## #install connector api using the command below.

!pip install mysql-connector-python

# Import the MySQL Connector/Python

import mysql.connector as connector

# Establish connection between Python and MySQL database via connector API

connection=connector.connect(

user="root", # use your own

password="", # use your own

)

print("Connection between MySQL and Python is established.\n")

# Create cursor object to communicate with entire MySQL database

cursor = connection.cursor()

print("Cursor is created to communicate with the MySQL using Python.\n")

# If exist, drop the database first, and create again

try:

cursor.execute("CREATE DATABASE little\_lemon")

except:

cursor.execute("drop database little\_lemon")

cursor.execute("CREATE DATABASE little\_lemon")

print("The database little\_lemon is created.\n")

# Set little\_lemon database for use

cursor.execute("USE little\_lemon")

print("The database little\_lemon is set for use.\n")

# The SQL query for MenuItems table is:

create\_menuitem\_table="""

CREATE TABLE MenuItems (

ItemID INT AUTO\_INCREMENT,

Name VARCHAR(200), Type VARCHAR(100),

Price INT, PRIMARY KEY (ItemID) );"""

# Create MenuItems table

cursor.execute(create\_menuitem\_table)

print("MenuItmes table is created.\n")

# The SQL query for Menu table is:

create\_menu\_table="""

CREATE TABLE Menus (

MenuID INT, ItemID INT, Cuisine VARCHAR(100),

PRIMARY KEY (MenuID,ItemID)

);"""

# Create Menu table

cursor.execute(create\_menu\_table)

print("Menu table is created.\n")

# The SQL query for Bookings table is:

create\_booking\_table=""" CREATE TABLE Bookings (

BookingID INT AUTO\_INCREMENT, TableNo INT,

GuestFirstName VARCHAR(100) NOT NULL,

GuestLastName VARCHAR(100) NOT NULL,

BookingSlot TIME NOT NULL,

EmployeeID INT,

PRIMARY KEY (BookingID)

);"""

# Create Bookings table

cursor.execute(create\_booking\_table)

print("Bookings table is created.\n")

# The SQL query for Bookings table is:

create\_orders\_table="""

CREATE TABLE Orders (

OrderID INT, TableNo INT,

MenuID INT, BookingID INT,

BillAmount INT, Quantity INT,

PRIMARY KEY (OrderID,TableNo)

);"""

# Create Orders table

cursor.execute(create\_orders\_table)

print("Orders table is created.\n")

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*#

# Insert query to populate "MenuItems" table is:

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*#

insert\_menuitmes="""

INSERT INTO MenuItems (ItemID, Name, Type, Price)

VALUES

(1,'Olives','Starters',5),

(2,'Flatbread','Starters', 5),

(3, 'Minestrone', 'Starters', 8),

(4, 'Tomato bread','Starters', 8),

(5, 'Falafel', 'Starters', 7),

(6, 'Hummus', 'Starters', 5),

(7, 'Greek salad', 'Main Courses', 15),

(8, 'Bean soup', 'Main Courses', 12),

(9, 'Pizza', 'Main Courses', 15),

(10,'Greek yoghurt','Desserts', 7),

(11, 'Ice cream', 'Desserts', 6),

(12, 'Cheesecake', 'Desserts', 4),

(13, 'Athens White wine', 'Drinks', 25),

(14, 'Corfu Red Wine', 'Drinks', 30),

(15, 'Turkish Coffee', 'Drinks', 10),

(16, 'Turkish Coffee', 'Drinks', 10),

(17, 'Kabasa', 'Main Courses', 17);"""

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*#

# Insert query to populate "Menu" table is:

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*#

insert\_menu="""

INSERT INTO Menus (MenuID,ItemID,Cuisine)

VALUES

(1, 1, 'Greek'),

(1, 7, 'Greek'),

(1, 10, 'Greek'),

(1, 13, 'Greek'),

(2, 3, 'Italian'),

(2, 9, 'Italian'),

(2, 12, 'Italian'),

(2, 15, 'Italian'),

(3, 5, 'Turkish'),

(3, 17, 'Turkish'),

(3, 11, 'Turkish'),

(3, 16, 'Turkish');"""

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*#

# Insert query to populate "Bookings" table is:

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*#

insert\_bookings="""

INSERT INTO Bookings (BookingID, TableNo, GuestFirstName,

GuestLastName, BookingSlot, EmployeeID)

VALUES

(1,12,'Anna','Iversen','19:00:00',1),

(2, 12, 'Joakim', 'Iversen', '19:00:00', 1),

(3, 19, 'Vanessa', 'McCarthy', '15:00:00', 3),

(4, 15, 'Marcos', 'Romero', '17:30:00', 4),

(5, 5, 'Hiroki', 'Yamane', '18:30:00', 2),

(6, 8, 'Diana', 'Pinto', '20:00:00', 5);"""

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*#

# Insert query to populate "Orders" table is:

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*#

insert\_orders="""

INSERT INTO Orders (OrderID, TableNo, MenuID, BookingID, Quantity, BillAmount)

VALUES

(1, 12, 1, 1, 2, 86),

(2, 19, 2, 2, 1, 37),

(3, 15, 2, 3, 1, 37),

(4, 5, 3, 4, 1, 40),

(5, 8, 1, 5, 1, 43);"""

print("Inserting data in MenuItems table.")

