Ted Sanjeevi
CS340
Database Project

The content of this database is about the Pokemon cartoon series. It has information about Pokemon from the first generation of the series. This database is a collection of information about each species of Pokemon. In the show/game there is also a similar database called the “PokeDex” which is used to gain information on Pokemon such as element types, skills, evolution forms and where they can be found. This database is used as a reference to look up information on old Pokemon and to log information on new Pokemon.

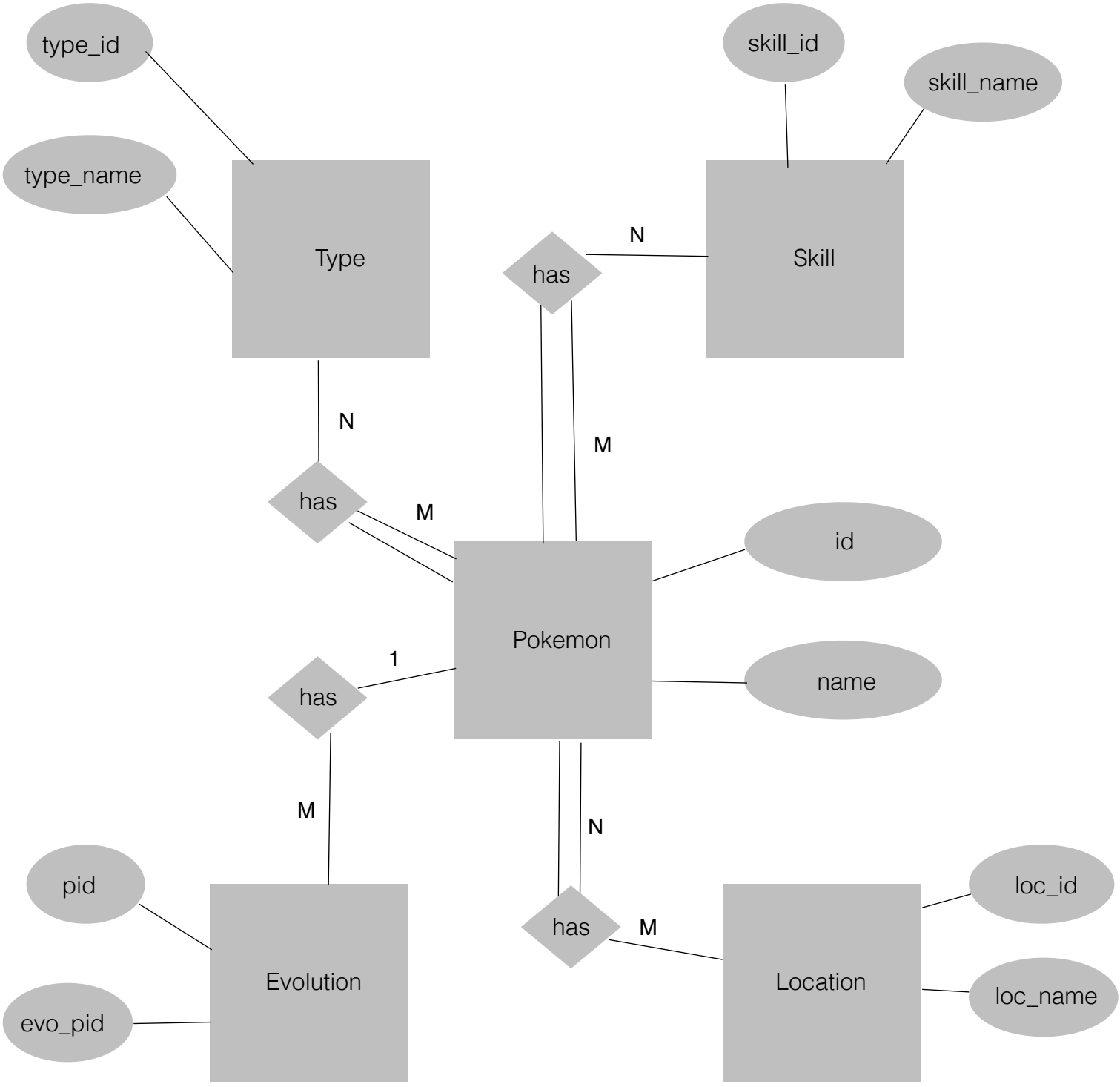
The database contains Pokemon names, Types, Skills, Evolutions, and Locations. The Pokemon table contains the Pokemon name and an ID that is auto incrementing. This ID is not related to the actual cartoon IDs. The name and id must be unique because there only needs to be one of each Pokemon. The Types, Skills, Evolution, and Location tables all have auto incrementing IDS with unique names because each entry only needs to appear once.

