

```
In [1]: !pip install mysql-connector-python
```

```
Collecting mysql-connector-python
  Downloading mysql_connector_python-8.0.25-cp37-cp37m-win_amd64.whl (793 kB)
Requirement already satisfied: protobuf>=3.0.0 in c:\users\hssai\anaconda3\lib\site-packages (from mysql-connector-python) (3.11.3)
Requirement already satisfied: setuptools in c:\users\hssai\anaconda3\lib\site-packages (from protobuf>=3.0.0->mysql-connector-python) (45.2.0.post20200210)
Requirement already satisfied: six>=1.9 in c:\users\hssai\anaconda3\lib\site-packages (from protobuf>=3.0.0->mysql-connector-python) (1.14.0)
Installing collected packages: mysql-connector-python
Successfully installed mysql-connector-python-8.0.25
```

```
In [2]: # import necessary libraries

import pandas as pd
import mysql.connector
from mysql.connector import Error
```

## Connecting to MySQL Server

```
In [5]: # connecting to MySQL Server

def create_server_connection(host_name, user_name, user_password):
    connection = None
    try:
        connection = mysql.connector.connect(
            host= '127.0.0.1',
            user='root',
            passwd= pw
        )
        print("MySQL Database connection successful")
    except Error as err:
        print(f"Error: '{err}'")

    return connection
```

```
In [6]: # testing if the server is connected
connection = create_server_connection("localhost", "root", "Azerty123@")

MySQL Database connection successful
```

## Creating a New Database

```
In [9]: # After we have established a connection  
# next, is to create a new database on the server  
  
def create_database(connection, query): # function takes two arguments, connection and query  
    cursor = connection.cursor()  
    try:  
        cursor.execute(query)  
        print("Database created successfully")  
    except Error as err:  
        print(f"Error: '{err}'")
```

```
In [10]: # define the query to create the database and call the function  
  
create_database_query = "CREATE DATABASE school"  
create_database(connection, create_database_query)
```

Database created successfully

## Connecting to the Database

```
In [11]: # After we have created a database in MySQL server, is the same function created before but now we take one more argument  
  
def create_db_connection(host_name, user_name, user_password, db_name):  
    connection = None  
    try:  
        connection = mysql.connector.connect(  
            host = '127.0.0.1',  
            user = 'root',  
            passwd = pw,  
            database = 'school'  
        )  
        print('MySQL Database connection successful')  
    except Error as err:  
        print(f"Error: '{err}'")  
  
    return connection
```

## Creating a Query Execution Function

```
In [12]: # create a query execution function, that's going to take the SQL queries stored in python as strings
# and pass them to the cursor.execute() to execute them on the server.

def execute_query(connection, query):
    cursor = connection.cursor()
    try:
        cursor.execute(query)
        connection.commit() # this method to make sure that the commands detailed in our SQL queries
        print("Query successful")
    except Error as err:
        print(f"Error: '{err}'")
```

## Creating Tables

```
In [14]: # after creating the database, I will start building the necessary tables
# we use triple quote notation for multi-line strings to store our SQL query.

create_teacher_table = """
CREATE TABLE teacher(
    teacher_id INT Primary Key,
    first_name VARCHAR(40) NOT NULL,
    last_name VARCHAR(40) NOT NULL,
    language_1 VARCHAR(3) NOT NULL,
    language_2 VARCHAR(3),
    dob DATE,
    tax_id INT UNIQUE,
    phone_no VARCHAR(20)
);
"""

# 1. create a connection to the database

connection = create_db_connection('127.0.0.1', 'root', pw, 'school')

# 2. execute our defined query

execute_query(connection, create_teacher_table)
```

MySQL Database connection successful  
Query successful

```

In [15]: # create client table
create_client_table = """
CREATE TABLE client (
    client_id INT PRIMARY KEY,
    client_name VARCHAR(40) NOT NULL,
    address VARCHAR(60) NOT NULL,
    industry VARCHAR(20)
);
"""

# create participant table
create_participant_table = """
CREATE TABLE participant (
    participant_id INT PRIMARY KEY,
    first_name VARCHAR(40) NOT NULL,
    last_name VARCHAR(40) NOT NULL,
    phone_no VARCHAR(20),
    client INT
);
"""

# create course table
create_course_table = """
CREATE TABLE course (
    course_id INT PRIMARY KEY,
    course_name VARCHAR(40) NOT NULL,
    language VARCHAR(3) NOT NULL,
    level VARCHAR(2),
    course_length_weeks INT,
    start_date DATE,
    in_school BOOLEAN,
    teacher INT,
    client INT
);
"""

# 1. create a connection to the database

connection = create_db_connection('127.0.0.1', 'root', pw, 'school')

# 2. execute our defined queries

execute_query(connection, create_client_table)
execute_query(connection, create_participant_table)
execute_query(connection, create_course_table)

