



ОНЛАЙН-ОБРАЗОВАНИЕ


Онлайн-образование

Не забыть включить запись!





Меня хорошо видно && слышно?

Ставьте  , если все хорошо
Напишите в чат, если есть проблемы

Секционирование таблиц

курс “Разработчик MS SQL Server”
занятие № 30



Коробков Виктор

Правила вебинара



Активно участвуем



Задаем вопрос в чат или голосом



Off-topic обсуждаем в Slack



Вопросы вижу в чате, могу ответить не сразу

Цели вебинара | После занятия вы сможете

- 1 Решить, нужно ли секционирование на проекте
- 2 Выбрать ключ секционирования
- 3 Секционировать таблицу

Смысл | зачем вам это уметь

Чтобы оптимизировать большую таблицу и правильно выбрать ключ секционирования

Маршрут вебинара

1 Секционирование VS шардирование

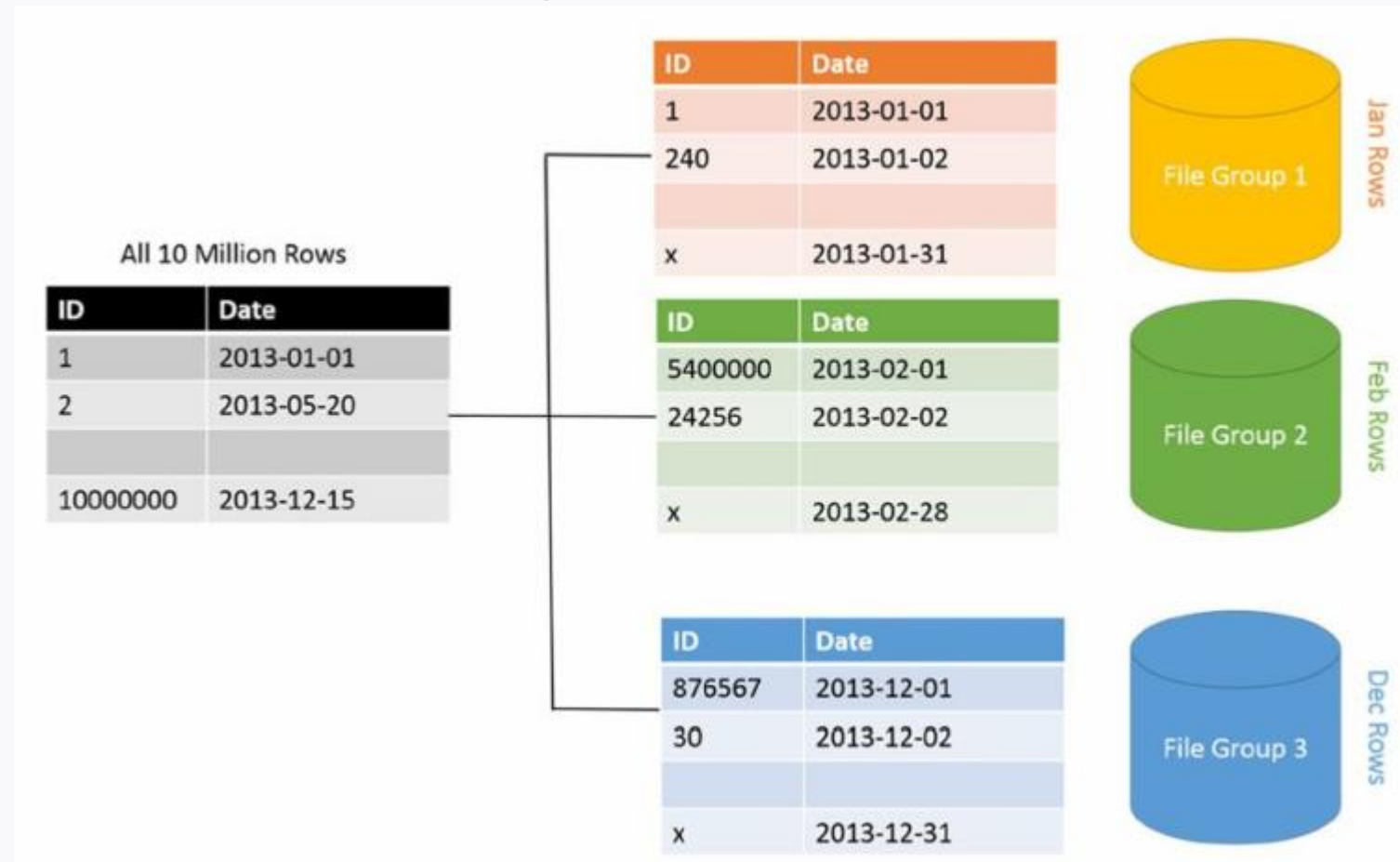
2 Ключ секционирования

3 Скользящее окно

4 Рефлексия

Секционирование

Секционирование или Партиционирование (Partitioning) – разбиение таблицы на секции, по ключу секционирования.



Преимущества

- Быстро и эффективно переносить подмножества данных и обращаться к ним, сохраняя при этом целостность набора данных;
- Операции обслуживания **можно** выполнять быстрее с одной или несколькими секциями;
