Часть 11 2 Ограничения уровня таблицы в базах данных SQLite3

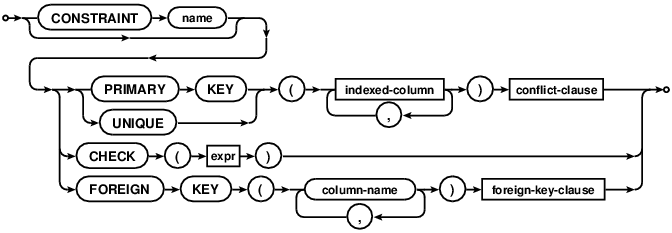
Виды ограничений уровня таблицы в базах данных SQLite

СОДЕРЖАНИЕ СТАТЬИ:

* [1 Виды ограничений уровня таблицы в базах данных SQLite](http://zametkinapolyah.ru/zametki-o-mysql/ogranicheniya-urovnya-tablicy-v-bazax-dannyx-sqlite.html#____SQLite)
* [2 Первичный ключ, как ограничение уровня таблицы в SQLite3](http://zametkinapolyah.ru/zametki-o-mysql/ogranicheniya-urovnya-tablicy-v-bazax-dannyx-sqlite.html#____SQLite3)
* [3 Внешний ключ, как ограничение уровня таблицы в SQLite](http://zametkinapolyah.ru/zametki-o-mysql/ogranicheniya-urovnya-tablicy-v-bazax-dannyx-sqlite.html#____SQLite-2)
* [4 Каскадное удаление и модификация данных в базе данных](http://zametkinapolyah.ru/zametki-o-mysql/ogranicheniya-urovnya-tablicy-v-bazax-dannyx-sqlite.html#i)

Мы подробно рассмотрели [ограничения уровня столбца в базах данных SQLite](http://zametkinapolyah.ru/zametki-o-mysql/ogranicheniya-urovnya-stolbca-v-bazax-dannyx-sqlite.html) и выяснили, что данные ограничения действуют внутри таблицы и[обеспечивают целостность данных](http://zametkinapolyah.ru/zametki-o-mysql/obespechenie-celostnosti-dannyx-v-bazax-dannyx-sqlite3.html) в столбцах базы данных, теперь давайте перейдем к более сложны ограничениям – **ограничениям уровня таблицы**.

На рисунке ниже вы можете увидеть синтаксис использования ограничений уровня таблицы в базах данных SQLIte3.



*Синтаксис использования ограничений уровня таблицы в базах данных SQLite*

Заметим, что ограничения уровня таблицы очень тесно связаны с понятием [нормализации отношения в базе данных](http://zametkinapolyah.ru/zametki-o-mysql/normalnye-formy-izbytochnost-dannyx-v-baze-dannyx-tranzitivnaya-zavisimost-proektirovanie-baz-dannyx.html), а также ограничения уровня таблицы реализуют на практике [связи между таблицами в базе данных](http://zametkinapolyah.ru/zametki-o-mysql/vidy-svyazej-mezhdu-tablicami-v-baze-dannyx-svyazi-v-relyacionnyx-bazax-dannyx-otnosheniya-kortezhi-atributy.html).

**Ограничения уровня таблицы дают возможность обеспечить целостность данных в базе данных**, эти ограничения позволяют нам избавиться от аномалий при использовании [команд группы манипуляции данными](http://zametkinapolyah.ru/zametki-o-mysql/7-komandy-manipulyacii-dannymi-v-sqlite3-dml-operatory-v-sqlite3.html), давайте посмотрим от чего защищают ограничения уровня таблицы:

* от аномалий [добавления данных в таблицы](http://zametkinapolyah.ru/zametki-o-mysql/dobavlenie-dannyx-i-strok-v-tablicy-baz-dannyx-sqlite.html) при использовании [оператора INSERT](http://zametkinapolyah.ru/zametki-o-mysql/chast-7-2-komanda-insert-v-sqlite3-operator-insert-v-sqlite3.html);
* от аномалий модификации данных в таблицах базы данных при использовании [команды UPDATE](http://zametkinapolyah.ru/zametki-o-mysql/chast-7-3-komanda-update-v-sqlite3-operator-update-v-sqlite.html);
* от аномалий [удаления данных из базы данных](http://zametkinapolyah.ru/zametki-o-mysql/udalenie-dannyx-i-strok-iz-tablicy-bazy-dannyx-sqlite.html) при использовании [оператора DELETE](http://zametkinapolyah.ru/zametki-o-mysql/komanda-delete-v-sqlite3-operator-delete-dml.html).

