

Частное учреждение образования
«Колледж бизнеса и права»

УТВЕРЖДАЮ
Ведущий методист
колледжа
_____ Е.В. Паскал
«___» _____ 2022

Специальность: «Программное обеспечение информационных технологий»	Учебная дисциплина: «Базы данных и системы управления базами данных»
--	--

ЛАБОРАТОРНАЯ РАБОТА № 20

Инструкционно-технологическая карта

Тема: Создание, удаление и изменение индексов.

Цель работы: научиться создавать, изменять и удалять и использовать индексы.

Время выполнения: 2 часа

Содержание работы

1. Теоретические сведения для выполнения работы
2. Порядок выполнения работы
3. Пример выполнения работы
4. Контрольные вопросы
5. Литература

1. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ ДЛЯ ВЫПОЛНЕНИЯ РАБОТЫ

Индекс для таблицы создается с помощью инструкции CREATE INDEX. Эта инструкция имеет следующий синтаксис:

```
CREATE [UNIQUE] [CLUSTERED | NONCLUSTERED] INDEX index_name
ON table_name (column1 [ASC | DESC],...)
[INCLUDE (column_name [,... ])]
[WITH [FILLFACTOR=n]
[.,] PAD_INDEX = {ON | OFF}]
[.,] DROP_EXISTING = {ON | OFF}]
[.,] SORT_IN_TEMPDB = {ON | OFF}]
[.,] IGNORE_DUP_KEY = {ON | OFF}]
[.,] ALLOW_ROW_LOCKS = {ON | OFF}]
[.,] ALLOW_PAGE_LOCKS = {ON | OFF}]
[.,] STATISTICS_NORECOMPUTE = {ON | OFF}]
[.,] ONLINE = {ON | OFF}]]
[ON file_group | "default"]
```

Параметр `index_name` задает имя создаваемого индекса. Индекс можно создать для одного или больше столбцов одной таблицы, обозначаемой параметром `table_name`. Столбец, для которого создается индекс, указывается параметром `column1`. Числовой суффикс этого параметра указывает на то, что индекс можно создать для нескольких столбцов таблицы. Компонент Database Engine также поддерживает создание индексов для представлений – индексированных представлений.

Можно проиндексировать любой столбец таблицы. Это означает, что столбцы, содержащие значения типа данных `VARBINARY(max)`, `BIGINT` и `SQL_VARIANT`, также могут быть индексированы.

Индекс может быть простым или составным.

Простой индекс создается по одному столбцу, а составной индекс – по нескольким столбцам. Для составного индекса существуют определенные ограничения, связанные с его размером и количеством столбцов. Индекс может иметь максимум 900 байтов и не более 16 столбцов.

Параметр `UNIQUE` указывает, что проиндексированный столбец может содержать только однозначные значения.

Параметр `CLUSTERED` задает кластеризованный индекс, а параметр `NONCLUSTED` (применяется по умолчанию) указывает, что индекс не изменяет порядок строк в таблице. Компонент Database Engine разрешает для таблицы максимум 249 некластеризованных индексов.

Параметр `INCLUDE` позволяет указать неключевые столбцы, которые добавляются к страницам листьев некластеризованного индекса. Имена столбцов в списке `INCLUDE` не должны повторяться, и столбец нельзя использовать одновременно как ключевой и неключевой.

Параметр `FILEFACTOR=n` задает заполнение в процентах каждой страницы индекса во время его создания. Значение параметра `FILEFACTOR` можно установить в диапазоне от 1 до 100. При значении `n=100` каждая страница индекса заполняется на 100%, т. е. существующая страница листа так же, как страница, не относящаяся к листу, не будет иметь свободного места для вставки новых строк. Поэтому это значение рекомендуется применять только для статических таблиц.

Параметр `PAD_INDEX` указывает, что значение параметра `FILEFACTOR` применяется как к страницам индекса, так и к страницам данных в индексе.

Параметр `DROP_EXISTING` позволяет повысить производительность при воспроизведении кластеризованного индекса для таблицы, которая также имеет некластеризованный индекс.

