# 09 | Modifying Data

Graeme Malcolm | Senior Content Developer, Microsoft
Geoff Allix | Principal Technologist, Content Master

# Module Overview

- Inserting Data into Tables
- Generating Identifiers
- Updating Data in Tables
- Deleting Data in Tables

# Inserting Data into Tables

- INSERT…VALUES
  - Inserts explicit values
  - You can omit identity columns, columns that allow NULL, and columns with default constraints.
  - You can also explicitly specify NULL and DEFAULT
- INSERT…SELECT / INSERT…EXEC
  - Inserts the results returned by the query or stored procedure into an existing table
- SELECT…INTO
  - Creates a new table from the results of a query
  - Not currently supported in Azure SQL Database

# Generating Identifiers
## Using Identity Columns

- IDENTITY property of a column generates sequential numbers automatically for insertion into a table
  - Can specify optional seed and increment values
  - Use system variables and functions to return last inserted identity:
    - **@@IDENTITY**: The last identity generated in the session
    - **SCOPE_IDENTITY()**: The last identity generated in the current scope
    - **IDENT_CURRENT('<*table_name*>')**: The last identity inserted into a table

```
INSERT INTO Sales.Orders (CustomerID, Amount)
VALUES
(12, 2056.99);
…
SELECT SCOPE_IDENTITY() AS OrderID;
```

# Generating Identifiers
## Using Sequences

- Sequences are objects that generate sequential numbers
  - Supported in SQL Server 2012 and later
  - Exist independently of tables, so offer greater flexibility than Identity
  - Use SELECT NEXT VALUE FOR to retrieve the next sequential number
    - Can be set as the default value for a column

```
CREATE SEQUENCE Sales.OrderNumbers AS INT
STARTS WITH 1 INCREMENT BY 1;
…
SELECT NEXT VALUE FOR Sales.OrderNumbers;
```

# DEMO

Inserting Data into Tables

# Updating Data in a Table
## The UPDATE Statement

- Updates all rows in a table or view
  - Set can be filtered with a WHERE clause
  - Set can be defined with a FROM clause
- Only columns specified in the SET clause are modified

```
UPDATE Production.Product
SET unitprice = (unitprice * 1.04)
WHERE categoryid =  1 AND discontinued = 0;
```

# Updating Data in a Table
## The MERGE Statement

- MERGE modifies data based on a condition
  - When the source  matches the target
  - When the source has no match in the target
  - When the target has no match in the source

```
MERGE INTO Production.Products as P
        USING Production.ProductsStaging as S
        ON P.ProductID=S.ProductID
WHEN MATCHED THEN
        UPDATE SET
        P.UnitPrice = S.UnitPrice, P.Discontinued=S.Discontinued
WHEN NOT MATCHED THEN
        INSERT (ProductName, CategoryID, UnitPrice, Discontinued)
        VALUES (S.ProductName, S.CategoryID, S.UnitPrice, S.Discontinued);
```

# Deleting Data From a Table

- DELETE without a WHERE clause deletes all rows
  - Use a WHERE clause to delete specific rows

```
DELETE FROM Sales.OrderDetails
WHERE orderid = 10248;
```

- TRUNCATE TABLE clears the entire table
  - Storage physically deallocated, rows not individually removed
  - Minimally logged
  - Can be rolled back if TRUNCATE issued within a transaction
  - TRUNCATE TABLE Will fail if the table is referenced by a foreign key constraint in another table

# DEMO

Updating and Deleting Data

# Modifying Data

- Inserting Data into Tables
- Generating Identifiers
- Updating Data in Tables
- Deleting Data in Tables

- Lab: Modifying Data

Microsoft

©2014 Microsoft Corporation. All rights reserved. Microsoft, Windows, Office, Azure, System Center, Dynamics and other product names are or may be registered trademarks and/or trademarks in the U.S. and/or other countries. The information herein is for informational purposes only and represents the current view of Microsoft Corporation as of the date of this presentation. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information provided after the date of this presentation. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS PRESENTATION.