```python
"""Small pokemon collection game using MySQL"""


import textwrap
import random

import mysql.connector as mycon


# Establish database connection
mcon = mycon.connect(
    host="localhost",
    user=input("Enter MySQL Username: "),
    passwd=input("Enter MySQL Password: "),
)
cursor = mcon.cursor()

# Database name
DATABASE = "pokemon_game"

# Table names
POKEMON_TABLE = "pokemon"

COLUMNS = ["ID", "Name", "Level", "Rarity"]

# Initialize the database and tables
cursor.execute(f"""CREATE DATABASE IF NOT EXISTS {DATABASE}""")
cursor.execute(f"USE {DATABASE}")

cursor.execute(
    f"""CREATE TABLE IF NOT EXISTS {POKEMON_TABLE}(
        id INTEGER PRIMARY KEY AUTO_INCREMENT,
        name VARCHAR(255) NOT NULL,
        level INTEGER,
        rarity VARCHAR(255)
    )"""
)

mcon.commit()


def execute(query, params=None):
    """Function to execute SQL queries on the database and automatically commit.

    Returns cursor.rowcount"""

    cursor.execute(query, params)
    if cursor.rowcount:
        mcon.commit()
    return cursor.rowcount


def printrow(row):
    print("|".join(map(lambda s: f"{s:^10}", row)))


AVAILABLE_POKEMON = {
    "pikachu": "common",
    "squirtle": "common",
    "ralts": "common",
    "charmander": "common",
    "bulbasaur": "common",
    "charizard": "rare",
    "eternatus": "rare",
    "rayquaza": "rare",
    "arceus": "rare",
}


def get_pokemon(_id=None):
    """Get data of a pokèmon from the database"""

    if _id is not None:
        execute(f"""SELECT * FROM {POKEMON_TABLE} WHERE id = %s""", (_id,))
        return cursor.fetchone()
    else:
        execute(f"""SELECT * FROM {POKEMON_TABLE}""")
        return cursor.fetchall()


def insert_pokemon(name, level):
    """Insert a pokemon into the database"""

    rarity = AVAILABLE_POKEMON[name]

    execute(
```

```
        f"""INSERT INTO {POKEMON_TABLE} (name, level, rarity)
        VALUES(%s, %s, %s)""",
        (name, level, rarity),
    )

    _id = cursor.lastrowid
    print()
    print(f"Successfully added a new Level {level} {name.title()} (#{_id})!")

    return _id


def delete_pokemon(_id):
    """Delete a pokemon from the database"""

    execute(f"""DELETE FROM {POKEMON_TABLE} WHERE id = %s""", (_id,))
    deleted = cursor.rowcount

    print()
    if deleted:
        print(f"Successfully deleted Pokèmon #{_id}")
    else:
        print("Could not find that Pokèmon!")

    return deleted


def take_input(text, *, check=lambda x: True, error_msg=""):
    """Keeps asking input until correct value is passed."""

    while True:
        value = input(text)
        if not check(value):
            print(error_msg)
            continue
        return value


while True:
    print(
        textwrap.dedent(
            f"""
+================================================+
|                Pokèmon Mini Game               |
+------------------------------------------------+
1. Add Pokèmon
2. Add Random Pokèmon
3. Remove Pokèmon

4. Display Pokèmon

5. Exit
================================================="""
        )
    )
    option = input("Enter option: ")
    try:
        option = int(option)
    except ValueError:
        print("Invalid input!")
        continue

    if option == 1:
        # Add pokemon
        name = take_input(
            "Input pokèmon name: ",
            check=lambda n: n.lower() in AVAILABLE_POKEMON,
            error_msg="Invalid pokèmon!",
        ).lower()
        level = take_input(
            "Input level of the pokèmon: ",
            check=lambda l: l.isdigit() and 0 <= int(l) <= 100,
            error_msg="Level must be an integer between 0 and 100",
        )
        insert_pokemon(name, level)

    elif option == 2:
        # Add random pokemon
        name = random.choice(list(AVAILABLE_POKEMON))
        level = random.randint(0, 100)
        insert_pokemon(name, level)

    elif option == 3:
        _id = take_input("Enter ID of the Pokèmon you want to remove: ")
        delete_pokemon(_id)

    elif option == 4:
```

```
174         pokemon = get_pokemon()
175         printrow(COLUMNS)
176         print("-" * 45)
177         for p in pokemon:
178             printrow(p)
179 
180     elif option == 5:
181         break
182 
183     else:
184         print("Invalid input!")
185         continue
```