Параметр `SORT_IN_TEMPDB` применяется для помещения в системную базу данных `tempdb` данных промежуточных операций сортировки, применяющихся при создании индекса.

Параметр `IGNORE_DUP_KEY` разрешает системе игнорировать попытку вставки повторяющихся значений в индексированные столбцы. индексированный столбец.

Когда параметр `ALLOW_ROW_LOCKS` активирован (имеет значение `ON`), система применяет блокировку строк. Подобным образом, когда активирован параметр `ALLOW_PAGE_LOCKS`, система применяет блокировку страниц.

Параметр `STATISTICS_NORECOMPUTE` определяет состояние автоматического перерасчета статистики указанного индекса.

Активированный параметр `ONLINE` позволяет создавать, пересоздавать (`rebuild`) и удалять индекс в диалоговом режиме.

Параметр `ON` создает указанный индекс или на файловой группе по умолчанию (значение `default`), или на указанной файловой группе (значение `file_group`).

2. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. Изучить теоретическую часть настоящей инструкционно-технологической карты.
2. Рассмотреть работу индексов, описанную в разделе «Примеры выполнения работы» настоящей инструкционно-технологической карты.
3. Получить у преподавателя индивидуальное задание и выполнить лабораторную работу в соответствии с вариантом задания согласно описанной в разделе «Пример выполнения работы» методике настоящей инструкционно-технологической карты.
4. Ответить на контрольные вопросы.

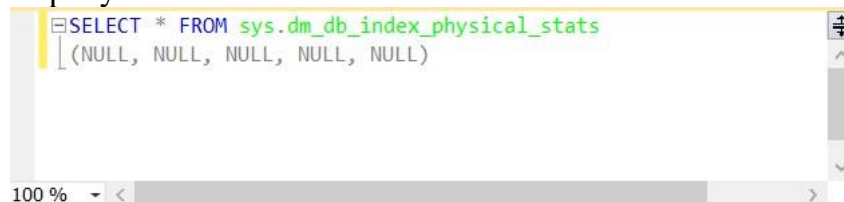
3. ПРИМЕР ВЫПОЛНЕНИЯ РАБОТЫ

1. Создание индексов

Индекс для таблицы создается с помощью инструкции `CREATE INDEX`. В SQL Server поддерживается создание индексов для представлений. Такие представления называются индексируемыми представлениями.

Индекс может быть простым или составным. Простой индекс создается по одному столбцу, а составной индекс — по нескольким столбцам. Для составного индекса существуют определенные ограничения, связанные с его размером и количеством столбцов. Индекс может иметь максимум 900 байтов и не более 16 столбцов.

Чтобы получить информацию обо всех индексах, имеющихся в базу данных можно использовать функцию `sys.dm_db_index_physical_stats ()`. Эта функция имеет пять параметров и все они имеют значения по умолчанию. Можно выставить все параметры по умолчанию и не накладывать фильтр на столбцы и строки, как показано на рисунке ниже.

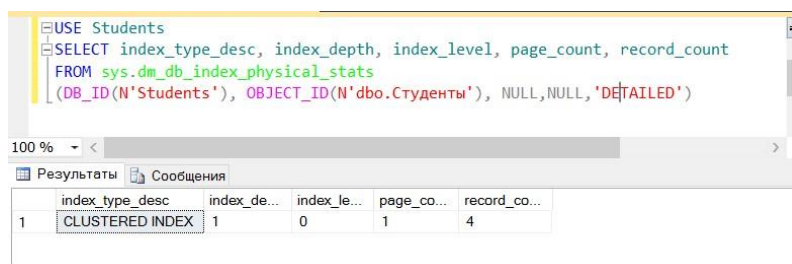


В результате функция возвращает 20 столбцов с данными для (почти) каждого уровня для каждого индекса по каждой таблице каждого раздела в каждой базе данных текущего экземпляра SQL Server.

