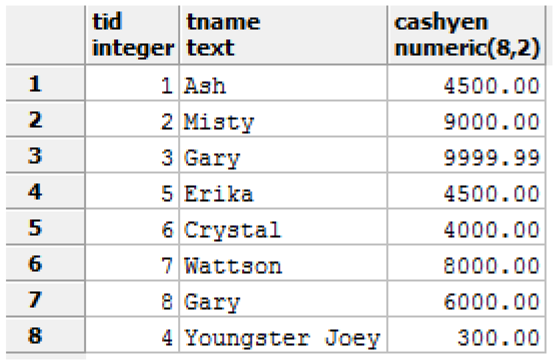
	tid integer	itemname text	quantity integer
1	1	Pokeball	10
2	6	Pokeball	30
3	6	Hyper Potion	20
4	2	Water Stone	1
5	2	Pokeball	5
6	7	Hyper Potion	2
7	4	Pokeball	3
8	3	Pokeball	30



	tid integer	itemname text	quantity integer
1	1	Pokeball	10
2	6	Pokeball	30
3	6	Hyper Potion	20
4	2	Water Stone	1
5	2	Pokeball	5
6	7	Hyper Potion	2
7	3	Pokeball	30
8	4	Pokeball	4



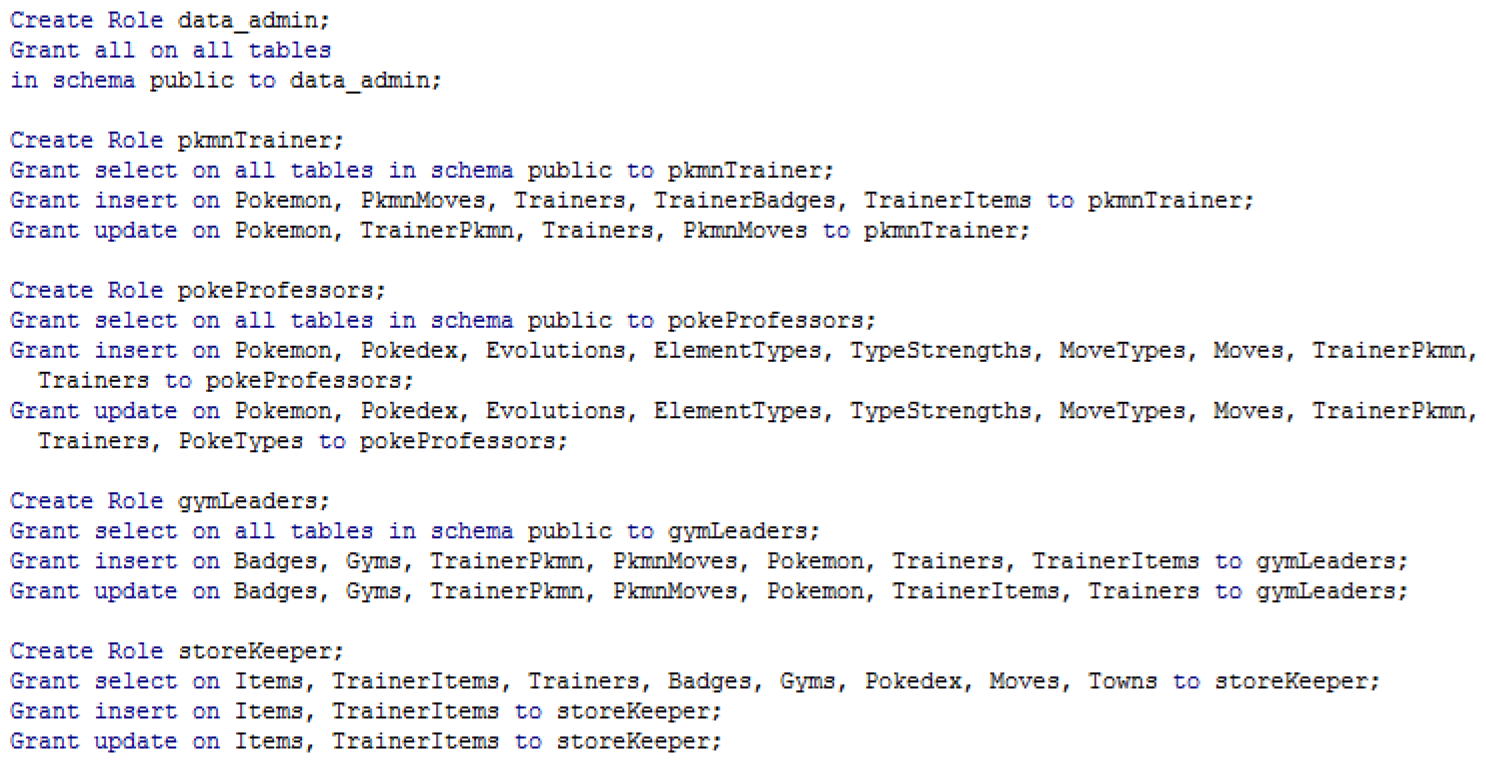
	tid integer	tname text	cashyen numeric(8,2)
1	1	Ash	4500.00
2	2	Misty	9000.00
3	3	Gary	9999.99
4	4	Youngster Joey	500.00
5	5	Erika	4500.00
6	6	Crystal	4000.00
7	7	Wattson	8000.00
8	8	Gary	6000.00



