КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
імені ТАРАСА ШЕВЧЕНКА

ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра прикладних інформаційних систем

Лабораторна робота №1

з курсу

«Технології обробки даних»

Студента 5 курсу
групи ІС-11
спеціальності 122 «Комп'ютерні науки»
ОП «Інформаційні системи»
Іванова Віктора Миколайовича

Київ – 2023

Робота в Python з MySQL

1. Підключіться до бази даних MySQL за допомогою mysql-connector-python.
2. Створіть таблицю students з наступними полями: id (INT, PRIMARY KEY, AUTO_INCREMENT), name (VARCHAR), age (INT), email (VARCHAR).
3. Додайте 5 студентів до таблиці students.
4. Виконайте запит для вибірки всіх студентів з таблиці students.
5. Виконайте запит для вибірки студента з таблиці students за ім'ям.
6. Оновіть вік одного зі студентів в таблиці students.
7. Видаліть студента з таблиці students за заданим ідентифікатором.
8. Використовуючи транзакції, додайте ще двох студентів до таблиці students, але якщо в процесі додавання виникне помилка, скасуйте зміни.
9. Створіть таблицю courses з полями id (INT, PRIMARY KEY, AUTO_INCREMENT), name (VARCHAR), description (VARCHAR), credits (INT).
10. Додайте 3 курси до таблиці courses.
11. Створіть таблицю student_courses для зв'язку між студентами та курсами з полями student_id (INT, FOREIGN KEY), course_id (INT, FOREIGN KEY).
12. Заповніть таблицю student_courses даними про курси, які вибрали студенти.
13. Виконайте запит для вибірки всіх студентів, які вибрали певний курс.
14. Виконайте запит для вибірки всіх курсів, які вибрали студенти за певним ім'ям.
15. Використовуючи JOIN, виконайте запит для вибірки всіх студентів та їх курсів, на які вони записані.

Код програми

```
import mysql.connector



# Підключення до бази даних

mydb = mysql.connector.connect(

    host="your_host",

    user="your_username",

    password="your_password",

    database="your_database"

)



# 1. Підключення до бази даних MySQL за допомогою mysql-connector-python

print("Connected to MySQL database")



# 2. Створення таблиці students
```

```
mycursor = mydb.cursor()

mycursor.execute("CREATE TABLE students (id INT PRIMARY KEY AUTO_INCREMENT,
name VARCHAR(255), age INT, email VARCHAR(255))")

print("Table 'students' created")



# 3. Додавання 5 студентів до таблиці students

students = [
    ("John Doe", 20, "john@example.com"),
    ("Jane Smith", 22, "jane@example.com"),
    ("Mike Johnson", 19, "mike@example.com"),
    ("Emily Davis", 21, "emily@example.com"),
    ("David Wilson", 23, "david@example.com")
]

sql = "INSERT INTO students (name, age, email) VALUES (%s, %s, %s)"

mycursor.executemany(sql, students)

mydb.commit()
```

```
print("5 students added to 'students' table")



# 4. Вибірка всіх студентів з таблиці students

mycursor.execute("SELECT * FROM students")

result = mycursor.fetchall()

print("All students in 'students' table:")

for row in result:

    print(row)



# 5. Вибірка студента з таблиці students за ім'ям

name = "John Doe"

sql = "SELECT * FROM students WHERE name = %s"

mycursor.execute(sql, (name,))

result = mycursor.fetchall()

print(f"Student with name '{name}':")
```

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

 # 6. Оновлення віку одного зі студентів в таблиці students

```
student_id = 1
new_age = 25
sql = "UPDATE students SET age = %s WHERE id = %s"
mycursor.execute(sql, (new_age, student_id))
mydb.commit()
print(f"Age of student with id {student_id} updated to {new_age}")
```

 # 7. Видалення студента з таблиці students за заданим ідентифікатором

```
student_id = 3
sql = "DELETE FROM students WHERE id = %s"
mycursor.execute(sql, (student_id,))
```

```
mydb.commit()

print(f"Student with id {student_id} deleted from 'students' table")



# 8. Використання транзакцій для додавання ще двох студентів до таблиці students

# з можливістю скасування змін, якщо виникне помилка

try:

    mydb.start_transaction()

    students = [

        ("Sarah Johnson", 19, "sarah@example.com"),

        ("Michael Brown", 22, "michael@example.com")

    ]

    sql = "INSERT INTO students (name, age, email) VALUES (%s, %s, %s)"

    mycursor.executemany(sql, students)

    mydb.commit()

    print("2 students added to 'students' table")
```

```
except Exception as e:
    mydb.rollback()
    print("Error occurred. Changes rolled back.")
    print(str(e))


# 9. Створення таблиці courses
mycursor.execute("CREATE TABLE courses (id INT PRIMARY KEY AUTO_INCREMENT,
name VARCHAR(255), description VARCHAR(255), credits INT)")
print("Table 'courses' created")


# 10. Додавання 3 курсів до таблиці courses
courses = [
    ("Mathematics", "Advanced calculus", 4),
    ("Physics", "Classical mechanics", 3),
    ("Computer Science", "Introduction to programming", 3)
]
```

```
sql = "INSERT INTO courses (name, description, credits) VALUES (%s, %s, %s)"

mycursor.executemany(sql, courses)

mydb.commit()

print("3 courses added to 'courses' table")
```

 # 11. Створення таблиці student_courses для зв'язку між студентами та курсами

```
mycursor.execute("CREATE TABLE student_courses (student_id INT, course_id INT,

FOREIGN KEY (student_id) REFERENCES students(id), FOREIGN KEY (course_id)

REFERENCES courses(id))")

print("Table 'student_courses' created")
```

 # 12. Заповнення таблиці student_courses даними про курси, які вибрали студенти

```
student_courses = [

    (1, 1),

    (1, 2),

    (2, 2),
```

```
    (3, 3),
    (4, 1),
    (5, 3)
]
sql = "INSERT INTO student_courses (student_id, course_id) VALUES (%s, %s)"
mycursor.executemany(sql, student_courses)
mydb.commit()
print("Data added to 'student_courses' table")

# 13. Вибірка всіх студентів, які вибрали певний курс
course_id = 2
sql = "SELECT s.* FROM students s INNER JOIN student_courses sc ON s.id =
sc.student_id WHERE sc.course_id = %s"
mycursor.execute(sql, (course_id,))
result = mycursor.fetchall()
print(f"All students who chose course with id {course_id}:")
```

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

 # 14. Вибірка всіх курсів, які вибрали студенти за певним ім'ям

```
name = "John Doe"
sql = "SELECT c.* FROM courses c INNER JOIN student_courses sc ON c.id = sc.course_id
INNER JOIN students s ON sc.student_id = s.id WHERE s.name = %s"
mycursor.execute(sql, (name,))
result = mycursor.fetchall()
print(f"All courses chosen by student with name '{name}':")
for row in result:
    print(row)
```

 # 15. Використання JOIN для вибірки всіх студентів та їх курсів

```
sql = "SELECT s.*, c.name FROM students s INNER JOIN student_courses sc ON s.id =
sc.student_id INNER JOIN courses c ON sc.course_id = c.id"
```

```
mycursor.execute(sql)

result = mycursor.fetchall()

print("All students and their courses:")

for row in result:

    print(row)



# Закриття з'єднання з базою даних

mydb.close()
```