- Можно повысить скорость выполнения запросов в зависимости от запросов, которые часто выполняются в вашей конфигурации оборудования.
- Таблица/индексы «размазываются» по разным файловым группам, которые могут находиться на разных физических дисках, что значительно повышает удобство администрирования и, теоретически, позволяет повысить производительность запросов к этим данным — можно либо читать только нужную секцию (меньше данных), либо читать всё параллельно (устройства разные, читается быстро).

Где использовать ???

Организация и поиск по архивным данным

DWH использование для подключения новых данных

Везде где нужно периодически добавить в таблицу много данных

Распределение нагрузки при использовании HDD дисков

[Секционирование в SQL Server](#)

Секционирование не всегда > производительность

Управление большими объемами информации

Секционирование НЕ сделает запросы быстрее

Возможно оно сделает их медленнее

И вам потребуются силы и время, чтобы они были такими же как До

Что произойдет, если запрос не будет включать в себя ЧЕТКОГО значения ключа секционирования?

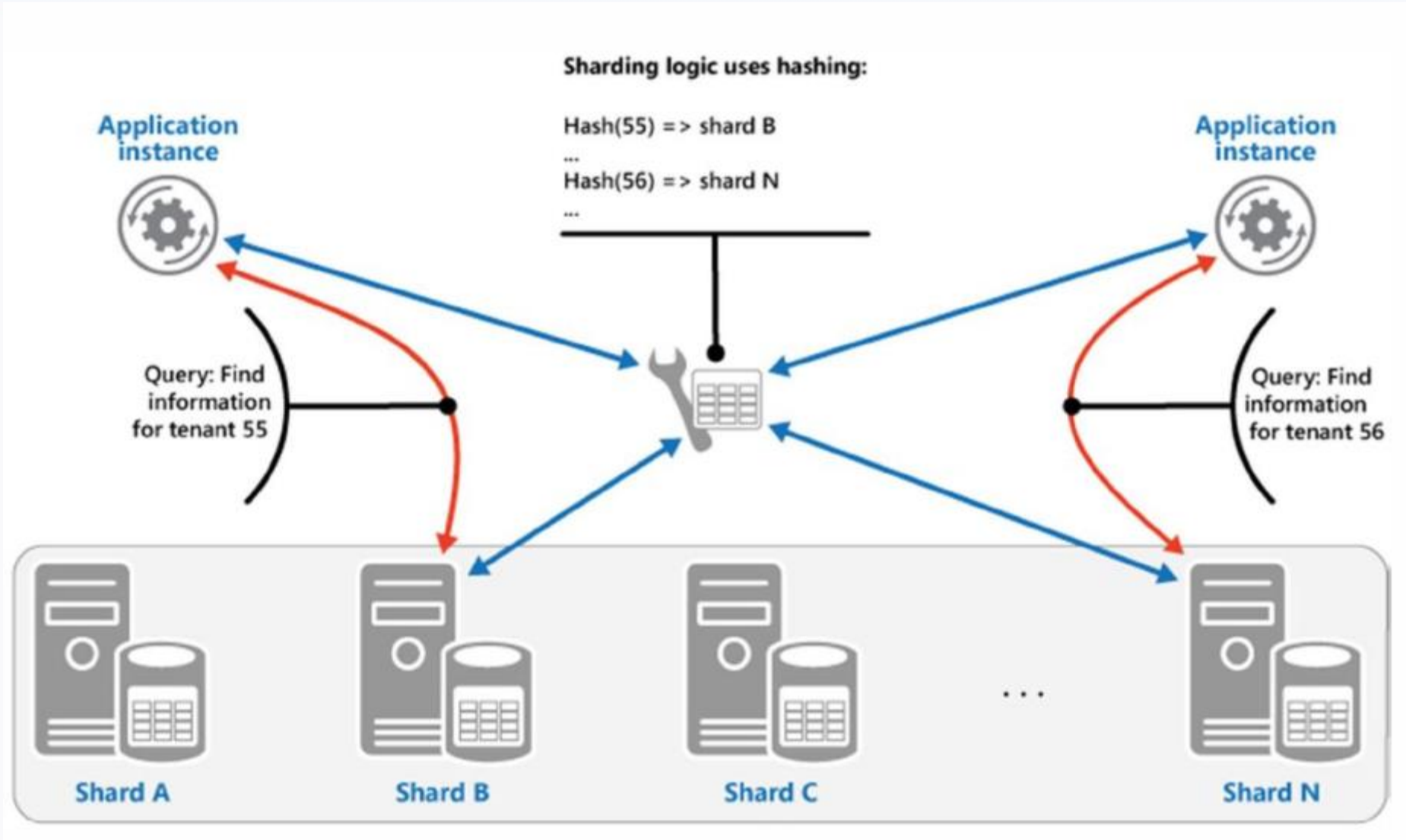
Секционирование не всегда > производительность

