Special Considerations

Kimberly L. Tripp

@KimberlyLTripp | www.SQLskills.com

What This Module Covers

Estimates and recompilation

Resource governor

Coding strategies

Estimates and Recompilation

Feature	Details
Filtered objects	
Partitioned objects	
CHOOSE	

Estimates and Recompilation

Feature

Filtered objects

Partitioned objects

CHOOSE

Details

SQL Server cannot save a single plan based on filtered indexes or filtered statistics

SQL Server won’t use a filtered object for parameters unless there is recompilation performed:

- **Using OPTION (RECOMPILE):** this is safest as it forces the statement to be recompiled and is not affected by database setting for parameterization
- **Using a dynamically constructed string:** this is plagued with potential coding and security (injection) problems and it only works when simple parameterization is on; forced statements will not leverage filtered objects

Estimates and Recompilation

Feature

Filtered objects

Partitioned objects

CHOOSE

Details

SQL Server CAN do partition elimination correctly (even if the first execution is only against one partition) so you will get the correct data

However, the first execution will set the “estimates” for the plan which could vary across the data sets (partitions) and lead to poor performance

Estimates and Recompilation

Feature

Filtered objects

Partitioned objects

CHOOSE

Details

Similar to partitioned views in that you will get the correct data but subsequent executions will have estimation issues and possibly poor performance

Resource Governor Resource Pools

When users are in a resource pool, they will only use plans from that resource pool

When procedures do not have a plan (in that pool) one will be created, causing a recompilation with the plan chosen by the parameters used for that execution

Subsequent executions from users within the same pool will reuse this plan (as long as their session settings are the same)

Resource Governor Resource Pools

Pro

- Different user types might warrant different plans and/or have different access patterns

Con

- If the different users aren’t needing different plans or have different access patterns then you might have inconsistencies in plans and performance
- This only adds to the complexity of troubleshooting

Resource Governor Resource Pools

Pro

- You can manage each pool independently and clear plans for just a specific pool using:

```
DBCC FREEPROCCACHE
  [ ( { pool_name } ) ]
  [WITH NO_INFOMSGS]
```

Con

- Using lots of pools may require more plan cache memory

Coding Strategies for Performance

Temporary tables

- Better for larger [e.g. over 10K rows] objects
- No restrictions on creation
- No restrictions on indexing
- Real statistics

Table variables

- Ideal for small objects
- All indexes must be unique
- Poor estimates can result in poor plans and poor performance:
  - Fixed estimate prior to SQL Server 2014 is 1 row
  - Fixed estimate in SQL Server 2014 is 100 rows

Coding Strategies for Performance

- Temp tables vs. table variables / table-value parameters (TVP)
- Table variables and table-value parameters (TVPs)
  - Until SQL Server 2012 SP3 and SQL Server 2014 CU3 table variables and TVPs can have horrible estimates unless adding OPTION (RECOMPILE)
    - Too many recompiles can waste CPU, consider trace flag 2453 which reduces the need to recompile when estimates are close to previously executed estimates
    - Check out Aaron Bertrand’s excellent post: *New Trace Flag to Fix Table Variable Performance* (http://bit.ly/1p9Mgzv)
- Multi-statement, user-defined functions:
  - Prior to SQL Server 2014, the fixed cardinality estimate is 1 row
  - In SQL Server 2014, the fixed cardinality estimate is 100 rows

Coding Strategies for Performance

Use data types consistently to avoid implicit conversions

Isolate columns to one side of an expression

Use variables where “average” estimation is desired otherwise use parameters (always use literals where possible)

Be careful of row-based operations with functions

There are many other potential coding strategies here: CTEs, joins vs. subqueries, etc.

- When To Break Down Complex Queries: http://bit.ly/1HbBB1Q

Realistic testing on realistic sample data is critical; developers that know SQL Server…

What This Module Covered

Estimates and recompilation

Resource governor

Coding strategies