Professional Association for SQL Server

24 HOURS OF PASS

Data Partitioning in VLDB

Dmitri Korotkevitch
Tampa, FL
MVP, MCM, MCPD

About me

- 15+ years in IT
- 11+ years working with SQL Server
- SQL Server MVP
- Microsoft Certified Master: SQL Server 2008
- MCPD – Enterprise Application Developer

MVP
Microsoft®
Most Valuable
Professional

- Blog: http://aboutsqlserver.com
  - Presentation deck would be available for download
- Email: dmitri@aboutsqlserver.com

Microsoft®
CERTIFIED
Master

Agenda

- Why we want to partition data
- How we can partition data in SQL Server
- Practical questions
    - Implementing tiered storage
    - Partitioning and HA strategies

Why we need to partition data

Everything is stored in one place

Same indexes, schema and compression

Index Maintenance overhead

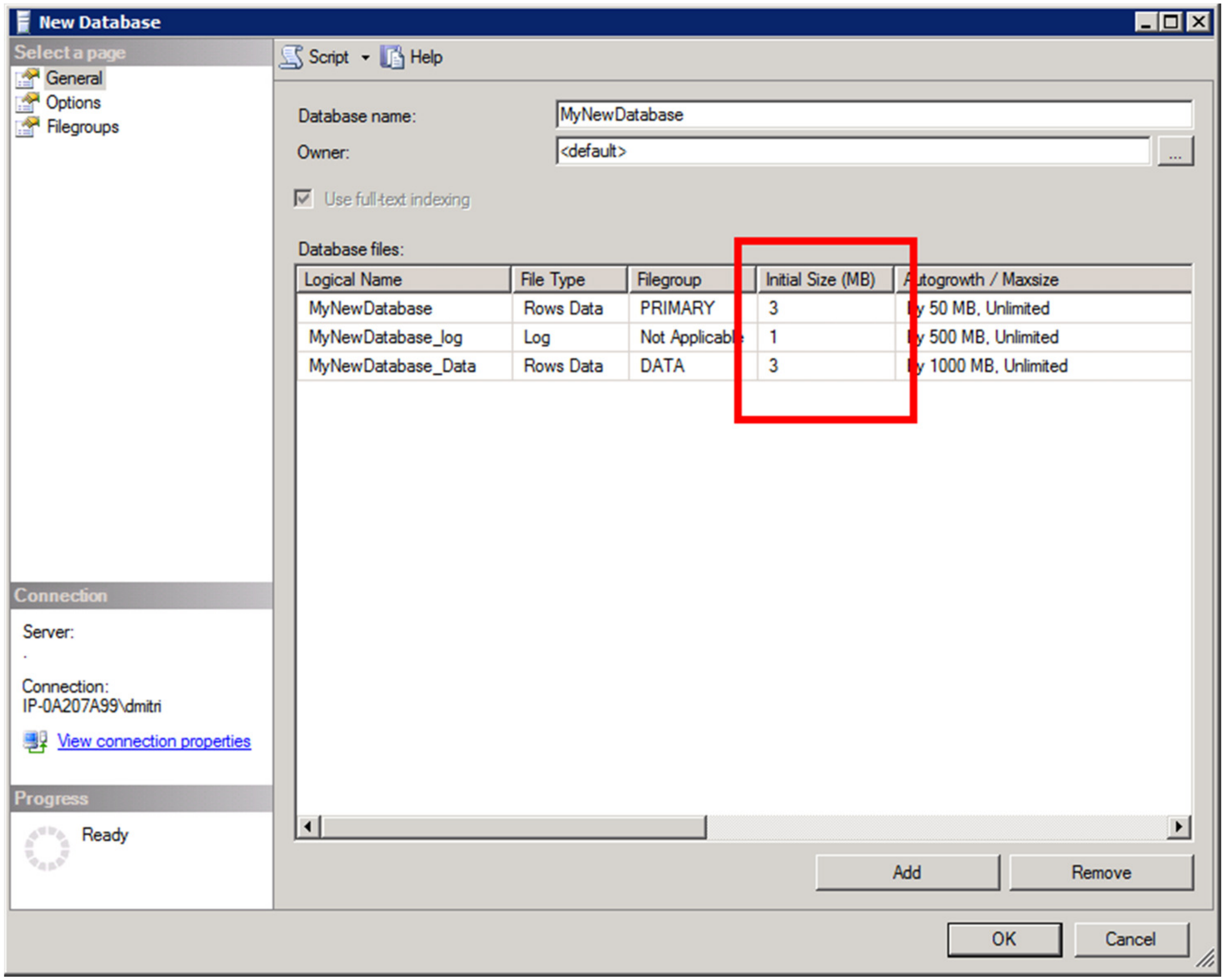
One large table with the data

Backup overhead

Extremely long Disaster Recovery

Inaccurate Statistics

When should I partition data?



Database name: MyNewDatabase

Owner: <default>

☑ Use full-text indexing

Database files:

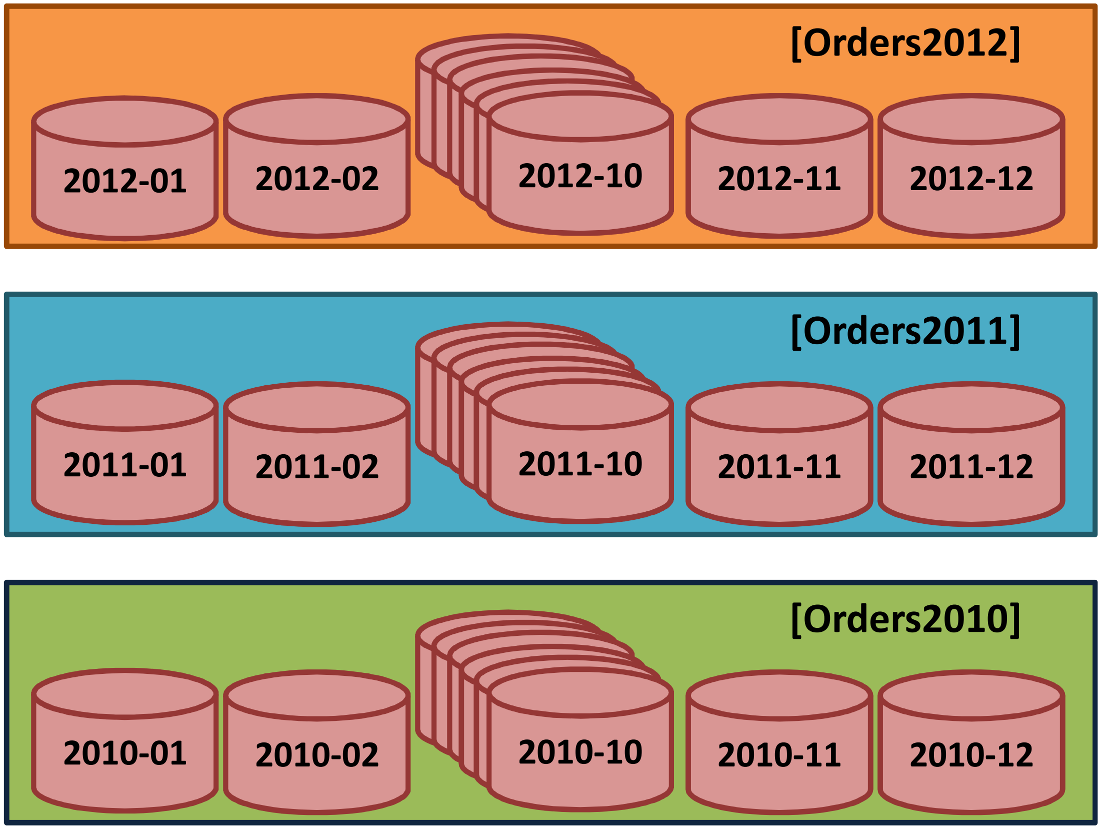
Logical Name	File Type	Filegroup	Initial Size (MB)	Autogrowth / Maxsize
MyNewDatabase	Rows Data	PRIMARY	3	By 50 MB, Unlimited
MyNewDatabase_log	Log	Not Applicabl	1	By 500 MB, Unlimited
MyNewDatabase_Data	Rows Data	DATA	3	By 1000 MB, Unlimited

Server:
.