```

MySQL Database connection successful  
Query successful  
Query successful  
Query successful

In [16]: *# after creating all table, the following step is to define the relationships between the table*

```
alter_participant = """
ALTER TABLE participant
ADD FOREIGN KEY(client)
REFERENCES client(client_id)
ON DELETE SET NULL;
"""

alter_course = """
ALTER TABLE course
ADD FOREIGN KEY(teacher)
REFERENCES teacher(teacher_id)
ON DELETE SET NULL;
"""

alter_course_again = """
ALTER TABLE course
ADD FOREIGN KEY(client)
REFERENCES client(client_id)
ON DELETE SET NULL;
"""

create_takescourse_table = """
CREATE TABLE takes_course (
    participant_id INT,
    course_id INT,
    PRIMARY KEY(participant_id, course_id),
    FOREIGN KEY(participant_id) REFERENCES participant(participant_id) ON DELETE
CASCADE,
    FOREIGN KEY(course_id) REFERENCES course(course_id) ON DELETE CASCADE
);
"""

# 1. create a connection

connection = create_db_connection("localhost", "root", pw, "scool")

# 2. execute the queries
execute_query(connection, alter_participant)
execute_query(connection, alter_course)
execute_query(connection, alter_course_again)
execute_query(connection, create_takescourse_table)
```

MySQL Database connection successful  
Query successful  
Query successful  
Query successful  
Query successful

## Populating the tables

```
In [18]: # insert data into teacher table
pop_teacher = """
INSERT INTO teacher VALUES
(1, 'James', 'Smith', 'ENG', NULL, '1985-04-20', 12345, '+491774553676'),
(2, 'Stefanie', 'Martin', 'FRA', NULL, '1970-02-17', 23456, '+49123456789
0'),
(3, 'Steve', 'Wang', 'MAN', 'ENG', '1990-11-12', 34567, '+447840921333'),
(4, 'Friederike', 'Müller-Rossi', 'DEU', 'ITA', '1987-07-07', 45678, '+49234
5678901'),
(5, 'Isobel', 'Ivanova', 'RUS', 'ENG', '1963-05-30', 56789, '+491772635467'),
(6, 'Niamh', 'Murphy', 'ENG', 'IRI', '1995-09-08', 67890, '+491231231232');
"""

connection = create_db_connection("localhost", "root", pw, "scool")

execute_query(connection, pop_teacher)
```

