

T-SQL Programming BY: Christopher Singleton

Joins Rows in both tables with the matching columns (Inner Join)

--Example Syntax of INNER JOIN:

Copies the contents of matching columns into one table.

```
SELECT Alias1.[Column] AS table1 Column (shown)  --Column names
      , Alias1.[Another Column]
      , Alias2.[Other table Column] AS table2 Column (Shown)
FROM [table1].[name] AS Alias1 --table1
     INNER JOIN [table2].[name] AS Alias2 --table2

      ON Alias1.ProductID = Alias2.ProductID  --Primary Key

      WHERE [Comments] LIKE '%heavy%'  --Filter

      ORDER BY Alias1.Column --Sort
```

--Example of typical INNER JOIN:

```
SELECT p.[Name] AS ProductName --Column names
      , p.[ProductID]
      , pr.[Comments] AS ProductReviewComments
FROM [Production].[Product] AS p  --table1

     INNER JOIN [Production].[ProductReview] AS pr --table2

      ON p.ProductID = pr.ProductID --Primary Key

      WHERE [Comments] LIKE '%heavy%'  --Filter

      ORDER BY p.ProductID --Sort
```

[Production].[Product] (table1 p)

	ProductID	Name	ProductNumber	MakeFlag	FinishedGoodsFlag
1	1	Adjustable Race	AR-5381	0	0
2	2	Bearing Ball	BA-8327	0	0
3	3	BB Ball Bearing	BE-2349	1	0
4	4	Headset Ball Bearings	BE-2908	0	0

[Production].[ProductReview] (table2 pr)

	ProductReviewID	ProductID	ReviewerName	Comments
1	1	709	John Smith	I can't believe I'm singing the
2	2	937	David	A little on the heavy side, but
3	3	937	Jill	Maybe it's just because I'm ne
4	4	798	Laura Noman	The Road-550-W from Advent

Result (combined tables)

	ProductName	ProductID	ProductReviewComments
1	Road-550-W Yellow, 40	798	The Road-550-W from Ac
2	HL Mountain Pedal	937	A little on the heavy side,

Joins the right tables rows into the left table (Left Join)

--Example of LEFT JOIN:

```
SELECT DISTINCT
    dp.[EnglishPromotionName]
    ,dr.[ResellerName]
    ,CONVERT(VARCHAR(10), frs.[OrderDate], 101) AS OrderDate
FROM [dbo].[DimPromotion] AS dp
LEFT JOIN [dbo].[FactResellerSales] AS frs ON dp.PromotionKey = frs.PromotionKey
LEFT JOIN [dbo].[DimReseller] AS dr ON frs.ResellerKey = dr.ResellerKey
WHERE dp.[EnglishPromotionName] <> 'No Discount' -- <> meaning !=
ORDER BY dp.[EnglishPromotionName]
```

[dbo].[DimPromotion] (table1 dp)

	PromotionKey	PromotionAlternateKey	EnglishPromotionName
1	1	1	No Discount
2	2	2	Volume Discount 11 to 14
3	3	3	Volume Discount 15 to 24
4	4	4	Volume Discount 25 to 40

[dbo].[FactResellerSales] (table2 frs)

	ResellerKey	ProductKey	PromotionKey	OrderDateKey	DueDateKey
1	676	349	1	20050701	20050713
2	676	350	1	20050701	20050713
3	676	351	1	20050701	20050713
4	676	344	1	20050701	20050713

[dbo].[DimReseller] (table3 dr)

	ResellerKey	GeographyKey	ResellerAlternateKey	ResellerName
1	1	637	AW00000001	A Bike Store
2	2	635	AW00000002	Progressive Spor
3	3	584	AW00000003	Advanced Bike C
4	4	572	AW00000004	Modular Cycle Sy

Result (combined tables)

	EnglishPromotionName	ResellerName	PromotionKey	OrderDate
1	Half-Price Pedal Sale	NULL	15	NULL
2	LL Road Frame Sale	NULL	12	NULL
3	Mountain Tire Sale	NULL	10	NULL
4	Mountain-100 Clearance Sale	Downtown Hotel	7	06/01/2006
5	Mountain-100 Clearance Sale	Elemental Sporting Goods	7	06/01/2006

Joins the Rows of two tables together (Full Outer Join)

--Example of FULL OUTER JOIN:

Copies all the contents of the specified columns from two tables into one table regardless.