Many to many relationship tables
Add Type to Pokemon table: Pokemon can have many types and each types can be many Pokemon. This table contains a Pokemon Id and an Type Id which are the primary keys and reference the Pokemon table and Type table.

Add Skill to Pokemon table: Pokemon can have many skills and each skill can be learned by many Pokemon. This table contains a Pokemon Id and an Skill Id which are the primary keys and reference the Pokemon table and Skill table.

Add location to Pokemon table: Pokemon can have many locations and each locations can have many Pokemon. This table contains a Pokemon Id and an Location Id which are the primary keys and references the Pokemon table and Location table.

One to many relationship
Add Evolution to Pokemon table: Pokemon can have more than one evolution but each evolution can only be from one Pokemon. This table contains a Pokemon Id and another Pokemon Id which are the primary keys. This table references the Pokemon table to match up Pokemon to their evolution forms.


type_id
type_name
Type
skill_id
skill_name
Skill
has
N
N
has
M
M
id
Pokemon
1
has
name
M
N
pid
evo_pid
Evolution
has
M
loc_id
loc_name
Location

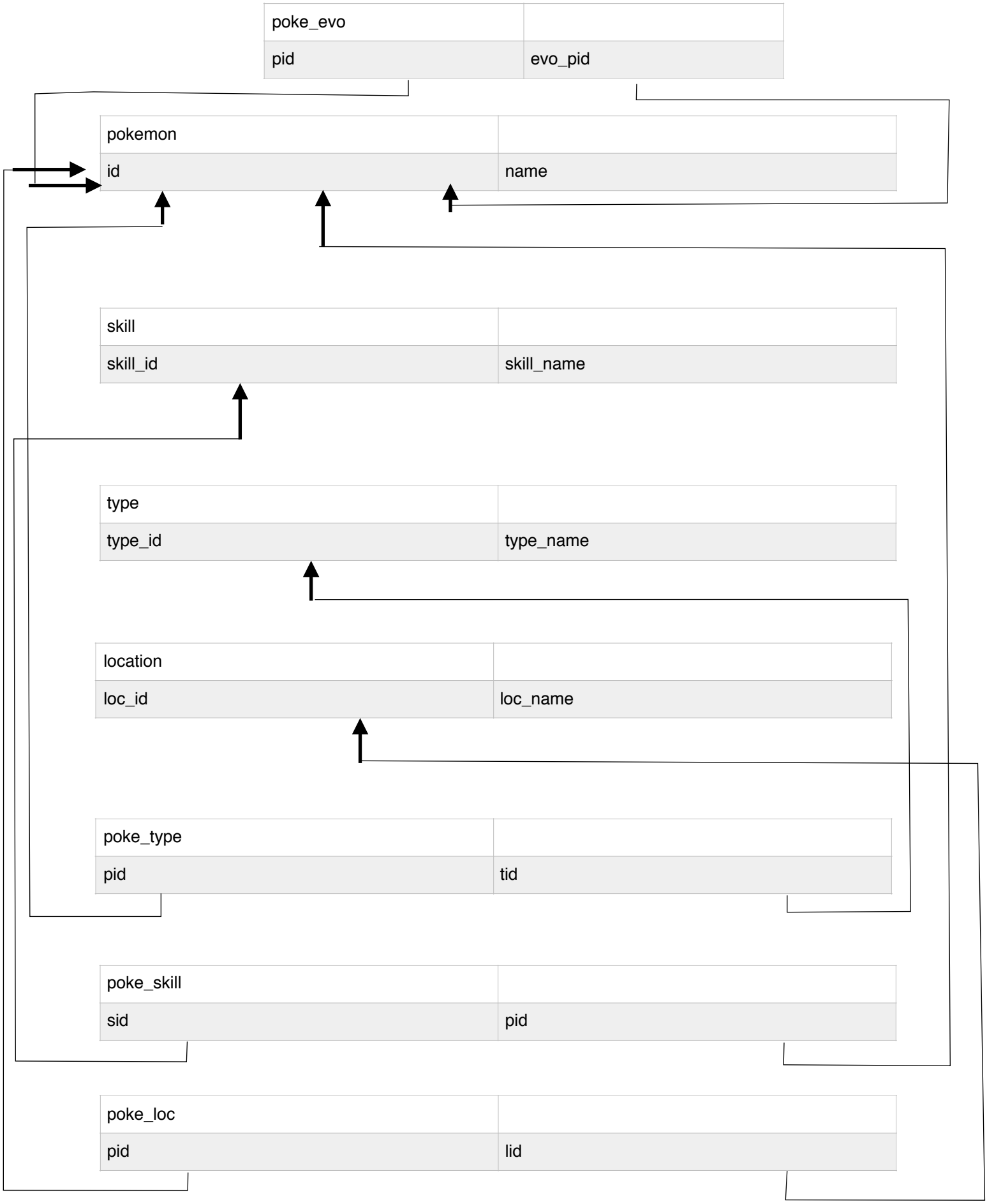

poke_evo
pid
evo_pid
pokemon
id
name
skill
skill_id
skill_name
type
type_id
type_name
location
loc_id
loc_name
poke_type
pid
tid
poke_skill
sid
pid
poke_loc
pid
lid

Table Creation

```
DROP TABLE IF EXISTS `poke_type`;
DROP TABLE IF EXISTS `poke_evo`;
DROP TABLE IF EXISTS `poke_skill`;
DROP TABLE IF EXISTS `poke_loc`;
DROP TABLE IF EXISTS `pokemon`;
DROP TABLE IF EXISTS `type`;
DROP TABLE IF EXISTS `skill`;
DROP TABLE IF EXISTS `location`;


CREATE TABLE `pokemon` (
	`id` int(11) NOT NULL AUTO_INCREMENT,
