

Сравнение СУБД

SQLite

SQLite — это библиотека, встраиваемая в приложение, которое её использует. Она предоставляет набор инструментов для простой (в сравнении с серверными БД) обработки любых видов данных.

Когда приложение использует SQLite, их связь производится с помощью функциональных и прямых вызовов файлов, содержащих данные (например, баз данных SQLite), а не какого-то интерфейса, что повышает скорость и производительность операций.

Преимущества

- **Файловая:** вся база данных хранится в одном файле, что облегчает перемещение.
- **Стандартизированная:** SQLite использует SQL, но некоторые функции опущены
- **Простота**

Недостатки

- **Отсутствие пользовательского управления**
- **Невозможность дополнительной настройки:** SQLite нельзя сделать более производительной (“затюнить”)

Когда стоит использовать

Встроенные приложения: все портируемые не предназначенные для масштабирования приложения — например, локальные однопользовательские приложения, мобильные приложения или игры.

- **Система доступа к дисковой памяти:** приложения, часто производящие прямые операции чтения/записи на диск, можно перевести на SQLite для повышения производительности.
- **Тестирование:** подойдёт для приложений, частью функционала которых является тестирование бизнес-логики.

Когда не стоит использовать SQLite

- **Многопользовательские приложения:** работа над приложением, доступом к БД в котором будут одновременно пользоваться несколько человек, лучше выбрать полнофункциональную СУБД.
- **Приложения, записывающие большие объемы данных:** SQLite допускает единовременное выполнение лишь одной операции записи.

MySQL

MySQL — это самая популярная из всех крупных серверных БД. Хотя MySQL и не пытается полностью реализовать SQL-стандарты, она предлагает широкий функционал. Приложения общаются с базой данных через процесс-демон.

Преимущества

- **Простота:** MySQL легко устанавливается.
- **Много функций:** MySQL поддерживает большую часть функционала SQL.
- **Безопасность:** в MySQL встроено много функций безопасности.
- **Мощность и масштабируемость:** MySQL может работать с действительно большими объёмами данных, и неплохо подходит для масштабируемых приложений.
- **Скорость:** пренебрежение некоторыми стандартами позволяет MySQL работать производительнее.

Недостатки

- **Известные ограничения:** по определению, в MySQL присутствуют определённые ограничения функциональности.
- **Вопросы надёжности:** некоторые операции реализованы менее надёжно, чем в других СУБД.

Когда стоит использовать MySQL

- **Распределённые операции:** когда вам нужен функционал больший, чем может предоставить SQLite, стоит использовать MySQL.
- **Высокая безопасность:** функции безопасности MySQL предоставляют надёжную защиту доступа и использования данных.
- **Веб-сайты и приложения:** большая часть веб-ресурсов вполне может работать с MySQL, несмотря на ограничения.
- **Кастомные решения:** если вы работаете над очень специфичным продуктом, MySQL подстроится под ваши потребности благодаря широкому спектру настроек и режимов работы.

Когда не стоит использовать MySQL

- **SQL-совместимость:** поскольку MySQL не пытается полностью реализовать стандарты SQL, она не является полностью совместимой с SQL.
- **Конкурентность:** хотя MySQL неплохо справляется с операциями чтения, одновременные операции чтения-записи могут вызывать проблемы.
- **Недостаток функций:** в зависимости от выбора движка MySQL может не предоставлять некоторых функций.

PostgreSQL

PostgreSQL — это самая продвинутая СУБД, ориентирующаяся в первую очередь на полное соответствие стандартам и расширяемость. PostgreSQL, или Postgres, пытается полностью соответствовать SQL-стандартам ANSI/ISO.

PostgreSQL отличается от других СУБД тем, что обладает объектно-ориентированным функционалом, в том числе полной поддержкой концепта ACID (Atomicity, Consistency, Isolation, Durability).

Будучи основанным на мощной технологии Postgres отлично справляется с одновременной обработкой нескольких заданий. Поддержка конкурентности реализована с использованием MVCC (Multiversion Concurrency Control), что также обеспечивает совместимость с ACID.

Преимущества

- **Полная SQL-совместимость.**
- **Сообщество:** PostgreSQL поддерживается опытным сообществом 24/7.
- **Расширяемость:** PostgreSQL можно программно расширить за счёт хранимых процедур.
- **Объектно-ориентированность:** PostgreSQL — не только реляционная, но и объектно-ориентированная СУБД.

Недостатки

- **Производительность:** В простых операциях чтения PostgreSQL может уступать своим соперникам.
- **Популярность:** из-за своей сложности инструмент не очень популярен.

Когда стоит использовать PostgreSQL

- **Целостность данных:** если приоритет стоит на надёжность и целостность данных, PostgreSQL — лучший выбор.
- **Сложные процедуры:** если ваша БД должна выполнять сложные процедуры, стоит выбрать PostgreSQL в силу её расширяемости.
- **Интеграция:** если в будущем вам предстоит перемещать всю базу на другое решение, меньше всего проблем возникнет с PostgreSQL.

Когда не стоит использовать PostgreSQL

- **Скорость:** если всё, что нужно — это быстрые операции чтения, не стоит использовать PostgreSQL.
- **Простые ситуации:** если вам не требуется повышенная надёжность, поддержка ACID и всё такое

Вывод

Я выбрал PostgreSQL, так как по идее Telegram bot подразумевает, что к БД будет обращаться много пользователей одновременно. Поэтому SQLite отпадает. На самом деле в моем случае все равно, что использовать PostgreSQL или MySQL. Просто я немного работал до этого с PostgreSQL, поэтому выбрал её.