

Виртуальный форум Microsoft

Данные. Технологии. SQL Server 2016.

SQL Server 2016
Always Encrypted
Stretch Database
R Services
Mobile BI
AlwaysOn ColumnStore
In-memory
Polybase
Azure Machine Learning
Azure SQL Database
Azure Data Lake
Azure HDInsight
Azure Synapse Analytics
Azure IoT Hub
Azure Cognitive Services
Azure DevOps
Azure Active Directory
Azure Key Vault
Azure Storage
Azure Cosmos DB
Azure Functions
Azure Logic Apps
Azure Event Hubs
Azure Service Bus
Azure Queue Storage
Azure Blob Storage
Azure File Storage
Azure Virtual Machines
Azure App Service
Azure Kubernetes Service
Azure Container Instances
Azure Container Registry
Azure Container Service
Azure Container Engine
Azure Container Network Interface
Azure Container Storage
Azure Container Service
Azure Container Engine
Azure Container Network Interface
Azure Container Storage





In-Memory OLTP в SQL Server 2016: Практическое Использование

Дмитрий Короткевич
<http://aboutsqlserver.com>

Добрый день!

20+ лет в IT

15+ лет работы с SQL Server

Microsoft Data Platform MVP

Microsoft Certified Master

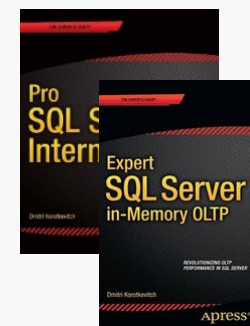
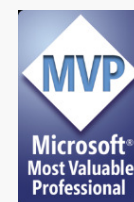
Автор:

Expert SQL Server In-Memory OLTP

Pro SQL Server Internals

Блог: <http://aboutsqlserver.com>

Email: dk@aboutsqlserver.com



Содержание

Использование memory-optimized объектов для замены временных таблиц

Использование memory-optimized табличных параметров

Использование In-Memory OLTP в системах со смешанной (OLTP & DW) нагрузкой

Demo

Использование
memory-optimized
объектов



Memory-Optimized Таблицы и Переменные

Как замена временных таблиц

- Улучшают производительность при OLTP нагрузке

- Уменьшают нагрузку на tempdb

- Потенциальные проблемы:

 - Отсутствие статистики

 - Неоптимальный bucket_count в hash индексах

Как временные таблицы в ETL процессах

- Уменьшают нагрузку на журнал транзакций

- Результаты зависят от схемы данных и ETL алгоритмов

Demo

Memory-Optimized
табличные параметры



Memory-Optimized Табличные Параметры

Наиболее оптимальный способ передачи пакета данных в T-SQL модуль

Замена дисковых на memory-optimized параметры
«прозрачна» для клиента

Потенциальные проблемы с bucket_count в hash индексах

In-Memory OLTP и Смешанная Нагрузка

«Горячие» данные: текущий операционный период

Постоянные изменения и добавления данных

OLTP нагрузка

«Холодные» данные: архивные операционные периоды

Статичные данные за длительный период времени

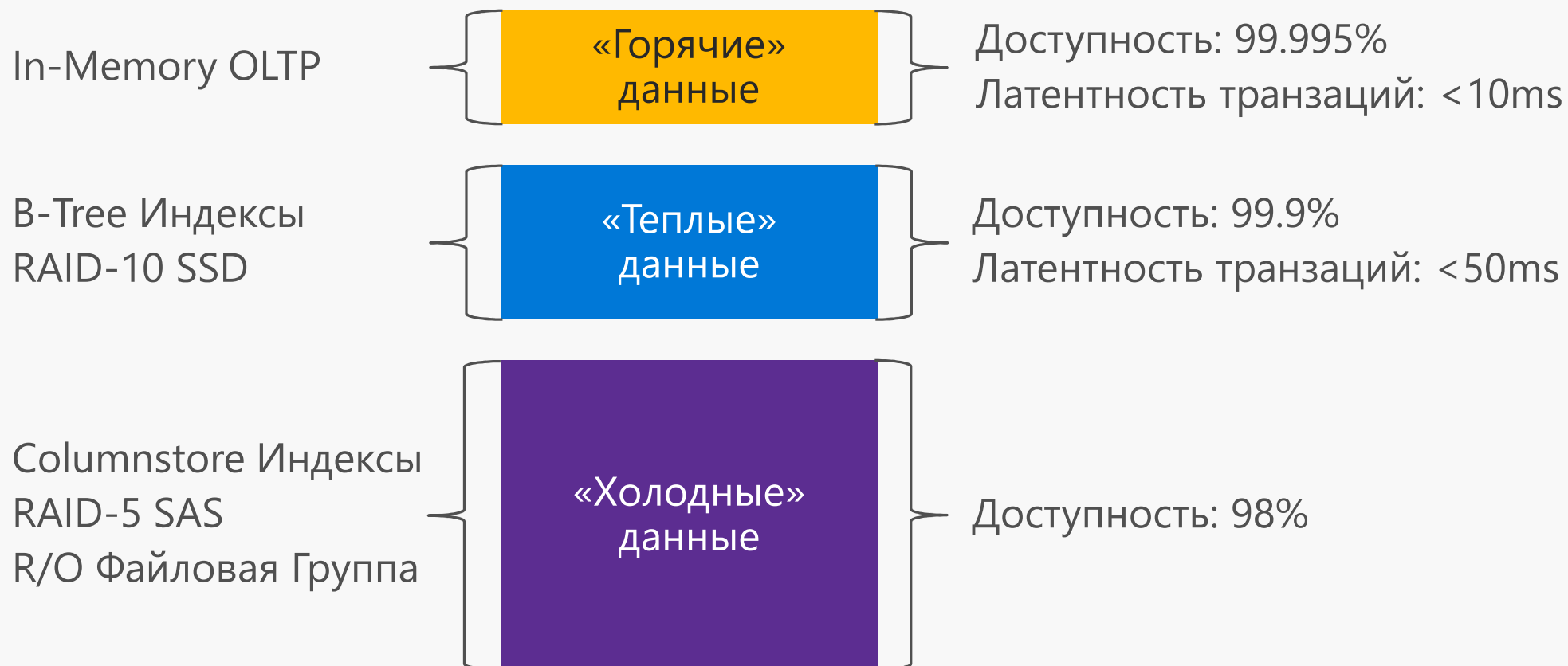
DW нагрузка (аналитика, отчетность)

«Теплые» данные: предыдущий операционный период

Статичные данные с незначительными модификациями

Смешанная нагрузка (OLTP + DW)