	tid integer	tname text	cashyen numeric(8,2)
1	1	Ash	4500.00
2	2	Misty	9000.00
3	3	Gary	9999.99
4	5	Erika	4500.00
5	6	Crystal	4000.00
6	7	Wattson	8000.00
7	8	Gary	6000.00
8	4	Youngster Joey	300.00

Security

There are five roles designed for this database. One is the database administrator (data_admin), who will have control over the entire database. Secondly, there is the Pokemon trainer, who needs to access and handle their own information and update the Pokemon they have, but should not be changing anything about them in the pokedex or about how they evolve, as they are not professionals. Next are Pokemon Professors (pokeProfessors), who should have a right to edit facts about the Pokemon species and help out trainers. After that, there are Gym Leaders, who should have the right to modify gym related tables. Finally, Storekeepers should be able to handle items, but they shouldn't be able to modify anything about Pokemon.



```
Create Role data_admin;
Grant all on all tables
in schema public to data_admin;

Create Role pkmnTrainer;
Grant select on all tables in schema public to pkmnTrainer;
Grant insert on Pokemon, PkmnMoves, Trainers, TrainerBadges, TrainerItems to pkmnTrainer;
Grant update on Pokemon, TrainerPkmn, Trainers, PkmnMoves to pkmnTrainer;

Create Role pokeProfessors;
Grant select on all tables in schema public to pokeProfessors;
Grant insert on Pokemon, Pokedex, Evolutions, ElementTypes, TypeStrengths, MoveTypes, Moves, TrainerPkmn,
  Trainers to pokeProfessors;
Grant update on Pokemon, Pokedex, Evolutions, ElementTypes, TypeStrengths, MoveTypes, Moves, TrainerPkmn,
  Trainers, PokeTypes to pokeProfessors;

Create Role gymLeaders;
Grant select on all tables in schema public to gymLeaders;
Grant insert on Badges, Gyms, TrainerPkmn, PkmnMoves, Pokemon, Trainers, TrainerItems to gymLeaders;
Grant update on Badges, Gyms, TrainerPkmn, PkmnMoves, Pokemon, TrainerItems, Trainers to gymLeaders;

Create Role storeKeeper;
Grant select on Items, TrainerItems, Trainers, Badges, Gyms, Pokedex, Moves, Towns to storeKeeper;
Grant insert on Items, TrainerItems to storeKeeper;
Grant update on Items, TrainerItems to storeKeeper;
```

Known Problems / The Future

Normally, even if a move has the effect of two different types, generally it is only classified under one type. However, this database allows for moves to have more than one type, as it would be easier to process type advantages and weaknesses this way. The database also does not currently account for the type of strength or weakness that one type has upon another type. For example, fairy Pokemon are immune to dragon moves, and water type gets double damage from electric type moves.

In the future, I would like to include stores and Pokemon Centers (places Pokemon get healed) in the database to give it a wider appeal. I would also have it be able to handle people other than trainers in the Pokemon world, such as Pokemon professors and Pokemon breeders. I would also like there to be more focus on the battle aspect. There could, for example, be a table denoting different Pokemon battles that took place in the world, and make note of who won. There could also be a table for the damage a Pokemon has taken, or any status effects that are currently damaging them (poisoned, paralyzed, fainted, etc).