

# **BLG317E Database Systems Term Project Report**

## **Members**

**Utku Biçer**

Frontend - Data Research - SQL Queries

**Ertuğrul Şentürk**

Backend - Data Research - SQL Queries

**Hüseyin Şimşek**

SQL Queries - Backend

**Muhammed Arif Göktürk**

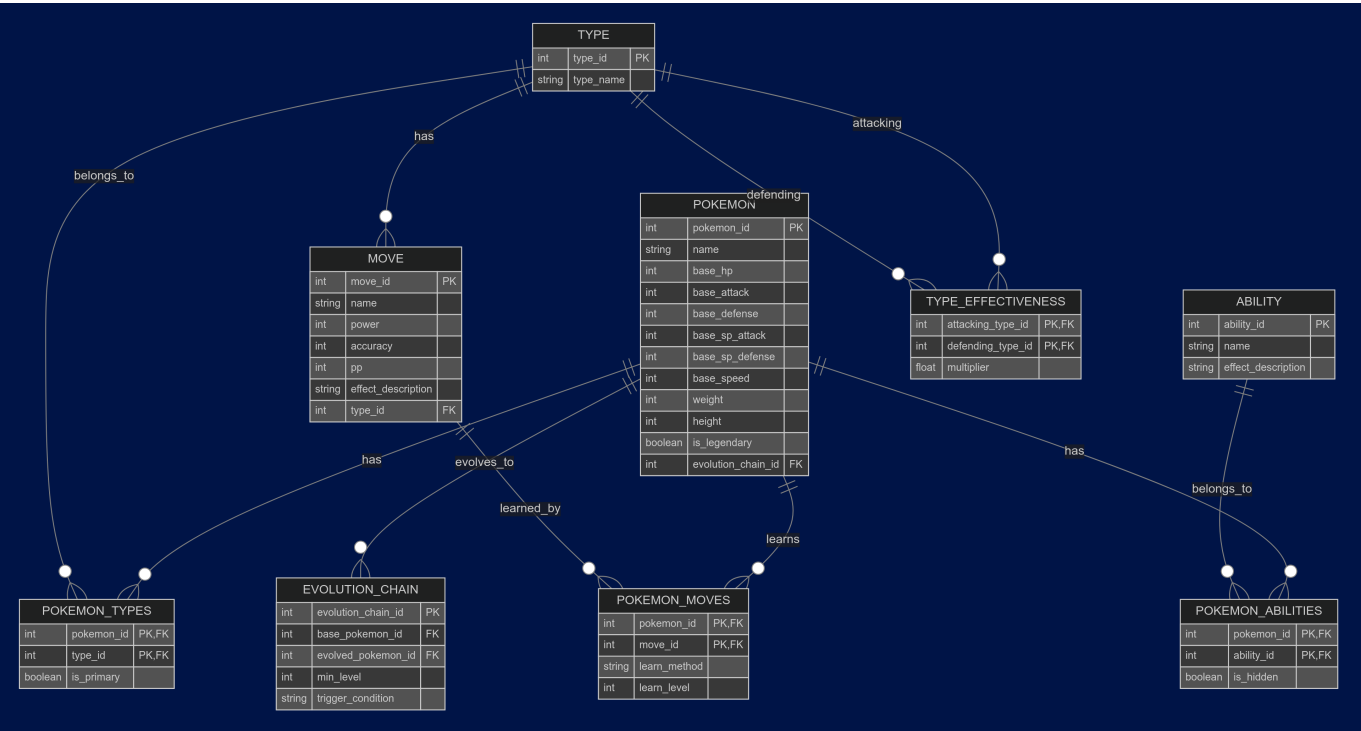
Frontend - SQL Queries

# Database Structure

## Overview

Our database consists of Pokemon data, including information about their height, weight, types, moves and abilities. It is sourced from an open source project, [PokeAPI](#).

## Entity Relationship Diagram (ERD)



# Application

The web app is built with the following stack:

- **Frontend:** Vite + React + shadcn
- **Backend:** NestJS
- **Database:** PostgreSQL

## Screenshots

### Main Page

itü Database Project

Team Members

Search Pokemon...

ID	Name	Weight	Height	
3	venusaur	1000	20	...
4	charmander	85	6	...
5	charmeleon	190	11	...
6	charizard	905	17	...
7	squirtle	90	5	...
8	wartortle	225	10	...
9	blastoise	855	16	...
10	caterpie	29	3	...
11	metapod	99	7	...
12	butterfree	320	11	...

0 of 1306 row(s) selected.

Rows per page10Page 1 of 131

Add Pokémon

Viewing a Pokemon

Pokémon Details

Name:

Utku's Pokemon

Weight (kg):

999

Height (m):

10

Types:

steel

fire

Abilities:

Moves:

pound

Type: normal

Power: 40

Accuracy: 100%

PP: 35

scratch

Type: normal

Power: 40

Accuracy: 100%

PP: 35

pin-missile

Type: bug

Power: 25

Accuracy: 95%

PP: 20

Edit

Adding a Pokemon

Create New Pokémon

Name:

Utku's Pokemon

Weight (kg):

999

Height (m):

10

Types:

normal

fighting

flying

poison

ground

rock

bug

ghost

steel

fire

water

grass

electric

psychic

ice

dragon

dark

fairy

stellar

unknown

shadow

Moves:

pound

karate-chop

double-slap

comet-punch

mega-punch

pay-day

fire-punch

ice-punch

thunder-punch

scratch

vice-grip

guillotine

razor-wind

swords-dance

cut

gust

wing-attack

whirlwind

fly

bind

slam

vine-whip

stomp

double-kick

mega-kick

jump-kick

rolling-kick

sand-attack

headbutt

horn-attack

fury-attack

horn-drill

tackle

body-slam

wrap

take-down

thrash

double-edge

tail-whip

poison-sting

twineedle

pin-missile

leer

bite

growl

roar

sing

supersonic

sonic-boom

disable

acid

ember

flamethrower

mist

water-gun

hydro-pump

surf

Create

Cancel

Editing a Pokemon

### Pokémon Details

Name:

Utku's Pokemon edited

Weight (kg):

999

Height (m):

1011

Types:

steel

fire

Abilities:

Moves:

**pound**

Type: normal

Power: 40

Accuracy: 100%

PP: 35

**scratch**

Type: normal

Power: 40

Accuracy: 100%

PP: 35

**pin-missile**

Type: bug

Power: 25

Accuracy: 95%

PP: 20

Update

Cancel

## Deleting a Pokemon

A screenshot of a web application showing a table with columns 'Height' and 'Actions'. The table has five rows with values 1011, 3, 0, 10, and 123. A context menu is open over the row with value 10, displaying the following options: 'Copy Pokemon name', 'View Pokemon', and 'Delete Pokemon'. A mouse cursor is hovering over the 'Delete Pokemon' option.

Height	Actions
1011	...
3	...
0	...
10	...
123	...

## Searching for a Pokemon

poke				
PK ID	Name	Weight	Height	
10282	Utku's Pokemon edited	999	1011	...
10164	slowpoke-galar	360	12	...
79	slowpoke	360	12	...
0 of 3 row(s) selected.				
		Rows per page	10	Page 1 of 1
		<< < > >>		

## SQL Queries & CRUD Operations

ability/:

- all : returns all rows from the ABILITY table. `SELECT * FROM ability;`
- from\_pokemon : returns the abilities which a specific pokemon has. `SELECT ability.ability_id, ability.name FROM ability JOIN pokemon_abilities ON ability.ability_id = pokemon_abilities.ability_id WHERE pokemon_abilities.pokemon_id = $1;`
- new : creates a new relation in the POKEMON\_ABILITY table from the pokemon ID, ability ID and is hidden values. `INSERT INTO pokemon_abilities VALUES ($1, $2, $3);`

effectiveness/:

- all : returns all rows from the EFFECTIVENESS table. `SELECT * FROM type_effectiveness;`
- from\_attack\_type : returns the effectiveness values of a attacking pokemon type ID. `SELECT * FROM type_effectiveness WHERE attacking_type_id = $1;`
- from\_defending\_type : returns the effectiveness values of a defending pokemon type ID. `SELECT * FROM type_effectiveness WHERE defending_type_id = $1;`

move/:

- all : returns all rows from the MOVE table. Joins the TYPE table to also fetch the type of the move. `SELECT move.move_id, move.name, move.power, move.accuracy, move.pp, type.type_id, type.type_name FROM move JOIN type ON move.type_id = type.type_id;`
- from\_pokemon : returns the moves which a specific pokemon has. `SELECT type.move_id, type.name FROM move JOIN pokemon_moves ON move.move_id = pokemon_moves.move_id WHERE pokemon_moves.pokemon_id = $1;`
- new : creates a new relation in the POKEMON\_MOVE table from the pokemon ID, move ID and learn level values. `INSERT INTO pokemon_moves VALUES ($1, $2, $3);`

pokemon/:

- all : returns all rows from the POKEMON table. `SELECT * FROM pokemon;`
- delete : deletes a row from the POKEMON table, and all relations that this pokemon has with TYPE, MOVE and ABILITY tables. `DELETE FROM pokemon WHERE pokemon_id = $1; DELETE FROM`

```
pokemon_types WHERE pokemon_id = $1; DELETE FROM pokemon_moves WHERE  
pokemon_id = $1; DELETE FROM pokemon_abilities WHERE pokemon_id = $1;
```

- **from\_ability** : returns the rows from the POKEMON table that has a specific ability which is specified by its ID. `SELECT pokemon.pokemon_id, name, height, weight FROM pokemon JOIN pokemon_abilities ON pokemon.pokemon_id = pokemon_abilities.pokemon_id WHERE pokemon_abilities.ability_id = $1;`
- **from\_move** : returns the rows from the POKEMON table that has a specified move which is specified by its ID. `SELECT pokemon.pokemon_id, name, height, weight FROM pokemon JOIN pokemon_moves ON pokemon.pokemon_id = pokemon_moves.pokemon_id WHERE pokemon_moves.move_id = $1;`
- **get** : returns the row from the POKEMON table which has a specified pokemon ID. `SELECT * FROM pokemon WHERE pokemon_id = $1;`
- **get\_max\_id** : returns the maximum of the ID values from the POKEMON table which will be used to generate a new and unique ID. `SELECT MAX(pokemon_id) FROM pokemon;`
- **new** : creates a new row in the POKEMONS table from the ID, name, weight and height properties. `INSERT INTO pokemon (pokemon_id, name, weight, height) VALUES ($1, $2, $3, $4);`
- **update** : updates the weight, height and name properties in the POKEMON table from an ID. `UPDATE pokemon SET (name = $2, height = $3, weight = $4) WHERE pokemon_id = $1;`

**type/:**

- **all** : returns all rows from the TYPE table rows. `SELECT * FROM type;`
- **from\_pokemon** : returns the types which a specific pokemon has. `SELECT type.type_id, type.type_name FROM type JOIN pokemon_types ON type.type_id = pokemon_types.type_id WHERE pokemon_types.pokemon_id = $1;`
- **new** : creates a new relation in the POKEMON\_TYPE table from the pokemon ID, type ID and is primary values. `INSERT INTO pokemon_types VALUES ($1, $2, $3);`