Лабораторная работа № 15 «Создание индексов. Резервное копирование и восстановление БД»

Задание. 1

Используя БД которая получилась при выполнении предыдущей лабораторной работы создайте некластеризованный индекс с именем «index\_name» для столбца «id\_topic» таблицы «posts» и уникальный индекс с именем index\_name2 для столбца password таблицы users. Затем сделайте реорганизацию созданных индексов. Все действия произведите в один SQL- запрос используя оператор GO.

Задание. 2

Создать резервную копию копию БД user\_base c помощью инструкции Backup

Удалить БД и восстановить с помощью инструкции RESTORE

****Индекс**** - это объект базы данных, который представляет собой структуру данных, состоящую из ключей, построенных на основе одного или нескольких столбцов таблицы или представления, и указателей, которые сопоставляются с местом хранения заданных данных. Индексы предназначены для более быстрого получения строк из таблицы, другими словами, индексы обеспечивают быстрый поиск данных в таблице, что значительно повышает производительность запросов и приложений. Индексы также могут быть использованы и для обеспечения уникальности строк таблицы, гарантируя тем самым целостность данных.

В Microsoft SQL Server существуют следующие типы индексов:

* Кластеризованный (*Clustered*) – это индекс, который хранит данные таблицы в отсортированном, по значению ключа индекса, виде. У таблицы может быть только один кластеризованный индекс, так как данные могут быть отсортированы только в одном порядке. По возможности каждая таблица должна иметь кластеризованный индекс, если у таблицы нет кластеризованного индекса, такая таблица называется «*кучей*». Кластеризованный индекс создается автоматически при создании ограничений PRIMARY KEY (*первичный ключ*) и UNIQUE, если до этого кластеризованный индекс для таблицы еще не был определен. В случае создания кластеризованного индекса для таблицы (*кучи*), в которой есть некластеризованные индексы, то после создания все их необходимо перестроить.
* Некластеризованный (*Nonclustered*) – это индекс, который содержит значение ключа и указатель на строку данных, содержащую значение этого ключа. У таблицы может быть несколько некластеризованных индексов. Создаваться некластеризованные индексы могут как на таблицах с кластеризованным индексом, так и без него. Именно этот тип индекса используется для повышения производительности часто используемых запросов, так как некластеризованные индексы обеспечивают быстрый поиск и доступ к данным по значениям ключа;
* Фильтруемый (*Filtered*) – это оптимизированный некластеризованный индекс, который использует предикат фильтра для индексирования части строк в таблице. Если хорошо спроектировать такой тип индекса, то он может повысить производительность запросов, а также снизить затраты на обслуживание и хранение индексов по сравнению с полнотабличными индексами;
* Уникальный (*Unique*) – это индекс, который обеспечивает отсутствие повторяющихся (*одинаковых*) значений ключа индекса, гарантируя тем самым уникальность строк по данному ключу. Уникальными могут быть как кластеризованные, так и некластеризованные индексы. Если создавать уникальный индекс по нескольким столбцам, индекс гарантирует уникальность каждой комбинации значений в ключе. При создании ограничений PRIMARY KEY или UNIQUE SQL сервер автоматически создает уникальный индекс для ключевых столбцов. Уникальный индекс может быть создан только в том случае, если у таблицы на текущий момент отсутствуют дублирующие значения по ключевым столбцам;
* Колоночный (*Columnstore*) – это индекс, основанный на технологии хранения данных в виде столбцов. Данный тип индекса эффективно использовать для больших хранилищ данных, поскольку он может увеличить производительность запросов к хранилищу до 10 раз и также до 10 раз уменьшить размер данных, так как данные в Columnstore индексе сжимаются. Существуют как кластеризованные колоночные индексы, так и некластеризованные;
* Полнотекстовый (*Full-text*) – это специальный тип индекса, который обеспечивает эффективную поддержку сложных операций поиска слов в символьных строковых данных. Процесс создания и обслуживания полнотекстового индекса называется «*заполнением*». Существует такие типы заполнения как: полное заполнение и заполнение на основе отслеживания изменений. По умолчанию SQL сервер полностью заполняет новый полнотекстовый индекс сразу после его создания, но на это может потребоваться значительный объем ресурсов, в зависимости от размеров таблицы, поэтому есть возможность откладывать полное заполнение. Заполнение на основе отслеживания изменений используется для обслуживания полнотекстового индекса после его первоначального полного заполнения;
* Пространственный (*Spatial*) – это индекс, который обеспечивает возможность более эффективного использования конкретных операций на пространственных объектах в столбцах с типом данных geometry или geography. Данный тип индекса может быть создан только для пространственного столбца, также таблица, для которой определяется пространственный индекс, должна содержать первичный ключ (*PRIMARY KEY*);
* XML – это еще один специальный тип индекса, который предназначен для столбцов с типом данных XML. Благодаря XML-индексу повышается эффективность обработки запросов к XML столбцам. Существует два вида XML-индекса: первичные и вторичные. Первичный XML-индекс индексирует все теги, значения и пути, хранимые в XML столбце. Он может быть создан, только если у таблицы есть кластеризованный индекс по первичному ключу. Вторичный XML-индекс может быть создан, только если у таблицы есть первичный XML-индекс и используется он для повышения производительности запросов по определенному типу обращения к XML-столбцу, в связи с этим существует несколько типов вторичных индексов: PATH, VALUE и PROPERTY;
* Также существуют специальные индексы для таблиц, оптимизированных для памяти (*In-Memory OLTP*) такие как: Хэш (*Hash*) индексы и некластеризованные индексы, оптимизированные для памяти, которые создаются для сканирования диапазона и упорядоченного сканирования.

Пример синтаксиса :

Создание:

CREATE UNIQUE CLUSTERED INDEX IX\_Clustered ON TestTable

(

ProductId ASC

)

GO

Удаление:

DROP INDEX IX\_NonClustered ON TestTable

**Реорганизация индекса:**

ALTER INDEX IX\_NonClustered ON TestTable

REORGANIZE

GO

****Перестроение индекса****

ALTER INDEX IX\_NonClustered ON TestTable

REBUILD

GO

Задание. 1

Используя БД которая получилась при выполнении предыдущей лабораторной работы создайте некластеризованный индекс с именем «index\_name» для столбца «id\_topic» таблицы «posts» и уникальный индекс с именем index\_name2 для столбца password таблицы users. Затем сделайте реорганизацию созданных индексов. Все действия произведите в один SQL- запрос используя оператор GO.

Задание. 2

Создать резервную копию копию БД user\_base c помощью инструкции Backup

Удалить БД и восстановить с помощью инструкции RESTORE