

# OUTPUT Clause and Derived Tables



Presented by Steve Stedman and Aaron Buma

# OUTPUT Clause and ~~Derived Tables~~



Presented by Steve Stedman and Aaron Buma

# OUTPUT Clause Correlated Subqueries and Subquery Extensions



Presented by Steve Stedman and Aaron Buma

# Welcome

- Welcome to all those joining us remotely.
  - For any questions via Google On Air Broadcasts, we will address most of these at the end of the training.
- Training provided by Emergency Reporting
  - <http://EmergencyReporting.com>

# Live Broadcast

- Using Google On Air Broadcasts
  - There is about a 40 to 50 second delay from live to what you see.
  - We are still learning how to properly use Google On Air Broadcasts. Please be patient.
  - Session will be available on my YouTube Channel about an hour after it the presentation ends.
    - <http://SteveStedman.com/YouTube>

# Questions

- We will have time for questions at the end of the session.
- Q&A available via Google On Air Hangout panel. Click the 3x3 grid icon near the top right, then select Q&A to see what people are asking, or to ask your own question.
- When you ask a question, it shows up for us about 40 to 50 seconds delayed.

# Agenda

- Correlated Subqueries and Subquery Extensions
  - Correlated Sub Queries
  - Sub Query Extensions (ANY, ALL, SOME)
  - Exists
- OUTPUT Clause

# Correlated Subqueries and Subquery Extensions



Presented by Aaron Buma

# Correlating Subqueries Together

- Also known as derived tables, a correlated subquery is a nested query
- It can be in the selected columns, in a join or as a filter in the where clause
- Subqueries return result sets to their outer query
- You can nest up to 32 queries together

## SubQuery Extensions

- Used in the where clause to filter on the query based on a single-column correlated query
- Operators: ALL, ANY, SOME, EXISTS

# ALL

- Syntax: 'WHERE 5 <= ALL (SELECT col<sub>1</sub> FROM..)
- All rows of col<sub>1</sub> must match the condition ( col<sub>1</sub> <= 5 ) in order for that correlated condition to be true

# ANY

- Syntax: 'WHERE  $5 \leq$  ANY (SELECT col<sub>1</sub> FROM..)
- Any of the rows of col<sub>1</sub> must match the condition ( col<sub>1</sub>  $\leq$  5 ) in order for that correlated condition to be true

# SOME

- Syntax: 'WHERE  $5 \leq$  SOME (SELECT col1 FROM..)
- Is the ISO standard of ANY, they will return the same results

# EXISTS

- Syntax: 'WHERE EXISTS (SELECT column FROM..)
- A hardcoded value (1,'x') can be used instead of a column because the columns data is not actually returned
- The first match resolves the condition as TRUE and the parser stops evaluating for the correlated value
- Is more performant than ANY or SOME because it doesn't have to process the all of the correlated results

# Demo

Basic Subqueries and  
Correlated Subqueries

# Demo

Subquery Extensions:  
ALL, ANY, SOME and EXISTS

# Review

- Correlated Query – Filtering a derived table or subquery based on a column in the main query.
- Using the 'ALL' operator returns TRUE only if all rows fit the condition
- Using 'ANY' or 'SOME' returns TRUE if one or more of the subquery rows match condition
- Using 'EXISTS' has the same results as 'ANY' or 'SOME' does not return data and will stop processing at first match with the condition, returning TRUE

# Up next OUTPUT Clause



**EMERGENCY  
REPORTING™**

This training is sponsored by Emergency Reporting.  
Visit <http://EmergencyReporting.com> for more details.

# OUTPUT Clause



**EMERGENCY  
REPORTING™**

Presented by Steve Stedman

# What is the OUTPUT clause

- A way to get a result set on queries that don't normally return a result set.
  - For instance INSERT, DELETE, UPDATE, and MERGE.

# OUTPUT Clause Applies To

- INSERT
- DELETE
- UPDATE
- MERGE
  - Using output with MERGE will be covered when we cover the MERGE Statement

# Availability of the OUTPUT Clause

- SQL Server 2005 and newer
- Available on all editions
  - Express
  - Standard
  - Enterprise
  - Developer
  - Azure

# INSERT

OUTPUT INSERTED.\*

- Displays the entire row that was just inserted.

# Returning the IDENTITY inserted without the OUTPUT clause

- @@IDENTITY returns the last identity value generated for any table in the current session, across all scopes. **You need to be careful here**, since it's across scopes. You could get a value from a trigger, instead of your current statement.
- SCOPE\_IDENTITY returns the last identity value generated for any table in the current session and the current scope. **Generally what you want to use.**
- IDENT\_CURRENT returns the last identity value generated for a specific table in any session and any scope. This lets you specify which table you want the value from, in case the two above aren't quite what you need (**very rare**). You could use this if you want to get the current IDENTITY value for a table that you have not inserted a record into.

# IDENTITY values of multi-row INSERT

- These only return one:
  - @@IDENTITY
  - SCOPE\_IDENTITY
  - IDENT\_CURRENT
- **OUTPUT INSERTED.\***
  - Will return all inserted.

# Demo

OUTPUT Clause – With an INSERT  
Statement

# UPDATE

[DELETED].\*, [INSERTED].\*

- DELETED returns the before value.
- INSERTED returns the after value.

# Demo

OUTPUT Clause – With an UPDATE  
Statement

# DELETE

[DELETED].\*

- DELETE returns the row that was deleted.

# Demo

OUTPUT Clause – With an DELETE  
Statement

# Demo

Auditing with the OUTPUT Clause

# OUTPUT Clause Gotchas

- Get results, even if the transaction fails.
- Columns returned from OUTPUT reflect the data as it exists after the INSERT, UPDATE, or DELETE statement has completed but before triggers are executed.

# Not supported in the following:

- DML statements that reference local partitioned views, distributed partitioned views, or remote tables.
- INSERT statements that contain an EXECUTE statement.
- A user-defined function cannot be created if it contains an OUTPUT INTO clause that has a table as its target.

# Review

- INSERT
- DELETE
- UPDATE
- MERGE
  - Using output with MERGE will be covered when we cover the MERGE Statement

# Quiz 1:

- All of the following are true, so which is better to use: ANY, SOME or EXISTS?
  - a) ANY because it comes first alphabetically
  - b) SOME because it is the ISO standard of ANY
  - c) EXISTS because it is more performant than ANY or SOME

# Quiz 2:

- Which is better to track or log changes to a table, the OUTPUT clause or a TRIGGER?
  - a) The OUTPUT clause has less overhead, therefore it is better.
  - b) A TRIGGER is the best option since it guarantees the information will be logged.
  - c) It depends on the specific environment, and your specific needs. In some cases the OUTPUT clause is better, and in other cases the TRIGGER is the best option.

# Any Questions?

- OUTPUT Clause
- Derived Table Queries (ie Subqueries)
  - Correlated Sub Queries
  - Sub Query Extensions (ANY, ALL, SOME)
  - Exists

# For More Information

- Visit <http://EmergencyReporting.com> to find out more about Emergency Reporting.
- Aaron on the web
  - <http://AaronBuma.com>
  - Twitter: @AaronDBuma
- Steve on the web
  - <http://SteveStedman.com>
  - twitter: @SqlEmt

# Tune in next week

- Thursday 2/17 at 9:00am (pacific time).
- Topics
  - Data Recursion
    - Self JOIN
    - Range Heirarchy
    - Recursive CTE
  - Merge Statement