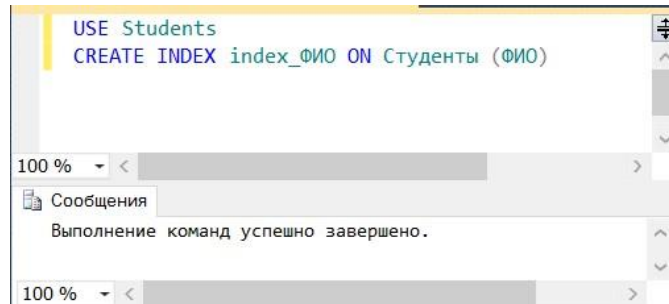
Ниже в таблице приведено описание пяти параметров для данной функции.

Номер параметра	Описание параметра	Пример использования параметра
1	идентификатор ID базы данных, который должен быть указан как число. Если указать NULL, что является значением по умолчанию, функция вернет информацию обо всех базах данных. Если ID не определено, SQL Server будет игнорировать следующие три параметра и предполагает, что они все принимают значения NULL	чтобы упростить получение ID базы данных можно задействовать функцию <code>db_id ()</code> . Например, <code>DB_ID(N'Students')</code>
2	это <code>object_id</code> , который также должен быть задан числом, а не именем. Если выставить NULL, то функция предоставит информацию обо всех объектах, и SQL Server назначит следующие два параметра, <code>index_id</code> и <code>partition_id</code> , также по умолчанию, что означает включение всех индексов и всех разделов	чтобы упростить получение ID объекта можно использовать функцию <code>object_id ()</code> , например, <code>OBJECT_ID(N'dbo.Студенты')</code>
3	позволяет устанавливать ID индекса для указанной таблицы, и значение по умолчанию NULL говорит о том, что пользователь получит на выходе все индексы.	—
4	указывает на номер раздела, и NULL означает, что нужна информация по всем разделам	—
5	это единственный параметр, для которого заданное по умолчанию значение NULL не предполагает возвращения максимума информации. Последний параметр указывает режим выборки, который SQL Server будет использовать при получении данных. Принимаемые параметры следующие: DEFAULT, NULL, LIMITED, SAMPLED или DETAILED. Установка значения NULL по умолчанию соответствует режиму LIMITED	—

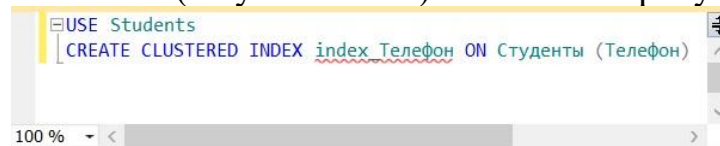
Для того, чтобы получить всю информацию об индексах в таблице Студенты базы данных Students необходимо выполнить код, представленный на рисунке ниже.



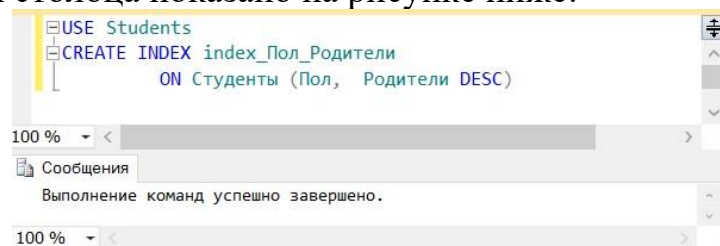
Создание простого некластеризованного (по умолчанию) индекса `index_ФИО` на столбец ФИО в таблице Студенты базы данных Students с порядком сортировки столбца по возрастанию ASC (по умолчанию) показано на рисунке ниже:



Создание простого кластеризованного индекса `index_Телефон` на столбец Телефон в таблице Студенты базы данных Students с порядком сортировки столбца по возрастанию ASC (по умолчанию) показано на рисунке ниже:



Создание составного некластеризованного индекса `index_Пол_Родители` на столбцы Пол и Родители в таблице Студенты базы данных Students с разным порядком сортировки столбца показано на рисунке ниже:



2. Изменение индексов

SQL Server является одной из немногих систем баз данных, которые поддерживают инструкцию `ALTER INDEX`. Эту инструкцию можно использовать для выполнения операций по обслуживанию индекса. Синтаксис инструкции `ALTER INDEX` очень сходен с синтаксисом инструкции `CREATE INDEX`.