MySQL Database connection successful

Query successful

In [19]: *# populate the remaining tables*

```
pop_client = """
INSERT INTO client VALUES
(101, 'Big Business Federation', '123 Falschungstraße, 10999 Berlin', 'NGO'),
(102, 'eCommerce GmbH', '27 Ersatz Allee, 10317 Berlin', 'Retail'),
(103, 'AutoMaker AG', '20 Künstlichstraße, 10023 Berlin', 'Auto'),
(104, 'Banko Bank', '12 Betrugstraße, 12345 Berlin', 'Banking'),
(105, 'WeMoveIt GmbH', '138 Anglistweg, 10065 Berlin', 'Logistics');
"""

pop_participant = """
INSERT INTO participant VALUES
(101, 'Marina', 'Berg', '491635558182', 101),
(102, 'Andrea', 'Duerr', '49159555740', 101),
(103, 'Philipp', 'Probst', '49155555692', 102),
(104, 'René', 'Brandt', '4916355546', 102),
(105, 'Susanne', 'Shuster', '49155555779', 102),
(106, 'Christian', 'Schreiner', '49162555375', 101),
(107, 'Harry', 'Kim', '49177555633', 101),
(108, 'Jan', 'Nowak', '49151555824', 101),
(109, 'Pablo', 'Garcia', '49162555176', 101),
(110, 'Melanie', 'Dreschler', '49151555527', 103),
(111, 'Dieter', 'Durr', '49178555311', 103),
(112, 'Max', 'Mustermann', '49152555195', 104),
(113, 'Maxine', 'Mustermann', '49177555355', 104),
(114, 'Heiko', 'Fleischer', '49155555581', 105);
"""

pop_course = """
INSERT INTO course VALUES
(12, 'English for Logistics', 'ENG', 'A1', 10, '2020-02-01', TRUE, 1, 105),
(13, 'Beginner English', 'ENG', 'A2', 40, '2019-11-12', FALSE, 6, 101),
(14, 'Intermediate English', 'ENG', 'B2', 40, '2019-11-12', FALSE, 6, 101),
(15, 'Advanced English', 'ENG', 'C1', 40, '2019-11-12', FALSE, 6, 101),
(16, 'Mandarin für Autoindustrie', 'MAN', 'B1', 15, '2020-01-15', TRUE, 3, 103),
(17, 'Français intermédiaire', 'FRA', 'B1', 18, '2020-04-03', FALSE, 2, 101),
(18, 'Deutsch für Anfänger', 'DEU', 'A2', 8, '2020-02-14', TRUE, 4, 102),
(19, 'Intermediate English', 'ENG', 'B2', 10, '2020-03-29', FALSE, 1, 104),
(20, 'Fortgeschrittenes Russisch', 'RUS', 'C1', 4, '2020-04-08', FALSE, 5, 103);
"""

pop_takescourse = """
INSERT INTO takes_course VALUES
(101, 15),
(101, 17),
(102, 17),
(103, 18),
(104, 18),
(105, 18),
(106, 13),
(107, 13),
(108, 13),
(109, 14),
```

```

(109, 15),
(110, 16),
(110, 20),
(111, 16),
(114, 12),
(112, 19),
(113, 19);
"""

connection = create_db_connection('localhost', 'root', pw, 'school')

execute_query(connection, pop_client)
execute_query(connection, pop_participant)
execute_query(connection, pop_course)
execute_query(connection, pop_takescourse)

```

```

MySQL Database connection successful
Query successful
Query successful
Query successful
Query successful

```

## Reading the data

```

In [20]: # using cursor.fetchall() to read the data from the database without making any changes

def read_query(connection, query):
    cursor = connection.cursor()
    result = None
    try:
        cursor.execute(query)
        result = cursor.fetchall()
        return result
    except Error as err:
        print(f"Error: '{err}'")

```



In [23]: *# create a simple query*

```
query1 = '''
SELECT * FROM teacher;
'''

connection = create_db_connection("localhost", "root", pw, 'school')
results = read_query(connection, query1)

for result in results:
    print(result)
```

MySQL Database connection successful

```
(1, 'James', 'Smith', 'ENG', None, datetime.date(1985, 4, 20), 12345, '+49177
4553676')
(2, 'Stefanie', 'Martin', 'FRA', None, datetime.date(1970, 2, 17), 23456, '+4
91234567890')
(3, 'Steve', 'Wang', 'MAN', 'ENG', datetime.date(1990, 11, 12), 34567, '+4478
40921333')
(4, 'Friederike', 'Müller-Rossi', 'DEU', 'ITA', datetime.date(1987, 7, 7), 45
678, '+492345678901')
(5, 'Isobel', 'Ivanova', 'RUS', 'ENG', datetime.date(1963, 5, 30), 56789, '+4
91772635467')
(6, 'Niamh', 'Murphy', 'ENG', 'IRI', datetime.date(1995, 9, 8), 67890, '+4912
31231232')
```

In [29]: *# doing join on the course and client tables*

```
query2 = '''
SELECT course.course_id, course.course_name, course.language, client.client_na
me, client.address
FROM course
INNER JOIN client
ON course.client = client.client_id
WHERE course.in_school = FALSE;
'''

connection = create_db_connection("localhost", "root", pw, 'school')
results = read_query(connection, query2)

for result in results:
    print(result )
```

MySQL Database connection successful

```
(13, 'Beginner English', 'ENG', 'Big Business Federation', '123 Falschungstra
ße, 10999 Berlin')
(14, 'Intermediate English', 'ENG', 'Big Business Federation', '123 Falschung
straße, 10999 Berlin')
(15, 'Advanced English', 'ENG', 'Big Business Federation', '123 Falschungstra
ße, 10999 Berlin')
(17, 'Français intermédiaire', 'FRA', 'Big Business Federation', '123 Falschu
ngstraße, 10999 Berlin')
(19, 'Intermediate English', 'ENG', 'Banko Bank', '12 Betrugstraße, 12345 Ber
lin')
(20, 'Fortgeschrittenes Russisch', 'RUS', 'AutoMaker AG', '20 Künstlichstraß
e, 10023 Berlin')
```

