**CS306-Project Phase III-Ege Tan-30977**

**Description:**

For project phase 3, I coded a Python application to interact with a MySQL database, focusing on performing CRUD (Create, Read, Update, Delete) operations on two specific tables, Students and Project\_Assigned. The goal is to design and implement a system that allows efficient manipulation of database records through a well-structured Python script, ensuring the integrity and security of database operations.I started by setting up the necessary database connection using MySQL Connector. The create\_connection function establishes a connection to the MySQL database using the provided credentials. It includes error handling to manage potential connection issues, like incorrect credentials or a non-existent database, ensuring robust and reliable database connectivity.The first set of operations I implemented are the create\_student and create\_project functions. These functions are responsible for inserting new records into the Students and Project\_Assigned tables, respectively. Each function establishes a database connection, creates a cursor object, constructs an SQL INSERT query with placeholders for the parameters, and executes the query. After executing the query, the transaction is committed to ensure the changes are saved to the database. Finally, the cursor and connection are closed to release the database resources.Next, I implemented the read\_students and read\_projects functions, which retrieve and display records from the respective tables. These functions also establish a database connection and create a cursor object. An SQL SELECT \* query is constructed to fetch all records. The results are fetched and returned as a list of tuples.Then, I worked on the update\_student\_gpa and update\_project\_deadline functions to modify existing records in the database. These functions follow a similar structure to the create functions. They establish a database connection, create a cursor object, construct an SQL UPDATE query, execute the query with the provided parameters, and commit the transaction. Finally, I implemented the delete\_student and delete\_project functions, which are responsible for removing records from the database. These functions also follow a similar structure, establishing a connection, creating a cursor, constructing an SQL DELETE query, executing the query, and committing the transaction.

**A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedStudents Table:**

**Project\_Assigned Table:**

**Printing the whole table :**SELECT \* FROM Students

SELECT \* FROM Project\_Assigned

def read\_students():

conn = create\_connection()

if conn:

cursor = conn.cursor()

query = "SELECT \* FROM Students"

cursor.execute(query)

rows = cursor.fetchall()

cursor.close()

conn.close()

print("SELECT \* FROM Students")

return rows

# Function to read all project records

def read\_assigned\_projects():

conn = create\_connection()

if conn:

cursor = conn.cursor()

query = "SELECT \* FROM Project\_Assigned"

cursor.execute(query)

rows = cursor.fetchall()

cursor.close()

conn.close()

print("SELECT \* FROM Project\_Assigned")

return rows

**Inserting New Row to Students and Project\_Assigned:**

INSERT INTO Students (studentid, sname, sgpa) VALUES ( 11 , Super\_User , 4.0 ),

INSERT INTO Project\_Assigned (projectid,pdeadline,studentid) VALUES (101,2023-05-20,11)

def create\_student(student\_id, name, gpa):

conn = create\_connection()

if conn:

cursor = conn.cursor()

query = "INSERT INTO Students (studentid, sname, sgpa) VALUES (%s, %s, %s)"

cursor.execute(query, (student\_id, name, gpa))

conn.commit()

cursor.close()

conn.close()

print("INSERT INTO Students (studentid, sname, sgpa) VALUES (",student\_id,name,gpa,")")

def create\_assigned\_project(project\_id, deadline, student\_id):

conn = create\_connection()

if conn:

cursor = conn.cursor()

query = "INSERT INTO Project\_Assigned (projectid, pdeadline, studentid) VALUES (%s, %s, %s)"

cursor.execute(query, (project\_id, deadline, student\_id))

conn.commit()

cursor.close()

conn.close()

print("INSERT INTO Project\_Assigned (projectid, pdeadline, studentid) VALUES (",project\_id,deadline,student\_id,")")

create\_student(11, "Super\_User", 4.0)

create\_assigned\_project(101, "2023-05-20", 11)

**Code Console Output:**

Students:

Connection established with the database

INSERT INTO Students (studentid, sname, sgpa) VALUES ( 11 , Super\_User , 4.0 )

(1, 'John Doe', Decimal('3.75'))

(2, 'Jane Smith', Decimal('3.90'))

(3, 'Alice Johnson', Decimal('3.60'))

(4, 'Bob Williams', Decimal('3.80'))

(5, 'Eva Brown', Decimal('3.95'))

(6, 'Charlie Davis', Decimal('3.70'))

(7, 'Grace Lee', Decimal('3.85'))

(8, 'David Taylor', Decimal('3.45'))

(9, 'Sophia White', Decimal('3.92'))

(10, 'Michael Jackson', Decimal('3.55'))

(11, 'Super\_User', Decimal('4.00'))

Projects:

Connection established with the database

INSERT INTO Project\_Assigned (projectid,pdeadline,studentid) VALUES (101,2023-05-20,11)

(1, datetime.date(2022, 3, 1), 1)

(2, datetime.date(2022, 3, 2), 2)

(3, datetime.date(2022, 3, 3), 3)

(4, datetime.date(2022, 3, 4), 4)

(5, datetime.date(2022, 3, 5), 5)

(6, datetime.date(2022, 3, 6), 6)

(7, datetime.date(2022, 3, 7), 7)

(8, datetime.date(2022, 3, 8), 8)

(9, datetime.date(2022, 3, 9), 9)

