

# **Тема: Индексирование: понятие индекса, типы индексных файлов.**

**Создание, активация и  
удаление индекса.**

**Переиндексирование**

Выполнил Булгаков Богдан  
Олегович

9пкс-222к-18

# Понятие индекса и индексных файлов.

- **Индекс** - структура данных, которая помогает СУБД быстрее обнаружить отдельные записи в файле и сократить время выполнения запросов пользователей.
- Индекс в базе данных аналогичен предметному указателю в книге. Это — вспомогательная структура, связанная с файлом и предназначенная для поиска информации по тому же принципу, что и в книге с предметным указателем. Индекс позволяет избежать проведения последовательного или пошагового просмотра файла в поисках нужных данных. При использовании индексов в базе данных искомым объектом может быть одна или несколько записей файла. Как и предметный указатель книги, индекс базы данных упорядочен, и каждый элемент индекса содержит название искомого объекта, а также один или несколько указателей (идентификаторов записей) на место его расположения.

- Структура индекса связана с определенным ключом поиска и содержит записи, состоящие из ключевого значения и адреса логической записи в файле, содержащей это ключевое значение. Файл, содержащий логические записи, называется **файлом данных**, а файл, содержащий индексные записи, — **индексным файлом**. Значения в индексном файле упорядочены по полю индексирования, которое обычно строится на базе одного атрибута.
- Индексные файлы (или индексы) занимают дополнительное место на жестком диске. Если строить индексы по всем полям таблицы и по всем сочетаниям полей, то они (индексы) займут на жестком диске больше места, чем сама база данных.

# Индексы: простые и сложные, по возрастанию и убыванию.

Индекс (или указатель) — это *порядковый номер записи*. При вводе исходных данных записям присваиваются порядковые номера по мере их ввода. Для удобства работы с записями их желательно упорядочить по значениям какого-либо поля, т. е. записи надо расположить или по возрастанию (по алфавиту), или по убыванию значений поля. Таким образом, индексный файл представляет собой

две колонки чисел: в первой колонке указываются номера, которые получили записи при их вводе в таблицу, а во второй колонке — номера, полученные записями после их упорядочения по значениям поля. Если выстраивать записи по номерам второй колонки индексного файла, то записи в таблице будут упорядочены — по **возрастанию** или **убыванию** — по значениям индексного поля, т. е. отсортированы.

Индексы можно строить как по значениям одного поля – **простой индекс**, так и по значениям нескольких полей – **сложный индекс**.

**Первичный индекс** – это индекс, соответствующий ключевому полю. Первичный индекс всегда один. В то же время допускается неограниченное число индексов для других полей.

Вторичный **индекс** – это индекс, созданный для неключевого поля

Вторичный индекс также является упорядоченным файлом, аналогичным первичному индексу. Однако связанный с первичным индексом файл данных всегда отсортирован по ключу этого индекса, тогда как файл данных, связанный со вторичным индексом, не обязательно должен быть отсортирован по ключу индексации

Индексы отличаются от первичных ключей тем, что не требуют непременно уникальности значений входящих в их состав полей. Индексы устанавливаются по полям, которые часто используются при поиске и сортировке данных: индексы помогут системе значительно быстрее найти нужные данные или отсортировать их в нужной последовательности.

# Особенности построения ИНДЕКСОВ.

В Delphi предусмотрено два способа построения индексов:

1) с помощью утилиты Database Desktop. В этом случае индексы строятся один раз во время проектирования приложения базы данных и при эксплуатации базы данных нельзя добавлять новые индексы;

2) с помощью команд. В этом случае программист предоставляет пользователю возможность в любой момент времени создавать и удалять индекс.

Каждый индекс принадлежит конкретной таблице. Таблица может иметь любое количество индексов. Имена индексов уникальные, т. е. у одной таблицы не может быть двух индексов с одинаковыми именами. В разных таблицах имена индексов могут повторяться.

# Работа с индексами с помощью команд

Для создания индексов конкретных полей таблицы служит команда CREATE INDEX:

```
CREATE INDEX имя_индекса ON имя_таблицы  
(имя_поля)
```

Индекс любого поля таблицы можно создавать программно – вызовом метода набора данных AddIndex.

Для работы с индексами предусмотрены три процедуры: создание индекса, определение текущего индекса и удаление индекса.

