



**Microsoft** Partner  
Silver Learning



# TRANSACTION-SQL

## Indexes



ITVVDN  
IT VIDEO DEVELOPERS NETWORK

# TRANSACT-SQL

Автор курса



Станислав Зуйко



MC ID: 12974808

# TRANSACT-SQL

После урока обязательно



Повторите этот урок в видео формате на [ITVDN.com](http://itvdn.com)

Доступ можно получить через руководство вашего учебного центра



Проверьте как Вы усвоили данный материал на [TestProvider.com](http://TestProvider.com)

# TRANSACTION-SQL

Tema

Indexes

# Indexes

## План

1. Определение
2. Страницы и экстенды
3. Куча
4. Кластерный индекс
5. Некластерный индекс
6. Создание индексов
7. Рекомендации

# Index

## Индекс

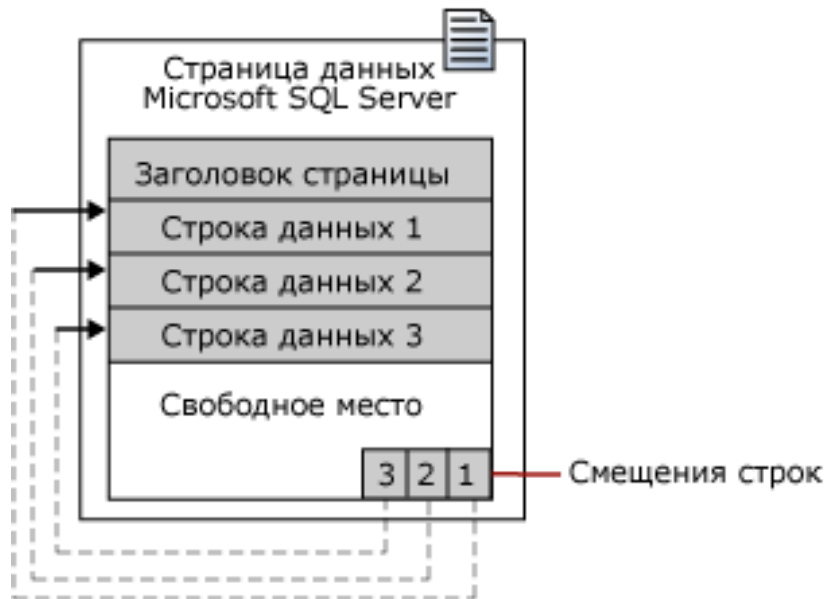
**Индекс** является структурой на диске, которая связана с таблицей (или представлением) и ускоряет получение строк из таблицы (или представления). Индекс содержит ключи, построенные из одного или нескольких столбцов в таблице (или представлении). Эти ключи хранятся в виде структуры сбалансированного дерева.

В тоже время индекс занимает дополнительное место в памяти и может иметь обратный эффект – не повышение, а понижение производительности.

# Организация хранения информации в SQL Server

## Страницы и экстененты

Основной единицей хранения данных в SQL Server является **страница**. Размер страницы составляет 8 КБ. Каждая страница начинается с 96-байтового заголовка, который используется для хранения системных данных о странице: номер страницы, тип страницы, объем свободного места на странице и идентификатор объекта, которому принадлежит страница.



### Типы страниц:

- Страницы данных (кроме больших типов данных типа `varchar(max)`, `varbinary(max)` и др.)
- Страницы индексов
- Страницы текста/изображения (для данных типа `varchar(max)`, `varbinary(max)` и др.)
- Index Allocation Map
- Прочие

# Организация хранения информации в SQL Server

## Страницы и экстененты

**Экстененты** являются основными единицами организации пространства. Экстент состоит из восьми непрерывных страниц или 64 КБ.



Каждая из восьми страниц в смешанном экстененте может находиться во владении разных объектов.

Однородные экстененты принадлежат одному объекту; все восемь страниц в кластере могут быть использованы только этим владеющим объектом.



# Heap

## Куча



Если у таблицы нет кластерного индекса, она является **кучей**. Записи не имеют определенного порядка хранения. Страницы не связаны в двунаправленный связанный список.

