To connect to a MySQL database in Python, you'll need the mysql-connector-python library. Here's a step-by-step guide:

### 1. Install the mysql-connector-python Library

If you don't have it already, install it using pip:

```
pip install mysql-connector-python
```

### 2. Import the Library

Import the library in your Python script:

```
import mysql.connector
```

### 3. Establish a Connection

Use the mysql.connector.connect() function to create a connection to the MySQL server. You'll need to provide the following information:

- host: The hostname or IP address of the MySQL server (e.g., "localhost").
- user: Your MySQL username.
- password: Your MySQL password.
- database: The name of the database you want to connect to (optional, but recommended).

```
try:
    connection = mysql.connector.connect(
        host="localhost",
        user="your_mysql_username",
        password="your_mysql_password",
        database="your_database_name"  # Replace with your database name
    )
    if connection.is_connected():
        print("Successfully connected to MySQL database!")
except mysql.connector.Error as err:
    print(f"Error connecting to MySQL: {err}")
    exit()
```

### 4. Create a Cursor Object

A cursor is used to execute SQL queries.

```
cursor = connection.cursor()
```

### 5. Execute Queries

Use the cursor's execute() method to run SQL queries. It's a good practice to use parameterized queries to prevent

SQL injection.

```
# Example: Create a table
cursor.execute("""
    CREATE TABLE IF NOT EXISTS employees (
        id INT AUTO_INCREMENT PRIMARY KEY,
        name VARCHAR(255) NOT NULL,
        age INT
    )
""")

# Example: Insert data using a parameterized query
employee_name = "John Doe"
employee_age = 30
cursor.execute("INSERT INTO employees (name, age) VALUES (%s, %s)", (employee_name, employee_age))
connection.commit()  # Important: Commit changes to the database!

# Example: Select data
cursor.execute("SELECT * FROM employees")
rows = cursor.fetchall()
for row in rows:
    print(row)

# Example: Select one row
cursor.execute("SELECT name FROM employees WHERE age = %s", (employee_age,))
one_row = cursor.fetchone()
print(one_row)
```

**6. Commit Changes**

If you're modifying data (INSERT, UPDATE, DELETE), you **must** commit the changes:

```
connection.commit()
```

**7. Close the Cursor and Connection**

Always close the cursor and connection when you're done:

```
cursor.close()
connection.close()
```

**Complete Example:**

```
import mysql.connector

def connect_and_query_mysql():
    connection = None  # Initialize connection
    try:
        # 1. Establish Connection
        connection = mysql.connector.connect(
            host="localhost",
            user="your_mysql_username",
            password="your_mysql_password",
            database="your_database_name"
        )
        if not connection.is_connected():
            print("Failed to connect to MySQL database.")
            return

        print("Successfully connected to MySQL database!")
        cursor = connection.cursor()

        # 2. Create table
        cursor.execute("""
            CREATE TABLE IF NOT EXISTS employees (
                id INT AUTO_INCREMENT PRIMARY KEY,
                name VARCHAR(255) NOT NULL,
                age INT
            )
        """)

        # 3. Insert data
        employee_name1 = "John Doe"
        employee_age1 = 30
        employee_name2 = "Jane Smith"
        employee_age2 = 25
        cursor.execute("INSERT INTO employees (name, age) VALUES (%s, %s)", (employee_name1,
employee_age1))
        cursor.execute("INSERT INTO employees (name, age) VALUES (%s, %s)", (employee_name2,
employee_age2))
        connection.commit()

        # 4. Select all data
        cursor.execute("SELECT * FROM employees")
        rows = cursor.fetchall()
        print("\nAll employees:")
        for row in rows:
            print(row)

        # 5. Select one employee
        cursor.execute("SELECT name FROM employees WHERE age = %s", (employee_age2,))
        one_row = cursor.fetchone()
        if one_row:
            print(f"\nEmployee with age {employee_age2}: {one_row[0]}")
        else:
            print(f"\nNo employee found with age {employee_age2}")

    except mysql.connector.Error as err:
        print(f"Error: {err}")
        if connection:
            connection.rollback()  # Rollback on error
    finally:
        # 6. Close the connection
        if connection and connection.is_connected():
            cursor.close()
            connection.close()
            print("MySQL connection closed.")

if __name__ == "__main__":
    connect_and_query_mysql()
```