In-Memory OLTP и Смешанная Нагрузка



Секционирование Данных

```
create table dbo.HotData
(
    TranDate datetime not null,
    TranId int not null primary key nonclustered hash,
    constraint CHK_HotData check (TranDate >= '2016-05-01')
);
with (memory_optimized=on);

create table dbo.WarmData
(
    TranDate datetime not null,
    TranId int not null,
    constraint CHK_WarmData
    check (TranDate >= '2016-01-01' and TranDate < '2016-05-01')
);
create unique clustered index IDX_WarmData
on dbo.WarmData(TranDate, TranId)
on psWarmData(TranDate);

create table dbo.ColdData
(
    TranDate datetime not null,
    TranId int not null,
    constraint CHK_WarmData check (TranDate < '2016-01-01')
);
create clustered columnstore index IDX_ColdData on dbo.ColdData
on psColdData(TranDate);
```

```
create view dbo.Data(TranDate, TranId)
as
    select TranDate, TranId
    from dbo.HotData
    where TranDate >= '2016-05-01'

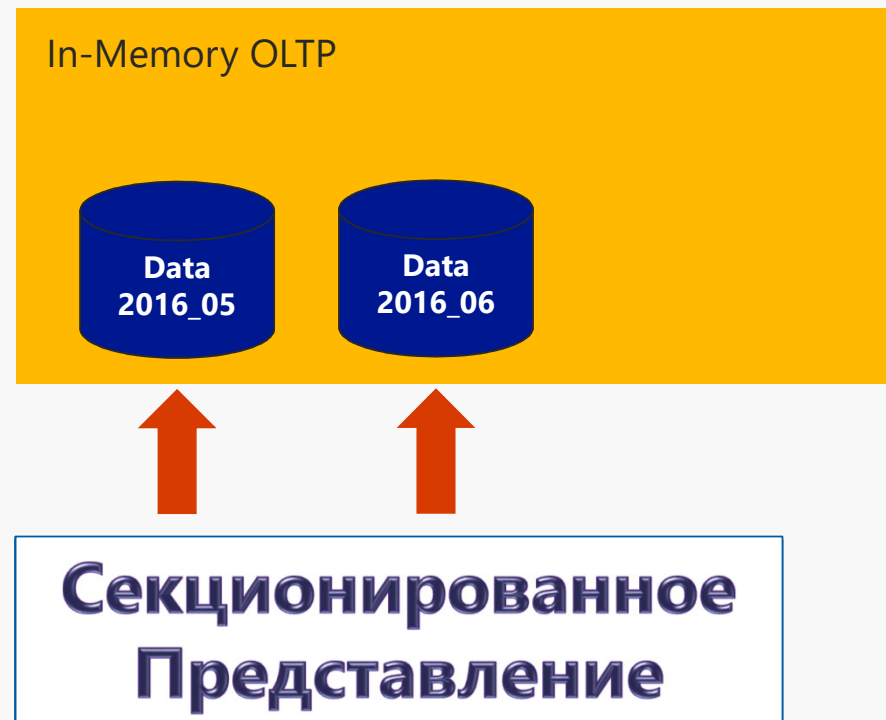
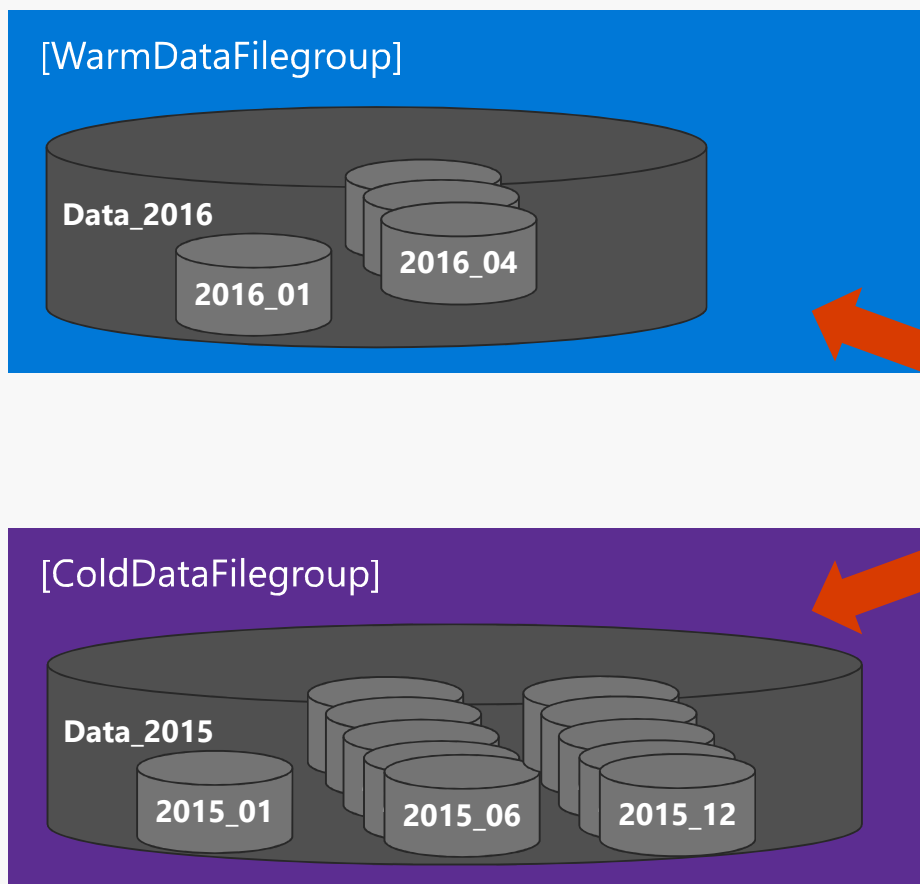
    union all

    select TranDate, TranId
    from dbo.WarmData

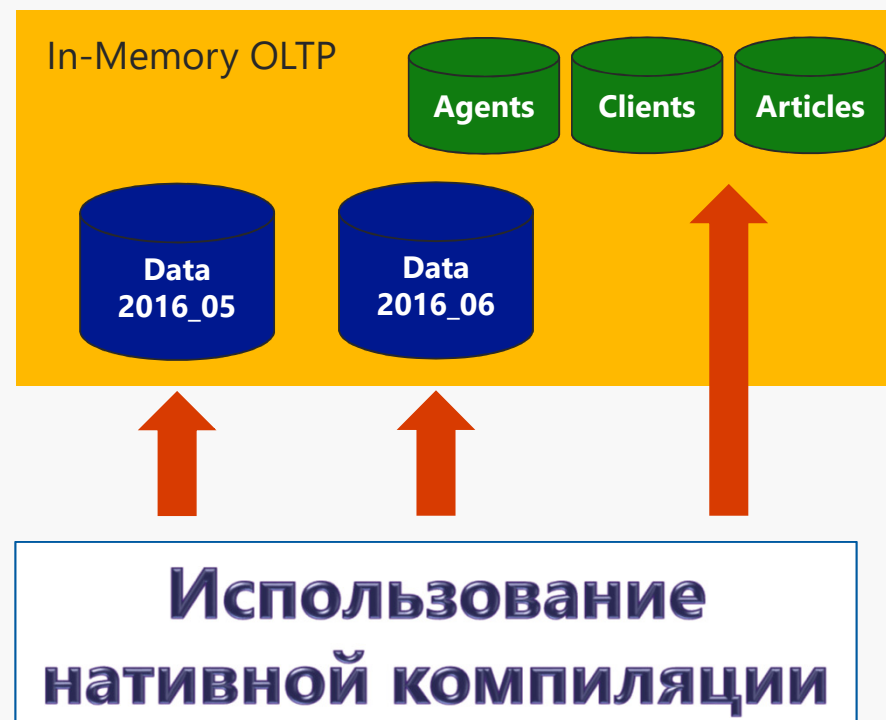
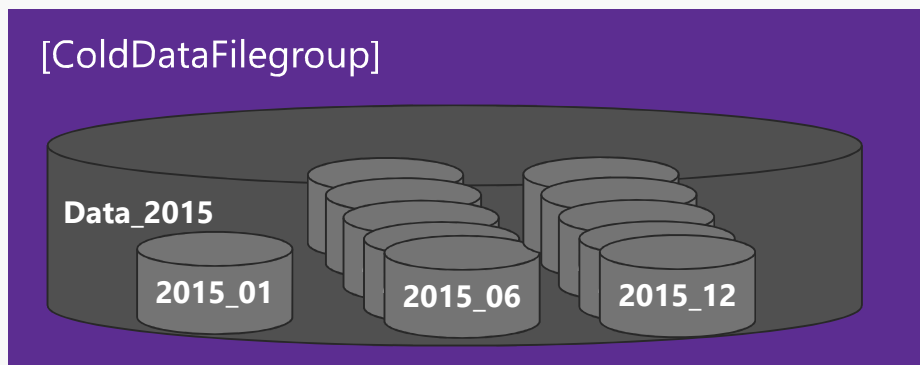
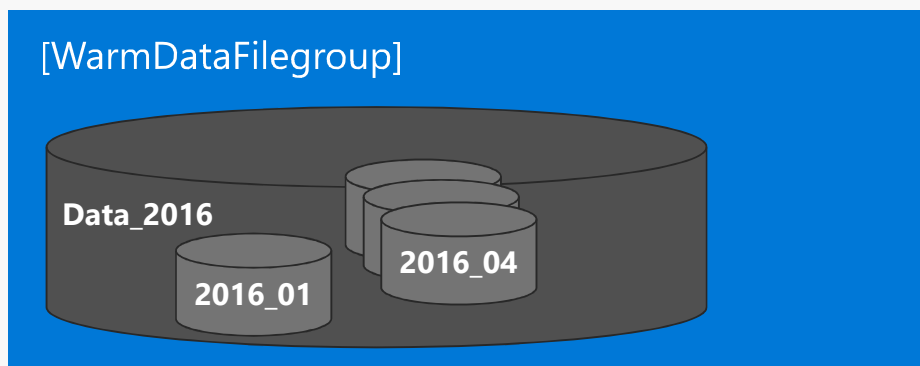
    union all

    select TranDate, TranId
    from dbo.ColdData
```