	`name` varchar(255) NOT NULL,
	PRIMARY KEY (`id`),
	UNIQUE KEY (`name`)
) ENGINE=InnoDB;

CREATE TABLE `type` (
	`type_id` int(11) NOT NULL AUTO_INCREMENT,
	`type_name` varchar(255) NOT NULL,
	PRIMARY KEY (`type_id`),
	UNIQUE KEY (`type_name`)
) ENGINE=InnoDB;

CREATE TABLE `skill` (
	`skill_id` int(11) NOT NULL AUTO_INCREMENT,
	`skill_name` varchar(255) NOT NULL,
	PRIMARY KEY (`skill_id`),
	UNIQUE KEY (`skill_name`)
) ENGINE=InnoDB;

CREATE TABLE `location` (
	`loc_id` int(11) NOT NULL AUTO_INCREMENT,
	`loc_name` varchar(255) NOT NULL,
	PRIMARY KEY (`loc_id`),
	UNIQUE KEY (`loc_name`)
) ENGINE=InnoDB;

CREATE TABLE `poke_type` (
	`pid` int(11) NOT NULL,
	`tid` int(11) NOT NULL,
	PRIMARY KEY (`pid`, `tid`),
	FOREIGN KEY (`pid`) REFERENCES `pokemon` (`id`)
		ON DELETE CASCADE,
	FOREIGN KEY (`tid`) REFERENCES `type` (`type_id`)
		ON DELETE CASCADE
) ENGINE=InnoDB;
```

```
CREATE TABLE `poke_evo` (
	`pid` int(11) NOT NULL,
	`evo_pid` int(11) NOT NULL,
	PRIMARY KEY (`pid`, `evo_pid`),
	FOREIGN KEY (`pid`) REFERENCES `pokemon` (`id`)
		ON DELETE CASCADE,
	FOREIGN KEY (`evo_pid`) REFERENCES `pokemon` (`id`)
		ON DELETE CASCADE
) ENGINE=InnoDB;

CREATE TABLE `poke_skill` (
	`sid` int(11) NOT NULL,
	`pid` int(11) NOT NULL,
	PRIMARY KEY (`sid`, `pid`),
	FOREIGN KEY (`sid`) REFERENCES `skill` (`skill_id`)
		ON DELETE CASCADE,
	FOREIGN KEY (`pid`) REFERENCES `pokemon` (`id`)
		ON DELETE CASCADE
) ENGINE=InnoDB;

CREATE TABLE `poke_loc` (
	`pid` int(11) NOT NULL,
	`lid` int(11) NOT NULL,
	PRIMARY KEY (`pid`, `lid`),
	FOREIGN KEY (`pid`) REFERENCES `pokemon` (`id`)
		ON DELETE CASCADE,
	FOREIGN KEY (`lid`) REFERENCES `location` (`loc_id`)
		ON DELETE CASCADE
) ENGINE=InnoDB;
```