Note: Where the data doesn't exist in non-matching columns, only creates nulls.

```
SELECT *
FROM [Person].[Person] as p
     FULL OUTER JOIN [Person].[EmailAddress] as pem
     ON p.[BusinessEntityID] = pem.[BusinessEntityID]
ORDER BY LastName
```

Typical SELECT Statements Used in SQL

--Wildcard (Selects Everything)

```
SELECT *
```

--Returns no duplicates.

```
SELECT DISTINCT FirstName
```

--Concatenated

```
SELECT CustomerID, Title+' '+FirstName+' '+LastName As CustomerName
SELECT 'Customer ID'+ CAST(CustomerID As Nvarchar (5))+' '+ is'+ ' '+Title+' '+FirstName+' '+LastName As CustomerInfo
```

--Use Alias' when performing a joins.

```
SELECT p.[Name] AS ProductName --Note: Alias Specifies the column located in which table, used for types of joins
```

--Change the datatype.

```
SELECT CAST(UnitPrice *(UnitPriceDiscount)* OrderQty As Decimal(7,2)) As TotalCost --Note: ANSI Standard
SELECT CONVERT(Decimal (7,2), UnitPrice *(UnitPriceDiscount)* OrderQty )As TotalCost
```

--Determine the inventory holding value, if null then return a zero.

```
SELECT (ISNULL(AvailableStockOnHand,0) + ISNULL(ReservedStock,0) + ISNULL(CommitedStock,0)) * ISNULL(AverageCost,0)
SELECT Item, COALESCE(ActualCost, AverageCost,0) AS ItemCost
```

Typical Filter (WHERE) Statements Used in SQL

```
WHERE ([StandardCost] < 50 AND [ListPrice] > 100) AND [Weight] > 1000
WHERE [Size] = 'XS' OR [Size] = 'S' OR [Size] = 'M' OR [Size] = 'L' OR [Size] = 'XL'
WHERE [NAME] LIKE '%Bike%'
WHERE c.[EnglishEducation] IN ('High School', 'Partial High School', 'Partial College')
```

Examples of Using SQL

```
--List all sales orders showing PurchaseOrderNumber, SalesOrderID, CustomerID, OrderDate,
--DueDate, and ShipDate. Format the datetime fields so that no time is displayed.
--add a field that calculates the difference between the due date and the ship date.
--Name the field ShipDays and show the result as a positive number.
--Be sure Datetime fields still show only the date.
```

```
SELECT [PurchaseOrderNumber], [SalesOrderID], [CustomerID],
CONVERT(NVARCHAR(10), [OrderDate], 101) AS OrderDate,
CONVERT(NVARCHAR(10), [DueDate], 101) AS DueDate,
CONVERT(NVARCHAR(10), [ShipDate], 101) AS ShipDate,
CAST(CONVERT(NVARCHAR(10), [ShipDate], 112) as INT) -
CAST(CONVERT(NVARCHAR(10), [OrderDate], 112) as INT) AS InProcessDays,
CAST(CONVERT(NVARCHAR(10), [DueDate], 112) as INT) -
CAST(CONVERT(NVARCHAR(10), [ShipDate], 112) as INT) AS ShipDayS
FROM [SalesLT].[SalesOrderHeader]
ORDER BY [SalesOrderID] ASC
```