Connection:
IP-0A207A99\dmitri

Ready

Partitioned Tables



[Orders2012]

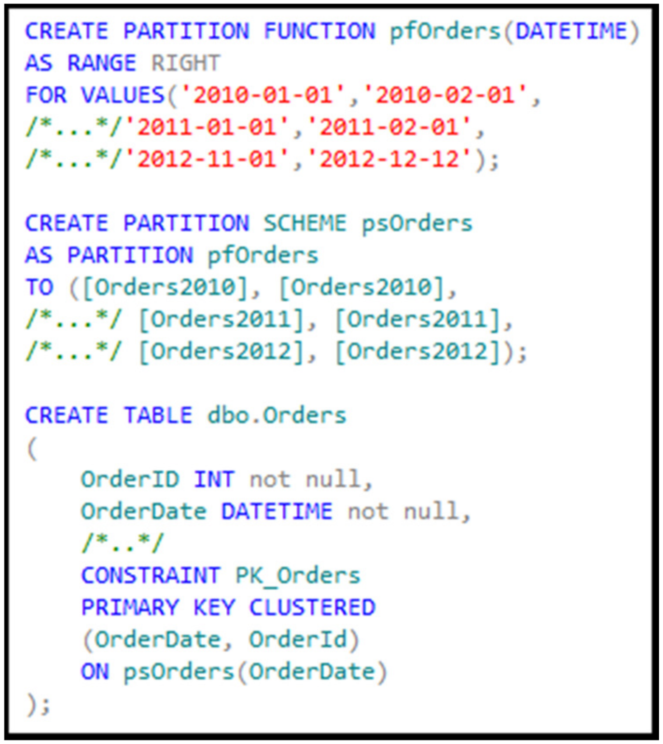
2012-01 2012-02 2012-10 2012-11 2012-12

[Orders2011]

2011-01 2011-02 2011-10 2011-11 2011-12

[Orders2010]