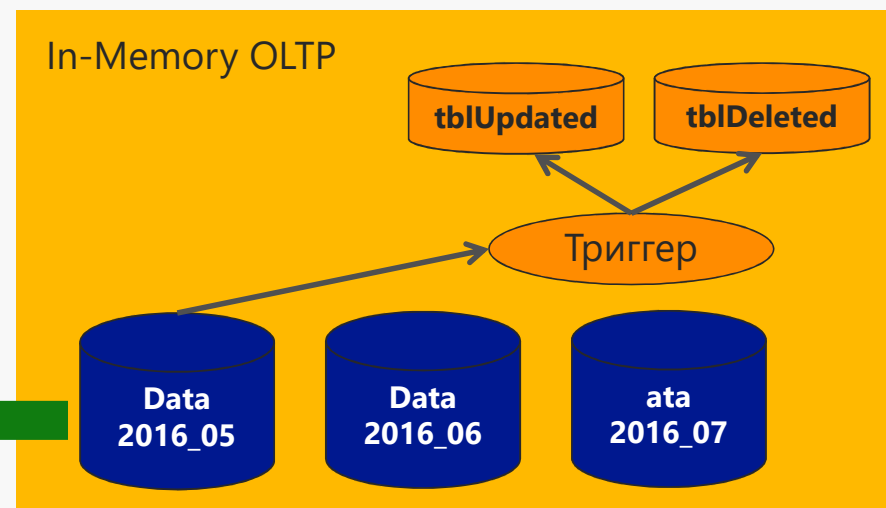
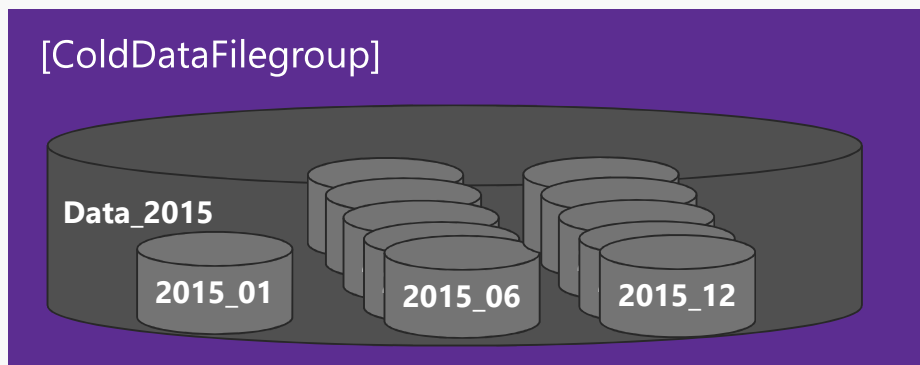
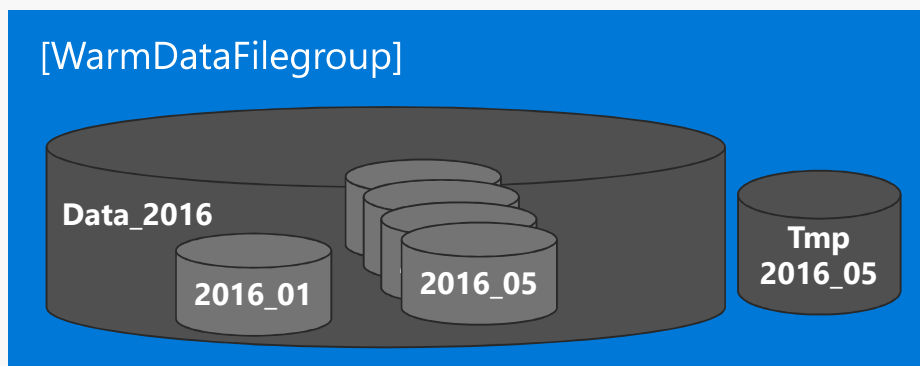
Имплементация (1)



