---------------------------------------------------------------------

-- T-SQL Fundamentals Fourth Edition

-- Chapter 04 - Subqueries

-- © Itzik Ben-Gan

---------------------------------------------------------------------

---------------------------------------------------------------------

-- Self-Contained Subqueries

---------------------------------------------------------------------

---------------------------------------------------------------------

-- Scalar Subqueries

---------------------------------------------------------------------

-- Order with the maximum order ID

USE TSQLV6;

DECLARE @maxid AS INT = (SELECT MAX(orderid)

FROM Sales.Orders);

SELECT orderid, orderdate, empid, custid

FROM Sales.Orders

WHERE orderid = @maxid;

SELECT orderid, orderdate, empid, custid

FROM Sales.Orders

WHERE orderid = (SELECT MAX(O.orderid)

FROM Sales.Orders AS O);

-- Scalar subquery expected to return one value

SELECT orderid

FROM Sales.Orders

WHERE empid =

(SELECT E.empid

FROM HR.Employees AS E

WHERE E.lastname LIKE N'C%');

SELECT orderid

FROM Sales.Orders

WHERE empid =

(SELECT E.empid

FROM HR.Employees AS E

WHERE E.lastname LIKE N'D%');

SELECT orderid

FROM Sales.Orders

WHERE empid =

(SELECT E.empid

FROM HR.Employees AS E

WHERE E.lastname LIKE N'A%');

---------------------------------------------------------------------

-- Multi-Valued Subqueries

---------------------------------------------------------------------

SELECT orderid

FROM Sales.Orders

WHERE empid IN

(SELECT E.empid

FROM HR.Employees AS E

WHERE E.lastname LIKE N'D%');

SELECT O.orderid

FROM HR.Employees AS E

INNER JOIN Sales.Orders AS O

ON E.empid = O.empid

WHERE E.lastname LIKE N'D%';

-- Orders placed by US customers

SELECT custid, orderid, orderdate, empid

FROM Sales.Orders

WHERE custid IN

(SELECT C.custid

FROM Sales.Customers AS C

WHERE C.country = N'USA');

-- Customers who placed no orders

SELECT custid, companyname

FROM Sales.Customers

WHERE custid NOT IN

(SELECT O.custid

FROM Sales.Orders AS O);

-- Missing order IDs

USE TSQLV6;

DROP TABLE IF EXISTS dbo.Orders;

CREATE TABLE dbo.Orders(orderid INT NOT NULL CONSTRAINT PK\_Orders PRIMARY KEY);

INSERT INTO dbo.Orders(orderid)

SELECT orderid

FROM Sales.Orders

WHERE orderid % 2 = 0;

SELECT n

FROM dbo.Nums

WHERE n BETWEEN (SELECT MIN(O.orderid) FROM dbo.Orders AS O)

AND (SELECT MAX(O.orderid) FROM dbo.Orders AS O)

AND n NOT IN (SELECT O.orderid FROM dbo.Orders AS O);

-- CLeanup

DROP TABLE IF EXISTS dbo.Orders;

---------------------------------------------------------------------

-- Correlated Subqueries

---------------------------------------------------------------------

-- Orders with maximum order ID for each customer

-- Listing 4-1: Correlated Subquery

USE TSQLV6;

SELECT custid, orderid, orderdate, empid

FROM Sales.Orders AS O1

WHERE orderid =

(SELECT MAX(O2.orderid)

FROM Sales.Orders AS O2

WHERE O2.custid = O1.custid);

SELECT MAX(O2.orderid)

FROM Sales.Orders AS O2

WHERE O2.custid = 85;

-- Percentage of customer total

SELECT orderid, custid, val,

CAST(100. \* val / (SELECT SUM(O2.val)

FROM Sales.OrderValues AS O2

WHERE O2.custid = O1.custid)

AS NUMERIC(5,2)) AS pct

FROM Sales.OrderValues AS O1

ORDER BY custid, orderid;

---------------------------------------------------------------------

-- EXISTS

---------------------------------------------------------------------

-- Customers from Spain who placed orders

SELECT custid, companyname

FROM Sales.Customers AS C

WHERE country = N'Spain'

AND EXISTS

(SELECT \* FROM Sales.Orders AS O

WHERE O.custid = C.custid);

-- Customers from Spain who didn't place Orders

SELECT custid, companyname

FROM Sales.Customers AS C

WHERE country = N'Spain'

