**1. Что такое SQLite?**

SQLite — это встраиваемая реляционная база данных, которая реализована в виде библиотеки C и предоставляет автономное хранилище данных. Она не требует установки сервера, конфигурации или управления.

**2. Что такое классы хранения? Какие классы хранения существуют в SQLite?**

**Классы хранения** (storage classes) в SQLite определяют, как данные сохраняются физически. SQLite использует динамическую типизацию и автоматически выбирает подходящий класс хранения для значений. Существуют следующие классы хранения:

* **NULL**: Значение отсутствует.
* **INTEGER**: Целое число.
* **REAL**: Число с плавающей точкой.
* **TEXT**: Текстовые строки, хранящиеся в кодировке UTF-8, UTF-16BE или UTF-16LE.
* **BLOB**: Двоичные данные, которые сохраняются как есть.

\***Тип BLOB** в SQLite определяет столбец, который хранит двоичные данные как есть. Тип BLOB используется для хранения данных, которые не могут быть интерпретированы как строки или числа, такие как изображения, аудио, видео и любые другие файлы в бинарном формате.

**3. Что такое аффинированные типы? Какие аффинированные типы существуют в SQLite?**

Аффинированные типы (affinity types) в SQLite — это концепция, позволяющая определять тип данных, который будет использоваться в столбцах таблицы

**Аффинированные типы** (affinity types) — это правила приведения типов данных в SQLite. Столбцы SQLite могут быть объявлены с типом данных, но фактически данные приводятся к одному из следующих аффинитетов:

* **TEXT**: Преобразует данные в строки.
* **NUMERIC**: Пытается привести данные к числовому типу, включая целые числа, числа с плавающей точкой, даты или время.
* **INTEGER**: Преобразует данные к целочисленному значению.
* **REAL**: Преобразует данные к числу с плавающей точкой.
* **BLOB**: Хранит данные как есть, без изменений.

**4. Как хранить дату в SQLite?**

SQLite не имеет встроенного типа данных для хранения даты и времени. Вместо этого используются следующие подходы:

1. **TEXT**: Формат ISO 8601 (YYYY-MM-DD HH:MM:SS.SSS).
2. **INTEGER**: Хранение метки времени Unix (количество секунд с 1 января 1970 года).
3. **REAL**: Хранение метки времени в формате Julian Date (с плавающей точкой).

**5. Поясните правила сравнения данных в SQLite.**

Правила при сравнении данных:

1. **Тип данных**: Данные приводятся к общему типу перед сравнением.
2. **NULL**: NULL считается меньше любого значения, кроме другого NULL (два NULL равны).
3. **Текст**: Сравнение строк происходит в лексикографическом порядке.
4. **Числа**: Целые и числа с плавающей точкой сравниваются по значению.
5. **BLOB**: Сравниваются побайтово.

**6. Перечислите объекты SQLite.**

* **Таблицы (Tables)**: Хранят данные в строках и столбцах.
* **Представления (Views)**: Сохранённые запросы, представляют виртуальные таблицы.
* **Индексы (Indexes)**: Улучшают производительность запросов.
* **Триггеры (Triggers)**: Автоматически выполняют действия при изменении данных.
* **Виртуальные таблицы**: Позволяют использовать нестандартные хранилища.
* **Внешние ключи**: Устанавливают связь между таблицами.

**7. Поясните порядок использования внешних ключей в SQLite.**

Для работы с внешними ключами в SQLite нужно:

1. Включить поддержку: PRAGMA foreign\_keys = ON;.
2. Определить внешний ключ в схеме таблицы:

|  |
| --- |
| CREATE TABLE child (  id INTEGER PRIMARY KEY,  parent\_id INTEGER,  FOREIGN KEY (parent\_id) REFERENCES parent (id) ); |

1. Убедиться, что родительские записи существуют перед добавлением дочерних записей.
2. При удалении или обновлении родительских записей использовать действия ON DELETE или ON UPDATE (например, CASCADE, SET NULL).

**8. Что такое PRAGMA?**

**PRAGMA** — это команда SQLite, которая позволяет изменять параметры конфигурации базы данных или получать информацию о её состоянии. Примеры:

* PRAGMA foreign\_keys; — проверяет, включены ли внешние ключи.
* PRAGMA cache\_size; — изменяет размер кэша.
* PRAGMA table\_info(table\_name); — отображает схему таблицы.

**9. Какие типы индексов существуют в SQLite?**

SQLite поддерживает:

1. **Обычные индексы**: Используются для ускорения запросов.
2. **Уникальные индексы**: Гарантируют уникальность значений в столбце.
3. **Виртуальные индексы**: Используются в виртуальных таблицах.
4. **Покрывающие индексы**: Содержат все столбцы, необходимые для выполнения запроса, что исключает необходимость обращения к самой таблице.

**10. Какие программные объекты SQLite вы знаете?**

Основные программные объекты:

* **Курсоры**: Для выполнения SQL-запросов и итерации по результатам.
* **Подключение к базе данных (Database Connection)**: Управляет соединением с базой данных.
* **Транзакции**: Для выполнения групп операций атомарно.
* **Подготовленные выражения (Prepared Statements)**: Оптимизированные запросы для многократного выполнения.
* **Функции пользователя (User-defined Functions)**: Кастомные функции, написанные на языке программирования (например, C или Python).