(10, datetime.date(2022, 3, 10), 10)

(101, datetime.date(2023, 5, 20), 11)

**Updating a Row From Students and Project\_Assigned:**

UPDATE Students SET sgpa = 0.0 WHERE studentid = 11

UPDATE Project\_Assigned SET pdeadline = 2023-06-01 WHERE projectid = 101

def update\_student\_gpa(student\_id, new\_gpa):

conn = create\_connection()

if conn:

cursor = conn.cursor()

query = "UPDATE Students SET sgpa = %s WHERE studentid = %s"

cursor.execute(query, (new\_gpa, student\_id))

conn.commit()

cursor.close()

conn.close()

print("UPDATE Students SET sgpa = ",new\_gpa," WHERE studentid = ",student\_id)

def update\_assigned\_project\_deadline(project\_id, new\_deadline):

conn = create\_connection()

if conn:

cursor = conn.cursor()

query = "UPDATE Project\_Assigned SET pdeadline = %s WHERE projectid = %s"

cursor.execute(query, (new\_deadline, project\_id))

conn.commit()

cursor.close()

conn.close()

print("UPDATE Project\_Assigned SET pdeadline = ",new\_deadline," WHERE projectid = ",project\_id)

update\_student\_gpa(11, 0.0)

update\_assigned\_project\_deadline(101, "2023-06-01")

**Code Console Output:**

Students:

Connection established with the database

UPDATE Students SET sgpa = 0.0 WHERE studentid = 11

(1, 'John Doe', Decimal('3.75'))

(2, 'Jane Smith', Decimal('3.90'))

(3, 'Alice Johnson', Decimal('3.60'))

(4, 'Bob Williams', Decimal('3.80'))

(5, 'Eva Brown', Decimal('3.95'))

(6, 'Charlie Davis', Decimal('3.70'))

(7, 'Grace Lee', Decimal('3.85'))

(8, 'David Taylor', Decimal('3.45'))

(9, 'Sophia White', Decimal('3.92'))

(10, 'Michael Jackson', Decimal('3.55'))

(11, 'Super\_User', Decimal('0.00'))

Projects:

Connection established with the database

UPDATE Project\_Assigned SET pdeadline = 2023-06-01 WHERE projectid = 101

(1, datetime.date(2022, 3, 1), 1)

(2, datetime.date(2022, 3, 2), 2)

(3, datetime.date(2022, 3, 3), 3)

(4, datetime.date(2022, 3, 4), 4)

(5, datetime.date(2022, 3, 5), 5)

(6, datetime.date(2022, 3, 6), 6)

(7, datetime.date(2022, 3, 7), 7)

(8, datetime.date(2022, 3, 8), 8)

(9, datetime.date(2022, 3, 9), 9)

(10, datetime.date(2022, 3, 10), 10)

(101, datetime.date(2023, 6, 1), 11)

**Deleting Row from Students and Project\_Assigned:**

DELETE FROM Students WHERE studentid = 11

DELETE FROM Project\_Assigned WHERE projectid = 101

def delete\_student(student\_id):

conn = create\_connection()

if conn:

cursor = conn.cursor()

query = "DELETE FROM Students WHERE studentid = %s"

cursor.execute(query, (student\_id,))

conn.commit()

cursor.close()

conn.close()

print("DELETE FROM Students WHERE studentid = ",student\_id)

def delete\_assigned\_project(project\_id):

conn = create\_connection()

if conn:

cursor = conn.cursor()

query = "DELETE FROM Project\_Assigned WHERE projectid = %s"

cursor.execute(query, (project\_id,))

conn.commit()

cursor.close()

conn.close()

print("DELETE FROM Project\_Assigned WHERE projectid = ",project\_id)

delete\_student(11)

delete\_assigned\_project(101)

**Code Console Output:**

Students:

Connection established with the database

DELETE FROM Students WHERE studentid = 11

(1, 'John Doe', Decimal('3.75'))

(2, 'Jane Smith', Decimal('3.90'))

(3, 'Alice Johnson', Decimal('3.60'))

(4, 'Bob Williams', Decimal('3.80'))

(5, 'Eva Brown', Decimal('3.95'))

(6, 'Charlie Davis', Decimal('3.70'))

(7, 'Grace Lee', Decimal('3.85'))

(8, 'David Taylor', Decimal('3.45'))

(9, 'Sophia White', Decimal('3.92'))

(10, 'Michael Jackson', Decimal('3.55'))

Projects:

Connection established with the database

DELETE FROM Project\_Assigned WHERE projectid = 101

(1, datetime.date(2022, 3, 1), 1)

(2, datetime.date(2022, 3, 2), 2)

(3, datetime.date(2022, 3, 3), 3)

(4, datetime.date(2022, 3, 4), 4)

(5, datetime.date(2022, 3, 5), 5)

(6, datetime.date(2022, 3, 6), 6)

(7, datetime.date(2022, 3, 7), 7)

(8, datetime.date(2022, 3, 8), 8)

(9, datetime.date(2022, 3, 9), 9)

(10, datetime.date(2022, 3, 10), 10)

**Before and After MySQL Outputs:**

**A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedInitial tables:**

**A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedAfter inserting new rows:**

**After updating the rows:**

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

**After deleting the rows:**

**A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated**