

Pokemon Database

Dani Anderson
Alexis Perez
Fischer Wells
CS3530-602 Database Theory

Table of Contents

Executive Summary.....	3
Vision and Objective.....	4
Mission Statement.....	5
Service.....	5
Use Case.....	6
Business Requirements.....	6
Comprehensive Pokémon Information Management.....	6
Regular Updates and Maintenance.....	7
Community Engagement.....	7
ER Diagram.....	8
ER Diagram Description.....	9
Development and Current Release.....	12
Minimum Viable Product (MVP).....	11
Expanding Functionality.....	11
Future Releases.....	12
Appendix.....	15
SQL Table Creation Scripts.....	14
SQL Data Population Scripts.....	16
Database Samples.....	18
Pokemon_Types.....	18
Moves.....	19
Pokemon.....	19
Abilities.....	19
Regions.....	20
Pokemon_Abilities.....	20
Time Log.....	20
Totals.....	21

Executive Summary

In the ever-evolving world of Pokémon, players continually seek to enhance their gaming experience by delving deeper into the vast array of Pokémon species, attributes, and strategies. For many, the challenge lies not only in mastering battles and capturing Pokémon but also in accessing reliable and comprehensive information about these creatures. Recognizing this need, our team has embarked on the creation of a dedicated Pokémon database designed to revolutionize the way players interact with and obtain information about Pokémon.

The primary motivation behind this project stems from the gaps and inefficiencies observed in existing Pokémon resources. While numerous platforms offer information on Pokémon, they often lack the depth, accuracy, and user-friendliness that players require. Our database aims to fill these voids by providing a centralized, well-organized repository of Pokémon data that is both exhaustive and easily navigable.

Our key objectives include comprehensive data collection, making sure we have all information available including stats, abilities, habitats and more. We also strive to have the most up-to-date and accurate data, and engage with the community to keep our database up to date. This will provide many benefits to players, providing quick and reliable access to detailed Pokémon

information, players can make informed decisions and refine their strategies, leading to a more rewarding gaming experience.

Our Pokémon database is more than just a repository of information; it is a tool designed to enhance the overall gaming experience for Pokémon enthusiasts. By addressing the limitations of existing resources and offering a comprehensive, user-friendly, and reliable platform, we aim to become the go-to source for all things Pokémon. As the Pokémon community continues to grow and evolve, our database will stand as a testament to our commitment to supporting players in their quest to become Pokémon Masters.

Vision and Objective

Our vision is to empower Pokémon players around the world by providing them with increased access to comprehensive Pokémon information, thereby enhancing their overall gaming experience. We aspire to create a platform that serves as the ultimate resource for Pokémon enthusiasts, enabling them to discover, learn, and strategize with ease. By offering accurate, up-to-date, and detailed data on every aspect of the Pokémon universe, we aim to support players in their journey to become the best Pokémon trainers they can be.

Our objective is to build a user-friendly and reliable Pokémon database that caters to the needs of players at all levels. We are committed to delivering an intuitive interface, robust search and filtering capabilities, and valuable insights that help players make informed decisions. By continuously updating our database to reflect the latest developments in the Pokémon franchise, we ensure that our users always have access to the most current information. Ultimately, our goal is to foster a vibrant community of knowledgeable and passionate Pokémon players who can share

their expertise and experiences, contributing to the collective growth and enjoyment of the game.

Mission Statement

Our mission is to empower Pokémon players worldwide by providing comprehensive, up-to-date information through an intuitive and reliable database, enhancing their gaming experience and fostering a vibrant community of knowledgeable enthusiasts.

Service

The Pokémon Database offers a comprehensive suite of user-friendly tools and services designed to meet the diverse needs of Pokémon players at all levels. Our system prioritizes accessibility and efficiency, enabling users to easily navigate and utilize the wealth of information available. Our key features include:

1. Extensive Pokémon Index: Detailed information on every Pokémon, including stats, abilities, evolutions, and habitats.
2. Regular Updates: Continuous updates to ensure the database reflects the latest game releases, updates, and meta shifts.

3. Transparency: Our database is open-source so that the community can validate, and contribute to, the database, allowing them to feel confident in the data that we make available.

Use Case

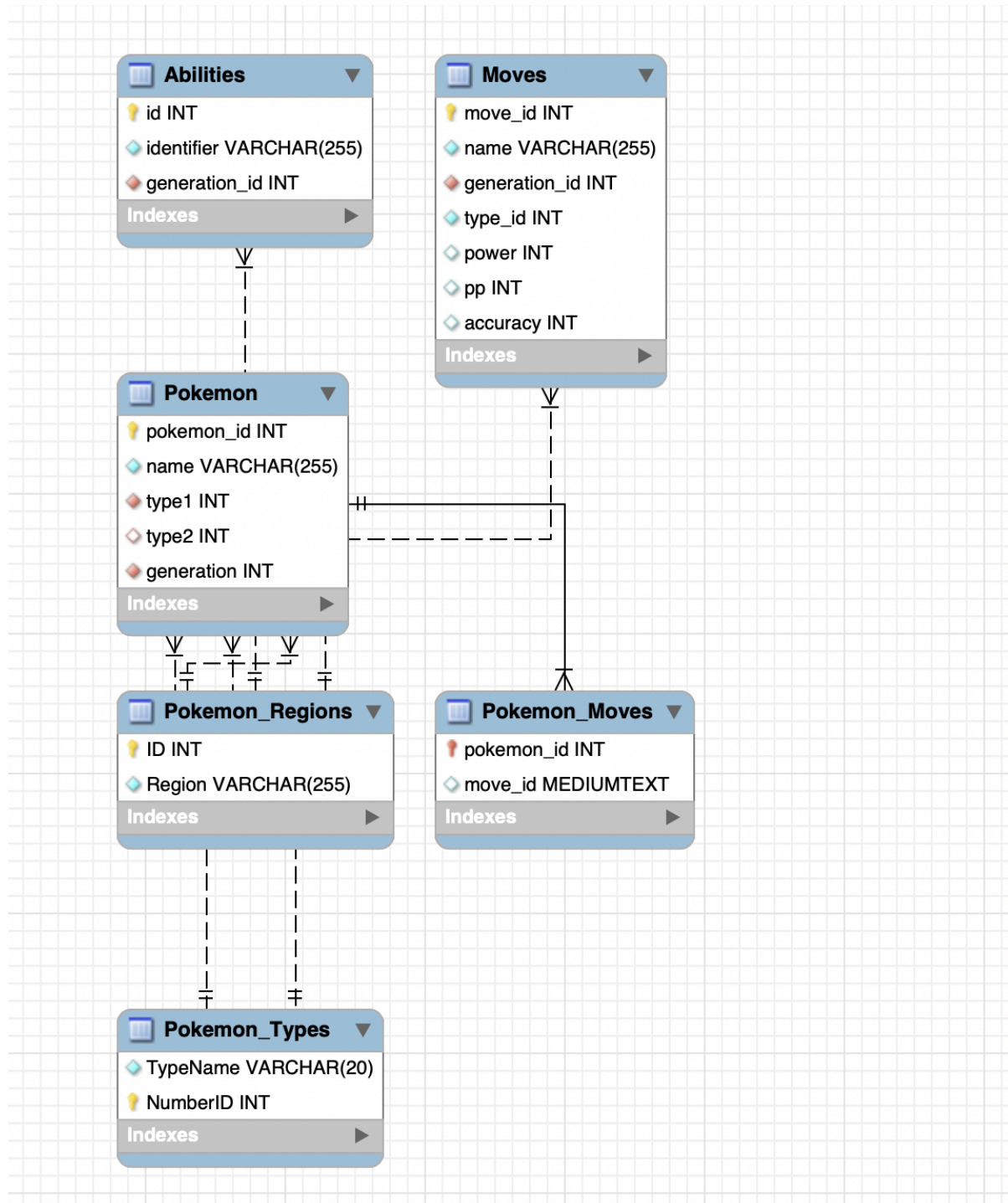
Consider a new Pokémon player progressing through the game, excitedly collecting various Pokémon, but soon feeling overwhelmed by the multitude of choices and strategies. Which Pokémon should they focus on evolving? Which Pokémon will synergize best to create a formidable team? Where can they find the specific Pokémon they want to catch? These common challenges can hinder the enjoyment and progression of the game. Our database is designed to address these issues comprehensively, providing clear and actionable information. It is equally valuable for advanced players seeking to refine their strategies and optimize their gameplay, ensuring that players of all levels can enjoy a more streamlined and enjoyable Pokémon experience.

