Statistics

Last updated by | Vitor Tomaz | Feb 24, 2023 at 3:27 AM PST

Contents

Statistics

This is a "How To" TSG about Query Optimizer statistics and their maintenance.

Issue

Statistics for query optimization contain statistical information about the distribution of values in one or more columns of a table or indexed view. The Query Optimizer uses these statistics to estimate the cardinality, or number of rows, in the query result. These cardinality estimates enable the Query Optimizer to create a high-quality query plan. For example, depending on your predicates, the Query Optimizer could use cardinality estimates to choose the `Index Seek` operator instead of the more resource-intensive `Index Scan` operator, if doing so improves query performance.

For most queries, the Query Optimizer already generates and maintains the necessary statistics for a high-quality query plan. If however the statistics get stale/outdated or are sampled in a non-optimal way, it may lead to bad cardinality estimates and bad query execution plans. By default, auto-update statistics is enabled on the databases but this might not be sufficient. In some cases you can improve query performance by updating the statistics on a table or indexed view. If auto-update statistics is disabled, it is usually necessary to update the statistics regularly through a scheduled job.

Investigation / Analysis

The topic of statistics and their maintenance is discussed in depth in the article Statistics. It contains everything you and your customer need to know about statistics, and helps you decide when and how to perform statistics maintenance. It covers concepts such as histogram and density, auto-create/auto-update/auto-update async, sampling vs. FULLSCAN, filtered statistics, and improving cardinality estimates.

Updating statistics requires I/O bandwidth on the database and causes queries to recompile. Therefore you shouldn't update statistics too frequently because there is a performance tradeoff between improving query plans, the I/O impact on active workloads, and the time it takes to recompile queries. The specific tradeoffs depend on your application.

Mitigation

For a complete, customer-ready index and statistics maintenance Stored Procedure, see Yochanan's blog article How to maintain Azure SQL Indexes and Statistics.

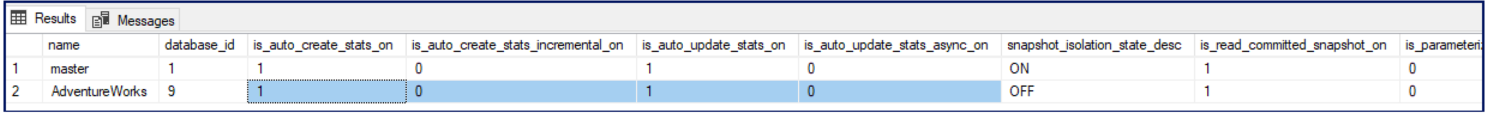
See below for sample SQL scripts to perform some manual analysis and maintenance steps.

Check which automatic maintenance options are active

The following query returns the current settings for several database options, including those related to statistics maintenance:

```
SELECT
    name,
    database_id,
    is_auto_create_stats_on,
    is_auto_create_stats_incremental_on,
    is_auto_update_stats_on,
    is_auto_update_stats_async_on,
    snapshot_isolation_state_desc,
    is_read_committed_snapshot_on,
    is_parameterization_forced,
    is_memory_optimized_enabled,
    is_accelerated_database_recovery_on,
    delayed_durability_desc,
    is_encrypted,
    is_published,
    is_distributor,
    is_cdc_enabled
FROM sys.databases
--WHERE name = 'AdventureWorks'
```

Sample output:



	name	database_id	is_auto_create_stats_on	is_auto_create_stats_incremental_on	is_auto_update_stats_on	is_auto_update_stats_async_on	snapshot_isolation_state_desc	is_read_committed_snapshot_on	is_parameteri
1	master	1	1	0	1	0	ON	1	0
2	AdventureWorks	9	1	0	1	0	OFF	1	0

Note the new option "is_auto_update_stats_async_on", which can be enabled through:

> ```
> ALTER DATABASE [AdventureWorks] SET AUTO_UPDATE_STATISTICS_ASYNC ON;
> ```

It comes with a companion option to reduce its priority against other workloads:

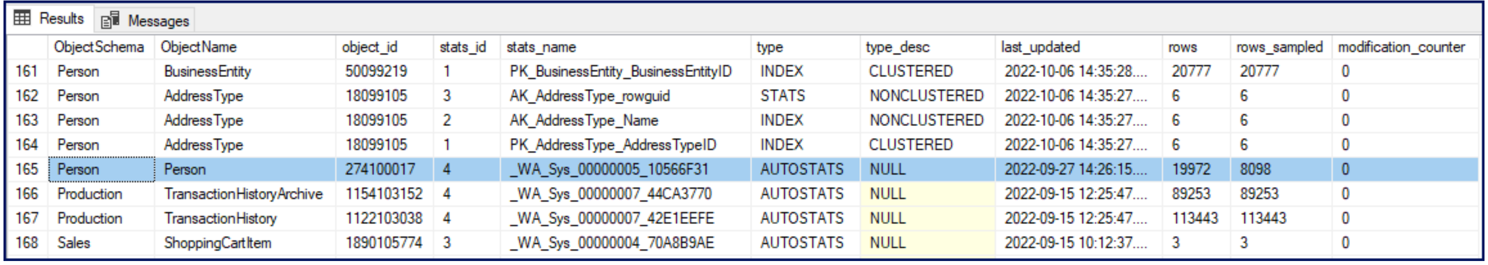
> ```
> ALTER DATABASE SCOPED CONFIGURATION SET ASYNC_STATS_UPDATE_WAIT_AT_LOW_PRIORITY = ON;
> ```

Find the last date and time when statistics were updated

The following query shows when the statistics were updated most recently, along with information about the sampling size and the number of data modifications since the last statistics update. Auto-update statistics is only triggered after hitting a certain modification threshold and this query provides the data to see this.

```
SELECT
    ObjectSchema = OBJECT_SCHEMA_NAME(s.object_id),
    ObjectName = object_name(s.object_id),
    s.object_id,
    s.stats_id,
    s.name AS 'stats_name',
    CASE WHEN (s.stats_id > 2 AND s.auto_created = 1) THEN 'AUTOSTATS' WHEN (s.stats_id > 2 AND s.auto_created
    i.type_desc,
    sp.last_updated,
    sp.rows,
    sp.rows_sampled,
    sp.modification_counter
FROM sys.stats s
OUTER APPLY sys.dm_db_stats_properties(s.object_id,s.stats_id) sp
LEFT JOIN sys.indexes i ON sp.object_id = i.object_id AND sp.stats_id = i.index_id
WHERE OBJECT_SCHEMA_NAME(s.object_id) != 'sys'
--AND (isnull(sp.modification_counter,0) > 0 )  -- filter on modification stats
ORDER BY sp.last_updated DESC;
```

Sample output:



	ObjectSchema	ObjectName	object_id	stats_id	stats_name	type	type_desc	last_updated	rows	rows_sampled	modification_counter
161	Person	BusinessEntity	50099219	1	PK_BusinessEntity_BusinessEntityID	INDEX	CLUSTERED	2022-10-06 14:35:28....	20777	20777	0
162	Person	AddressType	18099105	3	AK_AddressType_rowguid	STATS	NONCLUSTERED	2022-10-06 14:35:27....	6	6	0
163	Person	AddressType	18099105	2	AK_AddressType_Name	INDEX	NONCLUSTERED	2022-10-06 14:35:27....	6	6	0
164	Person	AddressType	18099105	1	PK_AddressType_AddressTypeID	INDEX	CLUSTERED	2022-10-06 14:35:27....	6	6	0
165	Person	Person	274100017	4	_WA_Sys_00000005_10566F31	AUTOSTATS	NULL	2022-09-27 14:26:15....	19972	8098	0
166	Production	TransactionHistoryArchive	1154103152	4	_WA_Sys_00000007_44CA3770	AUTOSTATS	NULL	2022-09-15 12:25:47....	89253	89253	0
167	Production	TransactionHistory	1122103038	4	_WA_Sys_00000007_42E1EEFE	AUTOSTATS	NULL	2022-09-15 12:25:47....	113443	113443	0
168	Sales	ShoppingCartItem	1890105774	3	_WA_Sys_00000004_70A8B9AE	AUTOSTATS	NULL	2022-09-15 10:12:37....	3	3	0

Update ALL Statistics WITH FULLSCAN

Use the stored procedure from Yochanan's blog article How to maintain Azure SQL Indexes and Statistics for updating all statistics. The proc has build-in logic to update statistics only if needed, thus has less impact on the concurrent workload.

If you don't care about impacting other workloads and simply want to update all the statistics on all the tables in your database, then run the following SQL script:

```
-- This will update all the statistics on all the tables in your database.
-- Remove the comments from EXEC sp_executesql in order to actually run the commands, instead of just printing

SET NOCOUNT ON

DECLARE updatestats CURSOR FOR
    SELECT table_schema, table_name FROM information_schema.tables where TABLE_TYPE = 'BASE TABLE'

OPEN updatestats

DECLARE @tableSchema NVARCHAR(128)
DECLARE @tableName NVARCHAR(128)
DECLARE @Statement NVARCHAR(300)

FETCH NEXT FROM updatestats INTO @tableSchema, @tableName

WHILE (@@FETCH_STATUS = 0)
BEGIN
   SET @Statement = 'UPDATE STATISTICS '  + '[' + @tableSchema + ']' + '.' + '[' + @tableName + ']' + ' WITH F
   PRINT @Statement -- comment this print statement to prevent it from printing whenever you are ready to exec
   --EXEC sp_executesql @Statement -- remove the comment on the beginning of this line to run the commands
   FETCH NEXT FROM updatestats INTO @tableSchema, @tableName
END

CLOSE updatestats
DEALLOCATE updatestats

SET NOCOUNT OFF

-- clear database procedure cache for the database after updating statistics
-- to force the creation on new, more accurate execution plans
ALTER DATABASE SCOPED CONFIGURATION CLEAR PROCEDURE_CACHE
```

Update statistics for individual objects

This is helpful if only a single table needs to be updated or its sampling ratio changed. It also allows you to deactivate the auto-update functionality by requesting the `NORECOMPUTE` option.

```
-- Update all statistics on a table using the default sampling ratio
UPDATE STATISTICS Sales.SalesOrderDetail;

-- Update statistics for a specific index
UPDATE STATISTICS Sales.SalesOrderDetail AK_SalesOrderDetail_rowguid;

-- Update statistics by using 50 percent sampling
UPDATE STATISTICS Production.Product(Products) WITH SAMPLE 50 PERCENT;

-- Update statistics by using FULLSCAN to force 100% scan of all rows in the "AK_Product_Name" index
UPDATE STATISTICS Production.Product(AK_Product_Name) WITH FULLSCAN;

-- Update statistics by using NORECOMPUTE to turn off automatic update statistics
-- to turn it on again, execute the same without the RECOMPUTE option
UPDATE STATISTICS Production.Product WITH NORECOMPUTE;
```

Create statistics on a filtered subset of data

If major, important queries select from a well-defined subset of rows, and that subset of rows has a unique data distribution, then filtered statistics can improve the query execution plans.

If for example one part of the application only works with certain product category IDs, you can create dedicated statistics for this scenario:

```
-- Create filtered statistics
CREATE STATISTICS Sub_1_2_3_Weights ON Production.Product (Weight)
WHERE ProductSubcategoryID IN (1,2,3);
```

More Information

The following Statistics limitations exist when creating or rebuilding indexes:

- When an index is created or rebuilt, statistics are created or updated by scanning all the rows in the table, which is equivalent to using the FULLSCAN clause in CREATE STATISTICS or UPDATE STATISTICS. However, when a partitioned index is created or rebuilt, statistics are not created or updated by scanning all rows in the table. Instead, the default sampling ratio is used. To create or update statistics on partitioned indexes by scanning all rows in the table, use CREATE STATISTICS or UPDATE STATISTICS with the FULLSCAN clause.
- Similarly, when the index creation or rebuild operation is resumable, statistics are created or updated with the default sampling ratio. If statistics were created or last updated with the PERSIST_SAMPLE_PERCENT clause set to ON, resumable index operations use the persisted sampling ratio to create or update statistics.
- When an index is reorganized, statistics are not updated.

In these scenarios, it might be necessary to run an `UPDATE STATISTICS ... WITH FULLSCAN` after such index operations.

Public Doc Reference

- UPDATE STATISTICS (Transact-SQL)
- Statistics
- How to maintain Azure SQL Indexes and Statistics

How good have you found this content?

🙂 🙁