2010-01 2010-02 2010-10 2010-11 2010-12

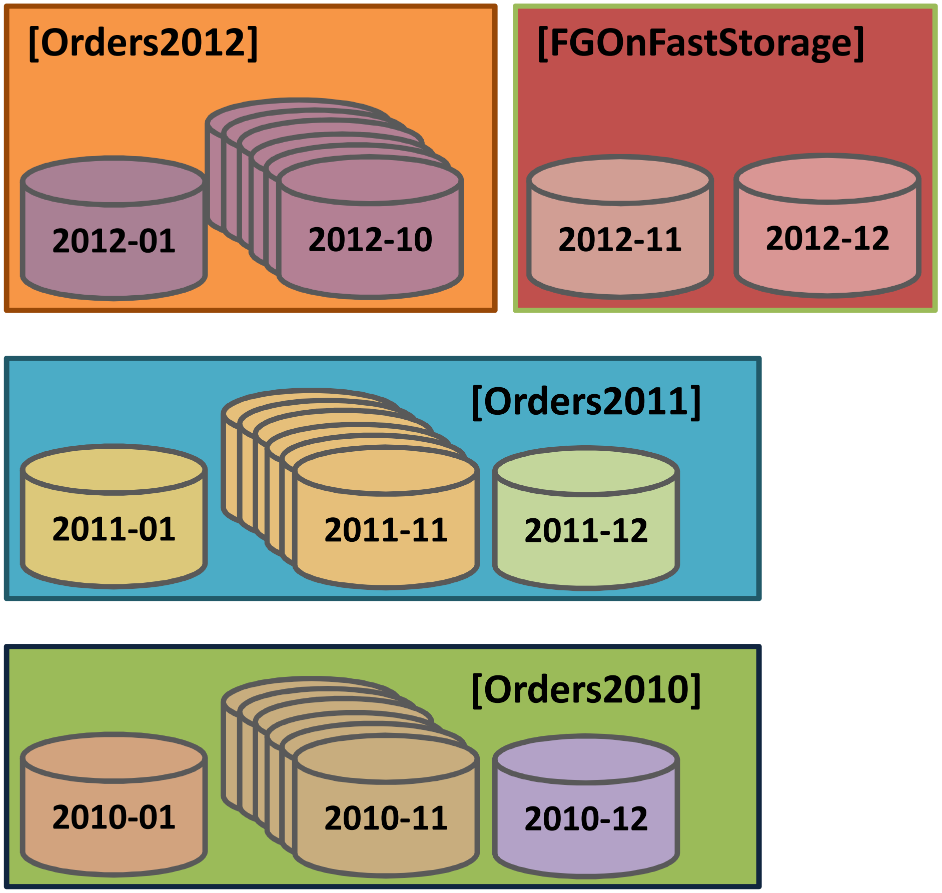


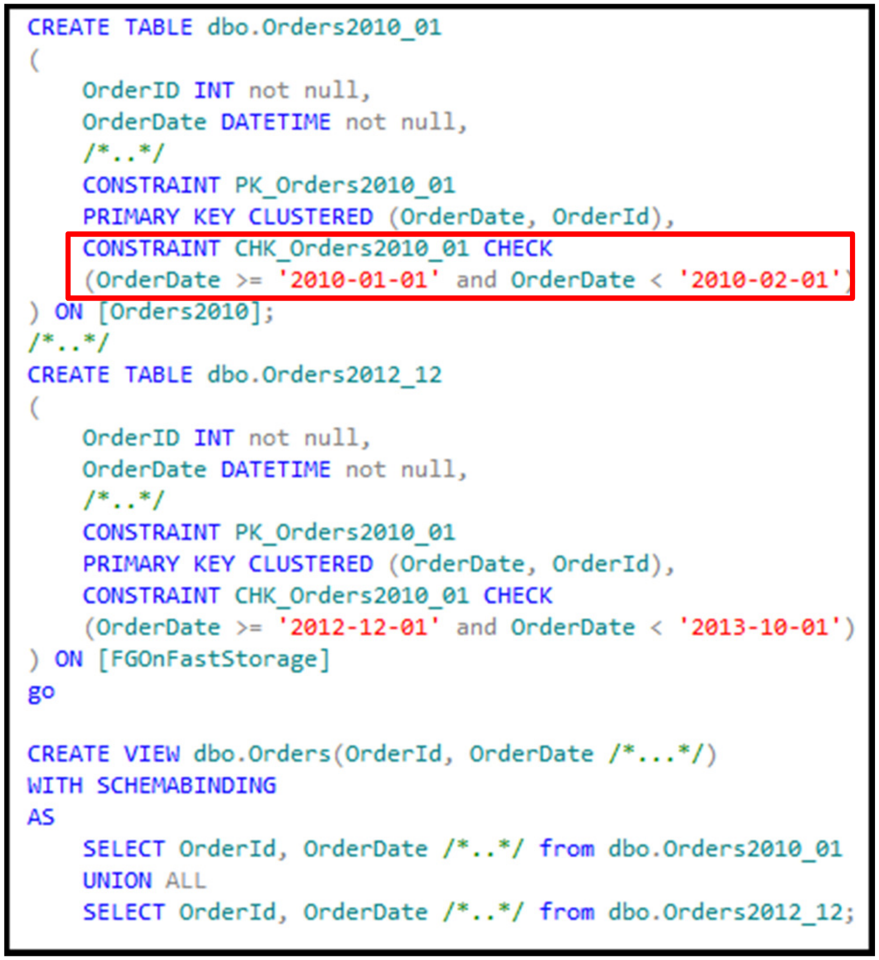
```
CREATE PARTITION FUNCTION pfOrders(DATETIME)
AS RANGE RIGHT
FOR VALUES('2010-01-01','2010-02-01',
/*...*/'2011-01-01','2011-02-01',
/*...*/'2012-11-01','2012-12-12');

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)
);
```

Partitioned Views





```
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;
```