General Use Query

filter by type:

```
SELECT p1.name , type.type_name, skill.skill_name, location.loc_name, p2.name
        FROM pokemon p1
        LEFT JOIN poke_type ON p1.id = poke_type.pid
        LEFT JOIN type ON poke_type.tid = type.type_id
        LEFT JOIN poke_skill ON p1.id = poke_skill.pid
        LEFT JOIN skill ON poke_skill.sid = skill.skill_id
        LEFT JOIN poke_loc ON p1.id = poke_loc.pid
        LEFT JOIN location ON poke_loc.lid = location.loc_id
        LEFT JOIN poke_evo ON p1.id = poke_evo.pid
        LEFT JOIN pokemon p2 ON poke_evo.evo_pid = p2.id
        WHERE type.type_id = [ type id ]
        ORDER BY p1.name ASC
```

filter by skill:

```
SELECT p1.name , type.type_name, skill.skill_name, location.loc_name, p2.name
        FROM pokemon p1
        LEFT JOIN poke_type ON p1.id = poke_type.pid
        LEFT JOIN type ON poke_type.tid = type.type_id
        LEFT JOIN poke_skill ON p1.id = poke_skill.pid
        LEFT JOIN skill ON poke_skill.sid = skill.skill_id
        LEFT JOIN poke_loc ON p1.id = poke_loc.pid
        LEFT JOIN location ON poke_loc.lid = location.loc_id
        LEFT JOIN poke_evo ON p1.id = poke_evo.pid
        LEFT JOIN pokemon p2 ON poke_evo.evo_pid = p2.id
        WHERE skill.skill_id = [ skill id ]
        ORDER BY p1.name ASC”
```

filter by pokemon:

```
SELECT p1.name , type.type_name, skill.skill_name, location.loc_name, p2.name
        FROM pokemon p1
        LEFT JOIN poke_type ON p1.id = poke_type.pid
        LEFT JOIN type ON poke_type.tid = type.type_id
        LEFT JOIN poke_skill ON p1.id = poke_skill.pid
        LEFT JOIN skill ON poke_skill.sid = skill.skill_id
        LEFT JOIN poke_loc ON p1.id = poke_loc.pid
        LEFT JOIN location ON poke_loc.lid = location.loc_id
        LEFT JOIN poke_evo ON p1.id = poke_evo.pid
        LEFT JOIN pokemon p2 ON poke_evo.evo_pid = p2.id
        WHERE p1.id = [ pokemon id]
```

filter by location:

```
SELECT p1.name , type.type_name, skill.skill_name, location.loc_name, p2.name
        FROM pokemon p1
        LEFT JOIN poke_type ON p1.id = poke_type.pid
        LEFT JOIN type ON poke_type.tid = type.type_id
```

```
LEFT JOIN poke_skill ON p1.id = poke_skill.pid
LEFT JOIN skill ON poke_skill.sid = skill.skill_id
LEFT JOIN poke_loc ON p1.id = poke_loc.pid
LEFT JOIN location ON poke_loc.lid = location.loc_id
LEFT JOIN poke_evo ON p1.id = poke_evo.pid
LEFT JOIN pokemon p2 ON poke_evo.evo_pid = p2.id
WHERE location.loc_id = [ location ]
```

Delete pokemon
'DELETE FROM `pokemon` WHERE id = [pokemon id]

Select pokemon id from pokemon table
SELECT id FROM pokemon WHERE id = [pokemon id]

Select type id from type table
SELECT type_id FROM type WHERE type_id = [type id]

Insert in to poke_type table
INSERT INTO poke_type (pid, tid) VALUES [pokemon id] , [type id] ’

Insert in to type table
INSERT INTO type (type_name) VALUES [new element type]

Insert in to poke_skill table
INSERT INTO poke_skill (sid, pid) VALUES [skill id] , [pokemon id]

Insert in to skill table
INSERT INTO skill (skill_name) VALUES [new skill name]“

Insert in to location table
INSERT INTO location (loc_name) VALUES [new location name]

Insert in to poke_loc table
INSERT INTO poke_loc (lid, pid) VALUES [location id], [pokemon id]

Insert in to poke_evo table
INSERT INTO poke_evo (pid, evo_pid) VALUES ([pokemon id], [pokemon id])

Insert in to pokemon table
INSERT INTO pokemon (name) VALUES [new pokemon name]