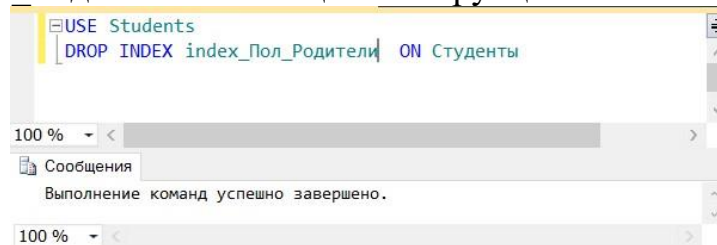
Инструкция `ALTER INDEX` поддерживает три другие параметра:

- ✓ параметр `REBUILD`, используемый для пересоздания индекса;
- ✓ параметр `REORGANIZE`, используемый для реорганизации страниц листьев индекса;
- ✓ параметр `DISABLE`, используемый для отключения индекса. Отключенный индекс недоступен для применения, пока он не будет снова включен. Кроме того, отключенный индекс не изменяется при внесении изменений в соответствующие данные. По этой причине, чтобы снова использовать отключенный индекс, его нужно полностью создать вновь. Для включения отключенного индекса применяется параметр `REBUILD` инструкции `ALTER TABLE`.

3. Удаление и переименование индексов

Для удаления индексов в текущей базе данных применяется инструкция `DROP INDEX`. Удаление кластеризованного индекса таблицы может быть очень ресурсоемкой операцией, т. к. потребуется пересоздать все некластеризованные индексы в связи с тем, что все некластеризованные индексы используют ключ индекса кластеризованного индекса, как указатель в своих страницах листьев.

В следующем примере показано удаление некластеризованного составного индекса `index_Пол_Родители` с помощью инструкции `DROP INDEX`.



Инструкция `DROP INDEX` имеет дополнительный параметр, `MOVE TO`, значение которого аналогично параметру `ON` инструкции `CREATE INDEX`. Иными словами, с помощью этого параметра можно указать, куда переместить строки данных, находящиеся в страницах листьев кластеризованного индекса. Данные перемещаются в новое место в виде кучи.

Индексы можно переименовывать с помощью системной процедуры `sp_rename`.

Замечание. Инструкцию `DROP INDEX` нельзя использовать для удаления индексов, которые создаются неявно системой для ограничений целостности, таких индексов, как `PRIMARY KEY` и `UNIQUE`. Чтобы удалить такие индексы, нужно удалить соответствующее ограничение.

4. Рекомендации по созданию и использованию индексов

Каждый индекс занимает определенный объем дискового пространства, следовательно, существует вероятность того, что общее количество страниц индекса базы данных может превысить количество страниц данных в базе. В отличие от получения выгоды при использовании индекса для выборки данных, вставка и удаление данных такой выгоды не предоставляют по причине необходимости обслуживания индекса. Чем больше индексов имеет таблица, тем больший требуется объем работы по их реорганизации. Общим правилом будет разумно выбирать индексы для частых запросов, а затем оценивать их использование.

4. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Дайте определение понятия «Индекс».
2. Поясните понятие простого и составного индекса.
3. Назовите системную функцию, с помощью которой можно получить информацию о существующих индексах. Поясните ее параметры.
4. Назовите команды создания, удаления и изменения индекса. Как осуществляется группировка, если среди значений столбца встречается значение `NULL`?

5. ЛИТЕРАТУРА

1. Петкович, Д. Microsoft SQL Server 2012. Руководство для начинающих: пер. с английского / Д. Петкович. – СПб.: БХВ-Петербург, 2013. – 816 с.: ил.
2. Сеть разработчиков Microsoft [Электронный ресурс]. – Режим доступа: <https://msdn.microsoft.com/ru-ru/library>

Преподаватель

В.Ю.Купцова

Рассмотрено на заседании цикловой
комиссии программного обеспечения
информационных технологий №10
Протокол № __ от «__» _____ 2022
Председатель ЦК В.Ю.Михалевич