А теперь посмотрим, **какие ограничения уровня таблицы есть в базах данных SQLite3**:

1. [Первичный ключ](http://zametkinapolyah.ru/zametki-o-mysql/chast-3-3-klyuchi-i-klyuchevye-atributy-v-bazax-dannyx.html) или PRIMARY KEY является ограничение уровня таблицы в базе данных SQLite. Если вы помните, то [вторая нормальная форма](http://zametkinapolyah.ru/zametki-o-mysql/vtoraya-normalnaya-forma-2nf.html) требует, чтобы у каждой таблицы был первичный ключ. Первичный ключ в SQLite может быть, как простым, так и составным.
2. Внешний ключ является ограничением уровня таблицы, внешние ключи предназначены для реализации различных видов связи между таблицами. Если отношение находится в [первой нормальной форме](http://zametkinapolyah.ru/zametki-o-mysql/pervaya-normalnaya-forma-1nf.html), то вы навряд ли будете использовать внешние ключи. А вот если ваша база данных нормализована до [третьей нормальной формы](http://zametkinapolyah.ru/zametki-o-mysql/tretya-normalnaya-forma-3nf.html), то таблицы справочники будут связаны с родительскими таблицами внешними ключами, это справедливо и для второй нормальной формы.
3. Автор сам не совсем понимает, почему разработчики SQLite отнесли ограничение CHECK к ограничениям уровня таблицы (вы можете в этом убедиться, открыв документацию). Хотя в документации указано, что не имеет никакого значения, как объявить ограничение CHECK: для таблицы или для столбца. Результат будет одинаковым.
4. SQLite позволяет выполнять операции каскадного удаления и каскадной модификации данных, чтобы избежать трудностей удаления, если ваша база данных находится во второй или третьей нормальной форме, а так же для поддержания целостности данных и устранения аномалий, связанных с этими операциями.

Первичный ключ, как ограничение уровня таблицы в SQLite3

Первичный ключ в теории баз данных – это столбец, который позволяет однозначно идентифицировать таблицы внутри базы данных. Но ни одна [СУБД](http://zametkinapolyah.ru/zametki-o-mysql/arxitektura-subd-arxitektura-baz-dannyx-logicheskaya-struktura-subd-opisanie-dannyx-v-baze-dannyx-bazy-dannyx-sxema-dannyx.html) в мире, в том числе и SQLite, не знает ни про какого Кодда и его [реляционную теорию баз данных](http://zametkinapolyah.ru/zametki-o-mysql/bazy-dannyx-vidy-i-tipy-baz-dannyx-struktura-relyacionnyx-baz-dannyx-proektirovanie-baz-dannyx-setevye-i-ierarxicheskie-bazy-dannyx.html), и даже не знает о том, что есть какой-то там первичный ключ.

Для SQLite первичный ключ или PRIMARY KEY – это всего лишь правило, которое не должен нарушать никто и если вы сказали SQLite, что этот столбец PRIMARY KEY, то она позаботится о том, чтобы значения в этом столбце были уникальными и вечными.

**PRIMARY KEY на самом деле не только ограничение уровня таблицы в базе данных SQLite**, но еще и индекс таблицы, который позволяет значительно ускорить процесс выборки данных из базы данных при помощи [команды SELECT](http://zametkinapolyah.ru/zametki-o-mysql/chast-7-1-komanda-select-v-sqlite3-operator-select-v-sqlite3.html). Если вы знакомы с индексами таблиц в базе данных, то знаете, что любому индексу можно дать имя, даже нужно давать имя, ограничению PRIMARY KEY мы тоже можем дать имя.

Внешний ключ, как ограничение уровня таблицы в SQLite

**Внешний ключи – это ограничение уровня таблицы в реляционной базе данных**. В теории, внешний ключ – это связь между таблицами, как вы помните, связи между таблицами делятся на два вида: связь один ко многим и связь многие ко многим. Связь один к одному мы в расчет не берем, так как обычно такая связь означает, что две таблицы нужно объединить в одну.

SQLite, опять же, не знает про виды связей и внешний ключ для нее – ограничение. Конструкция FOREIGEN KEY говорит SQLite о том, что данный столбец будет связан со столбцом другой таблицы. А конструкция REFERENCE указывает на столбец другой таблицы: то есть дает ссылку, по которой SQLite понимает, что эти таблицы связаны.

Далее мы подробно остановимся на ограничении внешнего ключа в базах данных SQLite. Сейчас важно понять, **что внешний ключ – это ограничение уровня таблицы, которое указывает SQLite на то, что данные в таблицах связаны между собой очень тесно** и SQLite начинает следить за тем, чтобы целостность связей между этими таблицами не была нарушена.

Каскадное удаление и модификация данных в базе данных

Дополнительным механизмом поддержания целостности данных в базах данных SQLite3 является каскадное удаление таблиц и каскадная модификация таблиц в базах данных. Когда мы организуем связи между таблицами при помощи внешнего ключа, SQLite начинает следить за целостностью связей между таблицами и не дает удалить значение из родительской таблицы, пока не будут удалены значения из дочерних таблиц.

Каскадное удаление и модификация данных в таблицах имеют два назначения: первое заключается в том, чтобы облегчить процесс удаления данных из таблицы базы данных (так как не всегда удобно выполнять несколько раз команду DELETE) и облегчить процесс модификации данных в таблицах базы данных (так как команду UPDATE тоже не очень удобно выполнять несколько раз); второе назначение заключается в обеспечение целостности данных и связей между таблицами.

Мы в отдельности рассмотрим каскадное удаление данных при помощи конструкции ON DELETE и каскадную модификацию данных при помощи конструкции ON UPDATE.