Т. 12. №1 2026
https://doi.org/10.33619/2414-2948/122

УДК 004.021:421 https://doi.org/10.33619/2414-2948/122/11

ИСПОЛЬЗОВАНИЕ МОДУЛЯ SQLITE ДЛЯ СОЗДАНИЯ БАЗ ДАННЫХ С ПОМОЩЬЮ PYTHON

***©Кадыркулова Н. К.**, SPIN-код: 7273-7751, Ошский технологический университет им. М. М. Адышева, г. Ош, Кыргызстан, kadyrkulova74@mail.ru*
***©Жеенбекова Н. Б.**, Ошский технологический университет им. М. М. Адышева, г. Ош, Кыргызстан, gbnurpery@gmail.com*

Using the SQLite module to create databases with Python

***©Kadyrkulova N.**, SPIN-code: 7273-7751, Osh Technological University named after M. M. Adyshev, Osh, Kyrgyzstan, kadyrkulova74@mail.ru*
***©Zheenbekov N.**, Osh Technological University named after M. M. Adyshev, Osh, Kyrgyzstan, gbnurpery@gmail.com*

Аннотация. Рассматривается модуль SQLite, предназначенный для работы с базами данных. Основная цель статьи — показать, как использовать модуль SQLite в Python для управления созданием базы данных. Предметом исследования является реализация и использование баз данных в среде программирования Python. Модуль SQLite используется для работы с базами данных, не требующими выделенного сервера. Актуальность статьи обусловлена тем, что SQLite — удобный формат файлов для хранения и управления данными. В результате была разработана программа, использующая модуль SQLite для создания приложений на языке программирования Python.

Abstract. Discusses the SQLite module, which is designed to work with databases. The main goal of the article is to show how to use the SQLite module in Python to manage the creation of a database. The subject of the study is the provision and use of databases in the Python programming environment. The SQLite module is used to work with databases that do not require a dedicated server. The relevance of the article is due to the fact that SQLite is a convenient file format for storing and managing data. As a result, a program was developed that uses the SQLite module to create applications in the Python programming language.

Ключевые слова: база данных, Python, SQL, система управления базами данных (СУБД), SQLite.

Keywords: database, Python, SQL, database management system (DBMS), SQLite.

Работа с базами данных одно из важнейших и фундаментальных направлений программирования. В настоящее время различные приложения, веб-сайты и аналитические системы основаны на базе данных. Поэтому усвоение информации на языке предикатов стало очень важным не только для ИТ-специалистов, но и для студентов инженерных, экономических и естественных наук.

База данных — это структура, предназначенная для хранения больших объемов информации и набора программных модулей для управления, выбора, сортировки и сравнения информации. Информация в базе данных хранится в одной или нескольких таблицах. Данные и таблица состоят из набора одинаковых записей, расположенных рядом друг с другом. База данных (БД) — это электронное хранилище информации, к которому можно получить доступ

Т. 12. №1 2026
https://doi.org/10.33619/2414-2948/122

с помощью одного или нескольких компьютеров. Базы данных обычно создаются для хранения и доступа к данным, содержащим информацию о предметной области, человеческой деятельности или области реального мира [1-2].

Как правило, база данных моделирует определенную предметную область или ее фрагмент. Часто файл базы данных действует как постоянный репозиторий данных. Программное обеспечение, которое манипулирует информацией в базе данных, называется системой управления базами данных (СУБД). Он может делать выбор на основе различных критериев и отображать запрашиваемую информацию в удобной для пользователя форме. Основными компонентами информационных систем, построенных на базе баз данных, являются файлы базы данных и программное обеспечение (клиентские приложения), которые позволяют выполнять действия, необходимые для решения ее проблем. Информация, хранящаяся в базе данных, постоянно обновляется. Актуальность зависит от того, как часто это делается. Информация об объектах также может быть изменена и дополнена. В нашем случае создана база данных по студентам факультета. Была разработана удобная технология для переключения между окнами для поиска информации о студентах. Цель исследования — использовать модуль SQLite для создания баз данных в среде Python. Стандартный пакет Python включает библиотеку для решения различных проблем. Качественные библиотеки для Python доступны в интернете в различных предметных областях: инструменты обработки текста и Интернет-технологии, обработка изображений, инструменты для создания приложений, механизмы доступа к базам данных, пакеты для научных вычислений, пакеты для создания графических интерфейсов и т. д. [3-5].

Среда Python может использоваться в различных областях, таких как: системное управление, Программирование встроенных систем, разработка прикладного программного обеспечения, разработка игр, тестирование, компьютерная графика, базы данных, визуализация данных и т. д. Язык Python предлагает широкий спектр возможностей для работы с базами данных различных структур. Это также один из способов использования базы данных и хранения информации. Python полезен не только для хранения данных, но и для выбора конкретных записей для различных параметров.

