**Лабораторное занятие №5.**

**Использование индексов.**

**Лабораторное (практическое) занятие** рассчитано на 2 часа, относится к теме «Эволюция компьютерных сетей»

**Формируемые компетенции:** ОК.1-9 ПК 2.1 - 2.4

**Цель:** Изучить операции для создания индексов. Получить навыки использования программы "SQLiteStudio".

**Методическое и техническое обеспечение:** IBM PC-совместимый компьютер, SQLiteStudio.

## Исходные данные

Студент получает индивидуальный вариант исходных данных с кратким описанием предметной области, который используется при выполнении всех лабораторных работ. При этом каждая очередная лабораторная работа является продолжением выполненной ранее и поэтому они должны обязательно выполняться последовательно. Вариант задания приведен в лабораторной работе №1.

**Теоретические знания**

Индекс создается для одной конкретной таблицы. Имя индекса должно быть уникальным среди имен всех индексов базы данных. Индекс может быть уникальным (ключевое слово UNIQUE). Это означает, что в индексе не может быть двух разных строк с одинаковыми значениями столбцов. Столбцы, входящие в состав уникального индекса, не могут также иметь пустого значения NULL. Ключевое слово ASCENDING (сокращенно ASC) означает, что записи индекса упорядочиваются по возрастанию значений столбцов, входящих в состав индекса. Этот вариант по умолчанию. Ключевое слово DESCENDING (сокращение DESC) указывает, что записи индекса упорядочиваются по уменьшению значений столбцов индекса. В состав индекса не могут входить столбцы, имеющие тип данных BLOB, а также столбцы любого типа данных, являющиеся массивами. Для одной таблицы можно создать не более 256 индексов.

CREATE [UNIQUE] [ASC[ENDING] | DESC[ENDING]]

INDEX <имя индекса> ON <таблица>

{ (<столбец> [, <столбец> ...]) | COMPUTED BY (выражение) };

## Примеры

Если для таблицы PEOPLE требуются и возрастающий, и убывающий индексы по столбцу, хранящему фамилии людей LAST\_NAME, то нужно создать два индекса, выполнив следующие операторы:

CREATE ASCENDING INDEX IDX\_ASC\_PEOPLE ON PEOPLE (LAST\_NAME);

CREATE DESCENDING INDEX IDX\_DESC\_PEOPLE ON PEOPLE (LAST\_NAME);

**ОПЕРАЦИЯ CREATE INDEX**

Когда создается новый индекс, появляются новые записи в системных таблицах RDB$INDICES и RDB$INDEX\_SEGMENTS. Когда изменения подтверждаются (commit), IB строит индекс. Большинство системных утилит создает специальные транзакции для обновления метаданных (включая создание индекса), которые подтверждаются (commit) сразу после выполнения соответствующего оператора DDL (Data Definition Language), но время подтверждения зависит от используемого интерфейса. Изменени метаданных порождают "задания", которые ассоциируются с транзакцией и выполняются по commit.

Когда создание вашего индекса подтверждено (committed), IB читает все записи таблицы в их естественном порядке собирая пары из ключевого значени и идентификатора записи (так называемого db-key). Затем этот набор сортируется по ключевому значению и создается нижняя часть индекса, со сжатием дубликатов ключевых значений. Перед тем как завершится commit, индекс уже построен и прилинкован в список индексов на корневой странице отношений (relations). Для нескольких тысяч записей процесс происходит очень быстро. Если у вас несколько миллионов записей, то это происходит дольше. Вставки и обновления записей в таблице "замрут" (как мне кажется) на время создания индекса.

Примечание КД. Оценить скорость создания индексов можно в статье по тестированию скорости вставки.

Версионирование записей IB усложняет задачу, поскольку некоторые записи могут иметь разные значения индексируемых полей. Код, который строит индексы, стартует специальную транзакцию, которая видит все существующие на данный момент версии записей (естественно, committed).

**КАК ПРОИСХОДИТ ИНДЕКСИРОВАНИЕ МНОГИХ ВЕРСИЙ ЗАПИСЕЙ?**

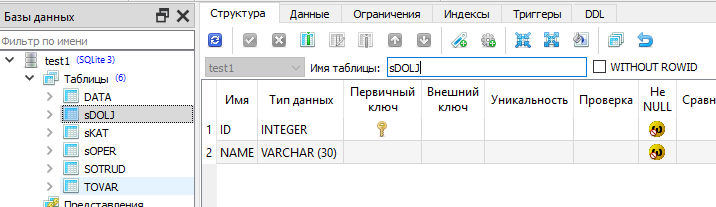
Когда запись обновляется или удаляется, ее старая версия сохраняется с пометкой в виде идентификатора создавшей версию транзакции, и тут же создается новая версия (или stub удаления) с тем же идентификатором транзакции. В зависимости от активности в базе данных, и длительности транзакций, одна запись может иметь несколько версий. Каждая версия может иметь разные значени индексируемых полей. Индекс создается с ключом для КАЖДОЙ версии. Когда старые версии записей уничтожаются, так же уничтожаются и эти ключевые записи в индексе.

Так что если вы создали запись с индексируемым полем со значением А, затем изменили его на Б, затем опять поменяли на А, удаление ключевой записи в индексе для более старой версии не удалит ключевую запись для новой. Интересный аспект этой архитектуры заключается в том, что индекс часто содержит ключевых записей больше, чем записей в соответствующей таблице.

Примечание. О влиянии количества дубликатов или версий ключа на скорость удаления читайте статью по проблемам удаления большого количества записей.

**Работа в SQLiteStudio**

Индексы создаются при автоматически при объявления первичных ключей. (рисунок 1).



Для того чтобы добавить новый индекс необходимо перейти на вкладку «**Индексы**» и нажать на кнопку «**Создать новый индекс**», как представлено на рисунке 2.

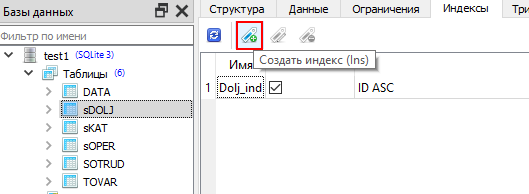


Рисунок 2 – создание нового индекса

В открывшемся окне (рисунок 3) в поле «Table Name» необходимо выбрать таблицу, для которой требуется сделать индекс, в поле «**Имя индекса**» вписываем имя создаваемого индекса. Выставляем галочку «**Уникальный индекс**» и из перечня полей выбираем нужные и нажимаем на кнопку «**ОК**».

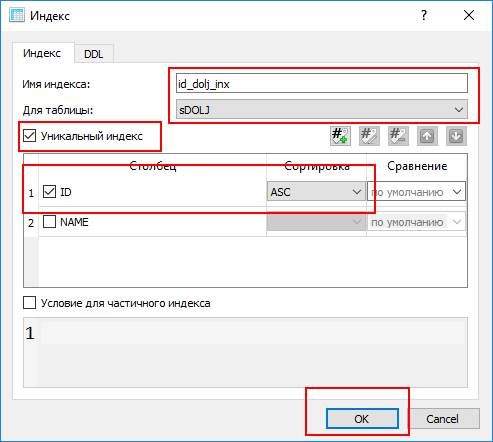


Рисунок 3 – создание нового индекса

**Задание**

Создать требуемые индексы для ранее разработанной базы данной. В отчете описать каждый созданный индекс, его назначение.