Управление большими объемами информации
Секционирование НЕ сделает запросы быстрее
Возможно оно сделает их медленнее
И вам потребуются силы и время, чтобы они были такими же как До

Что произойдет, если запрос не будет включать в себя ЧЕТКОГО значения ключа секционирования?

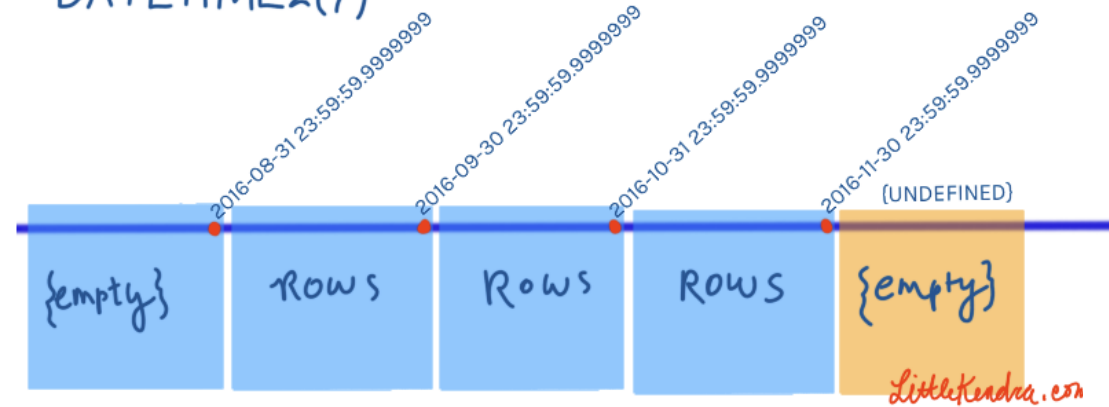
Запрос уйдет на все секции

Шардирование



Функции секционирования

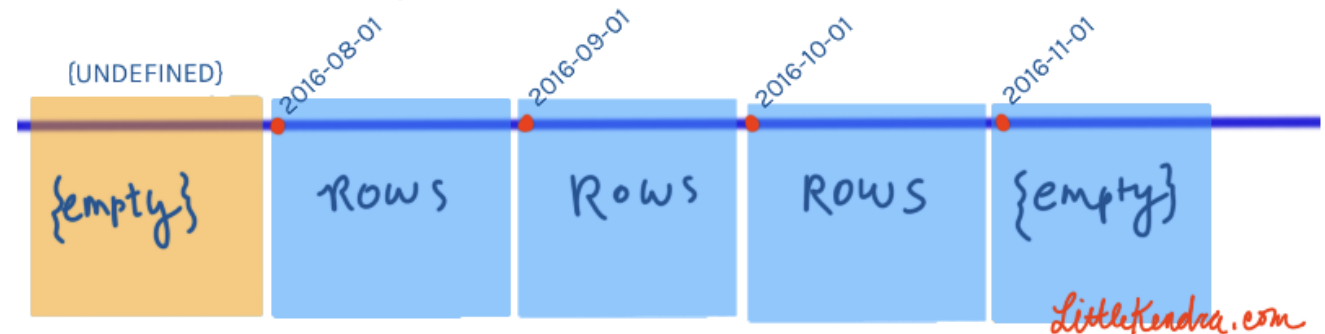
Left Based Partition Function
Upper Boundaries (Inclusive)
DATETIME2(7)



<https://www.sql.ru/blogs/t-sql/1144>

<https://medium.com/@selcukkilinc23/what-it-means-range-right-and-left-in-table-partitioning-2d654cb99ade>

Right Based Partition Function
Lower Boundaries (Inclusive)
DATETIME2(7)



Функции секционирования

```
CREATE PARTITION FUNCTION YearPartitions (date)

AS RANGE RIGHT FOR VALUES ( '2010-01-01', '2015-01-01','2020-01-01')

GO
```

Range RIGHT	
Date	Target partition
< 2010-01-01	Partition_1
>=2010-01-01 and < 2015-01-01	Partition_2
>=2015-01-01 and <2020-01-10	Partition_3
>=2020-01-01	Partition_4

Функции секционирования

```
CREATE PARTITION FUNCTION PF_Dictionary (nvarchar(100))  
  
AS RANGE LEFT FOR VALUES ( 'A', 'B', 'C', 'D')  
  
GO
```

Range LEFT	
<u>nvarchar</u>	Target partition
<= 'A'	Partition_1
>'A' and <= 'B'	Partition_2
>'B' and <='C'	Partition_3
>'C' and <='D'	Partition_4
>D	Partition_5

Ключ секционирования

Как выбрать?

Дата для архивов

Поле, которое входит во все или 99% запросов

Как сделать секционирование

- Создать файловую группу и файл (опционально)
- Выбрать ключ секционирования
- Создать схему и функцию партиционирования
- Создать таблицу по этой схеме
- Все индексы в идеале по этой схеме

Создание секционированных таблиц и индексов

Split & Merge

Split

```
CREATE PARTITION FUNCTION myRangePF1 (int)
AS RANGE LEFT FOR VALUES ( 1, 100, 1000 );
GO
--Split the partition between boundary_values 100 and 1000
--to create two partitions between boundary_values 100 and 500
--and between boundary_values 500 and 1000.
ALTER PARTITION FUNCTION myRangePF1 ( )
SPLIT RANGE (500);
```

