import mysql.connector as connector

# Establish connection b/w Python and MySQL database via connector API

connection=connector.connect(

user="root", # use your own

password="", # use your own

)

# Create a cursor object to communicate with entire MySQL database

cursor = connection.cursor()

# Set the little\_lemon database for use

cursor.execute("use little\_lemon")

# Confirm the database in use

connection.database

## **Task 1 solution**

Little Lemon need a list of all guests and their booking slots. Extract the information from the bookings table using a MySQL SELECT statement and display it in a neat format.

# The query to retrieve all bookings is:

all\_bookings = """SELECT \* FROM Bookings;"""

# Execute query

cursor.execute(all\_bookings)

# Fetch all results that satisfy the query

results = cursor.fetchall()

# Retrieve the column names

cols = cursor.column\_names

# print column names and the records from results using for loop

print("Data in the 'Bookings' table")

print(cols)

for result in results:

print(result)

## **Task 2 solution**

Change the guest Diana Pinto’s booking to table 10 in Little Lemon’s MySQL database using an SQL UPDATE statement in Python. Once you’ve completed the statement, rerun the code from Task 1 to view the results.

# The update query is:

update\_bookings="""UPDATE Bookings

SET TableNo=10

WHERE BookingID = 6;"""

# Execute the query to update the table

print("Executing update query")

cursor.execute(update\_bookings)

# Commit change

print("Comitting change to the table")

connection.commit()

print("Record is updated in the table")

## **Task 3 solution:**

Little Lemon has encountered a conflict with two bookings. To resolve the conflict, you need to update the record for the guest Joakim Iversen.

Update Joakim’s booking in the MySQL database using Python as follows:

* Change the guest’s table number to 11.
* Change the EmployeeID of the guest’s waiter to 6.

Your statements must alter two columns in the row that corresponds with the guest of Booking ID 2.

# The update query is:

update\_bookings="""UPDATE Bookings

SET TableNo=11, EmployeeID=6

WHERE BookingID = 2;"""

# Execute the query to update the table

print("Executing update query")

cursor.execute(update\_bookings)

# Commit change

print("Comitting change to the table")

connection.commit()

print("Record is updated in the table")

## **Task 4 solution**

Little Lemon restaurant didn’t receive their regular supply of ingredients today. This means that they can’t provide any Greek cuisine for their guests. They need to delete all Greek cuisine from their menu until the supply of ingredients is restored. See if you can help Little Lemon to carry out this task.

# The SQL query is:

delete\_query\_greek="""DELETE FROM Menus WHERE Cuisine = 'Greek'"""

# Execute the query

print("Executing 'DELETE' query")

cursor.execute(delete\_query\_greek)

# Commit change

print("Comitting change to the table")

connection.commit()

print("The table is updated after deletion of the requested records")

# The query to retrieve records from Menus table is:

all\_menus = """SELECT \* FROM Menus;"""

# Execute query

cursor.execute(all\_menus)

# Fetch all results that satisfy the query

results = cursor.fetchall()

# Retrieve column names

cols = cursor.column\_names

# Print column names and records from results using for loop

print("""Data in the "Menu" table:""")

print(cols)

for result in results:

print(result)

delete\_query\_null="""DELETE FROM Bookings WHERE TableNo IS NULL;"""

cursor.execute(delete\_query\_null)

connection.commit()

# Let's close the cursor and the connection

if connection.is\_connected():

cursor.close()

print("The cursor is closed.")

connection.close()

print("MySQL connection is closed.")

else:

print("Connection is already closed")