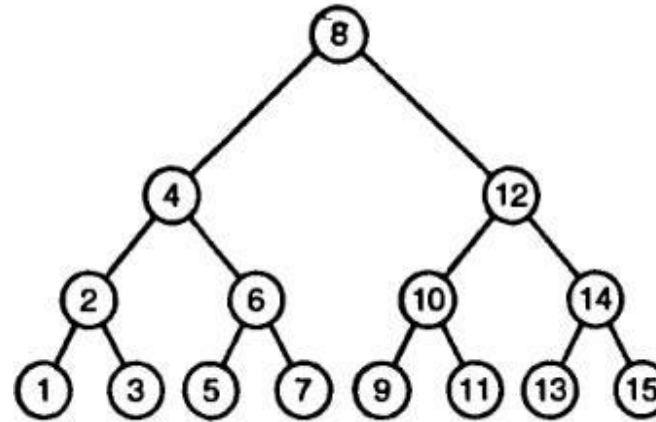
Для того, чтобы найти данные в куче, SQL Server использует **Index Allocation Map (IAM)** – это страница, которая содержит карту всех экстендов, которые содержат данные указанной таблицы.

SQL Server делает запрос к системной таблице sysindexes для нахождения FirstIAM страницы. Последовательность записей, получаемых конечным пользователем зависит от того, в каком порядке данные хранятся в IAM, а не от порядка вставки записей в таблицу.

При выполнении запроса к таблице, у которой нет индексов, выполняется **сканирование таблицы**. SQL Server не знает, что в таблице существует только одна запись, удовлетворяющая условию и будет просматривать все экстенды (которые содержат данные указанной таблицы), все страницы, все записи.

# B-tree

## Сбалансированное дерево



Индекс состоит из набора страниц, узлов индекса, которые организованы в виде структуры сбалансированного дерева.

Корневой узел является «точкой входа» в индекс, состоит только из одной страницы и содержит указатели на ключи следующих уровней индекса

Листья индекса могут содержать как сами данные таблицы (кластерный индекс), так и просто указатель на строки с данными в таблице (некластерный индекс).

