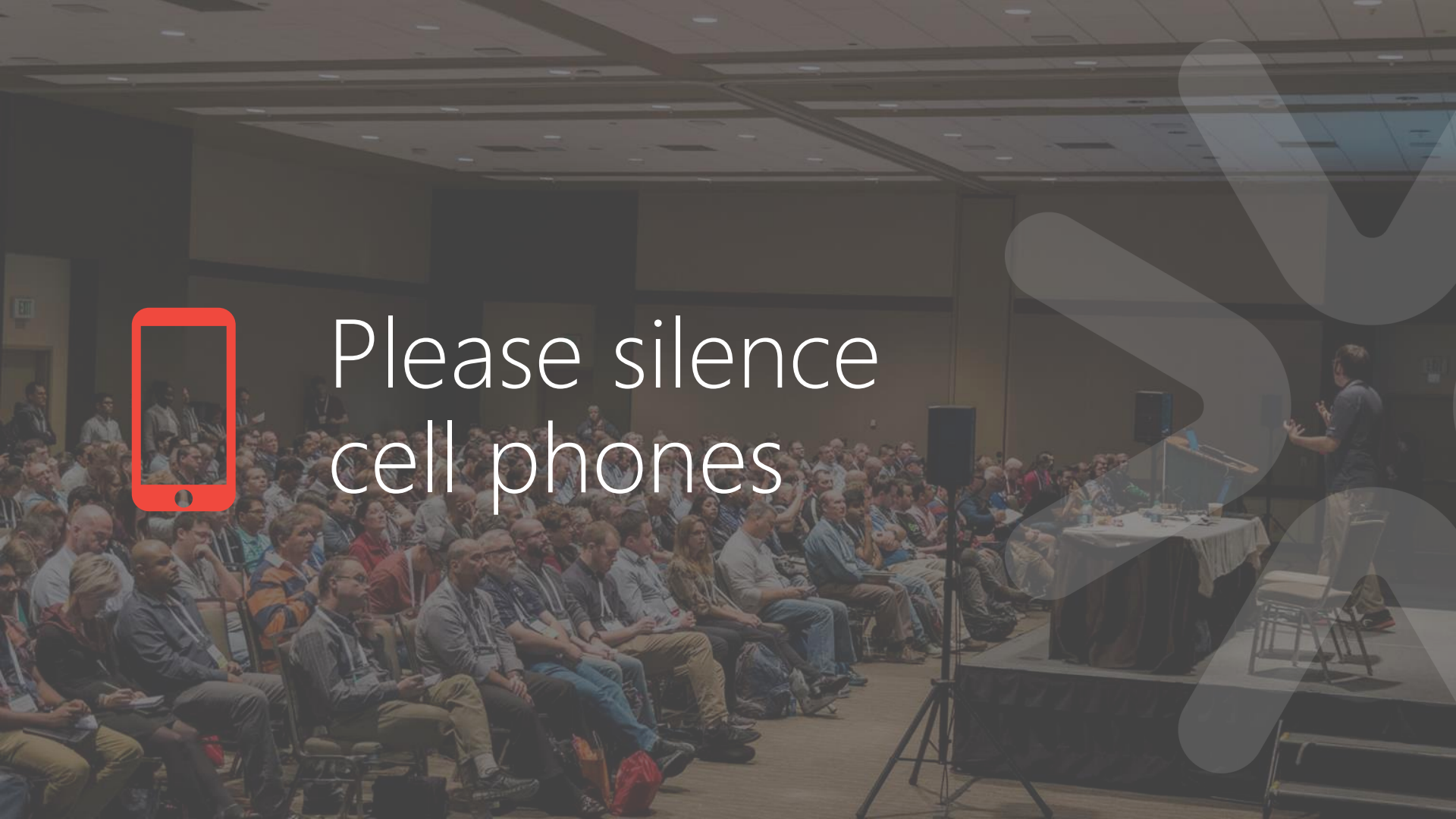
PASS SUMMIT2018

An Introduction to Partitioning

Andrew Pruski

@dbafromthecold


Please silence
cell phones

Explore everything PASS has to offer

Free Online Resources

Newsletters

PASS.org

24HOURS OF PASS

Free online webinar events

PASS LOCAL GROUPS

Local user groups around the world

PASS SQLSATURDAY

Free 1-day local training events

PASS VIRTUAL GROUPS

Online special interest user groups

PASS MARATHON

Business analytics training

PASS VOLUNTEERS

Get involved

Session evaluations

Your feedback is important and valuable.

