



## Почему MariaDB хорошо подходит для бэкенда на Python

### Описание

Выбор правильной базы данных для вашего приложения очень важен, особенно для внутренних приложений, которые напрямую взаимодействуют с базой данных. Существует множество вариантов, каждый из которых подходит для определенных случаев использования и имеет различные цены, скорости и возможности масштабирования. Python обычно используется в приложениях с большим объемом данных, поскольку он имеет мощные библиотеки для работы с данными, поэтому база данных, которую вы используете для приложения Python, имеет большое значение. В этой статье MariaDB сравнивается с другими технологиями баз данных, такими как MySQL и PostgreSQL, и обсуждается, почему это отличный выбор при использовании Python.

## Почему MariaDB так хорошо подходит для Python?

Благодаря своим мощным библиотекам для работы с данными и машинного обучения, Python является популярным языком программирования для специалистов по работе с данными – особенно в области инженерии данных и науки о данных. Обе области имеют дело с большими объемами данных, что увеличивает потребности в хранении, памяти и вычислительной мощности. Стоимость лицензирования часто растет вместе с объемом данных, поэтому имеет смысл использовать базу данных с открытым исходным кодом. Вы также можете сэкономить на аппаратном обеспечении. Поскольку вы полностью контролируете, где будет развернуто приложение, вы можете найти самые дешевые решения в облаке или на месте.

Существует множество баз данных с открытым исходным кодом, включая реляционные, NoSQL и графовые базы данных. Реляционные базы данных, как правило, отлично подходят для обеспечения структуры и согласованности, в то время как базы данных NoSQL, которые позволяют быстро развиваться, разработаны для большей гибкости. Хотя Python прекрасно работает с базами данных NoSQL, реляционные базы данных лучше подходят для аналитики и обработки больших объемов данных, которые обычно используются в Python. Давайте сравним MariaDB с некоторыми другими популярными реляционными базами данных с открытым исходным кодом.

## MariaDB vs MySQL vs PostgreSQL

Наиболее популярными реляционными базами данных с открытым исходным кодом являются MariaDB, SQLite, MySQL и PostgreSQL. Каждая из них обладает схожим набором основных технических возможностей, но MariaDB имеет некоторые уникальные особенности, которые делают ее наиболее подходящей для приложений Python. Более того, MariaDB является открытым исходным кодом, поэтому ее возможности не зависят от внешнего субъекта, как MySQL, который сейчас принадлежит Oracle. Хотя сообщество разработчиков с открытым исходным кодом все еще может предлагать и добавлять функции в MySQL, Oracle (у которой есть конкурирующая база данных) решает, куда ей двигаться.

С другой стороны, MariaDB была создана после того, как Oracle приобрела MySQL, чтобы продолжить подход к разработке под руководством сообщества. В результате MariaDB имеет более богатый набор функций, включая большее количество движков хранения данных, которые обеспечивают более высокую производительность запросов и репликации, чем MySQL. Эта более высокая

производительность важна при работе с большими массивами данных. Причины использования MariaDB вместо PostgreSQL немного более тонкие, так как PostgreSQL также создается сообществом. Однако PostgreSQL использует собственную BSD-подобную лицензию, PostgreSQL License, которая является более свободной, чем лицензия GNU для MariaDB, и позволяет пользователям создавать расширения с закрытым исходным кодом.

Хотя они приносят пользу пользователю, создающему расширение, эти функции не всегда используются в основной базе данных PostgreSQL, и разработчики могут даже взимать плату за использование своих расширений. Лицензирование MariaDB и MySQL GNU не позволяет приватизировать новые возможности – все новые функциональные возможности находятся в свободном доступе. MariaDB и PostgreSQL имеют самые богатые наборы функций. Однако у MariaDB есть несколько удобных функций для бэкенда на Python. Например, MariaDB использует один язык для взаимодействия со всеми своими различными движками хранения данных – OLAP и OLTP системы управляются одним и тем же синтаксисом, что снижает нагрузку на разработчиков.

Ваш бэкенд на Python может записывать транзакционные данные в MariaDB, которая может копировать эти данные в механизм хранения, более подходящий для аналитических запросов. Разработчики могут писать аналитические запросы к реплике, используя тот же синтаксис для повышения производительности. MariaDB также недавно представила форматирование строк по принципу f-String, аналогичное Python. Это избавляет разработчиков, использующих MariaDB с Python, от необходимости мысленно переключаться между языками.

## MariaDB Python Connector

До 2020 года программисты на Python подключались к MariaDB через пакет MySQL Python. Это было возможно, поскольку MariaDB является форком MySQL, но это означало, что соединения MariaDB вели себя так же, как и MySQL. В 2020 году был выпущен родной коннектор MariaDB, чтобы устранить зависимость от MySQL, предоставив больше контроля сообществу MariaDB. Вы можете установить коннектор с помощью pip, менеджера пакетов Python, и использовать его для всех распространенных случаев использования CRUD.

Управление всеми операциями осуществляется через объект курсора. По умолчанию курсор MariaDB принимает запросы в виде подготовленных запросов,

чтобы вы могли санировать любые динамические части запроса. Это безопаснее, чем форматирование строк для построения запросов, которое делает ваше приложение уязвимым для атак SQL-инъекций.

## Принцип работы коннектора

Коннектор прост в использовании. Сначала вы импортируете библиотеку коннектора MariaDB в свое приложение и используете следующую функцию для подключения к серверу базы данных MariaDB:

```
import mariadb

try:
    connection = mariadb.connect(
        user=username,
        password=password,
        host=mariadb_host,
        port=3306,
        database="sales"
    )
except mariadb.Error as err:
    print(f"An error occurred whilst connecting to MariaDB: {err}")
```

После подключения все запросы выполняются через объект курсора. Вы получаете объект курсора, а затем используете его для отправки запросов.

```
cursor = conn.cursor()
```

Чтобы отправить запрос в виде подготовленного оператора, используйте вопросительный знак в тексте запроса и передайте необходимые значения в виде кортежа.

```
ur.execute( "SELECT * FROM sales WHERE sale_date >= ? and price > ?", (sale_da
```

Входные значения из кортежа заменяют вопросительные знаки в порядке слева направо. Эти входные значения также saniруются для предотвращения SQL-инъекций. Встроенная защита от инъекций SQL выгодна для такого языка, как Python, разработанного с учетом интересов начинающих программистов.

## Python и MariaDB для WordPress

Еще одним преимуществом использования MariaDB в качестве бэкенда для

---

приложения на Python является то, что базу данных легко подключить к фронтенду WordPress. Ваш сайт WordPress может легко получить доступ к наборам данных, обработанных в Python. Например, вы можете выполнить некоторый анализ на Python, сохранить результаты в MariaDB, а затем отобразить их в виде графика на странице WordPress.

## **Заключение**

Несколько реляционных баз данных с открытым исходным кодом, включая MariaDB, MySQL и Postgres, могут служить в качестве бэкенда Python. Однако MariaDB является наиболее гибким и многофункциональным вариантом, благодаря своей природе с открытым исходным кодом. При работе с большими массивами данных MariaDB предоставляет множество движков для хранения данных, что делает ее быстрее альтернатив, и поддерживает множество вариантов использования, от транзакционной обработки до аналитических запросов. Гибкость, скорость и встроенная интеграция с Python с помощью коннектора MariaDB Python делают ее отличным выбором в качестве бэкенда для приложений на Python, обрабатывающих большие массивы данных.

### **Дата Создания**

02.05.2023