Comparing approaches

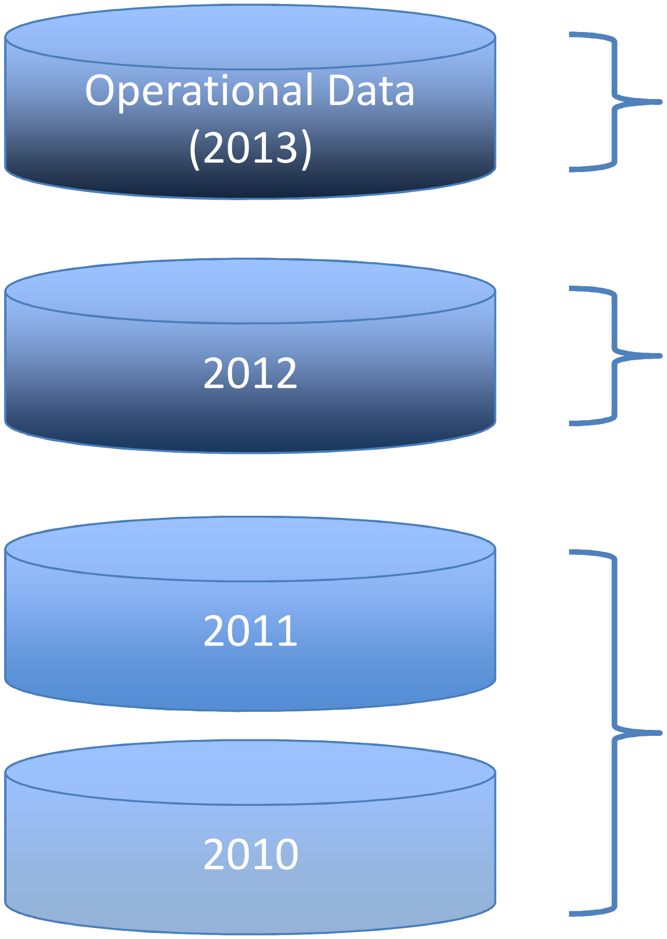
Partitioned Tables

- Enterprise and Developer editions only
- 1,000/ 15,000 partitions max
- *Almost* impossible to move data between filegroups keeping table online
- Statistics on the table level
- Same schema and indexes
- Cannot rebuild index online within the single partition
- Replication friendly

Partitioned Views

- All editions of SQL Server
- 255 tables/partitions max
- Data can be moved between filegroups online (Enterprise Edition)
- Separate statistics per table
- Different schema and indexes per table
- Online index rebuild within the table (Enterprise Edition) (* it depends)
- Not so “replication friendly”

Tiered Storage implementation



Operational Data
(2013)