Submit by 5pm Friday, November 16th to win prizes.

3 Ways to Access:

Go to passSummit.com

Download the GuideBook App
and search: PASS Summit 2018

Follow the QR code link displayed on session signage throughout the conference venue and in the program guide

PASS

Andrew Pruski

SQL Server DBA

/andrewpruski

@dbafromthecold

SQL Server DBA

Working with SQL Server for 10 years

Microsoft Data Platform MVP

Since December 2017

Based in Dublin, Ireland

Originally from Swansea, Wales. Living in Ireland for 5 years

Session Aim

To give you a base of knowledge to work with partitioning in SQL Server

Agenda

Partitioning Definition

Partitioning Key

Partition Functions & Schemes

Indexing Considerations

Splitting, Merging & Switching Partitions

Implementing Sliding Windows

Filegroup Restores

Partitioning Definition

Splitting a table horizontally into different units

Units can be spread across different physical locations

Limit of 15,000 partitions per table

Primarily for maintenance of data

Specialist functions available to manage data

Benefits

Partitioned tables appear as normal tables

Data is automatically mapped to the correct partition

Specialist operations allow for easy management of data

Individual partitions can be compressed

Individual partitions can be rebuilt

Drawbacks

- Requires management of partitions and filegroups
- Specialist operations can be blocked by DML operations
- Foreign keys referencing partitioned table will prevent switch operations
- Performance of queries not referencing the partitioning key will be affected

Creating a partitioned table

PASS

Partitioning key

Column in the table which defines partition boundaries
How is the data going to be split?
Archiving/retention policy for the data?
How is the table going to be queried?
All column types except timestamp, ntext, text, image, xml, varchar(max), nvarchar(max), or varbinary(max)

Partition Functions

Maps rows in the table to a partition

```
CREATE PARTITION FUNCTION [NAME](DATATYPE)
      AS RANGE RIGHT | LEFT
FOR VALUES (n,n1,n2...nx);
```

Left / Right Range Types

Defines which side of the boundary the value specified belongs

```
CREATE PARTITION FUNCTION [MyPartitionFunction](DATE)
        AS RANGE RIGHT | LEFT
FOR VALUES (2016-01-01,2017-01-01,2018-01-01);
```

RIGHT

2017-01-01 <= x < 2018-01-01

LEFT

2017-01-01 < x <= 2018-01-01

Partition Schemes

Maps partitions to filegroups

```
CREATE PARTITION SCHEME [NAME]
      AS PARTITION [FUNCTION NAME]
[ALL] TO (FILEGROUP,FILEGROUP,FILEGROUP...);
```

Creating a partitioned table

```
CREATE TABLE dbo.PartitionedTable
      (ID   INT IDENTITY(1,1),
       ColA VARCHAR(10),
       ColB VARCHAR(10),
       CreatedDate DATE)
ON PartitionScheme(CreatedDate);
```

Demo

PASS

Indexing

PASS

Clustered indexes

Create on the partition scheme specifying the partitioning key

Unique – the partitioning key has to be explicitly specified
Nonunique – the partitioning key will be added by SQL if not explicitly specified

Nonclustered indexes

An index that is created using the same partition scheme as the base table is *aligned*

An index that is created on a different filegroup or using a different partition scheme is *non-aligned*

Nonclustered indexes

Unique - the partitioning key has to be explicitly specified

Nonunique - the partitioning key will be added by SQL if not explicitly specified as an included column

Demo

PASS

Merging & Splitting

PASS

Merging Partitions

- Removes a partition
- Effectively “merges” two partitions into one
- Meta-data only operation if performed on an empty partition
- Data *will* be moved if partition is not empty, causing blocking and transaction log growth

