

TABLE VARIABLES AND TEMP TABLES WHAT YOU NEED TO KNOW!

WAYNE SHEFFIELD
PRINCIPAL CONSULTANT
SQL SOLUTIONS GROUP



SQL Server® 2008

Twitter: @DBAWayne
LinkedIn: <http://www.linkedin.com/in/WayneSheffield>
Blog: <http://blog.waynesheffield.com/wayne>

- What are Table Variables?
- What are Temporary Tables?
 - Local & Global
- Investigate myths.
- Changes in SQL Server 2012, 2014 and Windows Azure SQL Database (SQL Azure)

- Presentation and all demo scripts are available for download.
- For the purpose of this presentation, Session = Connection

What do you think?

- Do you prefer to use table variables?
- Do you prefer to use temporary tables?
- Can table variables have any indexes?
- Are table variables a memory-only process?
- Can temporary tables be in memory only?
- Are table variables logged?
- In which version of SQL Server were Table Variables added?

Introduced in SQL Server 2000.

```
DECLARE @TestTable TABLE (
    RowID      INTEGER IDENTITY          PRIMARY KEY CLUSTERED,
    SSN        CHAR(9) NOT NULL         UNIQUE,
    Age        TINYINT NULL             CHECK (Age > 18),
    rowguid    UNIQUEIDENTIFIER NOT NULL DEFAULT(NEWID()));
```

FOREIGN KEY constraints are not allowed.

- Explicit transaction rollbacks do not affect Table Variables.

	ID	SomeCol
1	0	Row1
2	1	Row2
3	2	Row3

```
DECLARE @TransactionTest TABLE (  
    ID          INTEGER PRIMARY KEY,  
    SomeCol    VARCHAR(20));
```

```
INSERT INTO @TransactionTest (ID, SomeCol) VALUES (0, 'Row1');  
INSERT INTO @TransactionTest (ID, SomeCol) VALUES (1, 'Row2');
```

```
BEGIN TRANSACTION;  
INSERT INTO @TransactionTest (ID, SomeCol) VALUES (2, 'Row3');  
ROLLBACK TRANSACTION;  
SELECT * FROM @TransactionTest;
```

- Transactions: duration of statement
 - less locking and logging resources.
- Fewer stored procedure recompilations
 - compared to temporary tables.

- Multi-Statement Table-Valued Functions
 - Must return a table variable
- Stored procedure parameter (SQL 2008).

- SELECT <ColumnList> INTO @TableVariable1
- In SQL 2000: INSERT INTO @TableVariable1
EXECUTE stored_procedure
- SET @TableVariable1 = @TableVariable2
- TRUNCATE @TableVariable1
- SET IDENTITY_INSERT @TableVariable1
- Named Constraints
- UDFs referenced by check constraints, default values or computed columns

Table Variables:

```

DECLARE @TestTable TABLE (
    RowID      INTEGER IDENTITY
    SSN        CHAR(9) NOT NULL
    Age        TINYINT NULL
    rowguid    UNIQUEIDENTIFIER NOT NULL DEFAULT(NEWID()));
PRIMARY KEY CLUSTERED,
UNIQUE,
CHECK (Age > 18),

```

```
declare @MyTableVariable TABLE (RowID int PRIMARY KEY (NON)CLUSTERED);
```

```
declare @temp TABLE (
    RowID int NOT NULL,
    ColA int NOT NULL,
    ColB char(1) UNIQUE (NON)CLUSTERED,
    PRIMARY KEY (NON)CLUSTERED (RowID, ColA));
```



```
declare @temp TABLE (
    RowID INT NOT NULL,
    ColA INT NOT NULL,
    ColB char(1),
    ColC char(1),
    ColD char(1),
    PRIMARY KEY (NON)CLUSTERED (RowID, ColA),
    UNIQUE (NON)CLUSTERED (ColB, ColC),
    UNIQUE (NON)CLUSTERED (ColC, ColD));
```

- No Statistics



- Optimizer: table variables have one row.
- OPTION RECOMPILE
 - No histogram.
 - Specified for each query.
- SQL 2012 SP2 – TF 2453

Collation: String collation from the current database.


Parallelism: SELECT statements only.

Referencing a table variable:

- In the FROM clause, by name
- Outside the FROM clause, by alias

Not supported in the SQL Server optimizer's cost-based reasoning model. (Affects queries with joins, parallelism decisions, index selection choices.)

Any questions about Table Variables?

- A temporary table has the same physical makeup as any other table created with the CREATE TABLE statement, except that:
- Name is limited to 116 characters. 
 - Identify different temporary tables created by different sessions at the same time with the same name.
 - A hexadecimal suffix is appended to the end of the name.

- Local temporary tables (created with a single “#”) are
 - Visible in the current connection, from the point the table is created.
 - Inside nested stored procedures (or dynamic sql) called after the table was created.
- Global temporary tables (created with a double “##”) are visible to all sessions.


- Temporary tables will be dropped when:
 - DROP TABLE.
 - Stored procedure finishes.
 - Session is closed.
- Global temporary tables are dropped when:
 - The session that created it ends, and all other sessions have stopped referencing it.
 - This referencing is performed at the statement level – NOT at the batch or session level.



- Table-Valued Function.
- Stored procedure parameters.
- Foreign Key constraints.
- Columns with UDDTs
- Columns with XML schema collection

How to add to tempdb.

- Startup stored procedure
- Model database

- Temporary tables can have named constraints and indexes. 
- Duplicate constraint name error
 - Build constraint without a name
 - Creates a system-assigned name

Temporary tables can be created with:

- CREATE TABLE
- SELECT <ColumnList> INTO #table

Temporary tables can be populated with:


- INSERT statement (INSERT INTO VALUES / SELECT / EXECUTE stored_procedure)
- SELECT <ColumnList> INTO #table

- IF EXISTS (SELECT 1 FROM tempdb.sys.tables WHERE name LIKE '#TestTable%')
- IF EXISTS (SELECT * FROM tempdb.INFORMATION_SCHEMA.Tables WHERE Table_Name Like '#TestTable%')
- IF EXISTS (SELECT * FROM #TestTable)
- IF EXISTS (SELECT * FROM sys.objects WHERE NAME = '#TestTable')

Why these methods of checking for #temp table existence don't work:

Local temporary tables (created with a single “#”) **are visible in the current connection**, from the point the table is created and inside nested stored procedures called after the table was created.

```
IF OBJECT_ID('tempdb..#TestTable') > 0
```

- In SQL Server, temporary tables are created with positive object_id values.
- However, SQL Server 2012 creates local temporary tables with negative object_id values. 
- Therefore, the best method to use is:

```
IF OBJECT_ID('tempdb..#TestTable') IS NOT NULL
```

Collation: Strings from tempdb

**Parallelism: SELECT, INSERT, UPDATE,
DELETE statements.**

- Table variables cannot have any indexes.

BUSTED

- Table variables don't participate in transactions, therefore nothing is written to the transaction log when changes are made to them.

BUSTED

- Table variables reside in memory only.

BUSTED

Myths – Confirm or bust!

- Temporary tables are physical only.
- <http://support.microsoft.com/default.aspx?scid=kb;en-us;305977&Product=sql2k> - “A table variable is not a memory-only structure. Because a table variable might hold more data than can fit in memory, it has to have a place on disk to store data. Table variables are created in the tempdb database similar to temporary tables. If memory is available, both table variables and temporary tables are created and processed while in memory (data cache).”

BUSTED

- BOL: “Temporary tables cannot be partitioned.”

BUSTED

Summary of differences

Feature	Table Variables	Temporary Tables
Scope	Current batch	Current session, nested stored procedures. Global: all sessions.
Usage	UDFs, Stored Procedures, Triggers, Batches.	Stored Procedures, Triggers, Batches.
Creation	DECLARE statement only.	CREATE TABLE statement. SELECT INTO statement.
Table name	Maximum 128 characters.	Maximum 116 characters (#); 128 (##).
Column data types	Can use user-defined data types. Can use XML collections.	User-defined data types and XML collections must be in tempdb to use.
Collation	String columns inherit collation from current database.	String columns inherit collation from tempdb database.

Summary of differences

Feature	Table Variables	Temporary Tables
Indexes	Can only have indexes that are automatically created as part of the DECLARE statement.	Indexes can be added after the table has been created.
Data insertion	INSERT statement (SQL 2000: cannot use INSERT/EXEC).	INSERT statement, including INSERT/EXEC. SELECT INTO statement.
Constraints	PRIMARY KEY, UNIQUE, NULL, CHECK & DEFAULT, but they must be incorporated with the creation of the table in the DECLARE statement. FOREIGN KEY not allowed.	PRIMARY KEY, UNIQUE, NULL, CHECK & DEFAULT. Can be part of the CREATE TABLE statement, or can be added after the table has been created. FOREIGN KEY not allowed.

Summary of differences

Feature	Table Variables	Temporary Tables
Truncate table	Not allowed.	Allowed.
Parallelism	SELECT statement only.	SELECT, INSERT, UPDATE & DELETE statements.
Insert explicit values into identity columns (SET IDENTITY_INSERT).	The SET IDENTITY_INSERT statement is not supported.	The SET IDENTITY_INSERT statement is supported.
Stored procedure recompilations	Not applicable.	Creating temp table and data inserts can cause stored procedure recompilations.
Implicit Transactions	Last only for length of update against the table variable. Uses less resources than temp tables.	Last for the length of the transaction. Uses more resources than table variables.

Summary of differences

Feature	Table Variables	Temporary Tables
Destruction	Automatically at the end of the batch.	Explicitly with DROP TABLE statement. Automatically when session ends. (Global: also when there are no other sessions with active statements using table.)
Rollbacks (Explicit ROLLBACK TRANSACTION)	Not affected (Data not rolled back).	Affected (Data is rolled back).
Statistics	Optimizer cannot create any statistics on columns, so it treats a table variable as having 1 record when creating execution plans.	Optimizer can create statistics on columns. Uses actual row count for generation execution plan.

Summary of differences

Feature	Table Variables	Temporary Tables
Pass as parameter to stored procedures	SQL 2008+ only; with predefined user-defined table type.	Not allowed to pass, but they are still in scope to nested procedures.
Explicitly named objects (constraints).	Not allowed.	Allowed, but be aware of multi-user issues.
Dynamic SQL	Must declare table variable inside the dynamic SQL.	Can use temporary tables created prior to calling the dynamic SQL.

- Object_id values are negative for local temporary tables.
- If temporary table is made in a “contained” database:
 - Collation is that of the database; not that of the tempdb database.
 - Temporary Tables may not refer to User Defined Data Types, XML Schema Collections, or User Defined Functions.
 - Named constraints cannot be used.
- SP2 – TF 2453 allows Query Optimizer to see row count on table variables involved in a JOIN condition.

- What's different about this table variable?

```
DECLARE @test TABLE (  
    col1 INTEGER PRIMARY KEY,  
    col2 CHAR(10) INDEX ix_1 (col2, col3),  
    col3 VARCHAR(100));
```

- What's different about this table variable?

```
DECLARE @test TABLE (  
    col1 INTEGER PRIMARY KEY,  
    col2 CHAR(10) INDEX ix_1 (col2, col3),  
    col3 VARCHAR(100));
```

- Inline specification of clustered and non-clustered indexes.
 - Allowed for all tables, including table variables.
 - Equivalent to creating a table, then issuing a CREATE INDEX statement.
 - Not supported:
 - Included columns
 - Filter conditions

- In-Memory OLTP
 - Supports Memory-Optimized Table Type.
 - Can create table variable from this table type.
 - Does not support temporary tables.

```
CREATE TYPE dbo.imTV AS TABLE (  
    Col1 INTEGER NOT NULL,  
    INDEX ix_imTV1 HASH(Col1) WITH (BUCKET_COUNT=8)  
) WITH (MEMORY_OPTIMIZED=ON);  
GO  
DECLARE @imTV dbo.imTV;
```

Windows Azure SQL Database

- Cannot refer to User Defined Data Types.
- Cannot refer to User Defined Functions.
- Cannot use Global Temporary Tables.

- Eliminate repeated use of a sub-query or CTE.
- “Pre-aggregation” or interim calculation storage (“Divide’n’Conquer” techniques).
- Performing running totals (aka “quirky update”).
- Clustered index needed on a different set of columns than what the base table is using.
- Data access to remote servers - especially non-SQL Servers (Oracle, MySQL, etc.)
- “Staging Table” or “Pre-validation Table”.

- Required for multi-statement Table-Valued Functions.
- Required when passing a table to a stored procedure.
- Eliminating excessive and/or long-running procedure recompilations.
- Utilize "no data rollback on explicit ROLLBACK TRANSACTION statements" feature.
- Consider when:
 - No JOIN or WHERE clauses against the table variable.
 - Low number of rows

- Desired visibility to multiple sessions (global) or nested procedures (local or global, possible concurrency issues when global).
- Creation of Non-unique indexes desired.
- Post-table creation DDL operations on table are needed.
- Need for SQL to build statistics and to otherwise be aware of data distribution in columns.
- You want to increase ease of current and future troubleshooting as compared to Table Variables.

When will temporary storage be written to disk vs. kept in memory?

"It depends" on:

- 1) Size of data.
- 2) Amount of server memory available.

INF: Frequently Asked Questions - SQL Server 2000 - Table Variables	http://support.microsoft.com/default.aspx?scid=kb;en-us;305977&Product=sql2k
T-SQL BOL (SQL 2000), table data type	http://msdn.microsoft.com/en-us/library/aa260638(SQL.80).aspx
T-SQL BOL (SQL 2008), Declare @local_variable	http://msdn.microsoft.com/en-us/library/ms188927.aspx
T-SQL BOL (SQL 2008), CREATE TABLE	http://msdn.microsoft.com/en-us/library/ms174979.aspx
Table-Valued Parameters (Database Engine)	http://msdn.microsoft.com/en-us/library/bb510489.aspx
Troubleshooting stored procedure recompilation	http://support.microsoft.com/kb/243586/
Local Temporary Tables and Table Variables	http://www.sqlservercentral.com/articles/Table+Variables/63878/
Startup stored procedure	http://msdn.microsoft.com/en-us/library/ms191129.aspx
Data Definition Language (DDL)	http://msdn.microsoft.com/en-us/library/aa905918(SQL.80).aspx
Gail Shaw: A Trio of Table Variables	http://sqlinthewild.co.za/index.php/2010/10/12/a-trio-of-table-variables/
Dave Ballentyne: Mythbusting Table Variables have no Statistics	http://sqlblogcasts.com/blogs/sqlandthelike/archive/2011/12/02/mythbusting-table-variables-have-no-statistics.aspx
T-SQL BOL (SQL Server 2012), Modified Features (Contained Database)	http://msdn.microsoft.com/en-us/library/ff929143(v=SQL.110).aspx
T-SQL BOL (SQL Server 2014), What's New (Transact-SQL Enhancements)	http://msdn.microsoft.com/en-us/library/bb510411(v=sql.120).aspx#TSQL
T-SQL BOL (SQL Server 2012), table	http://msdn.microsoft.com/en-us/library/ms175010.aspx
Can you partition a temporary table?	http://jasonbrimhall.info/2014/05/23/can-you-partition-a-temporary-table/



<http://www.linkedin.com/in/WayneSheffield>



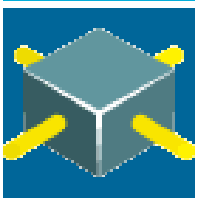
@DBAWayne



blog.waynesheffield.com/wayne



<http://www.sqlsaturday.com/<Event#>/schedule.aspx>
<http://bit.ly/TableVars>



http://www.sqlservercentral.com/Authors/Articles/Wayne_Sheffield/380617/