SQLite – это реляционная база данных, запросы к которой можно осуществлять при помощи языка запросов SQL. База данных не поддерживает все особенности SQL и уступает в функциональности другим развитым СУБД, но вполне подходит для хранения и извлечения информации.

Все базы данных хранятся в файлах, по одному файлу на базу. Количество баз данных, а так же таблиц в них, ограниченно только свободным местом, имеющимся на сайте. А максимально возможный объём одной базы данных составляет 2 Тб. Так как все данные хранятся в файлах, проблем с переносом базы данных с одного хостинга на другой не существует – достаточно лишь скопировать соответствующие файлы

**DDL** (**Data Definition Language) -**язык [определения](https://docs.microsoft.com/en-us/sql/t-sql/statements/statements?view=sql-server-2017) данных. Задачей DDL запросов является создание БД и описание ее структуры. Запросами такого вида устанавливаются правила того, в каком виде различные данные будут размещаться в БД.

**DML** (Data Manipulation Language) - язык [манипулирования](https://docs.microsoft.com/ru-ru/sql/t-sql/queries/queries?view=sql-server-2017) данными. В число запросов этого типа входят различные команды, используя которые непосредственно производятся некоторые манипуляции с данными. DML-запросы нужны для добавления изменений в уже внесенные данные, для получения данных из БД, для их сохранения, для обновления различных записей и для их удаления из БД. В число элементов DML-обращений входит основная часть SQL операторов.

**DCL** (Data Control Language) - язык [управления](https://docs.microsoft.com/en-us/sql/relational-databases/security/permissions-database-engine?view=sql-server-2017) данными. Включает в себя запросы и команды, касающиеся разрешений, прав и других настроек СУБД.

**TCL** (Transaction Control Language) - язык управления транзакциями. Конструкции такого типа применяют чтобы управлять изменениями, которые производятся с использованием DML запросов. Конструкции TCL позволяют нам производить объединение DML запросов в наборы транзакций.

* SQLite не требует отдельного процесса сервера или системы для работы (без сервера).
* SQLite поставляется с нулевой конфигурацией, что означает отсутствие необходимости в настройке или администрировании.
* Полная база данных SQLite хранится в одном кросс-платформенном диске.
* SQLite очень маленький и легкий, менее 400KiB полностью сконфигурированный или менее 250KiB с дополнительными функциями, опущенными.
* SQLite является автономным, что означает отсутствие внешних зависимостей.
* SQLite-транзакции полностью совместимы с ACID, обеспечивая безопасный доступ к нескольким процессам или потокам.
* SQLite поддерживает большинство функций языка запросов, найденных в стандарте SQL92 (SQL2).
* SQLite написан на ANSI-C и предоставляет простой и простой в использовании API.
* SQLite доступен в UNIX (Linux, Mac OS-X, Android, iOS) и Windows (Win32, WinCE, WinRT).

DDL запросы.

Такие запросы используются для создания таблиц. Каждая таблица характеризуется именем и описанием столбцов, которое содержит имя столбца и тип данных.

Пример запроса для создания таблицы:

create Table\_Name (\_id integer primary key autoincrement,

field\_name\_1 text,

field\_name\_2 text);

Первый столбец обозначен, как primary key (первичный ключ), т.е. уникальное число, которое однозначно идентифицирует строку.

Слово autoincrement указывает, что база данных будет автоматически увеличивать значение ключа при добавлении каждой записи, что и обеспечивает его уникальность.

Modification запросы.

Такие запросы используются для добавления, изменения или удаления записей.

Пример запроса на добавление строки:

insert into Table\_Name values(null, value1, value2);

В этом случае значения разместятся в соответствующие столбцы таблицы, первое значение задается для поля \_id и равно null, т. к. SQLite вычисляет значение этого поля самостоятельно. При добавлении можно указывать столбцы, в которые будут размещаться значения, остальные столбцы заполнятся значениями по умолчанию, в этом случае можно добавлять элементы в измененном порядке.

Query запросы. Такие запросы позволяют получать выборки из таблицы по различным критериям.

Пример запроса:

select from Table\_Name where (\_id = smth);

select Field\_Name\_1,

Field\_Name\_2 from Table\_Name

Field\_Name\_1 = smth);

Первый запрос выводит строку с \_id равным smth, второй - вы- водит два элемента Field\_Name\_1 и Field\_Name\_2 строк, в которых Field\_Name\_1 равен smth.

Преимущества

Простота использования

• Отсутствие необходимости настройки сервера СУБД

• Возможность простого распространения со своим продуктом

• Полностью свободная лицензия

• Кроссплатформенность (как

минимум Linux, Free BSD, QNX, Windows)

• Высокая скорость (на простых операциях в разы

быстрее My SQL и на порядок быстрее Postgre SQL)

• Поддержка большого подмножества SQL92

• Поддержка транзакций, триггеров, представлений (views),

вложенных запросов

• Безопасность. БД хранится в одном файле, права доступа

к которому можно контролировать стандартными

средствами ОС

Возможность использовать разные языки программирования:

– tcl/tk— модуль в пакете

– PHP 4, 5

– C — модуль в пакете

– C++

– OCaml

– Perl — модуль на CPAN: DBI&DBD и другие

– Delphi

– Ruby

– Python

– C#

– Java

– Parser 3

– и все прочие — запуском бинарника sqlite в shell/cmd конкретной ОС

• Очень экономичная, в плане ресурсов, архитектура. Вместо

стандартной цепочки:

приложение – клиент (stat или shared lib) – транспорт (tcp или unix

socket) – listener – DB engine – Backend (файлы)

Недостатки:

* Отсутствие системы пользователей - более крупные СУБД включают в свой состав системы управления правами доступа пользователей. Обычно применения этой функции не так критично, так как эта СУБД используется в небольших приложениях.
* Отсутствие возможности увеличения производительности - опять, исходя из проектирования, довольно сложно выжать что-то более производительное из этой СУБД.

Транзакция является рабочей единицей работы с базой данных (далее – БД). Это последовательность операций, выполняемых в логическом порядке пользователем, либо программой, которая работает с БД.

Мы можем сказать, что транзакция – это распространение изменений в БД. Например, если мы создаём, изменяем или удаляем запись, то мы выполняем транзакцию. Крайне важно контролировать транзакции для гарантирования.