# CLUSTERED INDEX

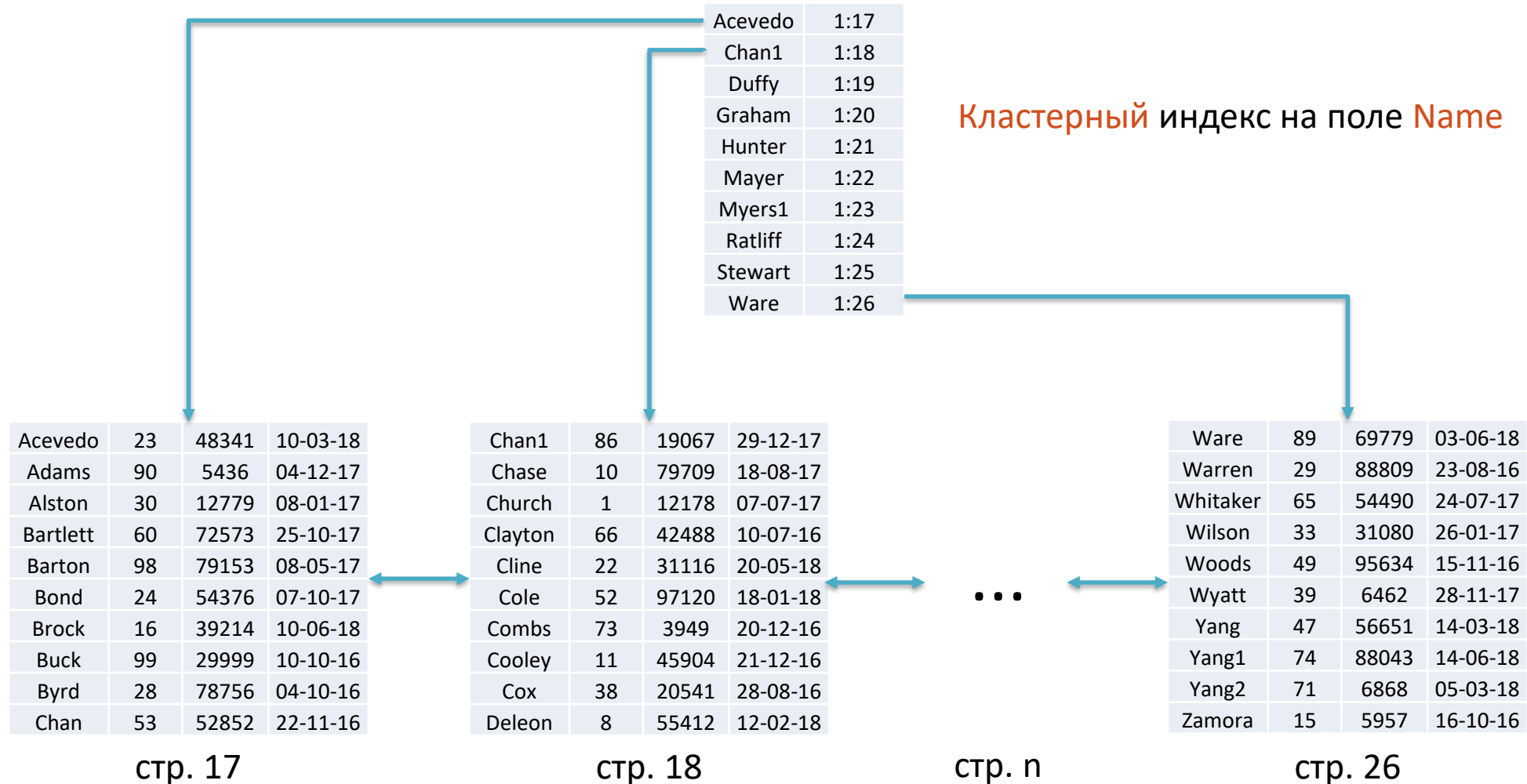
## Кластерный индекс

Кластерный индекс хранит реальные строки данных в листьях индекса. Важной характеристикой кластерного индекса является то, что все значения отсортированы в определенном порядке (либо возрастания, либо убывания). Таким образом, таблица (или представление) может иметь только один кластерный индекс.

В SQL Server кластерный индекс является уникальным индексом по определению. Если существуют записи с одинаковыми значениями, SQL Server делает их уникальными, добавляя номера из внутреннего (невидимого снаружи) счетчика.

