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

-- Microsoft SQL Server T-SQL Fundamentals

-- Chapter 03 - Joins

-- � Itzik Ben-Gan

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

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

-- CROSS Joins

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

USE Northwinds2022TSQLV7;

-- SQL-92

SELECT C.CustomerId, E.EmployeeId

FROM Sales.Customer AS C

  CROSS JOIN HumanResources.Employee AS E;

-- SQL-89

SELECT C.CustomerId, E.EmployeeId

FROM Sales.Customer AS C, HumanResources.Employee AS E;

-- Self Cross-Join

SELECT

  E1.EmployeeId, E1.EmployeeFirstName, E1.EmployeeLastName,

  E2.EmployeeId, E2.EmployeeFirstName, E2.EmployeeLastName

FROM HumanResources.Employee AS E1

  CROSS JOIN HumanResources.Employee AS E2;

GO

-- All numbers from 1 - 1000

-- Auxiliary table of digits

USE Northwinds2022TSQLV7 ;

DROP TABLE IF EXISTS dbo.Digits;

CREATE TABLE dbo.Digits(digit INT NOT NULL PRIMARY KEY);

INSERT INTO dbo.Digits(digit)

  VALUES (0),(1),(2),(3),(4),(5),(6),(7),(8),(9);

SELECT digit FROM dbo.Digits;

GO

-- All numbers from 1 - 1000

SELECT D3.digit \* 100 + D2.digit \* 10 + D1.digit + 1 AS n

FROM         dbo.Digits AS D1

  CROSS JOIN dbo.Digits AS D2

  CROSS JOIN dbo.Digits AS D3

ORDER BY n;

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

-- INNER Joins

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

USE Northwinds2022TSQLV7 ;

-- SQL-92

SELECT E.EmployeeId, E.EmployeeFirstName, E.EmployeeLastName, O.OrderId

FROM HumanResources.Employee AS E

  INNER JOIN Sales.[Order] AS O

    ON E.EmployeeId = O.EmployeeId;

-- SQL-89

SELECT E.EmployeeId, E.EmployeeFirstName, E.EmployeeLastName, O.OrderId

FROM HumanResources.Employee AS E, Sales.[Order] AS O

WHERE E.EmployeeId = O.EmployeeId;

GO

-- Inner Join Safety

/\*

SELECT E.empid, E.firstname, E.lastname, O.orderid

FROM HR.Employees AS E

  INNER JOIN Sales.Orders AS O;

GO

\*/

SELECT E.EmployeeId, E.EmployeeFirstName, E.EmployeeLastName, O.orderid

FROM HumanResources.Employee AS E, Sales.[Order] AS O;

GO

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

-- More Join Examples

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

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

-- Composite Joins

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

-- Audit table for updates against OrderDetails

USE Northwinds2022TSQLV7 ;

DROP TABLE IF EXISTS Sales.OrderDetailsAudit;

CREATE TABLE Sales.OrderDetailsAudit

(

  lsn        INT NOT NULL IDENTITY,

  orderid    INT NOT NULL,

  productid  INT NOT NULL,

  dt         DATETIME NOT NULL,

  loginname  sysname NOT NULL,

  columnname sysname NOT NULL,

  oldval     SQL\_VARIANT,

  newval     SQL\_VARIANT,

  CONSTRAINT PK\_OrderDetailsAudit PRIMARY KEY(lsn),

  CONSTRAINT FK\_OrderDetailsAudit\_OrderDetails

    FOREIGN KEY(orderid, productid)

    REFERENCES Sales.OrderDetail(orderid, productid)

);

SELECT OD.OrderId, OD.productid, OD.Quantity,

  ODA.dt, ODA.loginname, ODA.oldval, ODA.newval

FROM Sales.OrderDetail AS OD

  INNER JOIN Sales.OrderDetailsAudit AS ODA

    ON OD.OrderId = ODA.OrderId

    AND OD.productid = ODA.productid

WHERE ODA.columnname = N'qty';

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

-- Non-Equi Joins

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

-- Unique pairs of employees

SELECT

  E1.EmployeeId, E1.EmployeeFirstName, E1.EmployeeLastName,

  E2.EmployeeId, E2.EmployeeFirstName, E2.EmployeeLastName

FROM HumanResources.Employee AS E1

  INNER JOIN HumanResources.Employee AS E2

    ON E1.EmployeeId < E2.EmployeeId;

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

-- Multi-Join Queries

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

SELECT

  C.CustomerId, C.CustomerCompanyName, O.orderid,

  OD.productid, OD.Quantity

FROM Sales.Customer AS C

  INNER JOIN Sales.[Order] AS O

    ON C.CustomerId = O.CustomerId

  INNER JOIN Sales.OrderDetail AS OD

    ON O.orderid = OD.orderid;

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

-- Fundamentals of Outer Joins

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

-- Customers with no orders

SELECT C.CustomerId, C.CustomerCompanyName

FROM Sales.Customer AS C

  LEFT OUTER JOIN Sales.[Order] AS O

    ON C.CustomerId = O.CustomerId

WHERE O.orderid IS NULL;

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

-- Beyond the Fundamentals of Outer Joins

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

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

-- Including Missing Values

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

SELECT DATEADD(day, n-1, CAST('20140101' AS DATE)) AS orderdate

FROM dbo.Nums

WHERE n <= DATEDIFF(day, '20140101', '20161231') + 1

ORDER BY orderdate;

SELECT DATEADD(day, Nums.n - 1, CAST('20140101' AS DATE)) AS orderdate,

  O.orderid, O.CustomerId, O.EmployeeId

FROM dbo.Nums

  LEFT OUTER JOIN Sales.[Order] AS O

    ON DATEADD(day, Nums.n - 1, CAST('20140101' AS DATE)) = O.orderdate

WHERE Nums.n <= DATEDIFF(day, '20140101', '20161231') + 1

ORDER BY orderdate;

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

-- Filtering Attributes from Non-Preserved Side of Outer Join

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

SELECT C.CustomerId, C.CustomerCompanyName, O.orderid, O.orderdate

FROM Sales.Customer AS C

  LEFT OUTER JOIN Sales.[Order] AS O

    ON C.CustomerId = O.CustomerId

WHERE O.orderdate >= '20160101';

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

-- Using Outer Joins in a Multi-Join Query

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

SELECT C.CustomerId, O.orderid, OD.productid, OD.Quantity

FROM Sales.Customer AS C

  LEFT OUTER JOIN Sales.[Order] AS O

    ON C.CustomerId = O.CustomerId

  INNER JOIN Sales.OrderDetail AS OD

    ON O.orderid = OD.orderid;

-- Option 1: use outer join all along

SELECT C.CustomerId, O.orderid, OD.productid, OD.Quantity

FROM Sales.Customer AS C

  LEFT OUTER JOIN Sales.[Order] AS O

    ON C.CustomerId = O.CustomerId

  LEFT OUTER JOIN Sales.OrderDetail AS OD

    ON O.orderid = OD.orderid;

-- Option 2: change join order

SELECT C.CustomerId, O.orderid, OD.productid, OD.Quantity

FROM Sales.[Order] AS O

  INNER JOIN Sales.OrderDetail AS OD

    ON O.orderid = OD.orderid

  RIGHT OUTER JOIN Sales.Customer AS C

     ON O.CustomerId = C.CustomerId;

-- Option 3: use parentheses

SELECT C.CustomerId, O.orderid, OD.productid, OD.Quantity

FROM Sales.Customer AS C

  LEFT OUTER JOIN

      (Sales.[Order] AS O

         INNER JOIN Sales.OrderDetail AS OD

           ON O.orderid = OD.orderid)

    ON C.CustomerId = O.CustomerId;

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

-- Using the COUNT Aggregate with Outer Joins

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

SELECT C.CustomerId, COUNT(\*) AS numorders

FROM Sales.Customer AS C

  LEFT OUTER JOIN Sales.[Order] AS O

    ON C.CustomerId = O.CustomerId

GROUP BY C.CustomerId;

SELECT C.CustomerId, COUNT(O.orderid) AS numorders

FROM Sales.Customer AS C

  LEFT OUTER JOIN Sales.[Order] AS O

    ON C.CustomerId = O.CustomerId

GROUP BY C.CustomerId;