OLTP transaction latency <50ms

99.99% availability

RAID-10 with triple mirroring

2012

99.5% availability

RAID-5

2011

2010

98% availability

RAID-1 or RAID-0

Professional Association for SQL Server

24 HOURS OF PASS

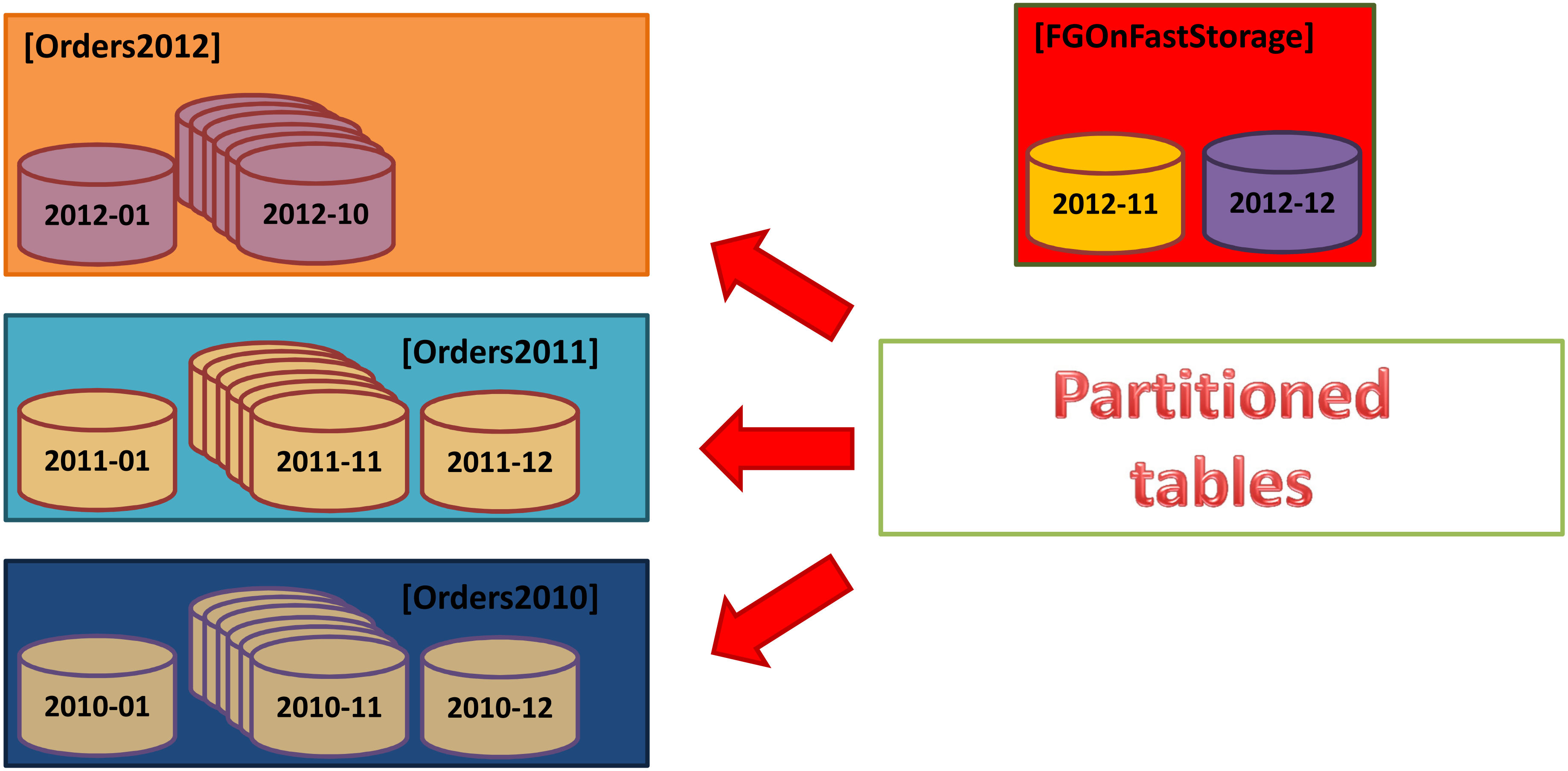
Demo

Moving data between different filegroups and disk arrays

Moving data online

SQL Server version and edition	Moving partition to different Filegroup	Moving table with LOB columns to different Filegroup	Moving table without LOB columns to different Filegroup	Moving data to different disk array within the same Filegroup
SQL Server 2012 Enterprise Edition	Not supported SQL Server acquires schema modification lock (SCH-M) during data movement	Supported	Supported	Supported in every edition
SQL Server 2005 - 2008R2 Enterprise Edition		Not supported	Supported	
Non-Enterprise Editions	N/A	Not supported	Not supported	

Mixed approach



[Orders2012]

2012-01

2012-10

[FGOnFastStorage]

2012-11

2012-12

[Orders2011]