SQL — это язык, который поддерживает базы данных, таблицы внутри них и т. д., структурированный язык запросов, позволяющий управлять. Язык структурированных запросов SQL разделен на следующие категории, и каждая категория имеет свой собственный оператор [6]:

ЯОД (язык определения данных) — это язык описания данных.

ЯМД (язык манипулирования данными) — это язык манипулирования данными.

ЯУД (язык управления данными) — язык определения доступа к данным.

ЯУТ (язык управления транзакциями) — это язык управления транзакциями.

Для работы с реляционными данными (СУБД) в Python мы можем использовать два метода: 1) работа с библиотекой, относящейся к базе данных, и использование SQL для работы с базой данных. Например, модуль sqlite3 используется для работы с SQLite. 2) Работа с использованием объектно-ориентированного подхода (ООП). Например, SQL Alchemy.

Давайте посмотрим еще более подробно на модуль SQLite. Модуль SQLite — это механизм транзакций без отдельного сервера для базы данных SQL. Версия Python 2.5 получила модуль sqlite3, поэтому мы можем загрузить дополнительные инструменты и создать базу данных в любой текущей версии Python. Модуль SQLite может предоставить простую интегрированную базу данных для приложения. Он основан на работе с файлами, а также предоставляет большой набор инструментов для работы с ними, что отличает его от сетевых систем управления базами данных (СУБД).

Т. 12. №1 2026
https://doi.org/10.33619/2414-2948/122

Когда вы работаете с SQLite, вы обращаетесь непосредственно к файлам, а не к портам из систем управления базами данных, что обеспечивает большую скорость. Вся база данных состоит из одного файла, что упрощает передачу его на разные компьютеры.

Эта СУБД отлично подходит для разработки и тестирования, не требует дополнительной установки и имеет хорошие возможности масштабирования в соответствии со стандартом SQL. Основными недостатками этой системы являются отсутствие пользовательской системы и возможность повышения производительности. Например, Mozilla использует базу данных SQLite в своем популярном браузере Firefox для хранения закладок и другой информации.

Модуль SQLite обеспечивает способ создания небольших и быстрых баз данных, но нам также известно, что существуют и другие модули расширения Python для других баз данных. Например, MySQL, zxjdbc, dcoracle2, Sybase и другие. Несмотря на достаточную теоретическую основу и постоянную реализацию, реляционная модель-не единственная модель, которая успешно используется сегодня. Например, язык программирования XML имеет интерфейс для работы с Python [3].

Древовидная модель данных более естественна для XML, основанного на множестве задач, или проще и последовательнее реляционные СУБД также исследуются для работы. Язык программирования Python является одним из тестов для этого исследования.

При решении конкретной проблемы мы должны выбрать наиболее подходящие инструменты для разработки программного обеспечения. Некоторые используют односторонний подход к этому выбору, Выбирая модель данных, которая не является оптимальной (для данной задачи или подзадачи).

В результате данные, которые по своей природе легко представить в другой модели, должны храниться и обрабатываться в выбранной модели, часто непреднамеренно моделируя естественные возможности и структуры хранения. Конечно, XML можно хранить в реляционной базе данных, а табличные данные-в XML, но это неестественно. По этой причине сложность и отказоустойчивость программного продукта возрастают, даже если используемые инструменты высокого качества. В Python вам необходимо импортировать модуль sqlite3 для работы с базой данных SQLite. Затем вы можете создать соединение с базой данных с помощью функции Connect() и указать путь к файлу базы данных. Если файла нет, SQLite сгенерирует его автоматически. Для этого начните с создания таблицы в базе данных. После того, как вы установили соединение с базой данных, вам нужно создать курсор. Курсор позволяет выполнять SQL-запросы и получать результаты. SQL-запрос может быть выполнен с использованием метода Execute (). Например, чтобы создать таблицу, мы можем записать следующий код (Рисунок 1).

```
import sqlite3
import json
import chardet
# pip install chardet

DATABASE = 'student_tests.db'

# Connect to the database
conn = sqlite3.connect(DATABASE)
```

Рисунок 1. Подключение к базе данных

Затем записываем следующий код для вызова метода cursor object() при создании одного или нескольких курсоров (Рисунок 2).

Т. 12. №1 2026
https://doi.org/10.33619/2414-2948/122

```
# Create a cursor
cursor = conn.cursor()
```

Рисунок 2. Вызов метода извлечения объекта

Чтобы использовать метод execute() при выполнении команды или запроса, вы должны написать код, как показано на (Рисунок 3).

```
# Check if the 'students' table exists
cursor.execute('''SELECT name FROM sqlite_master WHERE type='table' AND name='students' ''')
```

Рисунок 3. Способ выполнения команды или запроса

Чтобы получить результат в запросе, мы используем метод fetchone () и пишем код следующим образом (Рисунок 4).

```
# If the 'students' table does not exist, create it
if not cursor.fetchone():
    cursor.execute('''CREATE TABLE students (id INTEGER PRIMARY KEY, first_name TEXT, last_name TEXT, email TEXT)''')
```

