06 | Using Subqueries and APPLY

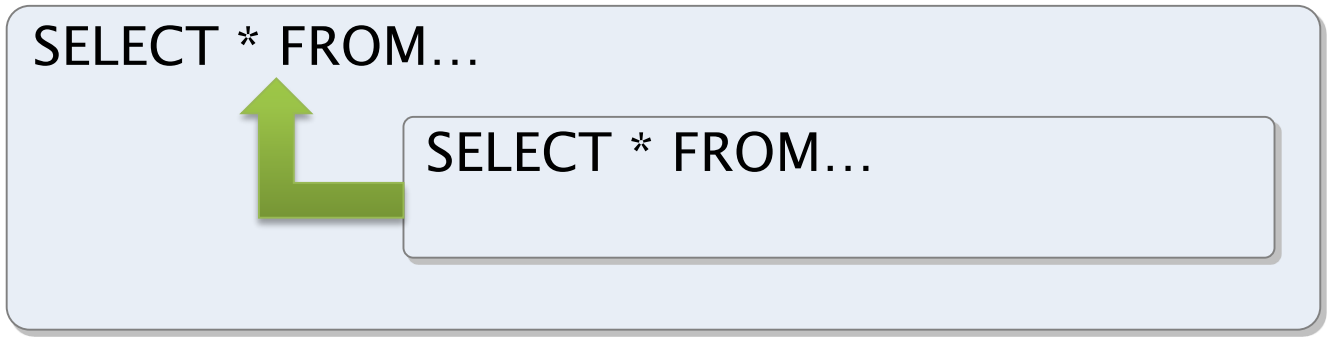
Graeme Malcolm | Senior Content Developer, Microsoft
Geoff Allix | Principal Technologist, Content Master

Module Overview

- Introduction to Subqueries
- Scalar or Multi-Valued?
- Self-Contained or Correlated?
- Using APPLY with Table-Valued Functions

Introduction to Subqueries

- Subqueries are nested queries: queries within queries
- Results of inner query passed to outer query
  - Inner query acts like an expression from perspective of outer query



Scalar or Multi-Valued?

- Scalar subquery returns single value to outer query
  - Can be used anywhere single-valued expression is used: SELECT, WHERE, and so on
- Multi-valued subquery returns multiple values as a single column set to the outer query
  - Used with IN predicate

```
SELECT orderid, productid, unitprice, qty
FROM Sales.OrderDetails
WHERE orderid =
      (SELECT MAX(orderid) AS lastorder
      FROM Sales.Orders);
```

```
SELECT custid, orderid
FROM Sales.orders
WHERE custid IN (
      SELECT custid
      FROM Sales.Customers
      WHERE countryregion = N'Mexico');
```

DEMO

Using Subqueries

Self-Contained or Correlated?

- Most subqueries are self-contained and have no connection with the outer query other than passing it results
- Correlated subqueries refer to elements of tables used in outer query
  - Dependent on outer query, cannot be executed separately
  - Behaves as if inner query is executed once per outer row
  - May return scalar value or multiple values

```
SELECT orderid, empid, orderdate
FROM Sales.Orders AS O1
WHERE orderdate = (SELECT MAX(orderdate)
                   FROM Sales.Orders AS O2
                   WHERE O2.empid = O1.empid)
ORDER BY empid, orderdate;
```

DEMO

Creating a Correlated Subquery

Using APPLY with Table-Valued Functions

- CROSS APPLY applies the right table expression to each row in left table
  - Conceptually similar to CROSS JOIN between two tables but can correlate data between sources

```
SELECT S.supplierid, s.companyname, P.productid,  P.productname, P.unitprice
FROM Production.Suppliers AS S
CROSS APPLY dbo.fn_TopProductsByShipper(S.supplierid) AS P
```

- OUTER APPLY adds rows for those with NULL in columns for right table
  - Conceptually similar to LEFT OUTER JOIN between two tables

DEMO

Using APPLY with Table-Valued Functions

Using Subqueries and APPLY

- Introduction to Subqueries
- Scalar or Multi-Valued?
- Self-Contained or Correlated?
- Using APPLY with Table-Valued Functions

- Lab: Using Subqueries and APPLY

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.