Merging Partitions

```
ALTER PARTITION FUNCTION [NAME]()
      MERGE RANGE (VALUE);
```

Splitting partitions

Creates a new partition with new boundary value

New boundary value must be distinct from other values

Takes a schema modification lock on the table

Meta-data only operation if partition is empty

SQL *will* move data to the new partition if the data crosses the new boundary value

Splitting partitions

```
ALTER PARTITION SCHEME [NAME]
      NEXT USED [FILEGROUP];

ALTER PARTITION FUNCTION [NAME]()
      SPLIT RANGE (VALUE);
```

Demo

PASS

Switching

PASS

Switching partitions

Move a partition from one table to another

Meta-data operation, runs immediately

Both tables must have the same structures

Destination partition must be empty or…
if destination table is not partitioned, it must be completely empty

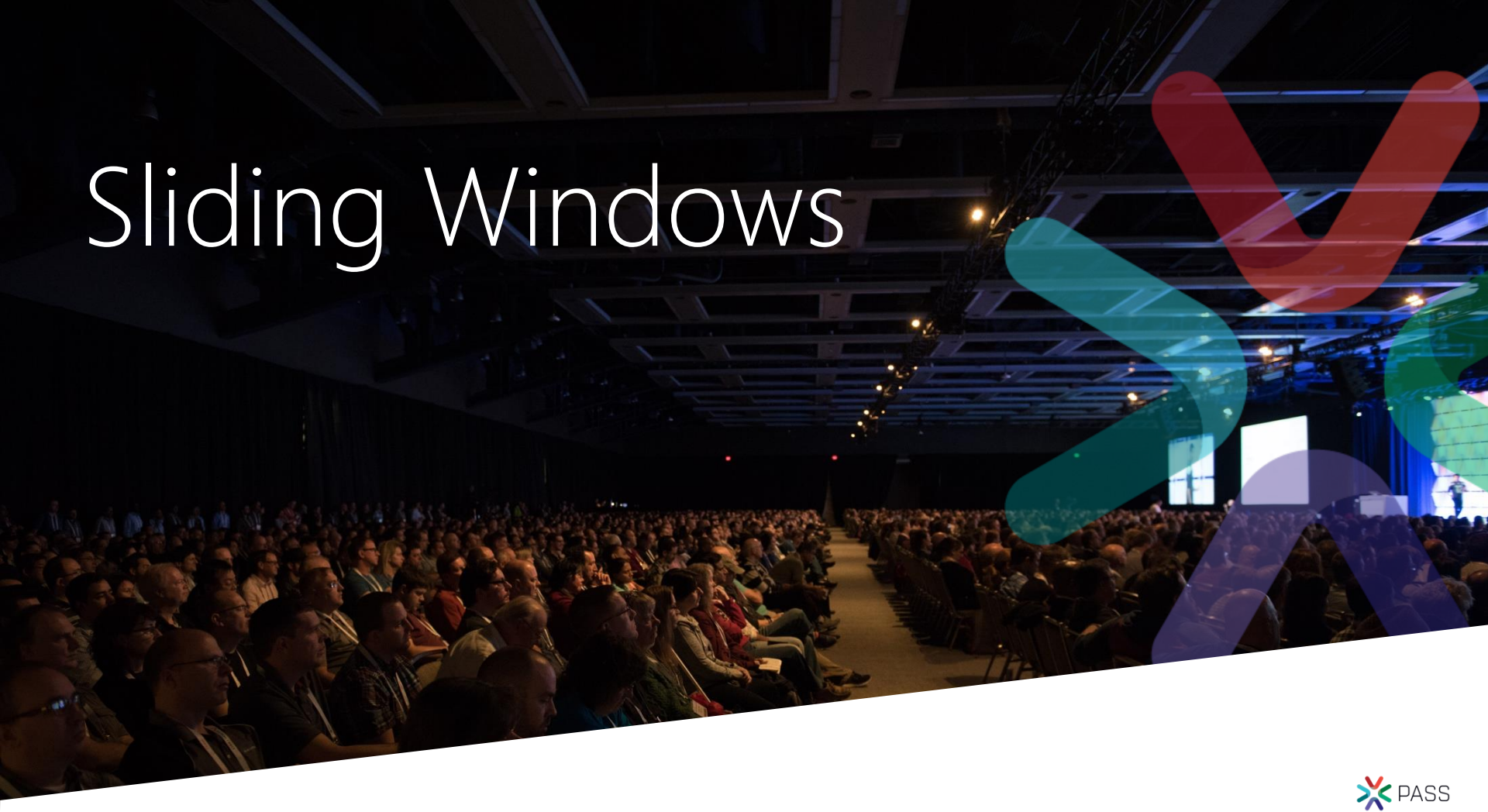
Switching partitions

```
ALTER TABLE [Source Table]
      SWITCH PARTITION Partition_Number
TO [Destination Table]
      PARTITION Partition_Number;
```