2011-01

2011-11

2011-12

Partitioned tables

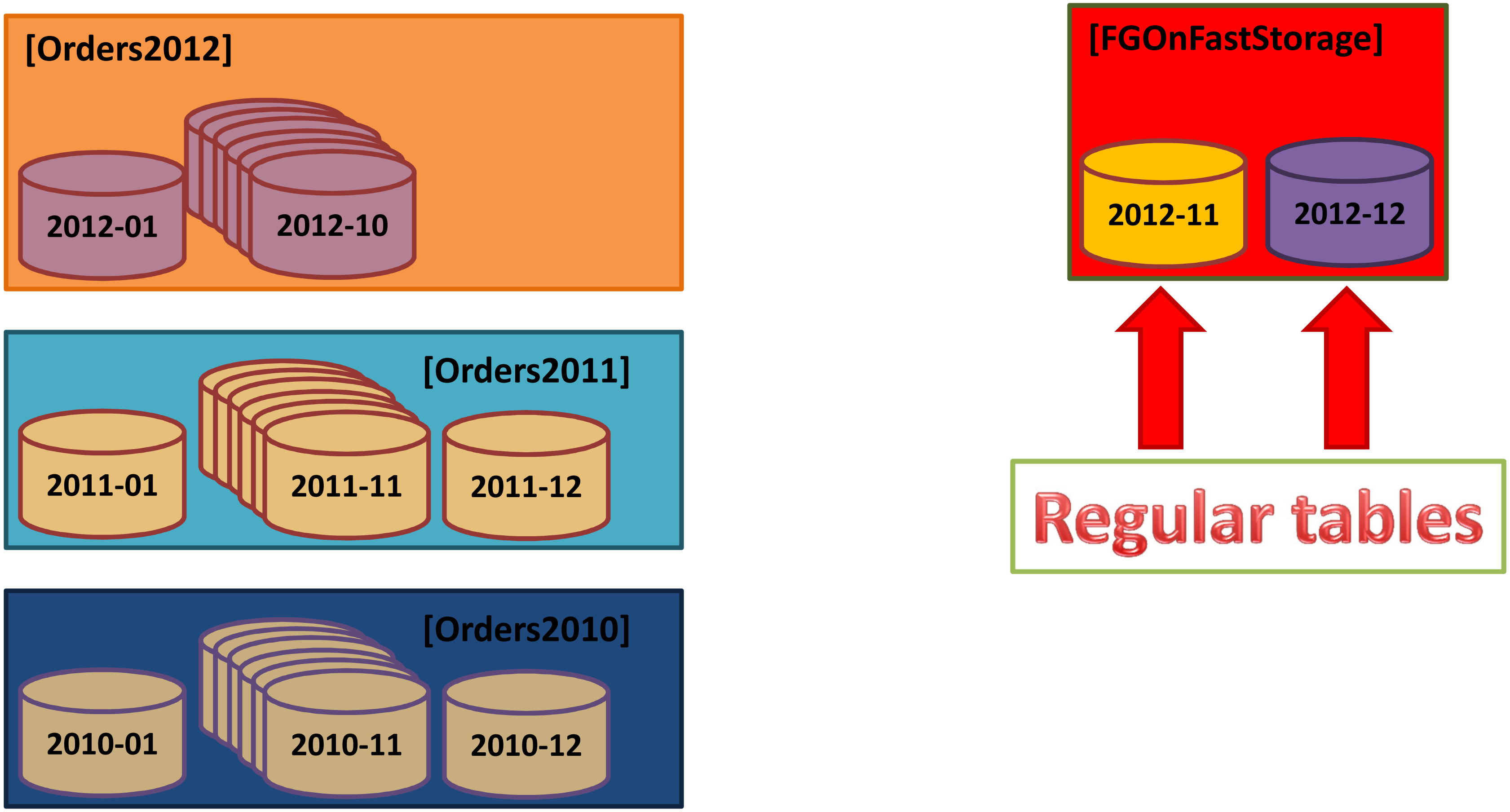
[Orders2010]