AND NOT EXISTS

(SELECT \* FROM Sales.Orders AS O

WHERE O.custid = C.custid);

---------------------------------------------------------------------

-- Returning "Previous" or "Next" Value

---------------------------------------------------------------------

SELECT orderid, orderdate, empid, custid,

(SELECT MAX(O2.orderid)

FROM Sales.Orders AS O2

WHERE O2.orderid < O1.orderid) AS prevorderid

FROM Sales.Orders AS O1;

SELECT orderid, orderdate, empid, custid,

(SELECT MIN(O2.orderid)

FROM Sales.Orders AS O2

WHERE O2.orderid > O1.orderid) AS nextorderid

FROM Sales.Orders AS O1;

---------------------------------------------------------------------

-- Running Aggregates

---------------------------------------------------------------------

SELECT orderyear, qty

FROM Sales.OrderTotalsByYear;

SELECT orderyear, qty,

(SELECT SUM(O2.qty)

FROM Sales.OrderTotalsByYear AS O2

WHERE O2.orderyear <= O1.orderyear) AS runqty

FROM Sales.OrderTotalsByYear AS O1

ORDER BY orderyear;

---------------------------------------------------------------------

-- Misbehaving Subqueries

---------------------------------------------------------------------

---------------------------------------------------------------------

-- NULL Trouble

---------------------------------------------------------------------

-- Customers who didn't place orders

-- Using NOT IN

SELECT custid, companyname

FROM Sales.Customers

WHERE custid NOT IN(SELECT O.custid

FROM Sales.Orders AS O);

-- Add a row to the Orders table with a NULL custid

INSERT INTO Sales.Orders

(custid, empid, orderdate, requireddate, shippeddate, shipperid,

freight, shipname, shipaddress, shipcity, shipregion,

shippostalcode, shipcountry)

VALUES(NULL, 1, '20220212', '20220212',

'20220212', 1, 123.00, N'abc', N'abc', N'abc',

N'abc', N'abc', N'abc');

-- Following returns an empty set

SELECT custid, companyname

FROM Sales.Customers

WHERE custid NOT IN(SELECT O.custid

FROM Sales.Orders AS O);

-- Exclude NULLs explicitly

SELECT custid, companyname

FROM Sales.Customers

WHERE custid NOT IN(SELECT O.custid

FROM Sales.Orders AS O

WHERE O.custid IS NOT NULL);

-- Using NOT EXISTS

SELECT custid, companyname

FROM Sales.Customers AS C

WHERE NOT EXISTS

(SELECT \*

FROM Sales.Orders AS O

WHERE O.custid = C.custid);

-- Cleanup

DELETE FROM Sales.Orders WHERE custid IS NULL;

---------------------------------------------------------------------

-- Substitution Error in a Subquery Column Name

---------------------------------------------------------------------

-- Create and populate table Sales.MyShippers

DROP TABLE IF EXISTS Sales.MyShippers;

CREATE TABLE Sales.MyShippers

(

shipper\_id INT NOT NULL,

companyname NVARCHAR(40) NOT NULL,

phone NVARCHAR(24) NOT NULL,

CONSTRAINT PK\_MyShippers PRIMARY KEY(shipper\_id)

);

INSERT INTO Sales.MyShippers(shipper\_id, companyname, phone)

VALUES(1, N'Shipper GVSUA', N'(503) 555-0137'),

(2, N'Shipper ETYNR', N'(425) 555-0136'),

(3, N'Shipper ZHISN', N'(415) 555-0138');

-- Shippers who shipped orders to customer 43

-- Bug

SELECT shipper\_id, companyname

FROM Sales.MyShippers

WHERE shipper\_id IN

(SELECT shipper\_id

FROM Sales.Orders

WHERE custid = 43);

-- The safe way using aliases, bug identified

SELECT shipper\_id, companyname

FROM Sales.MyShippers

WHERE shipper\_id IN

(SELECT O.shipper\_id

FROM Sales.Orders AS O

WHERE O.custid = 43);

-- Bug corrected

SELECT shipper\_id, companyname

FROM Sales.MyShippers

WHERE shipper\_id IN

(SELECT O.shipperid

FROM Sales.Orders AS O

WHERE O.custid = 43);

-- Cleanup

DROP TABLE IF EXISTS Sales.MyShippers;