Module 10

Using Subqueries

Module Overview

- Writing Self-Contained Subqueries
- Writing Correlated Subqueries
- Using the EXISTS Predicate with Subqueries

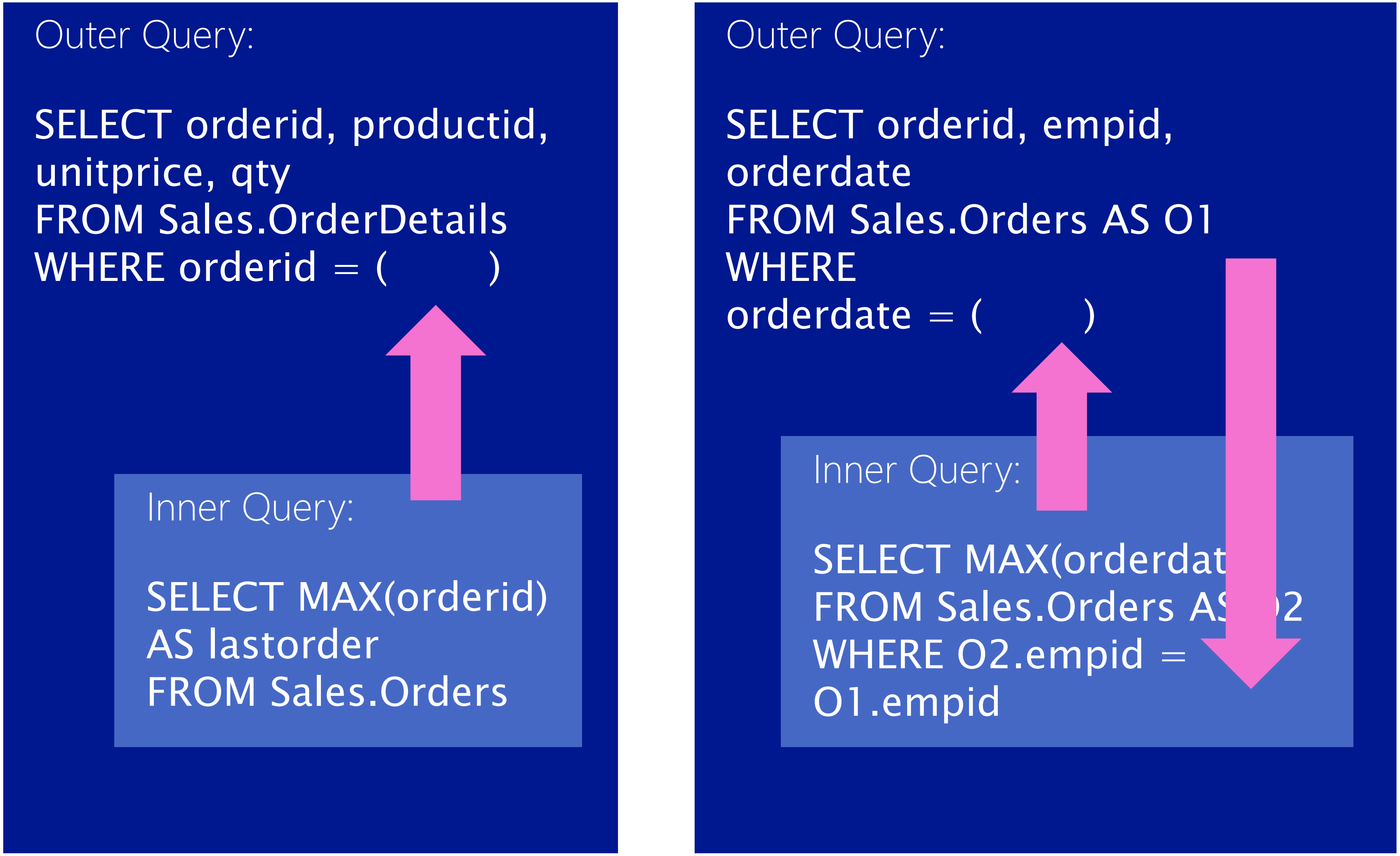
Lesson 1: Writing Self-Contained Subqueries

- Working with Subqueries
- Writing Scalar Subqueries
- Writing Multi-Valued Subqueries
- Demonstration: Writing Self-Contained Subqueries

Working with 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
- Subqueries can be self-contained or correlated
  - Self-contained subqueries have no dependency on outer query
  - Correlated subqueries depend on values from outer query
- Subqueries can be scalar, multi-valued, or table-valued