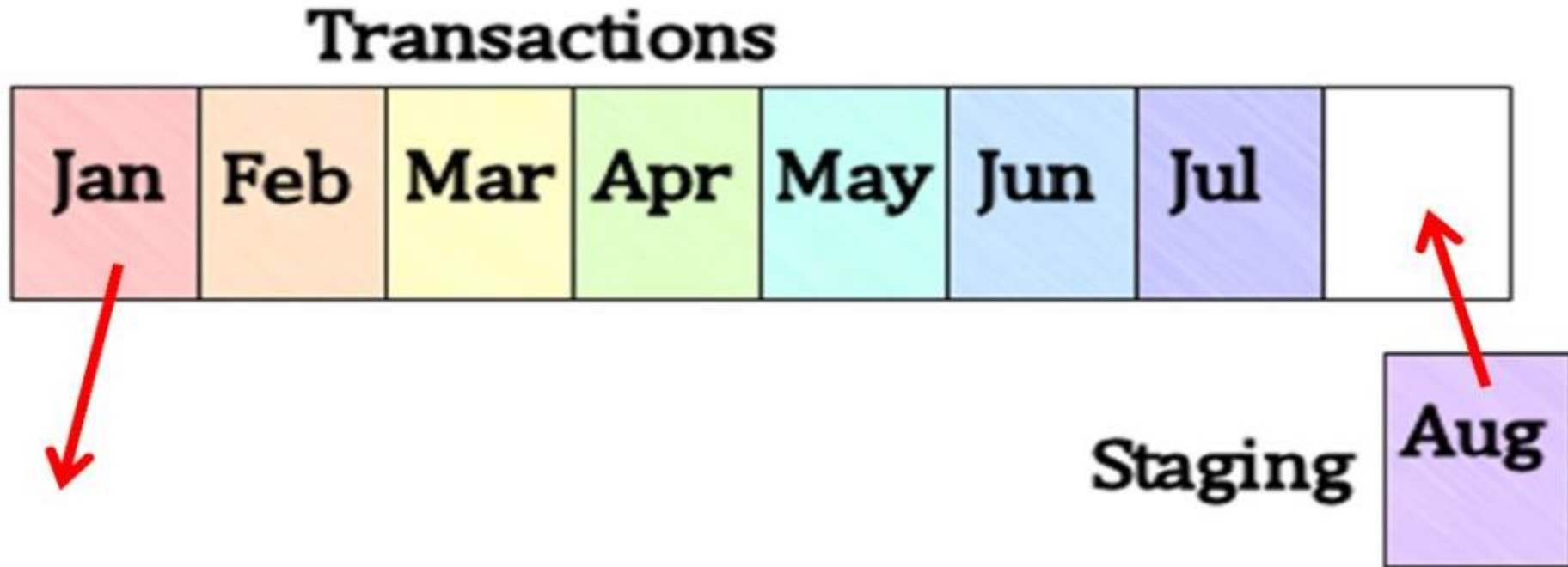
Split & Merge

Merge

```
CREATE PARTITION FUNCTION myRangePF1 (int)
AS RANGE LEFT FOR VALUES ( 1, 100, 1000 );
GO
--Merge the partitions between boundary_values 1 and 100
--and between boundary_values 100 and 1000 to create one partition
--between boundary_values 1 and 1000.
ALTER PARTITION FUNCTION myRangePF1 ()
MERGE RANGE (100);
```

ALTER PARTITION FUNCTION

Скользящее окно (sliding window)

