

Professional Association for SQL Server



**Все, что Вы хотели узнать о
секционировании данных, но боялись
спросить**

Короткевич Дмитрий
Тампа, FL
MVP, MCM, MCPD

О Докладчике



- 15+ лет в индустрии
- 11+ лет работы с SQL Server
- SQL Server MVP
- Microsoft Certified Master: SQL Server 2008
- MCPD – Enterprise Application Developer



- Blog: <http://aboutsqlserver.com>
 - Презентация и скрипты будут доступны для скачивания
- Email: dmitri@aboutsqlserver.com



Содержание

- Причины для секционирования данных
- Способы секционирования данных
- Практические вопросы
 - Перенос данных между дисковыми массивами без блокировки таблиц
 - Секционирование и стратегии высокой доступности

Для чего мы секционируем данные?

Все данные
хранятся в
одном месте

Резервное
копирование

Идентичные
схема, индексы
и компрессия

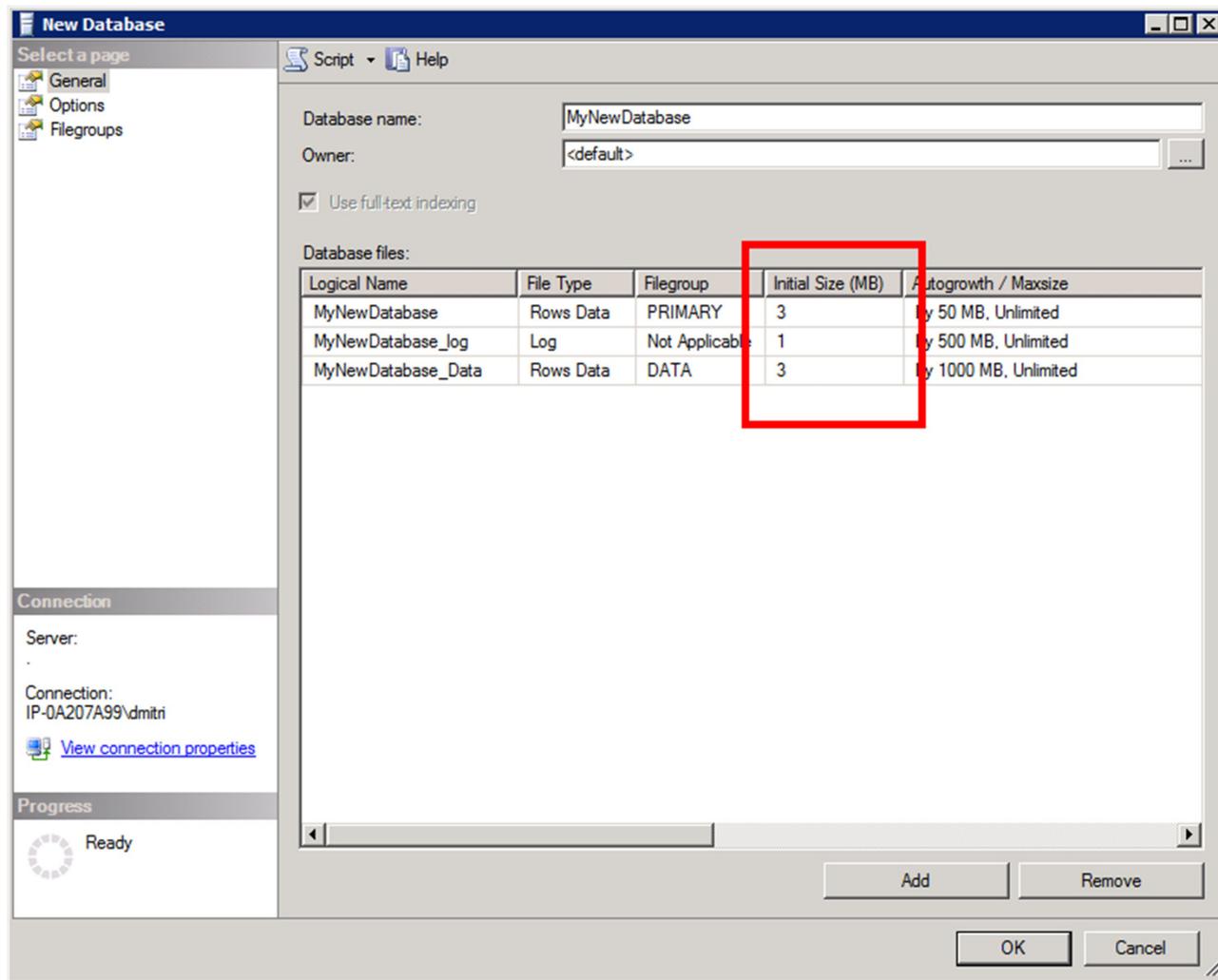
Большая
таблица с
данными