Имплементация (1)



Имплементация (1)

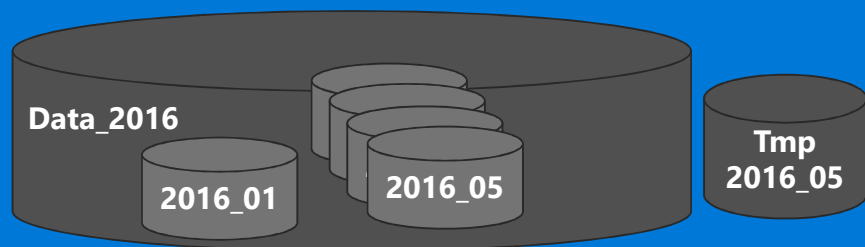


Копирование
Данных

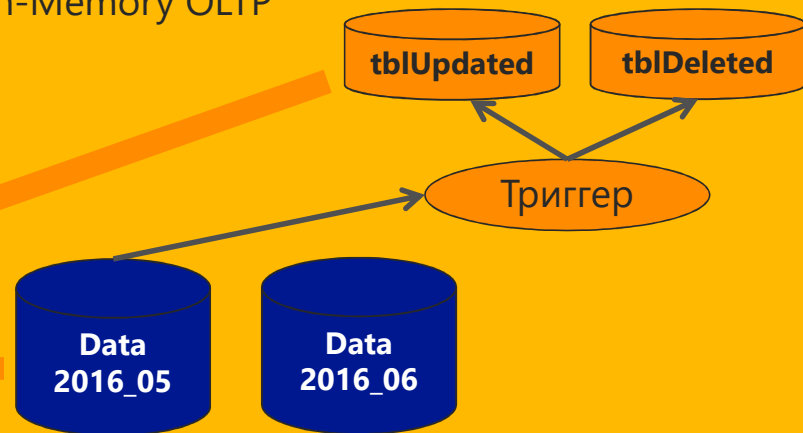
Имплементация

Обновление
и Удаление

[WarmDataFilegroup]



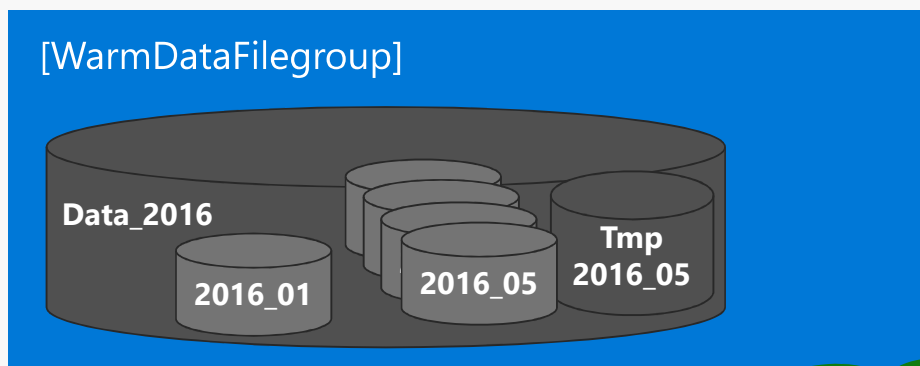
In-Memory OLTP



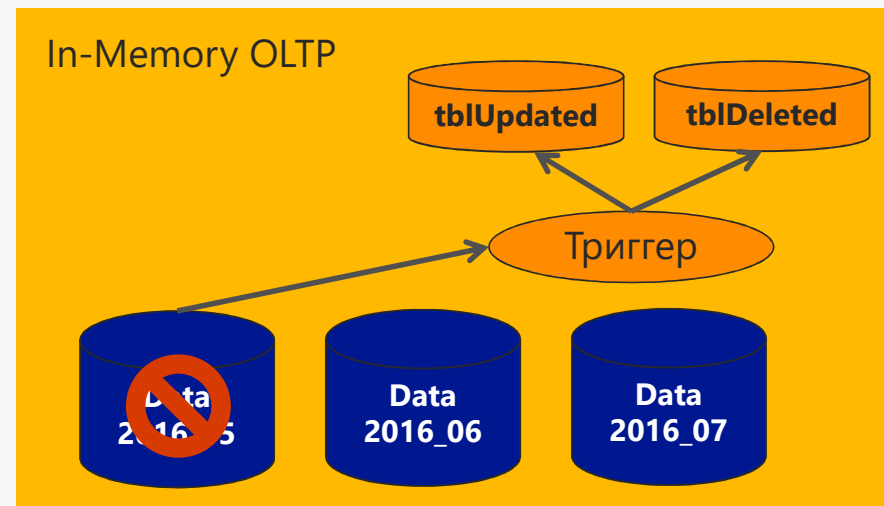
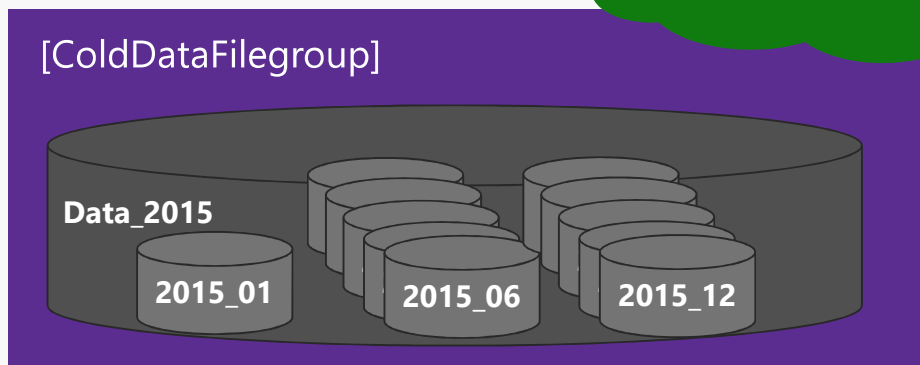
[ColdDataFilegroup]