Demo

PASS

Sliding Windows



PASS

Partition Sliding Windows

Method to remove old data and bring in new data periodically

Implements the SWITCH, MERGE, & SPLIT functions

Partitions in the table move “forward” but the overall number of partitions remains the same

Partition Sliding Windows

1. SWITCH oldest partition in live table to archive table
2. MERGE oldest partition
3. SPLIT new partition
4. Load new data into staging table
5. SWITCH data from staging table to live table
6. Update statistics on live table

Demo

PASS

Filegroup Restores

PASS

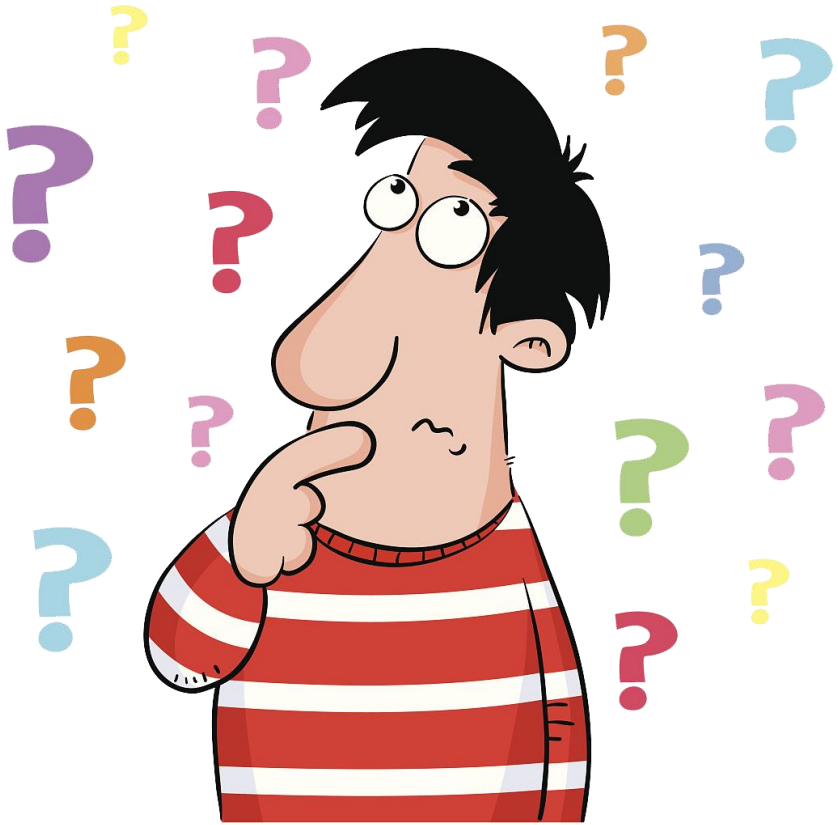
Filegroup Restores

Can be useful for VLDBs
Can be used to restore live partitions to development
Individual partitions are on different filegroups
Data in older partitions does not change or is not needed
Reduce recovery time for “active” data

Demo

PASS

A quick story



Questions?

PASS

PASS SUMMIT2018

Thank You

Learn more from Andrew Pruski

@dbafromthecold dbafromthecold@gmail.com