Долгое
восстановление
после сбоев

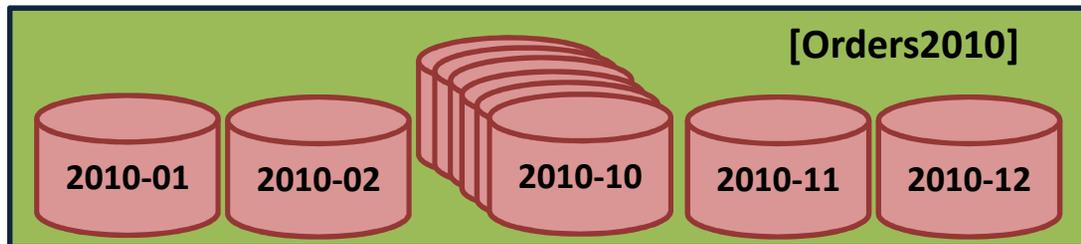
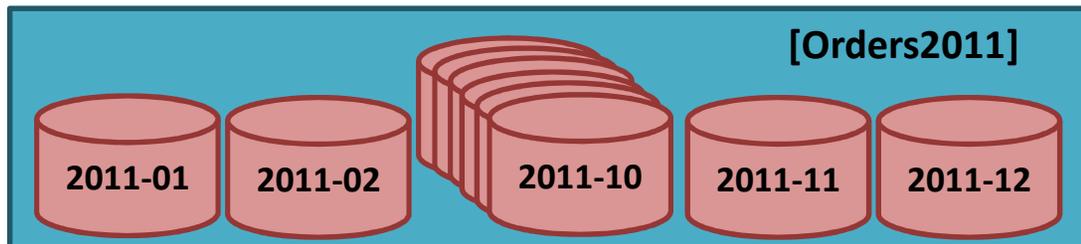
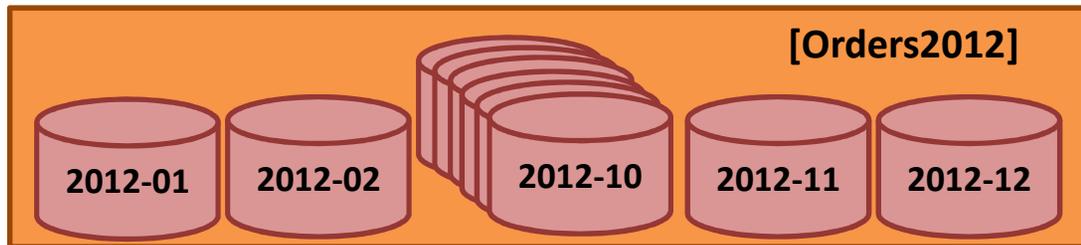
Проблемы с
поддержкой
индексов

Проблемы со
статистикой

Когда секционировать?



Секционированные таблицы

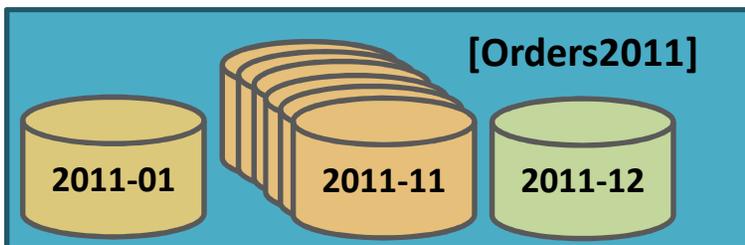
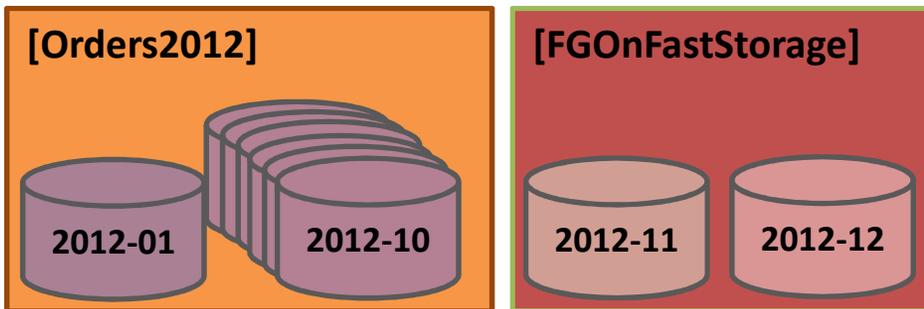


```
CREATE PARTITION FUNCTION pfOrders(DATETIME)
AS RANGE RIGHT
FOR VALUES('2010-01-01', '2010-02-01',
/*...*/ '2011-01-01', '2011-02-01',
/*...*/ '2012-11-01', '2012-12-12');

CREATE PARTITION SCHEME psOrders
AS PARTITION pfOrders
TO ([Orders2010], [Orders2010],
/*...*/ [Orders2011], [Orders2011],
/*...*/ [Orders2012], [Orders2012]);

CREATE TABLE dbo.Orders
(
    OrderID INT not null,
    OrderDate DATETIME not null,
    /*..*/
    CONSTRAINT PK_Orders
    PRIMARY KEY CLUSTERED
    (OrderDate, OrderId)
    ON psOrders(OrderDate)
);
```