2010-01

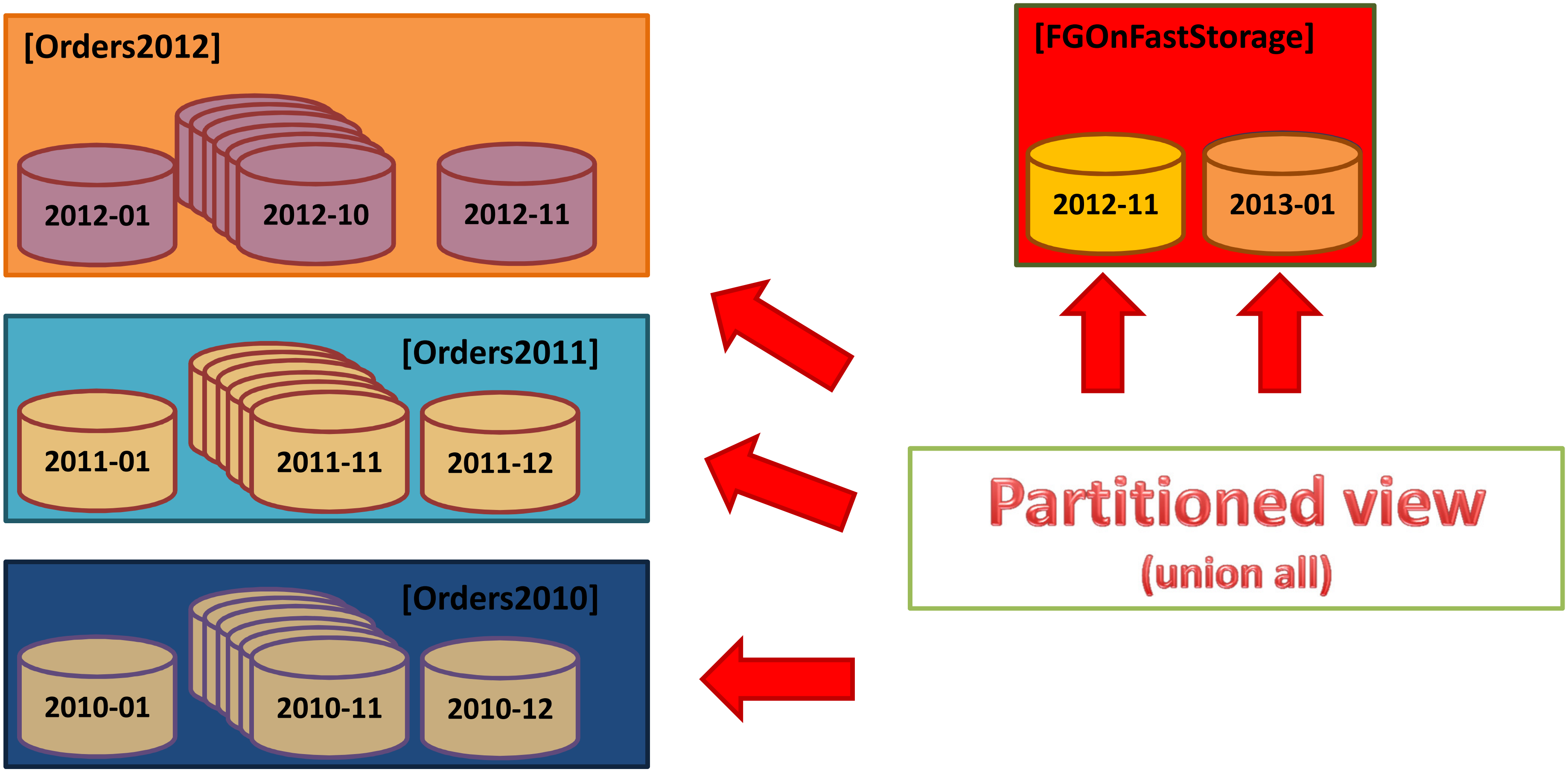
2010-11

2010-12

Mixed approach



Mixed approach



High Availability

- Dynamic files and filegroups creation
  - Failover cluster – only one copy of the database
  - AlwaysOn, Mirroring, Log Shipping – requires the same disk and folder structure

Professional Association for SQL Server

24 HOURS OF PASS

Demo

Partitioning and HA strategies

Partitioning and Filegroups

- Factors to consider
  - Disaster Recovery strategy
  - Disk subsystem limitation (size, requirements, budget)
- Place linked objects to the same FG
  - Example: *Orders* и *OrderItems*
- Separate read-only and read/write data
  - Filegroup can be marked as read-only and excluded from daily backups

Potential issues

- Increasing data size
  - Partitioned column would be part of the clustered and every non-clustered index
    - Example: Datetime (8 bytes) * 10M rows per day = ~28GB per NCI per year
- Different and in some cases suboptimal execution plans
  - More info:

    http://www.simple-talk.com/content/article.aspx?article=1587

    http://aboutsqlserver.com/publications

We discussed:

- Why we need partition data
- How we can partition data
- Tiered storage implementation and HA compatibility

Questions?

- Thank you very much for attending!

- Slides and scripts will be available for download
  - http://aboutsqlserver.com/presentations
- Email: dmitri@aboutsqlserver.com