## Formatting the output

```
In [33]: # returns a list of lists and then creates a pandas Dataframe

from_db = []

for result in results:
    result = list(result)
    from_db.append(result)

columns = ["course_id", "course_name", "language", "client_name", "address"]
df = pd.DataFrame(from_db, columns = columns )
display(df)
```

	course_id	course_name	language	client_name	address
0	13	Beginner English	ENG	Big Business Federation	123 Falschungstraße, 10999 Berlin
1	14	Intermediate English	ENG	Big Business Federation	123 Falschungstraße, 10999 Berlin
2	15	Advanced English	ENG	Big Business Federation	123 Falschungstraße, 10999 Berlin
3	17	Français intermédiaire	FRA	Big Business Federation	123 Falschungstraße, 10999 Berlin
4	19	Intermediate English	ENG	Banko Bank	12 Betrugstraße, 12345 Berlin
5	20	Fortgeschrittenes Russisch	RUS	AutoMaker AG	20 Künstlichstraße, 10023 Berlin

## Updating the Records

```
In [35]: # updating the address where client_id = 101

update = '''
UPDATE client
SET address = '23 Fingiertweg, 14534 Berlin'
WHERE client_id = 101;
'''

#1. create a connection

connection = create_db_connection('localhost', 'root', pw, 'school')

#2. execute the query

execute_query(connection,update)
```

MySQL Database connection successful  
Query successful

## Deleting the Records

In [41]: *# deleting a course\_id = 20 from the course database*

```
delete = '''
DELETE FROM course
WHERE course_id = 20;
'''

# create a connection

connection = create_db_connection('localhost', 'root', pw, 'school')

# execute the defined query

execute_query(connection, delete)
```

MySQL Database connection successful  
Query successful

In [45]: *# Let's confirm that we delete the course\_id = 20*

```
course_id_20 = '''
SELECT *
FROM course;
'''

connection = create_db_connection('localhost', 'root', pw, 'school')
result = read_query(connection, course_id_20)

from_db = []

for result in results:
    print(result)
```

MySQL Database connection successful  
(13, 'Beginner English', 'ENG', 'Big Business Federation', '123 Falschungstraße, 10999 Berlin')  
(14, 'Intermediate English', 'ENG', 'Big Business Federation', '123 Falschungstraße, 10999 Berlin')  
(15, 'Advanced English', 'ENG', 'Big Business Federation', '123 Falschungstraße, 10999 Berlin')  
(17, 'Français intermédiaire', 'FRA', 'Big Business Federation', '123 Falschungstraße, 10999 Berlin')  
(19, 'Intermediate English', 'ENG', 'Banko Bank', '12 Betrugstraße, 12345 Berlin')  
(20, 'Fortgeschrittenes Russisch', 'RUS', 'AutoMaker AG', '20 Künstlichstraße, 10023 Berlin')

## Creating Records from Lists

```
In [63]: def execute_list_query(connection, sql, val):
        cursor = connection.cursor()
        try:
            cursor.executemany(sql, val)
            connection.commit()
            print("Query successful")
        except Error as err:
            print(f"Error: '{err}'")
```

```
In [64]: # adding two new teachers to the database
        # use %s as placeholder for our value.

        sql = '''
            INSERT INTO teacher (teacher_id, first_name, last_name, language_1, language_2,
            dob, tax_id, phone_no)
            VALUES (%s, %s, %s, %s, %s, %s, %s, %s)
            '''

        val = [
            (7, 'Hank', 'Dodson', 'ENG', None, '1991-12-23', 11111, '+491772345678'),
            (8, 'Sue', 'Perkins', 'MAN', 'ENG', '1976-02-02', 22222, '+491443456432')
        ]
```

```
In [65]: #execute the query and add the teachers to our database
```

```
connection = create_db_connection('localhost', 'root', pw, 'school')
execute_list_query(connection, sql, val)
```

MySQL Database connection successful

Error: '1205 (HY000): Lock wait timeout exceeded; try restarting transaction'