Секционированные представления



```
CREATE TABLE dbo.Orders2010_01
(
    OrderID INT not null,
    OrderDate DATETIME not null,
    /*...*/
    CONSTRAINT PK_Orders2010_01
    PRIMARY KEY CLUSTERED (OrderDate, OrderId),
    CONSTRAINT CHK_Orders2010_01 CHECK
    (OrderDate >= '2010-01-01' and OrderDate < '2010-02-01')
) ON [Orders2010];
/*...*/
CREATE TABLE dbo.Orders2012_12
(
    OrderID INT not null,
    OrderDate DATETIME not null,
    /*...*/
    CONSTRAINT PK_Orders2010_01
    PRIMARY KEY CLUSTERED (OrderDate, OrderId),
    CONSTRAINT CHK_Orders2010_01 CHECK
    (OrderDate >= '2012-12-01' and OrderDate < '2013-10-01')
) ON [FGOnFastStorage]
go

CREATE VIEW dbo.Orders(OrderId, OrderDate /*...*/)
WITH SCHEMABINDING
AS
    SELECT OrderId, OrderDate /*...*/ from dbo.Orders2010_01
    UNION ALL
    SELECT OrderId, OrderDate /*...*/ from dbo.Orders2012_12;
```

Сравнение подходов

Таблицы

- Поддерживаются только в SQL Server Enterprise и Developer Edition
- Максимально 1,000/ 15,000 секций
- Нет возможности online переноса секций между файловыми группами
- Общая статистика на всю таблицу
- Одинаковая схема и индексы между секциями
- Нет поддержки online пересоздания индекса (rebuild) в пределах секции
- Поддержка репликации

Представления

- Поддерживаются во всех версиях SQL Server
- Максимально 255 таблиц/секций
- Секции (таблицы) могут быть перенесены между файловыми группами online (Enterprise Edition)
- Статистика на уровне таблиц/секций
- Схема и индексы могут отличаться в рамках секций (таблиц)
- Online пересоздание индекса в пределах секции (Enterprise Edition) (* зависит от версии)
- Проблемы с поддержкой репликации

Многоуровневое хранение данных



Источник: кинофильм «Кавказская пленница или новые приключения Шурика» (с) Мосфильм

Многоуровневое хранение данных

Оперативные данные
(2013 год)

OLTP транзакции <50ms
Доступность 99.99% времени
RAID-10 с тройным зеркалированием