Result

	PurchaseOrderNumber	SalesOrderID	CustomerID	OrderDate	DueDate	ShipDate	InProcessDays	ShipDayS
1	PO348186287	71774	29847	06/01/2004	06/13/2004	06/08/2004	7	5
2	PO19952192051	71776	30072	06/01/2004	06/13/2004	06/08/2004	7	5
3	PO19604173239	71780	30113	06/01/2004	06/13/2004	06/08/2004	7	5
4	PO19372114749	71782	29485	06/01/2004	06/13/2004	06/08/2004	7	5
5	PO19343113609	71783	29957	06/01/2004	06/13/2004	06/08/2004	7	5
6	PO19285135919	71784	29736	06/01/2004	06/13/2004	06/08/2004	7	5

Examples of Using SQL

```

SELECT p.[Name] AS ProductName
, CAST(oh.[OrderDate] AS DATE) AS OrderDate
, ps.[ProductSubcategoryID]
, pc.[Name] AS ProductCategoryName
, od.[OrderQty] AS TotalOrderQuantity
, CAST(oh.[Freight] AS DECIMAL(6,2)) AS FreightCost
, CAST(oh.[TaxAmt] AS DECIMAL(6,2)) AS TaxAmount
, CAST(oh.[SubTotal] AS DECIMAL(7,2)) AS SalesSubTotal
, CAST(oh.[TotalDue] AS DECIMAL(8,2)) AS SalesTotalDue
FROM [Production].[Product] AS p
INNER JOIN [Sales].[SalesOrderDetail] AS od
    ON p.[ProductID] = od.[ProductID]
INNER JOIN [Sales].[SalesOrderHeader] AS oh
    ON od.[SalesOrderID] = oh.[SalesOrderID]
INNER JOIN [Production].[ProductSubCategory] AS ps
    ON p.ProductSubcategoryID = ps.ProductSubCategoryID
INNER JOIN [Production].[ProductCategory] AS pc
    ON ps.ProductCategoryID = pc.ProductCategoryID
WHERE ps.[ProductSubcategoryID] IN (1, 2, 3)
    AND oh.[OrderDate] BETWEEN '2007-12-01' AND '2007-12-31'
ORDER BY CAST(oh.[TotalDue] AS DECIMAL(8,2)) DESC

```

--CAST Date (Do Away with Time)

--CAST to two Decimal Places.

--CAST to two Decimal Places.

--CAST to two Decimal Places.

--CAST to two Decimal Places.

--Proction.Product Table Alias

--INNER JOIN Sales.SalesOrderDetail Table.

--Common Key in both tables.

--INNER JOIN Sales.SalesOrderHeader Table.

--Common Key in both tables.

--INNER JOIN Production.ProductSubCategory Table.

--Common Key in both tables.

--INNER JOIN Production.ProductCategory Table.

--Common Key in both tables.

--Use the IN Keyword to filter all Bike Categories.

--December 2007 (One Whole Month Only)

--Sort by SalesTotalDue highest to lowest.