Рисунок 4. Метод получения результата

Чтобы завершить или отменить транзакцию, используйте метод соединения commit() или rollback (). Для этого необходимо написать код на (Рисунок 5).

```
# Insert the student
cursor.execute('''INSERT INTO students (first_name, last_name, email) VALUES (?, ?, ?)''', (first_name, last_name, email))
conn.commit()
print('Сохранено!')
```

Рисунок 5. Метод завершения или отмены транзакции

После того, все необходимые транзакции будут завершены, так как программа завершается записью команды close () (Рисунок 6).

```
# Close the connection
conn.close()
```

Рисунок 6. Транзакции

Кроме того, для Python был разработан стандарт DB-API версии 2.0, которому должны следовать разработчики всех модулей реляционных баз данных. Фактически, DB-API версии 2.0 описывает модуль интерфейса базы данных и имена функций и классов, которые должны содержать их семантику. Интерфейсный модуль должен содержать класс объектов подключения к базе данных и класс курсоров и реализован на уровне приложения связи через СУБД. Самым популярным и удобным модулем является SQLite, поэтому он небольшой, но быстро работает с базой данных. По этой причине язык программирования Python имеет возможность использовать множество встроенных модулей для работы с базами данных различных структур. Чтобы укрепить ваше понимание, вышеупомянутые программы-фреймворки могут помочь вам быть более уверенными в использовании Python для решения проблем с модулем SQLite.

Т. 12. №1 2026
https://doi.org/10.33619/2414-2948/122

Вывод

Следовательно, язык Python выбор конкретной базы данных напрямую зависит от ее дальнейшего использования и, кроме того, предоставляет только широкий набор инструментов для работы с каждой из них. Эта работа не только поможет освоить модуль SQLite, но и вдохновит на создание собственных проектов и исследований.

Список литературы:

1. Мансуров К. Т., Аленов Б. М. Использования компонента stringgrid для работы с базой данных // Вестник Жалал-Абадского государственного университета. 2023. №2(56). С. 365-369.

2. Кадыркулова Н. К., Алижанова Г. А. Маалыматтар базасын башкаруунун жана mysql маалымат базасын башкаруу системасынын өзгөчөлүктөрү // Вестник Жалал-Абадского государственного университета. 2023. №2(56). Р. 360-365.

3. Сузи Р. А. Язык программирования PYTHON. М., 2007. 326 с.

4. Хеллман Д. Стандартная библиотека Python3: справочник с примерами. СПб.: Диалектика, 2020. 1376 с.

5. Химич А. В. Технологии оцифровки как инструмент систематизации геоботанических данных // Университет на пути к новому качеству науки и образования: сборник статей национальной научно-практической конференции. Брянск, 2020. С. 477.

6. Чан Уэсли. Python. Создание приложений. М.: Вильямс, 2016. 808 с.

References:

1. Mansurov, K. T., & Alenov, B. M. (2023). Ispol'zovaniya komponenta stringgrid dlya raboty s bazoi dannykh. *Vestnik Zhalal-Abadskogo gosudarstvennogo universiteta*, (2(56)), 365-369. (in Russian).

2. Kadyrkulova, N. K., & Alizhanova, G. A. (2023). Maalymattar bazasyn bashkaruunun zhana mysql maalymat bazasyn bashkaruu sistemasynyn өzgөchөlүktөrү. *Vestnik Zhalal-Abadskogo gosudarstvennogo universiteta,* (2(56)), 360-365. (in Russian).

3. Suzi, R. A. (2007). Yazyk programmirovaniya PYTHON. Moscow. (in Russian).

4. Khellman, D. (2020). Standartnaya biblioteka Python3: spravochnik s primerami. St. Petersburg. (in Russian).

5. Khimich, A. V. (2020). Tekhnologii otsifrovki kak instrument sistematizatsii geobotanicheskikh dannykh. In Universitet na puti k novomu kachestvu nauki i obrazovaniya: sbornik statei natsional'noi nauchno-prakticheskoi konferentsii, Bryansk. (in Russian).

6. Chan, Uesli (2016). Python. Sozdanie prilozhenii. Moscow. (in Russian).

Поступила в редакцию
12.11.2025 г.

Принята к публикации
19.11.2025 г.

Ссылка для цитирования:

Кадыркулова Н. К., Жеенбекова Н. Б. Использование модуля sqlite для создания баз данных с помощью Python // Бюллетень науки и практики. 2026. Т. 12. №1. С. 84-88. https://doi.org/10.33619/2414-2948/122/11

Cite as (APA):

Kadyrkulova, N., & Zheenbekov, N. (2026). Using the SQLite module to create databases with Python. *Bulletin of Science and Practice, 12*(1), 84-88. (in Russian). https://doi.org/10.33619/2414-2948/122/11