2012 год

Доступность 99.5% времени
RAID-5

2011год

Доступность 98% времени
RAID-1 или RAID-0

2010год

Professional Association for SQL Server



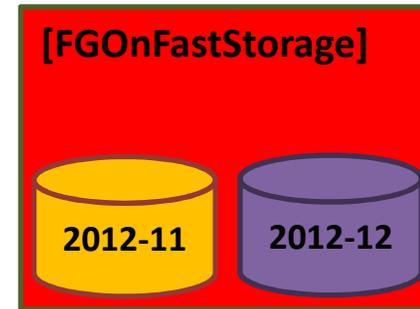
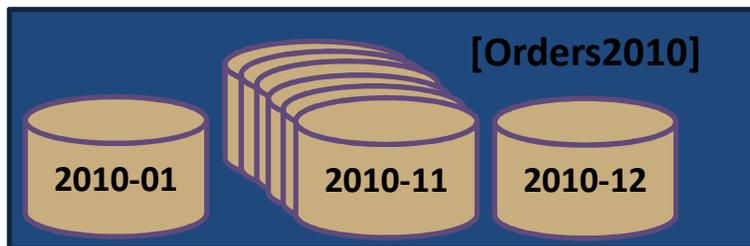
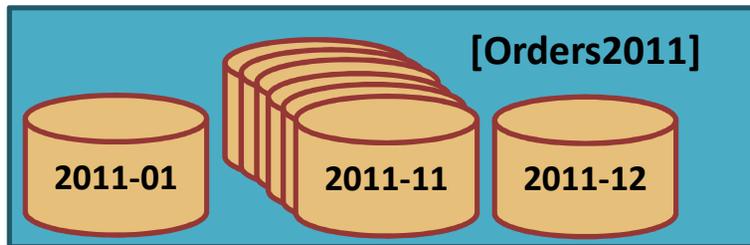
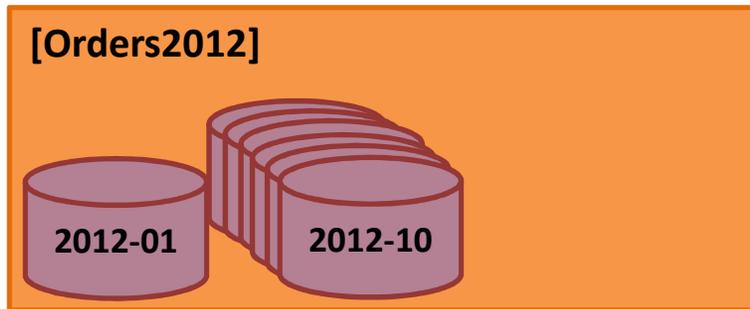
Демонстрация

Перенос данных между различными файловыми группами и дисковыми массивами

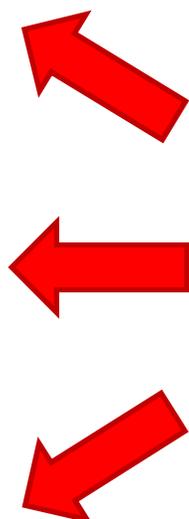
Перенос данных без блокировки таблиц

Версия и редакция SQL Server	Перенос секций между файловыми группами	Перенос таблицы с LOB столбцами между файловыми группами	Перенос таблицы без LOB столбцов между файловыми группами	Перенос данных между дисковыми массивами в пределах одной файловой группы
SQL Server 2012 Enterprise Edition	В общем случае не поддерживается Используется блокировка модификации схемы (SCH-M) на время переноса данных	Поддерживается	Поддерживается	Поддерживается во всех редакциях
SQL Server 2005 - 2008R2 Enterprise Edition		Не поддерживается	Поддерживается	
Non-Enterprise Editions	Неприменимо	Не поддерживается	Не поддерживается	

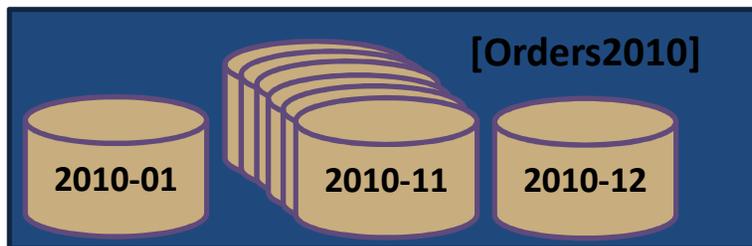
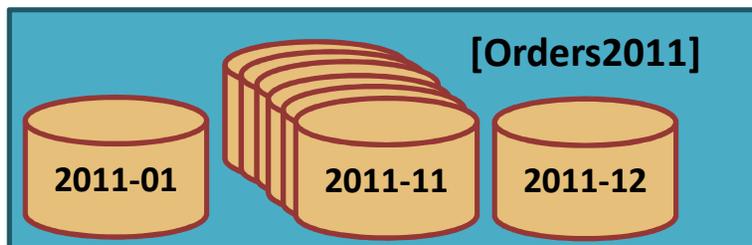
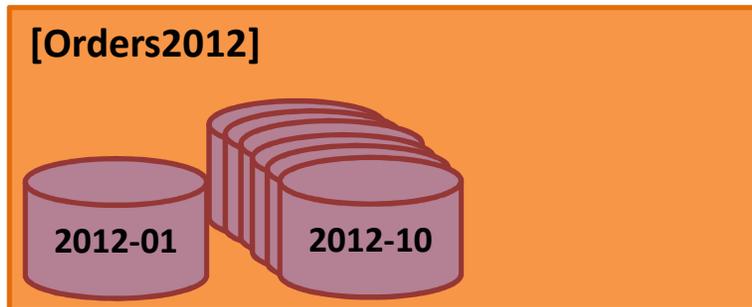
Смешанный подход