Result

	ProductName	OrderDate	ProductSubcategoryID	ProductCategoryName	TotalOrderQuantity	FreightCost	TaxAmount	SalesSub Total	SalesTotalDue
1	Touring-2000 Blue, 46	2007-12-01	3	Bikes	1	2942.89	9417.26	97647.72	110007.87
2	Touring-1000 Blue, 60	2007-12-01	3	Bikes	8	2942.89	9417.26	97647.72	110007.87
3	Touring-1000 Yellow, 50	2007-12-01	3	Bikes	10	2942.89	9417.26	97647.72	110007.87
4	Touring-1000 Yellow, 60	2007-12-01	3	Bikes	10	2942.89	9417.26	97647.72	110007.87
5	Touring-3000 Blue, 54	2007-12-01	3	Bikes	8	2942.89	9417.26	97647.72	110007.87
6	Touring-2000 Blue, 60	2007-12-01	3	Bikes	2	2942.89	9417.26	97647.72	110007.87
7	Touring-3000 Yellow, 54	2007-12-01	3	Bikes	2	2942.89	9417.26	97647.72	110007.87
8	Touring-3000 Yellow, 44	2007-12-01	3	Bikes	6	2942.89	9417.26	97647.72	110007.87
9	Touring-3000 Yellow, 50	2007-12-01	3	Bikes	1	2942.89	9417.26	97647.72	110007.87
10	Touring-3000 Yellow, 62	2007-12-01	3	Bikes	4	2942.89	9417.26	97647.72	110007.87
11	Touring-2000 Blue, 54	2007-12-01	3	Bikes	8	2942.89	9417.26	97647.72	110007.87
12	Touring-1000 Yellow, 46	2007-12-01	3	Bikes	7	2942.89	9417.26	97647.72	110007.87
13	Touring-1000 Blue, 50	2007-12-01	3	Bikes	3	2942.89	9417.26	97647.72	110007.87
14	Touring-1000 Blue, 46	2007-12-01	3	Bikes	6	2942.89	9417.26	97647.72	110007.87
15	Touring-3000 Blue, 50	2007-12-01	3	Bikes	2	2942.89	9417.26	97647.72	110007.87
16	Road-550-W Yellow, 42	2007-12-01	2	Bikes	1	2578.23	8250.33	84413.16	95241.72
17	Road-750 Black, 48	2007-12-01	2	Bikes	7	2578.23	8250.33	84413.16	95241.72

SubQueries (SELECT Clause)

--Names of resellers and the product names of products they purchased.
--Eliminate duplicate rows. Use an appropriate sort. No special predicated requested.

```
SELECT DISTINCT
    (SELECT ResellerName FROM [dbo].[DimReseller] AS r WHERE r.ResellerKey = frs.ResellerKey) AS r_ResellerName
    ,(SELECT EnglishProductName FROM [dbo].[DimProduct] AS p WHERE p.ProductKey = frs.ProductKey) AS p_ProductName
FROM [dbo].[FactResellerSales] AS frs
ORDER BY r_ResellerName, p_ProductName
```

SubQueries (FROM Clause)

```
SELECT c.CustFirstName, c.CustLastName, o.OrderNumber,
o.OrderDate, od.ProductNumber, p.ProductName, od.QuantityOrdered
FROM (
    (Customers AS c INNER JOIN Orders AS o ON c.CustomerID = o.CustomerID)
    INNER JOIN Order_Details od ON o.OrderNumber = od.OrderNumber
)
INNER JOIN Products p ON p.ProductNumber = od.ProductNumber
WHERE o.OrderDate = (SELECT MAX(OrderDate) FROM Orders AS O2 WHERE O2.CustomerID = c.CustomerID);
```

SubQueries (WHERE Clause)

--Find all current Products that are for sale and have not been sold to Resellers.

```
SELECT p.[ProductKey] AS p_ProductKey
    ,p.[EnglishProductName]
    ,p.[Status] AS p_Status
FROM [dbo].[DimProduct] AS p
WHERE p.ProductKey NOT IN
    (SELECT [ProductKey]
        FROM [dbo].[FactResellerSales] AS frs
        WHERE p.ProductKey = frs.ProductKey)
AND p.[Status] IS NULL
ORDER BY p.[EnglishProductName]
```

SubQueries (WHERE Clause Cont.)

-- List the products in the Accessory category that have a listprice higher than the average
-- listprice of Accessory items. Show product alternate key, product name, and listprice in the
-- results set. Order the information so it is easy to understand. Note: Both ways have the same results.

