Ответы 10

1. Секционирование таблиц - это процесс разбиения одной таблицы на несколько физических таблиц, называемых разделами, для увеличения производительности и улучшения управляемости данных. Каждый раздел обычно содержит часть данных, которые можно быстрее извлекать или обновлять, чем если бы эти данные хранились в одной большой таблице.
2. Секционирование таблиц целесообразно применять в следующих случаях:
   1. когда таблица содержит большое количество данных;
   2. когда таблицы часто используются для чтения или записи;
   3. когда необходимо улучшить производительность запросов, особенно если они работают с большими объемами данных;
   4. когда необходимо обеспечить лучшую управляемость данных, например, когда требуется удалить или переместить только часть данных.
3. Принципы секционирования для типов секционирования, используемых в заданиях лабораторной работы:
4. Секционирование по диапазону (RANGE) - разделение таблицы на разделы на основе значений в одном или нескольких столбцах. Каждый раздел охватывает диапазон значений столбца или столбцов.
5. Секционирование по списку (LIST) - разделение таблицы на разделы на основе значений в одном столбце. Каждый раздел содержит набор значений столбца.
6. Секционирование по хеш-функции (HASH) - разделение таблицы на разделы на основе хеш-функции, применяемой к одному или нескольким столбцам. Каждый раздел содержит строки с одинаковым хеш-значением.
7. Названия типов секционирования, которые не использовались в заданиях лабораторной работы:
8. Секционирование по времени (TIME) - разделение таблицы на разделы на основе времени создания или обновления строк.
9. Секционирование по раcпределению (RANGE LIST) - разделение таблицы на разделы на основе комбинации секционирования по диапазону и по списку. Каждый раздел содержит строки, которые соответствуют определенному диапазону значений столбца, а также определенному списку значений в другом столбце.
10. Оператор ALTER TABLE MERGE используется для объединения двух или более разделов таблицы в один. При выполнении оператора данные из разных разделов объединяются в один раздел, который заменяет исходные разделы. Это может быть полезно, если данные в таблице стали менее активными, и их можно объединить для уменьшения количества разделов и улучшения производительности.
11. Оператор ALTER TABLE SPLIT используется для разделения одного раздела таблицы на два или более раздела. При выполнении оператора данные из исходного раздела распределяются между новыми разделами на основеуказанных условий, например, значения определенного столбца. Это может быть полезно, если данные в таблице стали более активными, и их нужно разделить на более мелкие блоки для улучшения производительности.
12. Оператор ALTER TABLE EXCHANGE используется для обмена данными между разделом таблицы и отдельной таблицей. При выполнении оператора данные из исходного раздела заменяются данными из отдельной таблицы, а данные из исходного раздела переносятся в отдельную таблицу. Это может быть полезно, если требуется переместить часть данных из одной таблицы в другую, например, для архивирования или для обработки данных в отдельной таблице.

Ответы 11

1. СУБД SQLite - это легковесная, встроенная, серверная СУБД, которая хранит базы данных в одном файле на диске. Она не требует отдельного сервера баз данных и может быть использована как локальное хранилище в приложениях для различных платформ, таких как мобильные устройства или настольные приложения. SQLite поддерживает стандарт SQL и обладает хорошей производительностью в сравнении с другими встроенными СУБД.
2. Ограничения целостности в SQLite могут быть заданы с помощью следующих опций:

* NOT NULL - запрет на вставку значений NULL в столбец;
* UNIQUE - запрет на дублирование значений в столбце;
* PRIMARY KEY - комбинация NOT NULL и UNIQUE, используется для идентификации уникальных записей в таблице;
* FOREIGN KEY - ограничение, которое гарантирует существование связи между двумя таблицами;
* CHECK -ограничение, которое определяет условие, которое должно быть истинным для всех записей в столбце.

1. Collate в SQLite - это правило сравнения, используемое для сортировки и сравнения строк. В SQLite есть несколько видов collate:

* BINARY - сравнение производится байт за байтом;
* NOCASE - сравнение производится без учета регистра символов;
* RTRIM - пробелы в конце строк игнорируются при сравнении;
* UNICODE - сравнение производится на основе стандарта Unicode.

