

Лабораторна робота № 8

“Створення консольного інтерфейсу до сервера баз даних”

з курсу “Організація баз даних та знань”

Виконав:

Студент групи ФЕС-21с

Шавало А. А.

Викладач: асист. Галяткін О. О.

Львів 2024

Лабораторна робота №8. Створення консольного інтерфейсу до сервера баз даних

Мета роботи: навчитись створювати консольні інтерфейси до баз даних та керувати базами даних за допомогою програмних засобів.

Хід роботи

1.

- а) Підключається до бази даних розробленої у попередніх лабораторних роботах:

```
import mysql.connector

def connect_to_db(): 1usage new *
    connection = mysql.connector.connect(
        host='localhost',
        user='root',
        password='',
        database='lab5'
    )
    return connection
```

- б) Виводить список усіх таблиць в базі даних:

```
def show_tables(cursor): 1usage new *
    cursor.execute("SHOW TABLES")
    tables = cursor.fetchall()
    print("Таблиці бази даних:")
    for table in tables:
        print(table[0])
```

```
Таблиці бази даних:
cases
clients
lawyers
```

с) Виводить структуру усіх таблиць в базі даних:

```
def show_table_structure(cursor): 1usage new *
    cursor.execute("SHOW TABLES")
    tables = cursor.fetchall()
    for table in tables:
        print(f"\nСтруктура таблиці: {table[0]}")
        cursor.execute(f"DESCRIBE {table[0]}")
        structure = cursor.fetchall()
        for column in structure:
            print(column)
```

Структура таблиці: cases

Назва стовпця	Тип даних	NULL	Ключ	За замовчуванням
case_id	smallint unsigned	NO	PRI	NULL
client_id	smallint unsigned	NO	MUL	NULL
lawyer_id	smallint unsigned	NO	MUL	NULL
case_type	varchar(255)	NO		NULL
article_code	varchar(50)	YES		NULL
punishment_min	smallint	YES		NULL
punishment_max	smallint	YES		NULL
fee	decimal(10,2)	YES		NULL
case_status	varchar(50)	YES		NULL
case_result	varchar(255)	YES		NULL
received_punishment	smallint	YES		NULL

Структура таблиці: clients

Назва стовпця	Тип даних	NULL	Ключ	За замовчуванням
id	smallint unsigned	NO	PRI	NULL
full_name	varchar(255)	NO		NULL
birth_date	date	NO		NULL
address	varchar(255)	NO		NULL
phone	varchar(20)	YES		NULL
status	enum('фізична особа', 'юридична особа')	NO		NULL

- d) Виводить результати запиту, що містить використання псевдонімів стовпців та математичні дії або вбудовані функції;

```
def math_ps(cursor): 1usage new *
    cursor.execute("""
        SELECT
            case_type AS 'Тип справи',
            article_code AS 'Код статті',
            punishment_max AS 'Максимальний термін',
            punishment_min AS 'Мінімальний термін',
            (punishment_max + punishment_min) / 2 AS 'Середній термін'
        FROM cases;
    """)
    result = cursor.fetchall()

    print("\nЗапит з псевдонімами та математичними операціями:")
    print(
        f"{'Тип справи':<25} {'Код статті':<15} {'Максимальний термін':<20} {'Мінімальний термін':<20} {'Середній термін':<20}"
    )
    print("=" * 100)

    for row in result:
        print(f"{row[0]:<25} || {row[1]:<15} || {row[2]:<20} || {row[3]:<20} || {row[4]:<20}")
```

Запит з псевдонімами та математичними операціями:

Тип справи	Код статті	Максимальний термін	Мінімальний термін	Середній термін
Кримінальна справа	ККУ ст.115	15	7	11.0000
Цивільна справа	ЦКУ ст.203	0	0	0.0000
Адміністративна справа	КУпАП ст.173	0	0	0.0000
Кримінальна справа	ККУ ст.187	10	5	7.5000
Цивільна справа	ЦКУ ст.215	0	0	0.0000
Кримінальна справа	ККУ ст.289	7	3	5.0000
Адміністративна справа	КУпАП ст.122	0	0	0.0000
Цивільна справа	ЦКУ ст.376	0	0	0.0000
Кримінальна справа	ККУ ст.186	8	4	6.0000
Кримінальна справа	ККУ ст.187	10	5	7.5000

- e) Виводить результати запиту, що містить фільтрацію по певному стовпцю (рівне, більше, належить діапазону, тощо);

```
def filter(cursor): 1usage new *
    cursor.execute("SELECT * FROM cases WHERE punishment_min > 4 AND punishment_max < 11;")
    result = cursor.fetchall()
    print("\nЗапит з фільтрацією по стовпцю:")
    for row in result:
        print(row)
```

Запит з фільтрацією по стовпцю:

ID	Тип справи	Код статті	Максимальний термін	Мінімальний термін
4	4	4	Кримінальна справа	ККУ ст.187
10	10	10	Кримінальна справа	ККУ ст.187

- f) Виводить результати запиту, що містить використання операторів AND, OR, NOT:

```
def logical_operators(cursor):
    cursor.execute("""
        SELECT
            case_type AS 'Тип справи',
            article_code AS 'Код статті',
            punishment_max AS 'Максимальний термін',
            punishment_min AS 'Мінімальний термін',
            case_status AS 'Статус справи'
        FROM cases
        WHERE
            punishment_min > 5
            AND punishment_max < 15
            OR case_status NOT LIKE 'активна';
    """)

    result = cursor.fetchall()

    print("\nЗапит з використанням операторів AND, OR, NOT:")
    print(
        f"{'Тип справи':<30} {'Код статті':<23} {'Максимальний термін':<23} {'Мінімальний термін':<23} {'Статус справи':<20}"
    )
    print("=" * 110)

    for row in result:
        print(f"{row[0]:<30} || {row[1]:<20} || {row[2]:<20} || {row[3]:<20} || {row[4]:<20}")
```

Запит з використанням операторів AND, OR, NOT:				
Тип справи	Код статті	Максимальний термін	Мінімальний термін	Статус справи
Кримінальна справа	ККУ ст.115	15	7	завершена
Цивільна справа	ЦКУ ст.203	0	0	завершена
Адміністративна справа	КУпАП ст.173	0	0	завершена
Кримінальна справа	ККУ ст.187	10	5	в процесі
Цивільна справа	ЦКУ ст.215	0	0	завершена
Кримінальна справа	ККУ ст.289	7	3	в процесі
Адміністративна справа	КУпАП ст.122	0	0	завершена
Цивільна справа	ЦКУ ст.376	0	0	завершена
Кримінальна справа	ККУ ст.186	8	4	в процесі
Кримінальна справа	ККУ ст.187	10	5	в процесі

- g) Виводить результати запиту, що містить сортування результату:

```
def sorting(cursor):
    cursor.execute(
        "SELECT case_type AS 'Тип справи', article_code AS 'Код статті', punishment_max AS 'Максимальний термін', punishment_min AS 'Мінімальний термін', value AS 'Вартість'"
    )
    result = cursor.fetchall()

    print("\nЗапит з сортуванням результатів:")
    print(
        f"{'Тип справи':<30} {'Код статті':<23} {'Максимальний термін':<23} {'Мінімальний термін':<23} {'Вартість':<15}"
    )
    print("=" * 280)

    for row in result:
        print(f"{row[0]:<30} || {row[1]:<20} || {row[2]:<25} || {row[3]:<25} || {row[4]:<15}")
```

Запит з сортуванням результатів:				
Тип справи	Код статті	Максимальний термін	Мінімальний термін	Вартість
Кримінальна справа	ККУ ст.187	10	5	12000.00
Кримінальна справа	ККУ ст.115	15	7	10000.00
Кримінальна справа	ККУ ст.187	10	5	10000.00
Кримінальна справа	ККУ ст.186	8	4	9500.00
Кримінальна справа	ККУ ст.289	7	3	8000.00
Цивільна справа	ЦКУ ст.376	0	0	6000.00
Цивільна справа	ЦКУ ст.203	0	0	5000.00
Цивільна справа	ЦКУ ст.215	0	0	4000.00
Адміністративна справа	КУпАП ст.173	0	0	3000.00
Адміністративна справа	КУпАП ст.122	0	0	2000.00

2. Додати у програму 2-3 запити внесення нових даних у будь-якій таблиці БД.

```
def insert_data(cursor, connection): 1 usage new *
    cursor.execute("""
        INSERT INTO clients (full_name, birth_date, address, phone, status)
        VALUES ('Новий Клієнт', '1991-01-01', 'м. Київ, вул. Нова, 1', '+380631111111', 'фізична особа');
    """)
    connection.commit()
    print("\nНовий запис додано до таблиці clients.")
```

Новий запис додано до таблиці clients.

3. Додати у програму 2-3 запити оновлення даних у будь-якій таблиці БД.

```
def update_data(cursor, connection): 1 usage new *
    cursor.execute("""
        UPDATE cases SET case_status = 'завершена' WHERE case_id = 2;
    """)
    connection.commit()
    print("\nДані оновлено.")
```

Дані оновлено.

Висновок: Я реалізував програму на Python, яка підключається до бази даних та виконує кілька запитів, таких як виведення списку таблиць, їх структури, математичних операцій з використанням псевдонімів, фільтрацію за певними критеріями, логічні оператори, а також сортування результатів. Ця програма демонструє основні можливості роботи з базами даних, спрощуючи аналіз інформації та управління даними.