# Plan Caching in SQL Server 2008

Source : https://technet.microsoft.com/en-us/library/ee343986(v=sql.100).aspx

Before a query begins execution on a SQL Server, the batch gets compiled into a plan

Compiled plans are stored into a part of SQL Server's memory called plan cache.

Most memory used by SQL Server is allocated to the Buffer Pool, SQL Server steals a proportion of this memory for use in caching query plans

The overall amount of memory also affected by memory pressure being experienced by Server

Internally, SQL Server converts the name of the stored procedure to an ID, and subsequent plan reuse happens based on the value of that ID

When a stored procedure is compiled for the first time (or in fact any parameterized batch), **the values of the parameters supplied** with the execution call are used to optimize the statements within that stored procedure. This process is known as "parameter sniffing."

# Query plans and execution contexts

There are at most two instances of a query plan at any time in plan cache: one for all of the serial executions and one for all of the parallel executions.

Execution contexts hold the values needed for a specific execution of a query plan, such as parameter values and user-specific information. Once a reusable query plan is found, an available execution context is found (causing execution context reuse) or freshly generated the execution context used for a particular execution has slight impact on performance

# Query Plan Reuse

To avoid multiple query plans for a query that is executed with different parameter values, execute the query using **sp\_executesql** stored procedure. This method is useful if the same query plan is good for all or most of the parameter values

When a query contains the "KEEPFIXED PLAN" hint, its plan is not recompiled for plan optimality-related reasons

# Causes of Recompilations

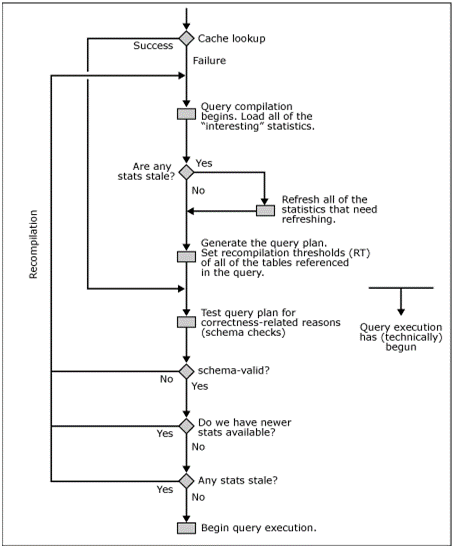
* Schemas of objects have changed since query/sp was last compiled
* Data in tables that query/sp references may have changed considerably since it was last compiled, data changes caused Recompilation Threshold (RT) to increase, when it reached certain limit, query plan will be recompiled
* Statistic has changed :
  + CREATE INDEX … WITH DROP EXISTING
  + sp\_createstats stored procedure
  + sp\_updatestats stored procedure
  + ALTER INDEX REBUILD (but not REORGANIZE)
  + When a table with no rows gets a row
  + When 500 rows are changed to a table that is less than 500 rows
  + When 20% + 500 are changed in a table greater than 500 rows

Statistics-related recompilations can be identified by the "EventSubClass" column of the profiler trace (to be described later in this paper) containing the string "Statistics changed".

* Sql Server restart

# High-level overview of query compilation

The following flowchart succinctly describes the batch compilation and recompilation process in SQL Server. The main processing steps are as :



Useful system table

* sys.dm\_exec\_cached\_plans
* sys.dm\_exec\_query\_plan
* sys.dm\_exec\_plan\_attributes
* sys.dm\_exec\_sql\_text
* sys.dm\_exec\_cached\_plan\_dependent\_object
* sys.syscacheobjects