

# Производительность

## Лекция №10

Бутенко И.В. 2017 год

# Вопросы разработчику

- Будет ли так же быстро работать этот запрос на больших объемах данных?
- Приведет ли изменение структуры БД к понижению производительности моего запроса?
- Не приведет ли мой запрос к понижению производительности ИС в целом?

# Варианты увеличения производительности

- Увеличение аппаратных мощностей
  - Количество и частота процессоров;
  - Количество оперативной памяти;
  - Быстрая и надежная дисковая подсистема.
- Организационно-административные меры
  - Четкий регламент всех трудоемких операций;
  - Повышение квалификации персонала
- Программно-архитектурные меры
  - Реструктуризация кода
  - Денормализация структур таблиц
- Программные меры
  - Оптимизация запросов БД
  - Оптимизация ХП и др объектов
  - Понижение уровня изоляции транзакций
  - Индексация

# Организационные меры

- Перераспределение «тяжелых» действий
- Формализация действий с прописыванием четких инструкций
- Оптимизация нагрузки на систему со стороны пользователей

# Архитектурные меры

- Выделение отдельных БД/СУБД под определенный класс задач (ХД, OLAP)
- Пересмотр структур БД
- Физическое перераспределение хранимых данных по разным устройствам.

# Программные меры

- ГДЕ проблема?
  - Анализ логов
  - MS Profiler
- ПОЧЕМУ она возникает?
- КАК ее можно исправить?
- Не приведет ли исправление этой проблемы к появлению новых в других местах?

# Сканирование

Последовательное считывание всех строк таблицы в целях выполнения запроса.

- Сравнительно долго выполняется
- При многопользовательских запросах неэффективно
- Вызывает блокировки (Repeatable read – Level2)

# Индекс

Набор ссылок на места физического размещения строк в структуре БД, упорядоченный по возрастанию или по убыванию.

В SQL Server 2005 бывают:

- Некластерный индекс (Nonclustered index)
- Кластерный индекс (Clustered index)

# Особенности индекса

- Индекс создается для одной таблицы. Не может быть создан один индекс на две и более таблиц.
- Столбец или столбцы, по которым произведена индексация, называются индексированными.
- Индекс на основе 2х и более столбцов называется составным (composite index).
- Индекс хранится отдельно от таблицы и представляет собой значения индексированного столбца и указатели на соответствующие строки данных.
- Индексы служат для повышения скорости поиска и выборки данных.
- Индексу может быть присвоено ограничение (constraint) уникальности (UNIQUE).

# Некластерный индекс

**Некластерный индекс** представляет собой набор всех значений индексируемого столбца, упорядоченных по возрастанию или убыванию и указатель на исходную строку таблицы.

- Некластерный индекс полностью аналогичен предметному указателю.
- Некластерных индексов может быть несколько для одной таблицы.

# Указатель на строку

Указатель на строку – row locator или row identifier (RID) позволяет определить местоположение конкретной строки в БД.

RID состоит из:

- Идентификационный номер файла (ID file) – указывает на файл данных
- ИД номер страницы (ID Page) – Файл данных состоит из множества 8 КВ страниц
- Номер слота строки на странице (slot number). Каждая страница содержит строки только одной таблицы, она разбивается на слоты, каждый слот используется для хранения только одной строки.

# Применение

Некластерный индекс создается в следующих ситуациях:

- Столбец содержит большое количество уникальных (неповторяющихся) значений
- Запросы к этой таблице не должны возвращать большой набор данных
- Индексируемый столбец часто включается в разделы `where` или `having`
- Столбец должен быть редко изменяем. При изменении данных столбца сервер так же изменяет и индексы

# Кластерный индекс

**Кластерный индекс** осуществляет физическое перестроение порядка строк в таблице.

- Аналог с энциклопедией, в которой все термины расположены в алфавитном порядке.
- У каждой таблицы может быть только один кластерный индекс.
- Если у таблицы предполагается создать только один индекс, рекомендуется сделать его кластерным.

При создании кластерного индекса, происходит перенаправление адресации всех некластерных индексов (не на RID таблицы а на ID строки в кластерном индексе). Такие образом все некластерные индексы ссылаются на кластерный.

# Ограничение уникальности

Ограничение уникальности индекса предназначено для обеспечения уникальности значений соответствующего индекса.

Перед созданием уникального индекса нужно убедиться, что в таблице нет и не будет повторяющихся значений по этому индексу.

# Фактор заполнения

Физически индексы хранятся на 8КВ страницах, эти страницы называются индексными - index pages.

- Страницы разбиваются на слоты (каждый слот либо пуст, либо содержит элемент индекса)
- каждая страница предназначена для хранения данных только одного индекса
- страницы, содержащие данные одного индекса связаны между собой в виде списка (1я на 2ю, 2я на 3ю и тд)
- слоты на странице располагаются друг за другом

При добавлении значительного количества строк в таблицу происходит заполнение индексных страниц, если пустых слотов нет, то происходит операция расщепления (split) страницы.

- Выделение новой страницы
- Перенос на нее части элементов расщепляемой страницы
- Изменяются ссылки страниц друг на друга

# Фактор заполнения

Фактор заполнения – параметр, определяющий плотность записи данных на странице.

- Фактор заполнения позволяет контролировать число свободных слотов на странице.
- Фактор заполнения большой – индекс занимает мало места, операции вставки медленнее (DSS системы)
- Фактор заполнения маленький – индекс занимает много места, операции вставки быстрее (OLTP системы)

# Управление индексами

Перед созданием индексов нужно провести АНАЛИЗ производительности.

Индекс создается автоматически:

- При определении в таблице первичного ключа (кластерный индекс)
- При определении ограничения UNIQUE
- В команде CREATE TABLE можно явно указать создание индекса.

# Создание индексов

```
CREATE [UNIQUE] [CLUSTERED |  
NONCLUSTERED]  
INDEX index_name ON table (column[ASC|DESC]  
[,...n])  
[ WITH  
  [PAD_INDEX] ,  
  [FILLFACTOR = fillfactor],  
  [IGNORE_DUP_KEY] ,  
  [DROP_EXISTING]  
]
```

# Работа с индексами

- Перестроение:

```
DBCC DBREINDEX (  
    [database.owner.table_name],  
    index_name, fillfactor )
```

- Просмотр данных о фрагментации

```
DBCC SHOWCONTIG (table_id, index_id)
```

- Удаление

```
DROP INDEX 'table.index'
```

# Статистики

**Статистика** представляет собой информацию о распределении в таблице данных, упорядоченных с помощью индекса.

```
CREATE STATISTICS stat_name ON table (column, [...n])  
[ WITH  
[FULLSCAN | SAMPLE number PERCENT]  
[NORECOMPUTE]  
]
```

- Обновление статистики: UPDATE STATISTICS table
- Просмотр статистики: DBCC SHOW\_STATISTICS (table, stat\_name)