# Создание индекса

Компоненты Table и Query имеют метод *AddIndex*, с помощью которого создается новый индекс. Формат метода:

```
procedure AddIndex (const Name, Fields .  
string; Options: TIndexOptions);
```

# Назначение аргументов:

*Name* — содержит уникальное имя индекса. Имя индекса пишется буквами английского алфавита и не содержит символа «пробел». Вместо пробела разрешено использование символа «подчеркивание». Имя индекса нужно для того, чтобы впоследствии его можно было уничтожить.

*Fields* — содержит имя поля (или полей), по значениям которого будет построен индекс. Если имен полей несколько, то они отделяются друг от друга символом «точка с запятой». Разрешается строить индекс только по физическим полям, которые объявлены в структуре таблицы.

*Options* — содержит одну или несколько стандартных констант, которые определяют свойства индекса. Допускается использование следующих констант:

- *ixPrimary* — задает первичный индекс;
- *ix Unique* — задает уникальный индекс;
- *ixDescending* — задает индекс по убыванию;
- *ixCaseinsensitive* — задает индекс, нечувствительный к регистру букв.

Аргумент *Options* имеет тип `TIndexOptions`, и константа (или все константы) заключаются в квадратные скобки. Если константы отсутствуют, то все равно надо указать пустые квадратные скобки. Порядок следования полей внутри сложного индекса имеет важное значение.

# Добавление нового индекса.

Добавление нового индекса происходит в режиме эксклюзивного доступа к таблице БД (свойство *Exclusive = True*) и осуществляется следующим методом:

***procedure* AddIndex** (const Name, Fields: String; Options: TIndex Options).

Параметр Name определяет имя индекса, а параметр Fields – список индексных полей. должны указываться только поля, объявленные в структуре БД, в противном случае будет возбуждена исключительная ситуация. Параметр является множеством, которое содержит значения, определяющие свойства индекса:

- ixCaseInsensitive – индекс чувствителен к регистру букв;
- ixDescending- индексные поля сортируются по убыванию значений;
- ixPrimary – создаётся первичный ключ;
- ixUnique – значения полей в индексе должны однозначно определять запись.

# Открытие и закрытие индексного файла.

В реальной работе иногда возникает необходимость вставки в ту или иную таблицу БД сразу множества записей. В этом случае можно временно отключить индекс оператором

```
ALTER INDEX Имя_индекса DEACTIVATE
```

А после завершения операции – вновь его подключить оператором

```
ALTER INDEX Имя_индекса ACTIVATE
```

Вставка с отключенным индексом (индексами) реализуется быстрее, т.к. каждая единичная вставка не вызывает изменения индекса (индексов). Кроме того, последующее включение индекса ведёт к его перерасчёту и способствует его оптимизации.. но в условиях многопользовательской работы отключение и повторное включение индекса проблематично, т.к. с индексом одновременно могут работать два и более пользователя. На практике для улучшения производительности индекса администратор БД должен время от времени его уничтожать оператором

```
DROP INDEX Имя_индекса
```

# Активация индекса.

Для одной таблицы можно создать несколько индексов. В каждый момент времени один из них можно сделать текущим, т.е. активным. Даже при существовании нескольких индексов таблица может не иметь текущего индекса. Текущий индекс важен, например, при выполнении поиска и сортировки записей набора данных компонента *Table*

# Удаление индекса и индексного файла.

Удаление существующего индекса происходит в режиме эксклюзивного доступа к таблице БД и осуществляется следующим методом:

*procedure* DeleteIndex (const Name: String).

Параметр Name определяет имя удаляемого индекса. Удаление индекса выполняется при эксклюзивном доступе к таблице, т.е. свойство *Exclusive* должно иметь значение *True*. При попытке удаления несуществующего индекса возбуждается исключительная ситуация *raise Exception*.

# Переиндексирование: назначение и команда.

Для смены текущего индекса компоненты *Table* и *Query* имеют свойство:

***property*** *IndexName: string;*

После замены текущего индекса набор данных, размещенный в компоненте *Table* (или в *Query*), будет автоматически отсортирован в соответствии с установленным индексом.

Если имя индекса не известно, то процедуру замены индекса можно производить, по именам полей, входящих в индекс. Для этих целей компоненты *Table* и *Query* снабжены свойством:

***property*** *IndexFieldNames: string;*

- Если индекса с указанными именами полей не существует, то возбуждается исключительная ситуация *raise Exception*.
- В Delphi индекс, построенный по полю первичного ключа, имени не имеет.
- Поэтому второй способ замены текущего индекса следует использовать в тех случаях, когда текущим индексом надо сделать первичный ключ (первичный индекс).

**Спасибо за  
внимание**