# CLUSTERED INDEX

## Кластерный индекс



# NONCLUSTERED INDEX

## Некластерный индекс

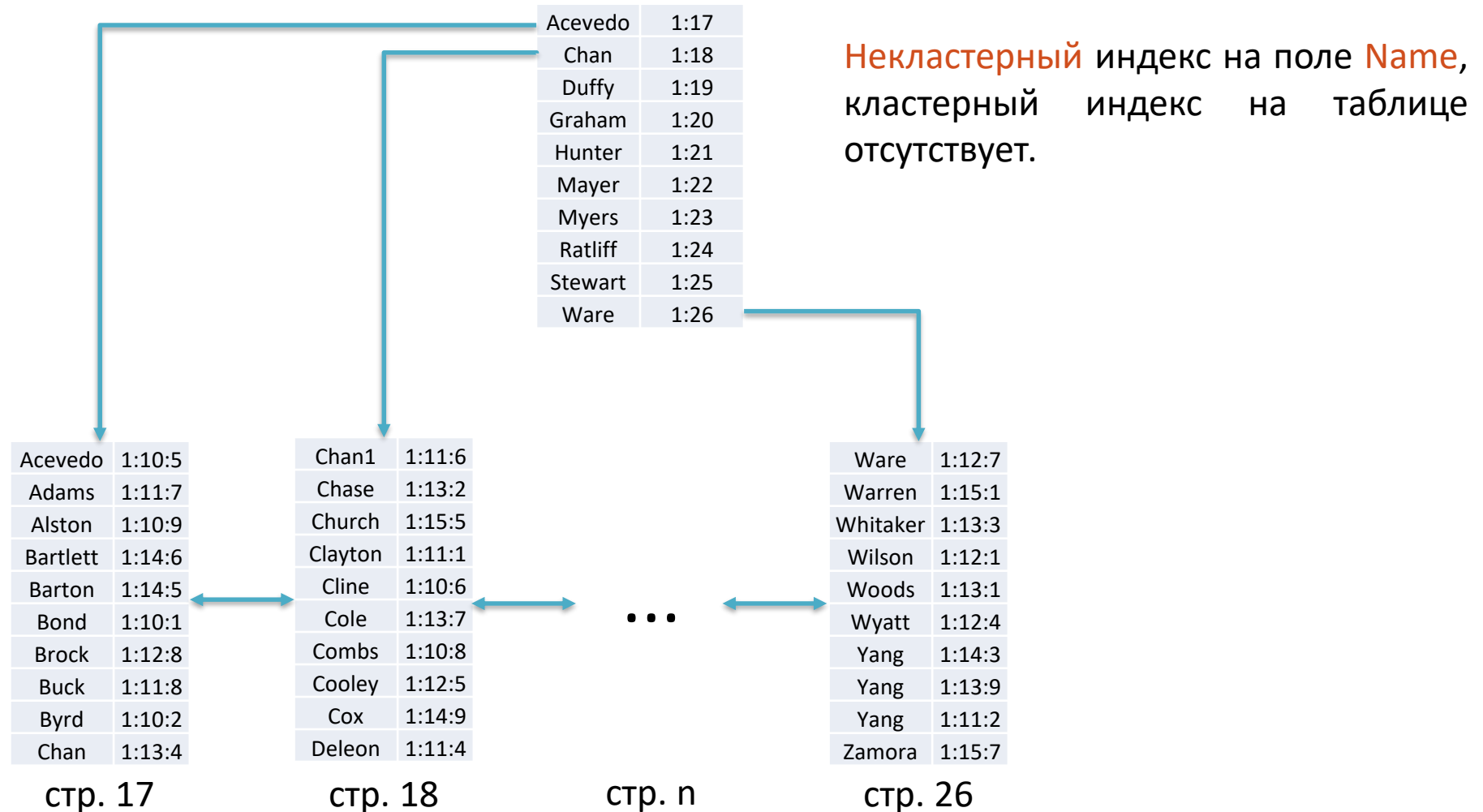
Некластерный индекс на листовом уровне содержит отсортированные значения индексированного столбца и row locator – указатель на остальные данные. Если у таблицы есть кластерный индекс (кластерная таблица), то row locator'ом является ключ кластерного индекса, если кластерного индекса нет (таблица-куча), то row locator – это row ID (указатель на определенную строку данных в определенной таблице).

