Aas Trailblazers

Unleashing insights

Table Partition

- Divide data into multiple chunks or partitions

- Each partition can be managed separately

- Created mostly on date column - closely tied to the order in which the data is loaded and filtered during query execution

- Benefit
  - Improves query performance by eliminating partitions not necessary
  - Simply data load by way of switch in and switch out
  - Makes data lifecycle management efficient

Table Partition

2007, 2008, 2009

A single table containing all the data

2007 **2008** **2009**

A yearly partitioned table

Table Partition

2007, 2008, 2009

A single table containing all the data

2007 **2008** **2009** **2010**

A yearly partitioned table

Switch-In

2010 ← Insert New Records

Stage Table

Table Partition

2007, 2008, 2009

A single table containing all the data

2007	2008	2009	2010

Switch-Out   A yearly partitioned table

2007   Delete or Archive

Stage Table

Range Partitioning

Range Right (beginning of period)

Partition Number	1	2	3	4	5
Value	< 2007/01/01	>=2007/01/01 and < 2008/01/01	>=2008/01/01 and < 2009/01/01	>=2009/01/01 and < 2010/01/01	>= 2010/01/01

Range Left (end of period)

Partition Number	1	2	3	4	
Value	<=2007/01/01	>2007/01/01 and <=2008/01/01	>2008/01/01 and <=2009/01/01	>2009/01/01 and <= 2010/01/01	>2010/01/01

Boundary Points
2007/01/01
2008/01/01
2009/01/01
2010/01/01

Demo

Guidance: Partitioning

- Don't over partition
  - Data is already spread across 60 distributions
  - Partitioning granularity likely to differ to SQL Server or other SMP systems
  - Columnstore index row groups up to 1,048,576 rows
- Partition for data lifecycle management
  - Sliding window development
  - Targeted index rebuilds