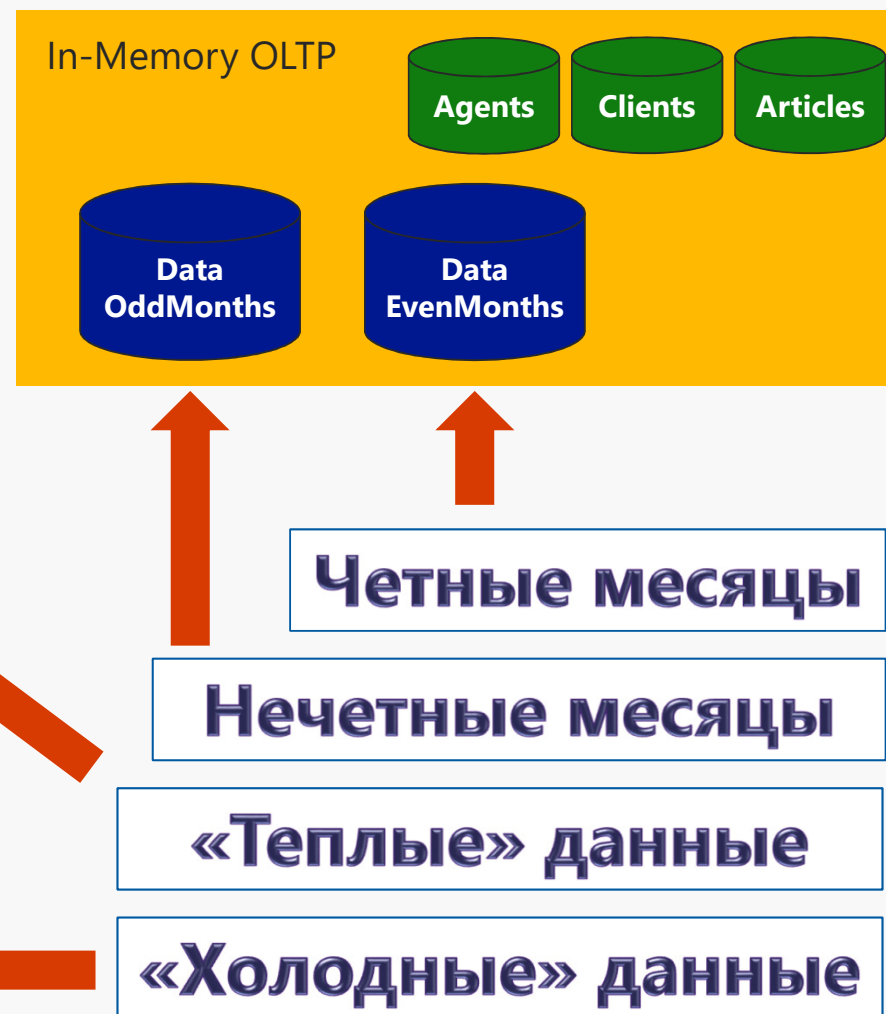
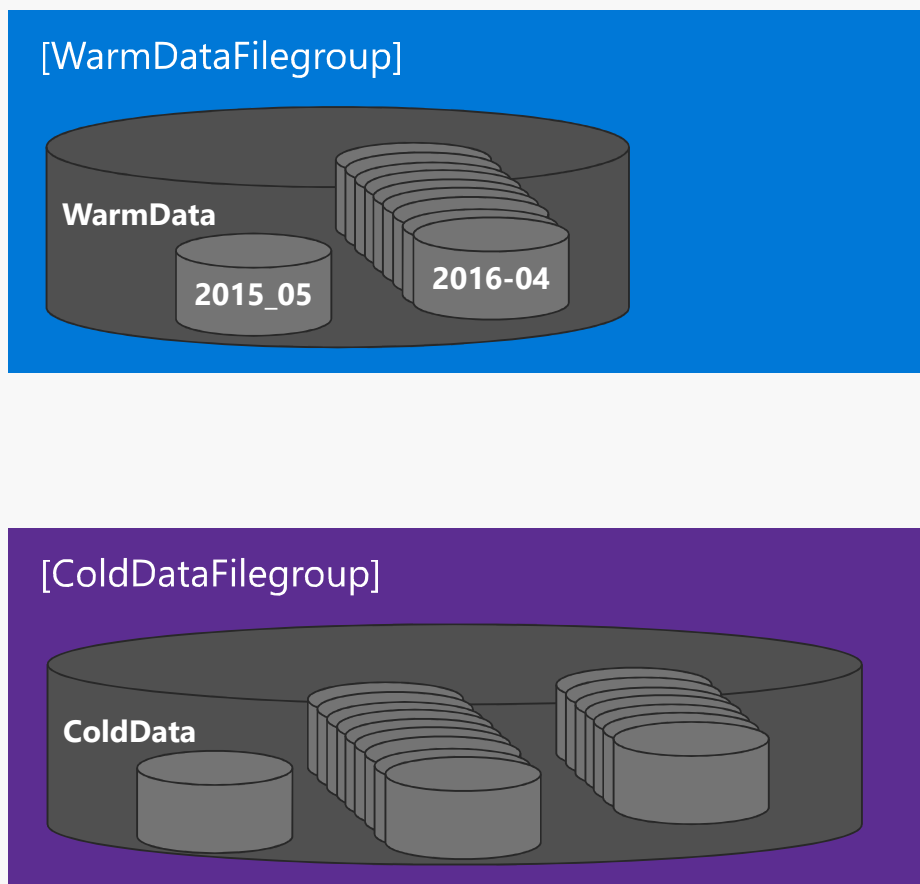
Имплементация (1)