Необходимо создавать кластерные ключи как можно более короткими. Каждый некластерный индекс будет использовать значения кластерного индекса (если последний определен). Следовательно увеличение размера кластерного индекса приводит к увеличению занимаемой памяти для всех не кластерных индексов.

Необходимо создавать кластерный индекс первым, так как если уже будут некластерные индексы, то всем им нужно будет перестроиться – на листовом уровне вместо row ID указателем на остальные данные должен стать ключ кластерного индекса.

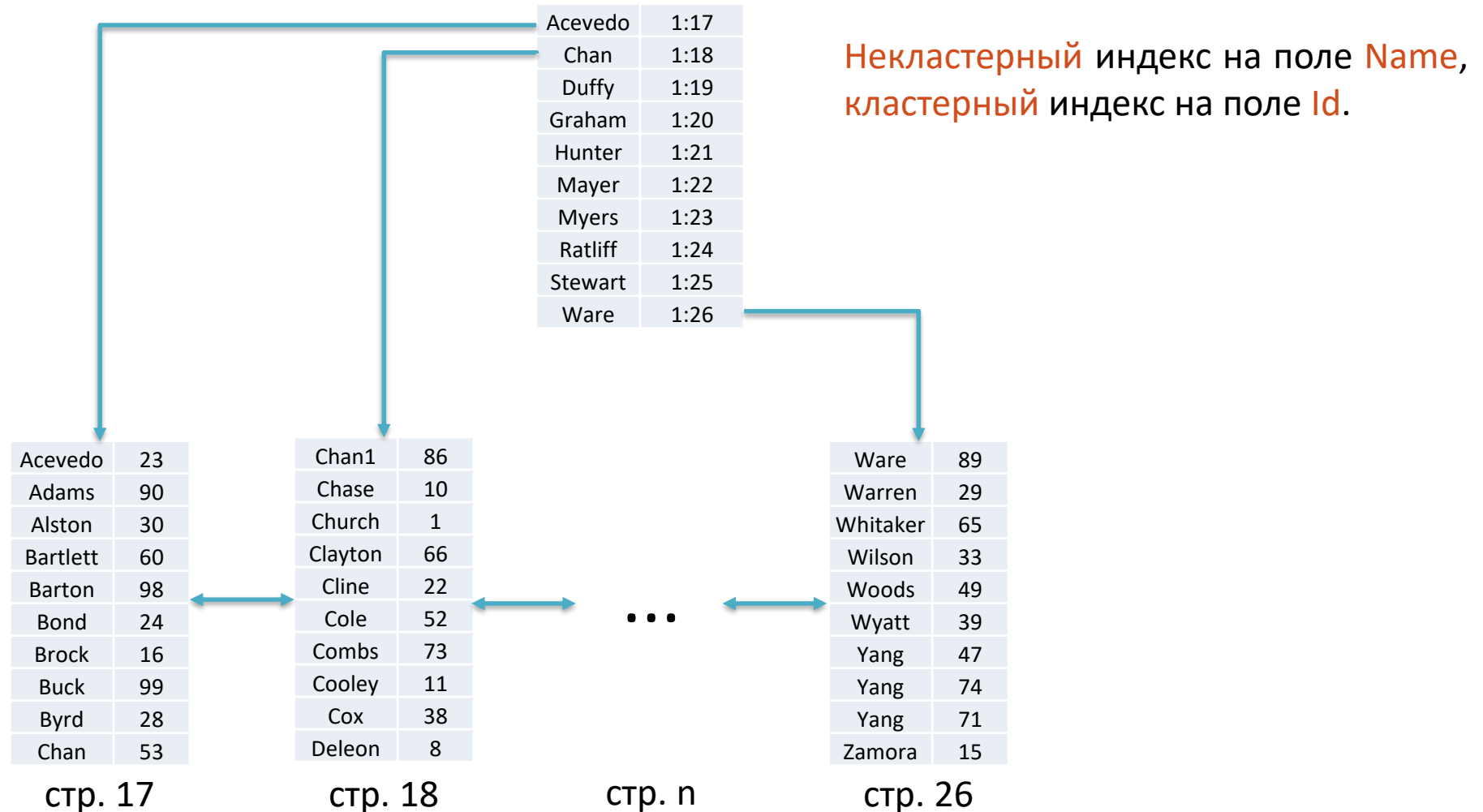
# NONCLUSTERED INDEX

## Некластерный индекс



# NONCLUSTERED INDEX

## Некластерный индекс



# CREATE INDEX

## Создание индексов

1. PRIMARY KEY по умолчанию создает уникальный кластерный индекс, но можно создать первичный ключ, который задаст некластерный индекс. Если у таблицы уже есть кластерный индекс, то создастся некластерный индекс.

```
CREATE TABLE Persons (  
    Id INTEGER PRIMARY KEY,  
    Name VARCHAR(255)  
);
```

```
CREATE TABLE Persons (  
    Id INTEGER PRIMARY KEY NONCLUSTERED,  
    Name VARCHAR(255)  
);
```



# CREATE INDEX

## Создание индексов

2. Ограничение UNIQUE по умолчанию создает уникальный некластерный индекс, но можно создать UNIQUE, который задаст кластерный индекс.

```
CREATE TABLE Persons (  
    Id INTEGER,  
    Name VARCHAR(255) UNIQUE  
);
```

```
CREATE TABLE Persons (  
    Id INTEGER,  
    Name VARCHAR(255) UNIQUE CLUSTERED  
);
```

# CREATE INDEX

## Создание индексов

3. Создавая индексы не через PRIMARY KEY или UNIQUE, можно создать не уникальный индекс (кластерный или некластерный). По умолчанию создается не уникальный некластерный индекс.

```
CREATE INDEX index_name  
ON table_name (column_name)
```

```
CREATE UNIQUE INDEX index_name  
ON table_name (column_name)
```

```
CREATE CLUSTERED INDEX index_name  
ON table_name (column_name)
```

```
CREATE UNIQUE CLUSTERED INDEX index_name  
ON table_name (column_name)
```