1. В SQLite внешние ключи могут использоваться для поддержки связей между таблицами. Если в таблице создается внешний ключ, SQLite автоматически проверяет существование связанной записи в другой таблице при вставке или обновлении данных. Если связанная запись отсутствует, SQLite либо откажет в выполнении операции, либо удалит или обновит связанные записи в зависимости от настроек внешнего ключа.
2. Pragma в SQLite - это команда SQL, используемая для управления поведением СУБД. Например, с помощью pragma можно получить информацию о базе данных, определить настройки оптимизации запросов или изменить параметры выполнения транзакций.
3. Динамическая типизация данных в SQLite означает, что тип данных для каждого столбца определяется на основе первого значения, которое будет вставлено в него. Это означает, что можно вставлять данные различных типов в один и тот же столбец. Например, если в столбец было вставлено целое число, то SQLite будет автоматически преобразовывать все последующие значения в этом столбце в целые числа. Однако, если в столбец была вставлена строка, то все последующие значения будут также рассматриваться как строки. Это может привести к ошибкам, если тип данных не соответствует ожидаемому. Чтобы избежать таких ошибок, в SQLite рекомендуется явно определять типы данных для каждого столбца при создании таблицы.

ОТВЕТЫ 12

1. Виртуальные таблицы в SQLite - это специальный тип таблиц, который не имеет физического представления на диске, а создается и обслуживается в памяти. Они могут быть созданы с помощью специальных модулей расширения, которые позволяют работать с данными из других источников, таких как файлы, сетевые ресурсы или даже оперативная память.
2. Внутренние таблицы (также называемые системными таблицами) в SQLite - это таблицы, которые используются для хранения метаданных и информации о базе данных, таких как схема таблиц, индексы, права доступа и т.д. Они не предназначены для непосредственной работы с данными.
3. В СУБД SQLite могут быть временными следующие объекты:

* Временные таблицы - создаются на время выполнения сессии и удаляются после ее завершения.
* Временные представления - создаются на время выполнения запроса и удаляются после его завершения.
* Временные индексы - создаются для улучшения производительности запросов, но удаляются после выполнения запроса.

1. Да, DML операции (INSERT, UPDATE, DELETE) могут выполняться над представлениями в SQLite. Однако, если представление содержит операторы JOIN, подзапросы или группировку данных, то эти операции могут быть ограничены или запрещены.
2. Индексирование в SQLite - это процесс создания структуры данных, которая позволяет быстро находить записи в таблице по определенным критериям. Индекс может быть создан для одного или нескольких столбцов в таблице. SQLite поддерживает несколько типов индексов, включая B-дерево, хеш-таблицу и R-дерево.
3. В SQLite существуют два вида триггеров:

* Перед триггеры (BEFORE) - выполняются передвставкой, обновлением или удалением данных в таблице.
* После триггеры (AFTER) - выполняются после вставки, обновления или удаления данных в таблице.

1. В СУБД SQLite триггеры выполняются в порядке их определения. То есть, если в таблице определено несколько триггеров, то они будут выполняться в том же порядке, в котором они были созданы.
2. Транзакция - это логически связанная группа операций с базой данных, которая должна быть выполнена целиком или не выполнена вообще. Транзакция начинается с команды BEGIN TRANSACTION и завершается командами COMMIT или ROLLBACK. Если транзакция не завершается командой COMMIT, то все изменения, сделанные в рамках транзакции, будут отменены с помощью команды ROLLBACK.
3. Вложенные транзакции - это транзакции, которые выполняются внутри другой транзакции. В SQLite вложенные транзакции не поддерживаются, но можно использовать сохранение точек восстановления (savepoints), чтобы имитировать вложенные транзакции.
4. Точки сохранения (savepoints) - это механизм в SQLite, который позволяет сохранять состояние транзакции на определенной точке и продолжать ее выполнение с этой точки в случае ошибки. Точка сохранения создается с помощью команды SAVEPOINT, а откат до точки сохранения выполняется с помощью команды ROLLBACK TO SAVEPOINT. Это может быть полезно, когда необходимо выполнить часть операций в рамках транзакции, но сохранить возможность отката только этой части операций, а не всей транзакции.