Business Requirements

- Comprehensive Pokémon Information Management

- Store and manage detailed attributes of each Pokémon, including stats, evolutions, abilities, habitats, and more.
- Enable tracking of changes in Pokémon data, such as updates from new game releases or meta shifts.
- Ensure the database accommodates the evolving needs and strategies of Pokémon players, allowing them to better plan team compositions, move sets, and counters for specific challenges.
- Regular Updates and Maintenance
 - With each Pokemon game, more Pokemon will need added to the database. We will regularly maintain the database to have the most current, accurate information.
- Community Engagement
 - By making the database open source we allow users to share their insights and data, contributing to a vibrant and knowledgeable Pokémon community.
 - The open source nature of the database also allows us the community to validate our data and identify gaps or errors in the data, which we will allow us to resolve any issues quickly.

ER Diagram



ER Diagram Description

- **Abilities Table:**

- id (INT, Primary Key): Unique identifier for each ability.
- identifier (VARCHAR(255)): Name of the ability.
- generation_id (INT): Identifier for the generation in which the ability was introduced.

- **Moves Table:**

- move_id (INT, Primary Key): Unique identifier for each move.
- name (VARCHAR(255)): Name of the move.
- generation_id (INT): Identifier for the generation in which the move was introduced.
- type_id (INT): Identifier for the type of the move.
- power (INT): Power of the move.
- pp (INT): Power points (PP) of the move.
- accuracy (INT): Accuracy of the move.

- **Pokemon Table:**

- pokemon_id (INT, Primary Key): Unique identifier for each Pokémon.
- name (VARCHAR(255)): Name of the Pokémon.
- type1 (INT): Identifier for the primary type of the Pokémon.
- type2 (INT): Identifier for the secondary type of the Pokémon (if any).
- generation (INT): Identifier for the generation in which the Pokémon was introduced.

- **Pokemon_Regions Table:**

- ID (INT, Primary Key): Unique identifier for each region.

- Region (VARCHAR(255)): Name of the region.
- Pokemon_Types Table:
- TypeName (VARCHAR(20)): Name of the type.
- NumberID (INT, Primary Key): Unique identifier for each type.
- **Pokemon_Moves Table:**
 - pokemon_id (INT, Foreign Key): References the pokemon_id in the Pokémon table.
 - move_id (MEDIUMTEXT, Foreign Key): References the move_id in the Moves table.
- **The relationship between the Pokémon and Abilities tables is one-to-many:**
 - Each Pokémon can have none, one, or many abilities listed in the Abilities table.
 - Each entry in the Abilities table must be related to one generation in the Pokémon table.
- **The relationship between the Pokémon and Pokémon_Moves tables is one-to-many:**
 - Each Pokémon can have none, one, or many moves listed in the Pokémon_Moves table.
 - Each entry in the Pokémon_Moves table must be related to one Pokémon in the Pokémon table.
- **The relationship between the Moves and Pokémon_Moves tables is one-to-many:**
 - Each move can be associated with none, one, or many Pokémon through the Pokémon_Moves table.
 - Each entry in the Pokémon_Moves table must be related to one move in the Moves table.
- **The relationship between the Pokémon and Pokémon_Regions tables is one-to-many:**
 - Each Pokémon can be found in one or many regions listed in the Pokémon_Regions table.

- Each entry in the Pokémon_Regions table must be related to one region in the Pokémon table.
- **The relationship between the Pokémon and Pokémon_Types tables is many-to-many:**
 - Each Pokémon can have one or two types listed in the Pokémon_Types table.
 - Each entry in the Pokémon_Types table must be related to one or many Pokémon in the Pokémon table.

Development and Current Release

Minimum Viable Product (MVP)

In this stage of our project, we aimed to deliver a Minimum Viable Product (MVP). This entails creating a fully functional database that can be navigated and queried using a SQL editor. The MVP provides essential access to detailed Pokémon information, allowing players to retrieve and utilize data efficiently. This foundational stage ensures that players can easily find and analyze key attributes, moves, abilities, and stats of various Pokémon, thereby enhancing their gaming experience and strategy development.