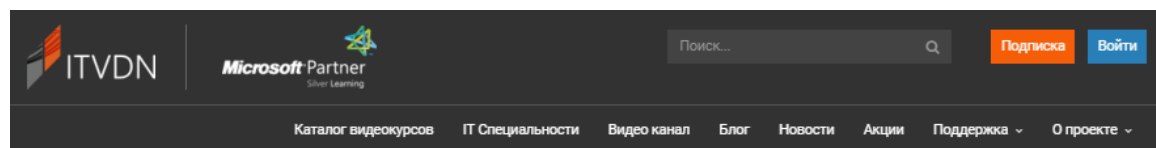
# CREATE INDEX

## Рекомендации

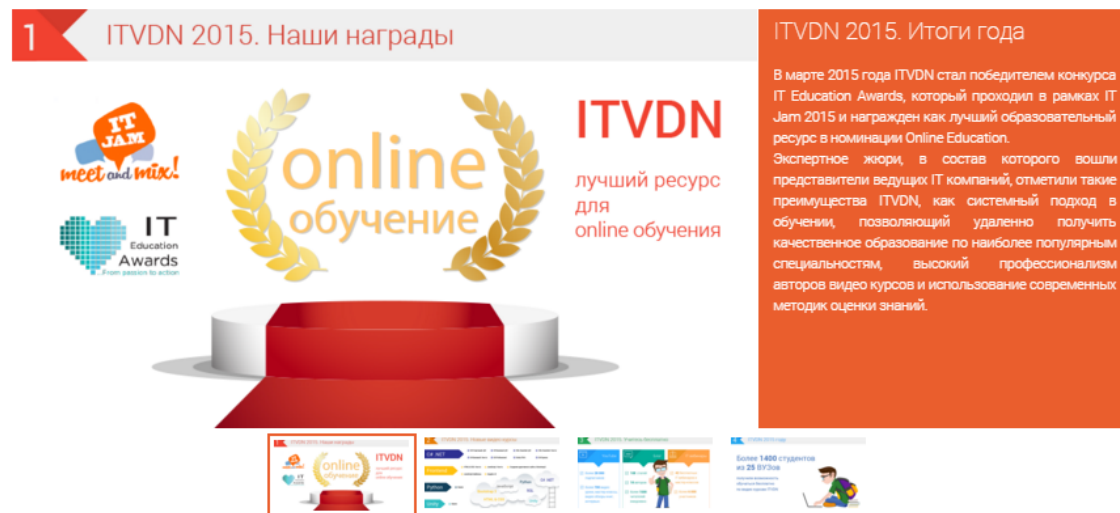
1. Для кластерного индекса наиболее подходящим столбцом будет уникальное, короткое, не поддерживающее NULL значений, которое не будет обновляться. Вот почему первичный ключ часто используется как кластерный индекс.
2. Создавайте индексы на столбцы, по которым происходит поиск, сортировка, группировка, соединение таблиц.
3. Не создавайте индексы на полях, которые редко используются в запросах.
4. Не создавайте индексы на полях, которые содержат несколько уникальных значений, например, столбец, содержащий только значения мужской или женский пол. Чем больше дубликатов в столбце, тем хуже работает индекс. Чем больше уникальных значения, тем выше работоспособность индекса. Когда возможно используйте уникальный индекс.
5. Для небольших таблиц поиск по индексу может занять больше времени, чем простое сканирование всех строк.
6. Для таблиц которые часто обновляются используйте как можно меньше индексов.

# Смотрите наши уроки в видео формате

ITVDN.com



Посмотрите этот урок в видео формате на образовательном портале [ITVDN.com](http://itvdn.com) для закрепления пройденного материала.



Все курсы записаны сертифицированными тренерами, которые работают в учебном центре CyberBionic Systematics

## Новые видео

Исключения	0
Итераторы и генераторы	0

## Популярные видео курсы

Видео курс C# Стартовый (для начинающих)	9 уроков (16 ч. 3 мин.)
Видео курс по шаблонам проектирования	29 уроков (16 ч. 7 мин.)

## Теги

.NET Developer
Frontend Developer



# Проверка знаний

## TestProvider.com

TestProvider | Мы помогаем людям оценить себя

Регистрация Войти

Главная Каталог Сертификация Microsoft Поддержка О нас

### Тестирование

Языки программирования и информационные технологии

**Microsoft**

C# ASP.NET MVC JavaScript Patterns Of Design SQL Architecture Guide WCF HTML&CSS XML SEO WPF HTML5&CSS3 JQuery XNA SharePoint GUI for Android Windows Azure Platform Microsoft Patterns&Practices TFS SCRUM ReSharper TDD WWF LINQ Entity Framework Windows Forms Refactoring Microsoft Expression Blend 4 Windows Phone 8 Windows 8 AppStore Visual Studio Tips&Tricks MSF MEF SilverLight AJAX MEF Service Oriented Architecture

Пройти тест

Наши партнеры

Microsoft Partner CyberBionic ITVDN PROMETRIC TEST CENTER PEARSON VUE Authorized Test Center Windows Azure Cloud Partner EBA

Дополнительные ресурсы:

Очное обучение On-line обучение Видео обучение

TestProvider – это online сервис проверки знаний по информационным технологиям. С его помощью Вы можете оценить Ваш уровень и выявить слабые места. Он будет полезен как в процессе изучения технологии, так и общей оценки знаний IT специалиста.

После каждого урока проходите тестирование для проверки знаний на [TestProvider.com](https://testprovider.com)

Успешное прохождение финального тестирования позволит Вам получить соответствующий Сертификат.



# TRANSACTION-SQL

Q&A

# Информационный видеосервис для разработчиков программного обеспечения