Online Index
Rebuild

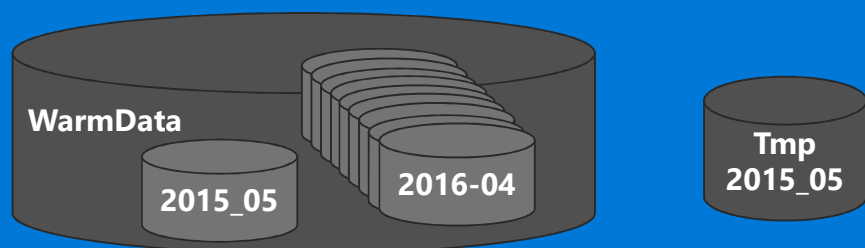


Имплементация (2)

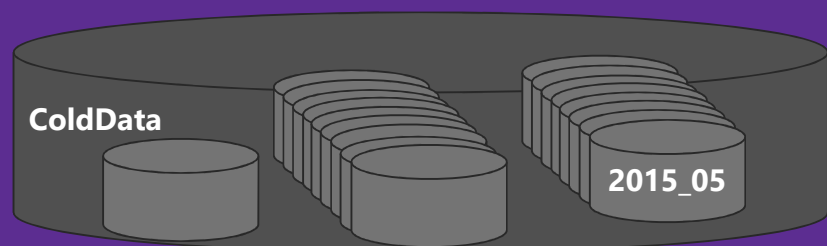


Имплементация (2)

[WarmDataFilegroup]



[ColdDataFilegroup]



In-Memory OLTP

Agents

Clients

Articles

Data
OddMonths

Data
EvenMonths

Online Index
Rebuild

Потенциальные Проблемы

ALTER TABLE ADD CONSTRAINT CHECK требует сканирование индекса и Sch-M блокировки

Вариант решения: Создание нескольких CHECK ограничений (январь); (январь..февраль); ...

CREATE INDEX ON [NewFg] WITH (DROP_EXISTING=ON)
не переносит LOB данные

Вариант решения: CREATE INDEX ON NewPartitionScheme() WITH (DROP_EXISTING=ON)

Demo

In-Memory OLTP в
системах со
смешанной нагрузкой



Мы Обсудили

Использование memory-optimized объектов для замены временных таблиц

Использование memory-optimized TVP

Использование In-Memory OLTP в системах со смешанной (OLTP & DW) нагрузкой

Примеры доступны: <http://aboutsqlserver.com>

Email: dk@aboutsqlserver.com

Виртуальный форум Microsoft

Данные. Технологии. SQL Server 2016.

SQL Server 2016
Always Encrypted
Stretch Database
R Services
Mobile BI
AlwaysOn ColumnStore
In-memory
Polybase
Azure Machine Learning
Azure SQL Data Warehouse
Always Encrypted
Stretch Database
R Services
Mobile BI
AlwaysOn ColumnStore
In-memory
Polybase
Azure Machine Learning
Azure SQL Data Warehouse