Comparing Self-Contained and Correlated Subqueries



Self-Contained Subquery:

Outer Query:

```
SELECT orderid, productid,
unitprice, qty
FROM Sales.OrderDetails
WHERE orderid = (        )
```

Inner Query:

```
SELECT MAX(orderid)
AS lastorder
FROM Sales.Orders
```

Correlated Subquery:

Outer Query:

```
SELECT orderid, empid,
orderdate
FROM Sales.Orders AS O1
WHERE
orderdate = (        )
```

Inner Query:

```
SELECT MAX(orderdat
FROM Sales.Orders AS O2
WHERE O2.empid =
O1.empid
```

Writing Scalar Subqueries

- Scalar subquery returns single value to outer query
- Can be used anywhere single-valued expression is used: SELECT, WHERE, and so on

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

- If inner query returns an empty set, result is converted to NULL
- Construction of outer query determines whether inner query must return a single value

Writing Multi-Valued Subqueries

- Multi-valued subquery returns multiple values as a single column set to the outer query
- Used with IN predicate
- If any value in the subquery result matches IN predicate expression, the predicate returns TRUE

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

- May also be expressed as a JOIN (test both for performance)

Demonstration: Writing Self-Contained Subqueries

In this demonstration, you will see how to:

- Write a nested subquery

Lesson 2: Writing Correlated Subqueries

- Working with Correlated Subqueries
- Writing Correlated Subqueries
- Demonstration: Writing Correlated Subqueries

Working with Correlated Subqueries

- Correlated subqueries refer to elements of tables used in outer query
- Dependent on outer query, cannot be executed separately
  - Harder to test than self-contained subqueries
- Behaves as if inner query is executed once per outer row
- May return scalar value or multiple values

```sql
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;
```

Writing Correlated Subqueries

- Write inner query to accept input value from outer query
- Write outer query to accept appropriate return result (scalar or multi-valued)
- Correlate queries by passing value from outer query to match argument in inner query

```sql
SELECT custid, orderid, orderdate
FROM Sales.Orders AS outerorders
WHERE orderdate =
            (SELECT MAX(orderdate)
             FROM Sales.Orders AS innerorders
             WHERE innerorders.custid = outerorders.custid)
ORDER BY custid;
```

Demonstration: Writing Correlated Subqueries

In this demonstration, you will see how to:

- Write a correlated subquery

Lesson 3: Using the EXISTS Predicate with Subqueries

- Working with EXISTS
- Writing Queries Using EXISTS with Subqueries
- Demonstration: Writing Subqueries Using EXISTS

Working with EXISTS

- When a subquery is used with the keyword EXISTS, it functions as an existence test
  - True or false only—no rows passed back to outer query
- EXISTS evaluates to TRUE or FALSE (not UNKNOWN)
  - If any rows are returned by the subquery, EXISTS returns TRUE
  - If no rows are returned, EXISTS returns FALSE
- Syntax:

```
WHERE [NOT] EXISTS (subquery)
```

Writing Queries Using EXISTS with Subqueries

- The keyword EXISTS does not follow a column name or other expression
- The SELECT list of a subquery introduced by EXISTS typically only uses an asterisk (*)

```
SELECT custid, companyname
FROM Sales.Customers AS c
WHERE EXISTS (
   SELECT *
   FROM Sales.Orders AS o
   WHERE c.custid=o.custid);
```

```
SELECT custid, companyname
FROM Sales.Customers AS c
WHERE NOT EXISTS (
   SELECT *
   FROM Sales.Orders AS o
   WHERE c.custid=o.custid);
```

Demonstration: Writing Subqueries Using EXISTS

In this demonstration, you will see how to:

- Write queries using EXISTS and NOT EXISTS

Lab: Using Subqueries

- Exercise 1: Writing Queries That Use Self-Contained Subqueries
- Exercise 2: Writing Queries That Use Scalar and Multiresult Subqueries
- Exercise 3: Writing Queries That Use Correlated Subqueries and an EXISTS Predicate

Logon Information
Virtual machine: **20761C-MIA-SQL**
User name: **AdventureWorks\Student**
Password: **Pa55w.rd**

Estimated Time: 60 minutes

Lab Scenario

As a business analyst for Adventure Works, you are writing reports using corporate databases stored in SQL Server. You have been handed a set of business requirements for data and will write T-SQL queries to retrieve the specified data from the databases. Due to the complexity of some of the requests, you will need to embed subqueries into your queries to return results in a single query.

Module Review and Takeaways

- Review Question(s)