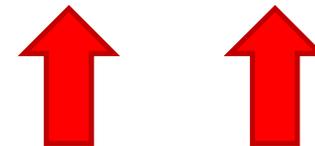
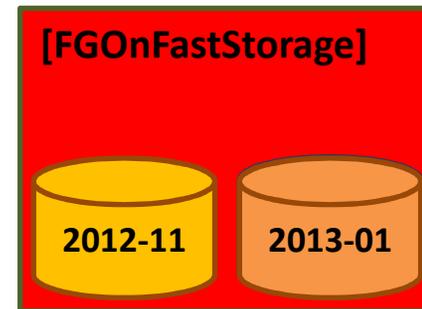
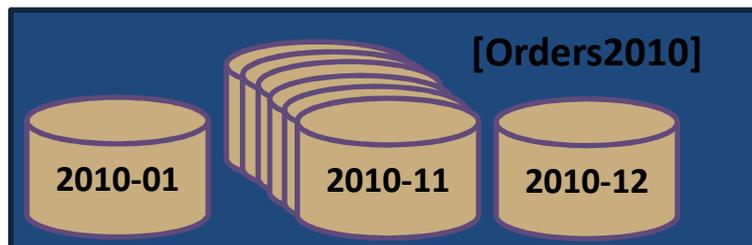
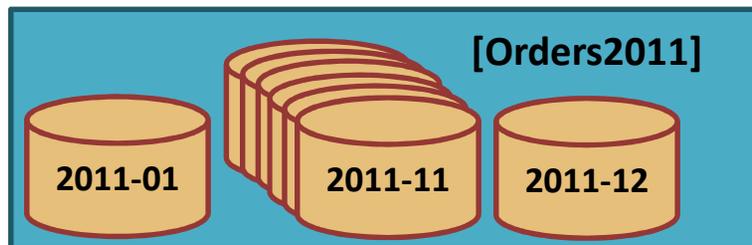
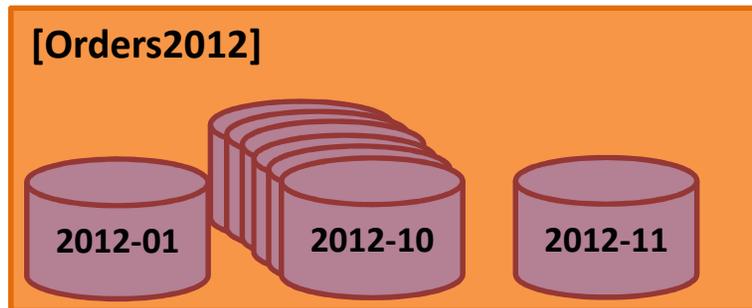
Секционированные
таблицы



Смешанный подход



Смешанный подход



**Секционированное
представление
(union all)**

Стратегии высокой доступности

- Динамическое создание файлов и файловых групп
 - Failover cluster – одна копия БД
 - AlwaysOn, Mirroring, Log Shipping – требуют одинаковой структуры дисков и папок

Professional Association for SQL Server



Демонстрация

Секционирование и стратегии высокой доступности

Секционирование и файловые группы

- Что следует принять во внимание:
 - Стратегию восстановления после сбоев (Disaster Recovery)
 - Ограничения в дисковой системе (объем, цена, требования)
- Связанные объекты
 - Например: *Orders* и *OrderItems*
- Разделение между модифицируемыми и статичными (read-only) объектами
 - Файловая группа со статичными объектами может быть помечена как Read Only

Потенциальные проблемы

- Увеличение объема данных
 - Столбец секционирования присутствует в кластерном и некластерном индексах
 - Пример: Datetime (8 bytes) * 10M записей в день = ~28GB на NCI в год
- Изменение планов выполнения запросов
 - Неоптимальные планы в некоторых запросах
- Более подробно:
 - <http://www.simple-talk.com/content/article.aspx?article=1587>
 - <http://aboutsqlserver.com/publications>

Мы обсудили

- Причины для секционирования данных
- Способы секционирования данных
- Практические вопросы переноса данных между дисковыми массивами и файловыми группами