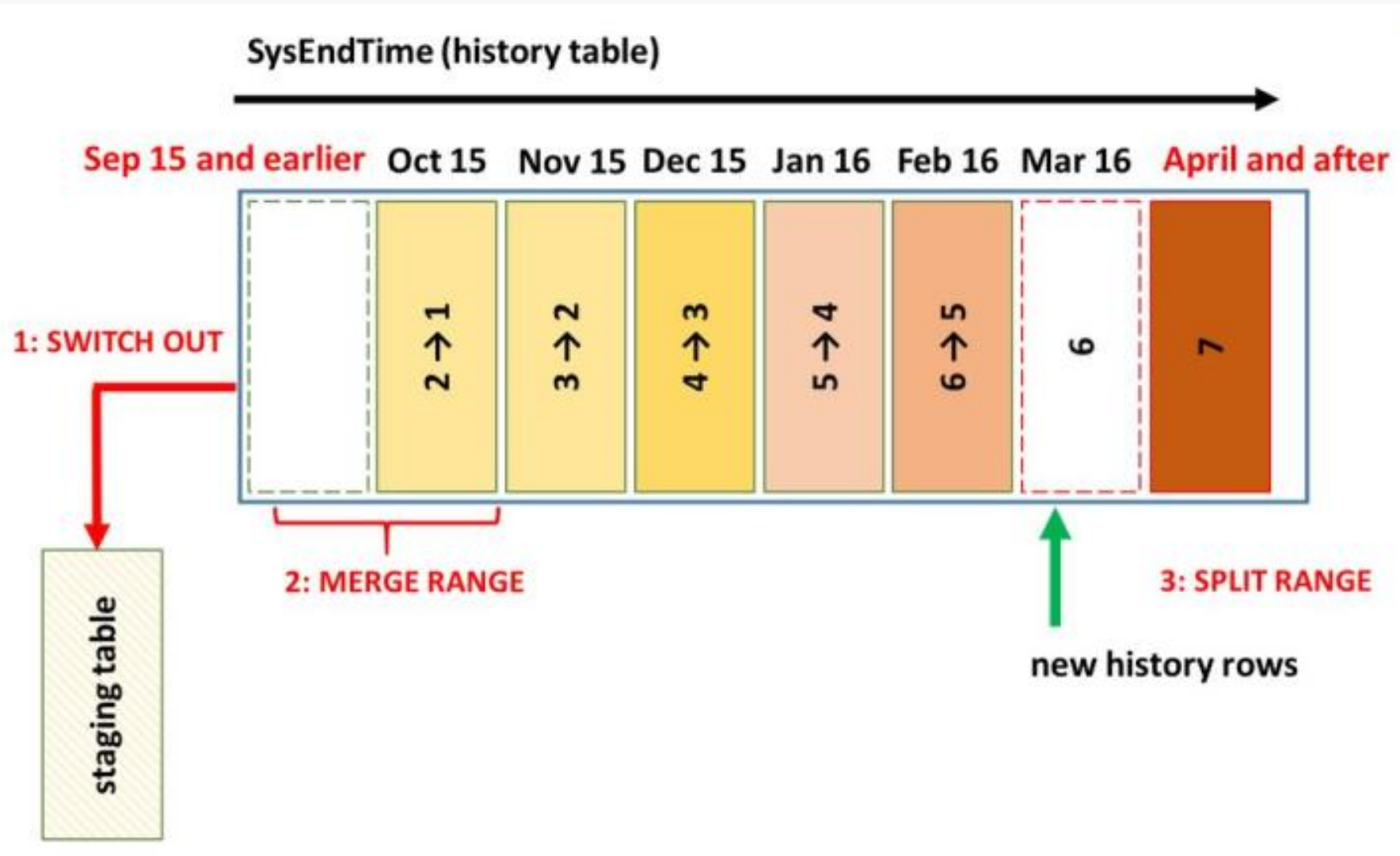


[Implementation of Sliding Window Partitioning in SQL Server to Purge Data](https://littlekendra.com/2016/02/02/sliding-window-table-partitioning-what-to-decide-before-you-automate/)

<https://littlekendra.com/2016/02/02/sliding-window-table-partitioning-what-to-decide-before-you-automate/>

!!! перемещаются не данные, а изменяются метаданные !!!

Скользящее окно (sliding window)



Скользящее окно (sliding window)

- До использования оператора SWITCH **должны существовать обе таблицы**. Перед выполнением операции переключения в базе данных должны существовать и таблица, откуда перемещается секция (исходная таблица), и таблица, получающая секцию (целевая таблица).
- **Секция-получатель должна существовать и должна быть пустой**. Если таблица добавляется как секция в уже существующую секционированную таблицу или секция перемещается из одной секционированной таблицы в другую, то секция-получатель должна существовать и быть пустой.
- **Несекционированная таблица-получатель должна существовать и должна быть пустой**. Если секция предназначена для формирования единой несекционированной таблицы, то необходимо, чтобы таблица, получающая новую секцию, существовала и являлась пустой несекционированной таблицей.
- Если секция переключена из одной секционированной таблицы в другую, то **обе таблицы должны быть секционированы по одному и тому же столбцу**.
- Исходная и целевая таблицы должны находиться в одной и той же файловой группе. Исходная и целевая таблицы в инструкции ALTER TABLE...SWITCH **должны храниться в одной и той же файловой группе**, так же как и их столбцы с большими значениями. Любые соответствующие индексы, секции индексов или индексированные представления секций также должны храниться в той же файловой группе.

Рефлексия

Есть ли шардинг в mssql?



Что произойдет, если мы отправим запрос не включающий ключа секционирования в секционированную таблицу?

Можно ли создать индекс без секций на секционированной таблице?

The background of the entire image is an aerial photograph of a city with many skyscrapers, overlaid with a semi-transparent blue layer. A network of thin, light-blue lines connects various points across the blue area, creating a digital or technological aesthetic.

Заполните, пожалуйста,
опрос о занятии по ссылке в чате

Спасибо за внимание!
Приходите на следующие вебинары



Коробков Виктор