Expanding Functionality

In future stages of our project, we plan to develop a user-friendly front-end interface connected to the database, enabling Pokémon players to access helpful data without requiring SQL knowledge or an editor setup. Additionally, we aim to establish a GitHub repository where anyone can join and

submit merge requests to suggest changes and improvements to the database. By making the project open source, we encourage community involvement and collaboration, with our team acting as moderators to approve merge requests that enhance the user experience and benefit the overall project. This approach ensures continuous improvement and inclusivity, fostering a vibrant and dynamic Pokémon database.

Future Releases

With future releases, we plan to significantly expand the functionality and scope of our Pokémon database. One of our primary goals is to implement additional tables to encompass a broader range of data, including cities, gyms, and game versions. By integrating this detailed information, we aim to provide a more comprehensive resource for players, allowing them to explore the Pokémon world more thoroughly and strategize their gameplay with even greater precision.

In addition to expanding the database structure, we are committed to building a robust front-end web application. This application will feature an intuitive interface that simplifies the process of searching and accessing the database, making it user-friendly for everyone, regardless of their technical background. This front-end will include advanced search capabilities, filters, and visualization tools to help users quickly find the information they need. Our objective is to make the wealth of data in our database easily accessible and navigable for all Pokémon enthusiasts.

Furthermore, we plan to publish basic strategy guides and develop better tools to encourage and manage contributions to our open-source database. By offering strategy guides, we aim to assist players in optimizing their gameplay and making informed decisions. To support community involvement, we will enhance our GitHub repository with clear guidelines and streamlined processes for submitting and reviewing merge requests. This will foster a collaborative environment where contributors can share their expertise and insights, with our team ensuring the quality and relevance of the additions. Through these efforts, we aim to create a dynamic, community-driven platform that continually evolves to meet the needs of Pokémon players worldwide.

Conclusion

Our Pokémon database project represents a significant step forward in providing players with the comprehensive, reliable, and easily accessible information they need to enhance their gaming experience. From the initial development of a robust and navigable database to the future expansion of functionality and community involvement, our goal is to create the ultimate resource for Pokémon enthusiasts. By integrating detailed data, developing user-friendly interfaces, and fostering an open-source environment, we are committed to supporting players at all levels. As we continue to evolve and grow, we look forward to empowering the Pokémon community with the tools and knowledge to excel in their journey to become Pokémon Masters.

Appendix

SQL Table Creation Scripts

```
CREATE DATABASE pokemon_db;  
USE pokemon_db;
```

```
CREATE TABLE Pokemon_Regions (  
    ID INT PRIMARY KEY,  
    Region VARCHAR(255) NOT NULL  
);
```

```
CREATE TABLE Abilities (  
    id INT PRIMARY KEY,  
    identifier VARCHAR(255) NOT NULL,  
    generation_id INT NOT NULL,  
    FOREIGN KEY (generation_id) REFERENCES  
Pokemon_Regions(ID)  
);
```

```
CREATE TABLE Pokemon_Types (  
    TypeName VARCHAR(20) NOT NULL,
```

```

    NumberID INT PRIMARY KEY
);

```

```

CREATE TABLE Moves (
    move_id INT PRIMARY KEY,
    name VARCHAR(255) NOT NULL,
    generation_id INT NOT NULL,
    type_id INT NOT NULL,
    power INT,
    pp INT,
    accuracy INT,
    FOREIGN KEY (generation_id) REFERENCES
Pokemon_Regions(ID)
);

```

```

CREATE TABLE Pokemon (
    pokemon_id INT PRIMARY KEY,
    name VARCHAR(255) NOT NULL,
    type1 INT NOT NULL,
    type2 INT DEFAULT NULL,
    generation INT NOT NULL,
    FOREIGN KEY (type1) REFERENCES
Pokemon_Types(NumberID),
    FOREIGN KEY (type2) REFERENCES
Pokemon_Types(NumberID),
    FOREIGN KEY (generation) REFERENCES
Pokemon_Regions(ID)
);

```

```

CREATE TABLE Pokemon_Moves (
    pokemon_id INT,
    move_id MEDIUMTEXT,
    PRIMARY KEY (pokemon_id),

```

```

    FOREIGN KEY (pokemon_id) REFERENCES
    Pokemon(pokemon_id)
);

```

SQL Data Population Scripts

```

-- Load data into Pokemon_Regions table
LOAD DATA INFILE '/var/lib/mysql-files/Pokemon_Regions.csv'
INTO TABLE Pokemon_Regions
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
IGNORE 1 ROWS
(ID, Region);

```

```

-- Load data into TypesOfPokemon table first
LOAD DATA INFILE '/var/lib/mysql-files/pokemon_types.csv'
INTO TABLE Pokemon_Types
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
IGNORE 1 ROWS
(TypeName, NumberID);

```

```

-- Load data into Pokemon table, treating empty strings as NULL
LOAD DATA INFILE '/var/lib/mysql-files/pokemon.csv'
INTO TABLE Pokemon
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
IGNORE 1 ROWS
(pokemon_id, name, type1, @type2, generation)
SET type2 = NULLIF(@type2, "");

```

```

-- Load data into Moves table
LOAD DATA INFILE '/var/lib/mysql-files/moves.csv'

```



```

INTO TABLE Moves
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
IGNORE 1 ROWS
(move_id, name, generation_id, type_id, @power, @pp,
accuracy)
SET power = NULLIF(@power, ""),
pp = NULLIF(@pp, "");

-- Load data into Abilities table
LOAD DATA INFILE '/var/lib/mysql-files/abilities.csv'
INTO TABLE Abilities
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
IGNORE 1 ROWS
(id, identifier, generation_id);

-- Loads data in Pokemon_Moves table
LOAD DATA INFILE '/var/lib/mysql-files/pokemon_moves.csv'
INTO TABLE Pokemon_Moves
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS
(pokemon_id, move_id);

-- Correct table name in the SELECT statement
SELECT * FROM Pokemon_Types LIMIT 10;
SELECT * FROM Pokemon LIMIT 10;
SELECT * FROM Moves LIMIT 10;
SELECT * FROM Abilities LIMIT 10;
SELECT * FROM Pokemon_Regions LIMIT 10;
SELECT * FROM Pokemon_Moves LIMIT 10;

```

Database Samples

Pokemon_Types

	TypeName	NumberID	
	Normal	1	
	Fire	2	
	Water	3	
	Electric	4	
	Grass	5	
	Ice	6	
	Fighting	7	
	Poison	8	
	Ground	9	
	Flying	10	
	NULL	NULL	

Moves

	move_id	name	generation_...	type_id	power	pp	accuracy
	1	pound	1	1	40	35	100
	2	karate-chop	1	2	50	25	100
	3	double-slap	1	1	15	10	85
	4	comet-punch	1	1	18	15	85
	5	mega-punch	1	1	80	20	85
	6	pay-day	1	1	40	20	100
	7	fire-punch	1	10	75	15	100
	8	ice-punch	1	15	75	15	100
	9	thunder-punch	1	13	75	15	100
	10	scratch	1	1	40	35	100

Pokemon

	pokemon_id	name	type1	type2	generation	
	1	bulbasaur	5	8	1	
	2	ivysaur	5	8	1	
	3	venusaur	5	8	1	
	4	charmander	2	NULL	1	
	5	charmeleon	2	NULL	1	
	6	charizard	2	10	1	
	7	squirtle	3	NULL	1	
	8	wartortle	3	NULL	1	
	9	blastoise	3	NULL	1	
	10	caterpie	12	NULL	1	

Abilities

	id	identifier	generation_...
	1	stench	3
	2	drizzle	3
	3	speed-boost	3
	4	battle-armor	3
	5	sturdy	3
	6	damp	3
	7	limber	3
	8	sand-veil	3
	9	static	3
	10	volt-absorb	3

			and ER Diagram	
8/5/24	Dani Anderson	1	Helped team fix Docker issue	
8/5/24	Dani Anderson	1	Worked with professor to figure out populating tables from CSV	

8/6/24	Fischer Wells	4	Finished Paper	
8/6/24	Dani Anderson	3	Assisted teammates with their responsibilities	
8/6/24	Fischer Wells	2	Made slideshow	
8/6/24	Alexis Perez	4	Fixed table bug	

Totals

Name	Time
Dani Anderson	10 hrs.
Alexis Perez	8 hrs.
Fischer Wells	8 hrs.