# Populate MenuItems table

cursor.execute(insert\_menuitmes)

print("Total number of rows in MenuItem table: {}\n".format(cursor.rowcount))

# Once the query is executed, you commit the change into the database

connection.commit()

print("Inserting data in Menus table.")

# Populate MenuItems table

cursor.execute(insert\_menu)

print("Total number of rows in Menu table: {}\n".format(cursor.rowcount))

connection.commit()

print("Inserting data in Bookings table.")

# Populate Bookings table

cursor.execute(insert\_bookings)

print("Total number of rows in Bookings table: {}\n".format(cursor.rowcount))

connection.commit()

print("Inserting data in Orders table.")

# Populate Orders table

cursor.execute(insert\_orders)

print("Total number of rows in Orders table: {}\n".format(cursor.rowcount))

connection.commit()

print("""The database "little\_lemon" is ready for use.""")

## **Task 1 solution**

Create a database connection pool with three connections available for the users to connect to the database with. You need to import MySQLConnectionPool class and pass the following arguments:

* pool\_name = “ll\_pool\_a”
* pool\_size = 3
* \*\*dbconfig

Your database configuration should resemble the following code:

from mysql.connector.pooling import MySQLConnectionPool

from mysql.connector import Error

dbconfig = {

"database":"little\_lemon",

"user" : "root",

"password" : ""

}

try:

pool = MySQLConnectionPool(pool\_name = "ll\_pool\_a",

pool\_size = 3, #default is 5

\*\*dbconfig)

print("The connection pool is created with a name: ",pool.pool\_name)

print("The pool size is:",pool.pool\_size)

except Error as er:

print("Error code:", er.errno)

print("Error message:", er.msg)

Use the actual name of the database together with authenticated username and password in the above configuration.

Use the Error class from mysql.connector to handle a possible error in an instance where the wrong parameters are passed on the database configuration.

TIP: Use try-except block from Python to implement the error handling. Once the connection pool is created, use the print statements to display the name of the pool and the number of connections in it.

## **Task 2 solution**

Get a connection from the database connection pool that you have created in the first task and retrieve the following columns from the Bookings table:

* BookingID
* GuestFirstName
* GuestLastName

Retrieve the required columns and put the connection back into the pool after you have completed the task.

TIP: Use the get\_connection module from the pool to use the connection. Use print statements to display the following message and close the connection to return to the pool.

from mysql.connector.pooling import MySQLConnectionPool

dbconfig = {

"database":"little\_lemon",

"user" : "root",

"password" : ""

}

pool = MySQLConnectionPool(pool\_name = "ll\_pool\_a",

pool\_size = 3, #default is 5

\*\*dbconfig)

# Get the connection from the connection pool "pool"

print("Getting a connection from the pool.")

connection1 = pool.get\_connection()

#print("A user with connection id {} is connected to the database.".format(

# connection1.connection\_id))

#db\_Info = connection1.get\_server\_info()

#print("MySQL server version is:", db\_Info)

# Create cursor object to communicate with entire MySQL database

print("Creating a cursor object.")

cursor = connection1.cursor()

# The SQL query is:

sql\_query = "SELECT BookingId, GuestFirstName, GuestLastName FROM Bookings;"

# Execute query

print("Executing the SQL query.")

cursor.execute(sql\_query)

# Fetch all results that satisfy the query

print("Fetching the query results.")

results = cursor.fetchall()

# Retrieve column names

print("Retrieving the column names.")

cols = cursor.column\_names

# Print column names and records in "results" using for loop

print("Printing the results.\n")

print("""Upcoming Bookings are:\n""")

print(cols)

for result in results:

print(result)

# Put the connection back to the pool

print("\nReturning the connection back to the pool.")

connection1.close()

print("The connection is placed back into the pool for the next user to connect.")

Task 3 solution

The following five guests want to connect to the database:

guests = ["Anna", "Marcos", "Diana", "Joakim", "Hiroki"]

You only have three connections in the database connection pool. Use the available connection in the pool to connect three guests and then add new connections to the pool to connect the two remaining guests. Ensure that all five guests are connected to the database at the same time, by adding more connections to the pool.

TIP: Use add\_connection module from the pool and add a new connection if all existing connections are in use. Use try-except from Python and print the message to inform the user when connected.

# Create a connection pool

from mysql.connector.pooling import MySQLConnectionPool

dbconfig = {

"database":"little\_lemon",

"user" : "root",

"password" : ""

}

pool = MySQLConnectionPool(pool\_name = "ll\_pool\_a",

pool\_size = 3, #default is 5

\*\*dbconfig)

# List of the guests who want to connect to the database

guests = ["Anna", "Marcos", "Diana", "Joakim", "Hiroki"]

# To add connection to the pool, the connection must be of MySQLConnection instance

# Also possible to create via connect module and need the import below

import mysql.connector as connector

# Assign connection to each user

for guest in guests:

try:

guest\_connected = pool.get\_connection()

print("[{}] is connected.\n".format(guest))

except:

print("No more connections are available.")

print("Adding new connection in the pool.")

# Create a connection

connection=connector.connect(user="root",password="")

# Add the connection into the pool

pool.add\_connection(cnx=connection)

print("A new connection is added in the pool.\n")

user\_connected = pool.get\_connection()

print("[{}] is connected.\n".format(guest))