### 

[**Function: get\_pokemon\_stats**](#_19dp81wl3men) **2**

[**Function: get\_pokemon\_by\_generation**](#_m0ym0uncq6ik) **2**

[**Function: get\_effectiveness**](#_xmi7l6txokgh) **3**

[**Function: get\_pokemon\_abilities**](#_ga9k9kvi7z78) **3**

[**Function: get\_pokemon\_by\_type**](#_lbncmmvj4vvi) **4**

[**Trigger: check\_weight\_height**](#_kbv6j05uwzq) **5**

[**Trigger: prevent\_duplicate\_type**](#_4000xhzqsdr) **5**

[**Trigger: check\_stat\_value**](#_dlkt48mnc214) **6**

### 

### **Function: get\_pokemon\_stats**

Purpose:

This function retrieves the stats of a given Pokémon based on its ID.

Parameters:

* p\_id (int): The ID of the Pokémon for which stats are to be retrieved.

Returns:

* stat\_name (name): The name of the stat.
* stat\_value (int): The value of the stat.

Example Usage:

SELECT \* FROM get\_pokemon\_stats(1);

Description:

The function joins the **pokemon\_stat** table with the **stat** table to fetch the names and values of the stats for a given Pokémon ID.

### **Function: get\_pokemon\_by\_generation**

Purpose:

This function retrieves all Pokémon that belong to a specific generation, including their names and other details.

Parameters:

* gen (int): The generation number.

Returns:

* pokemon\_id (int): The ID of the Pokémon.
* pokemon\_name (name): The name of the Pokémon.
* pokedex\_info (text): Information from the Pokédex about the Pokémon.
* weight (numeric): The weight of the Pokémon.
* height (numeric): The height of the Pokémon.

Example Usage:

SELECT \* FROM get\_pokemon\_by\_generation(1);

Description:

The function joins the **pokemon** table with the **species** table to fetch the details of Pokémon from a specific generation, ensuring that the species name (i.e., the Pokémon name) is included in the results.

### 

### **Function: get\_effectiveness**

Purpose:

This function retrieves the effectiveness multiplier of one type attacking another type.

Parameters:

* attacker (int): The ID of the attacking type.
* defender (int): The ID of the defending type.

Returns:

* multiplier (numeric): The effectiveness multiplier.

Example Usage:

SELECT get\_effectiveness(1, 2);

Description:

The function queries the effectiveness table to find the multiplier for the effectiveness of one type attacking another. If no specific entry is found, it returns a default multiplier of 1.

### **Function: get\_pokemon\_abilities**

Purpose:

This function retrieves all abilities of a given Pokémon, including whether they are hidden and the name of the Pokémon.

Parameters:

* p\_id (int): The ID of the Pokémon.

Returns:

* pokemon\_name (name): The name of the Pokémon.
* ability\_name (name): The name of the ability.
* effect (text): The effect of the ability.
* is\_hidden (boolean): Indicates if the ability is hidden.

Example Usage:

SELECT \* FROM get\_pokemon\_abilities(1);

Description:

The function joins the **pokemon, species, pokemon\_ability**, and **ability** tables to fetch the abilities of a given Pokémon, including the species name, ability names, their effects, and whether the abilities are hidden.

### **Function: get\_pokemon\_by\_type**

Purpose:

This function retrieves all Pokémon that belong to a specific type, including their names and other details.

Parameters:

* t\_id (int): The ID of the type.

Returns:

* pokemon\_id (int): The ID of the Pokémon.
* pokemon\_name (name): The name of the Pokémon.
* pokedex\_info (text): Information from the Pokédex about the Pokémon.
* weight (numeric): The weight of the Pokémon.
* height (numeric): The height of the Pokémon.

Example Usage:

SELECT \* FROM get\_pokemon\_by\_type(1);

Description:

The function joins the **pokemon**, **pokemon\_type**, and **species** tables to fetch details of Pokémon of a specific type, ensuring that the species name is included in the results.

### **Trigger: check\_weight\_height**

Purpose:

This trigger ensures that the weight and height of a Pokémon are positive values when a new Pokémon is inserted or an existing Pokémon is updated.

Trigger Timing:

* Before INSERT or UPDATE on the pokemon table.

Trigger Function:

* check\_pokemon\_weight\_height

Example Usage:

This trigger is automatically invoked when inserting or updating the pokemon table.

Description:

This trigger function checks the weight and height fields of the pokemon table before an INSERT or UPDATE operation. If either the weight or height is less than or equal to zero, an exception is raised, preventing the operation from proceeding. This ensures that only valid positive values for weight and height are stored in the database.

### **Trigger: prevent\_duplicate\_type**

Purpose:

This trigger ensures that a Pokémon cannot have duplicate entries for the same type in the pokemon\_type table.

Trigger Timing:

* Before INSERT on the pokemon\_type table.

Trigger Function:

* prevent\_duplicate\_pokemon\_type

Example Usage:

This trigger is automatically invoked when inserting into the pokemon\_type table.

Description:

This trigger function checks for existing entries in the pokemon\_type table with the same pokemon\_id and type\_id as the new row being inserted. If a duplicate entry is found, it raises an exception, preventing the insertion. This ensures that each Pokémon can only have unique types, maintaining the integrity of the pokemon\_type table.

### **Trigger: check\_stat\_value**

Purpose:

This trigger ensures that the stat values for Pokémon are within a valid range (e.g., 0 to 255) whenever an insert or update operation is performed on the pokemon\_stat table.

Trigger Timing:

* Before INSERT or UPDATE on the pokemon\_stat table.

Trigger Function:

* validate\_stat\_value

Example Usage:

This trigger is automatically invoked when inserting or updating the pokemon\_stat table.

Description:

This trigger function validate\_stat\_value checks whether the value being inserted or updated in the pokemon\_stat table falls within the valid range of 0 to 255. If the value is outside this range, an exception is raised with an appropriate error message. This ensures that only valid stat values are allowed in the database, maintaining data integrity and consistency.