

Практическое занятие №15

Тема: Составление программ для работы с базами данных в IDE PyCharm Community.

Цель: закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, работы с БД в IDE PyCharm Community.

Постановка задачи: Средствами Python реализовать программу для работы с однотабличной БД. Программа должна обеспечивать функционал по вводу данных в БД (10 позиций), их поиску, удалению и редактированию. При организации поиска, удаления и редактирования использовать WHERE, предусмотреть по три SQL-запроса для каждой операции.

Вариант 16

Приложение КОНТРОЛЬ ИСПОЛНЕНИЯ ПОРУЧЕНИЙ для некоторой организации. БД должна содержать таблицу Поручения со следующей структурой записи: Порядковый номер поручения, Название поручения, Дата выдачи поручения, Срок исполнения, Исполнитель.

Текст программы:

```
import sqlite3
import os

root = os.path.abspath('.')

db_path = os.path.join(root, 'assignments.db')

def create_table():
    conn = sqlite3.connect(db_path)
    try:
        cursor = conn.cursor()

        cursor.execute("DROP TABLE IF EXISTS Assignments")

        cursor.execute("CREATE TABLE IF NOT EXISTS Assignments
                        (id INTEGER PRIMARY KEY AUTOINCREMENT,
                         title TEXT,
                         issue_date TEXT,
                         due_date TEXT,
                         executor TEXT)")
        conn.commit()
    except sqlite3.Error as e:
        print("Ошибка при работе с базой данных ", e)
    finally:
        conn.close()

def insert_data(info)
    conn = sqlite3.connect(db_path)
    try:
        cursor = conn.cursor()
        cursor.executemany("INSERT INTO Assignments (title, issue_date, due_date, executor) VALUES
(?, ?, ?, ?)", info)
```

```

    conn.commit()
    print("Данные успешно добавлены в базу данных.")
except sqlite3.Error as e
    print("Ошибка при добавлении данных ", e)
finally
    conn.close()

def display_all()
    conn = sqlite3.connect(db_path)
    try
        cursor = conn.cursor()
        cursor.execute("SELECT * FROM Assignments")
        rows = cursor.fetchall()
        for row in rows
            print(row)
    except sqlite3.Error as e
        print("Ошибка при выводе данных ", e)
    finally
        conn.close()

def search_by_executor(executor)
    conn = sqlite3.connect(db_path)
    try
        cursor = conn.cursor()
        cursor.execute("SELECT * FROM Assignments WHERE executor=?", (executor,))
        rows = cursor.fetchall()
        for row in rows
            print(row)
    except sqlite3.Error as e
        print("Ошибка при поиске данных ", e)
    finally
        conn.close()

def delete_by_id(assignment_id)
    conn = sqlite3.connect(db_path)
    try
        cursor = conn.cursor()
        cursor.execute("DELETE FROM Assignments WHERE id=?", (assignment_id,))
        conn.commit()
        print(f"Запись с ID '{assignment_id}' успешно удалена.")
    except sqlite3.Error as e
        print("Ошибка при удалении записи ", e)
    finally
        conn.close()

def update_due_date(assignment_id, new_due_date)
    conn = sqlite3.connect(db_path)
    try
        cursor = conn.cursor()
        cursor.execute("UPDATE Assignments SET due_date=? WHERE id=?", (new_due_date, assignment_id))
        conn.commit()
        print(f"Срок исполнения для записи с ID '{assignment_id}' успешно обновлен.")
    except sqlite3.Error as e

```

```

    print("Ошибка при обновлении записи ", e)
finally
    conn.close()

create_table()

info = [
    ('Подготовка отчета', '2024-05-01', '2024-05-10', 'Иванов'),
    ('Проверка документации', '2024-05-03', '2024-05-15', 'Петров'),
    ('Разработка презентации', '2024-05-05', '2024-05-20', 'Сидоров'),
    ('Анализ данных', '2024-05-07', '2024-05-25', 'Кузнецов'),
    ('Тестирование системы', '2024-05-10', '2024-05-30', 'Смирнов')
]

insert_data(info)

print("Таблица целиком ")
display_all()

print("\nПоиск по исполнителю 'Иванов' ")
search_by_executor('Иванов')

print("\nПоиск по исполнителю 'Петров' ")
search_by_executor('Петров')

print("\nУдаление записи с ID 1 ")
delete_by_id(1)

print("\nТаблица после удаления ")
display_all()

print("\nОбновление срока исполнения для записи с ID 2 ")
update_due_date(2, '2024-06-01')

print("\nТаблица после обновления ")
display_all()

```

Протокол работы программы:

Данные успешно добавлены в базу данных.

Таблица целиком:

```

(1, 'Подготовка отчета', '2024-05-01', '2024-05-10', 'Иванов')
(2, 'Проверка документации', '2024-05-03', '2024-05-15', 'Петров')
(3, 'Разработка презентации', '2024-05-05', '2024-05-20', 'Сидоров')
(4, 'Анализ данных', '2024-05-07', '2024-05-25', 'Кузнецов')
(5, 'Тестирование системы', '2024-05-10', '2024-05-30', 'Смирнов')

```

Поиск по исполнителю 'Иванов':

```

(1, 'Подготовка отчета', '2024-05-01', '2024-05-10', 'Иванов')

```

Поиск по исполнителю 'Петров':

(2, 'Проверка документации', '2024-05-03', '2024-05-15', 'Петров')

Удаление записи с ID 1:

Запись с ID '1' успешно удалена.

Таблица после удаления:

(2, 'Проверка документации', '2024-05-03', '2024-05-15', 'Петров')

(3, 'Разработка презентации', '2024-05-05', '2024-05-20', 'Сидоров')

(4, 'Анализ данных', '2024-05-07', '2024-05-25', 'Кузнецов')

(5, 'Тестирование системы', '2024-05-10', '2024-05-30', 'Смирнов')

Обновление срока исполнения для записи с ID 2:

Срок исполнения для записи с ID '2' успешно обновлен.

Таблица после обновления:

(2, 'Проверка документации', '2024-05-03', '2024-06-01', 'Петров')

(3, 'Разработка презентации', '2024-05-05', '2024-05-20', 'Сидоров')

(4, 'Анализ данных', '2024-05-07', '2024-05-25', 'Кузнецов')

(5, 'Тестирование системы', '2024-05-10', '2024-05-30', 'Смирнов')

Вывод: В процессе выполнения практического занятия закрепила усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составления программ, работы с БД в IDE PyCharm Community.

Выполнены разработка кода, отладка, тестирование, оптимизация программного кода. Готовые программные коды выложены на GitHub.