----- One Way -----

```
SELECT dp.ProductAlternateKey
      ,dp.EnglishProductName
      ,dp.ListPrice
FROM [dbo].[DimProduct] AS dp
WHERE dp.ListPrice > (SELECT Avg(ListPrice)
                     FROM [dbo].[DimProduct] AS p
                     WHERE p.ProductSubCategoryKey IN
                        (SELECT psc.ProductSubCategoryKey
                         FROM [dbo].[DimProductSubcategory] AS psc
                         WHERE psc.ProductSubCategoryKey = p.ProductSubCategoryKey
                          AND psc.ProductCategoryKey = 4))

ORDER BY dp.ListPrice
```

----- Correct Way -----

```
SELECT dp.ProductAlternateKey
      ,dp.EnglishProductName
      ,dp.ListPrice
FROM [dbo].[DimProduct] AS dp
WHERE dp.ListPrice > (SELECT Avg(ListPrice)
                     FROM [dbo].[DimProduct] AS p
                     INNER JOIN [dbo].[DimProductSubcategory] AS PS
                        ON PS.ProductSubcategoryKey = P.ProductSubcategoryKey
                     INNER JOIN [dbo].[DimProductCategory] AS PC
                        ON PC.ProductCategoryKey = PS.ProductCategoryKey
                     WHERE PC.ProductCategoryKey = 4)

ORDER BY dp.ListPrice
```

Aggregation (SUM, AVG, MIN, MAX, YEAR)

--List the total dollar amount (SalesAmount) for 2008 sales to resellers in Germany.
--Show only the total sales--one row, one column--rounded to two decimal places.

```
SELECT CAST(SUM([SalesAmount]) AS Decimal(8,2)) AS Total_ResellerSales
FROM [dbo].[FactResellerSales] AS frs
    INNER JOIN [dbo].[DimReseller] AS dr ON frs.ResellerKey = dr.ResellerKey
    INNER JOIN [dbo].[DimGeography] AS dg ON dr.[GeographyKey] = dg.[GeographyKey]
WHERE dg.EnglishCountryRegionName = 'Germany'
    AND frs.[OrderDateKey] BETWEEN 20080101 AND 20081231
    --(frs.[OrderDateKey] >= 20080101 AND frs.[OrderDateKey] <= 20081231)
```

--List the lowest list price, the average list price, the highest list price, and product count for road bikes.

```
SELECT MIN(ListPrice) AS MinListPrice
    ,AVG(DISTINCT ListPrice) AS AvgListPrice
    ,MAX(ListPrice) AS MaxListPrice
    ,COUNT(ProductSubcategoryKey) AS NumOfRoadBikes
FROM [dbo].[DimProduct]
WHERE ProductSubcategoryKey = 2
```

--Another way to COUNT:

```
SELECT MIN(ListPrice) AS MinListPrice
    ,AVG(DISTINCT ListPrice) AS AvgListPrice
    ,MAX(ListPrice) AS MaxListPrice
    ,COUNT([EnglishProductName]) AS NumOfRoadBikes
FROM [dbo].[DimProduct]
WHERE [EnglishProductName] LIKE 'Road%' AND ProductSubcategoryKey = 2
```


Aggregation (SUM, AVG, MIN, MAX, YEAR, Cont.)

List total internet sales in 2007 and 2008.

Also, list total internet sales between 2007 and 2008.

Round to two decimal places and give alias for three columns.

```
SELECT CAST(SUM([SalesAmount]) AS Decimal(10,2)) AS TotalSalesAmount2007  --2007 Outer query
      ,(SELECT CAST(SUM([SalesAmount]) AS Decimal(10,2))                  --2008 Subquery
        FROM [dbo].[FactInternetSales]
        WHERE YEAR([OrderDate]) = 2008) AS TotalSalesAmount2008
      ,CAST(SUM([SalesAmount]) - (SELECT SUM([SalesAmount])
                                FROM [dbo].[FactInternetSales]
                                WHERE YEAR([OrderDate]) = 2008) AS Decimal(8,2)) AS YearlySalesDifference  --2007-2008 = Difference Calculated Subquery
FROM [dbo].[FactInternetSales]
WHERE YEAR([OrderDate]) = 2007
```

Result

	TotalSalesAmount2007	TotalSalesAmount2008	YearlySalesDifference
1	9791060.30	9770899.74	20160.56