# C:\Users\stefano.maggi.CONBIPELSPA\Desktop\prom38.png

**Date**: 16/10/2012

**Procedure:** GROUP BY techniques (SQL Server)

**Source:** [**LINK**](http://weblogs.sqlteam.com/jeffs/archive/2005/12/14/8546.aspx)

**Permalink:** [**LINK**](http://heelpbook.altervista.org/2012/group-by-techniques-sql-server/)

**Created by:** HeelpBook Staff

**Document Version:** 1.0

# [**GROUP BY techniques (SQL Server)**](http://heelpbook.altervista.org/2012/group-by-techniques-sql-server/)

One aspect of the versatile **SELECT** statement that seems to confuse many people is the **GROUP BY** clause.  It is very important to group your rows in the proper place.

Always push **GROUP BY** aggregations as far into your nested **SELECT** statements as possible – if you have a situation in which you are grouping by long lists of columns that are not part of primary keys, you are probably have not structured your query correctly.

Here’s a classic example that returns total sales per customer, in addition to returning the customer’s name and address:

SELECT

  C.CustomerID, C.CustomerName,

  C.CustomerType, C.Address1, C.City,

  C.State, SUM(S.Sales) as TotalSales

FROM

  Customers C

INNER JOIN Sales S

  ON C.CustomerID = S.CustomerID

GROUP BY

  C.CustomerID, C.CustomerName,

  C.CustomerType, C.Address1, C.City, C.State

I can’t say how many times I see **SELECT’s** written this way and it is simply wrong. You should only be grouping on **CustomerID**, and not on all those other columns.  Push the grouping down a level, into a derived table:

SELECT

  C.CustomerID, C.CustomerName,

  C.CustomerType, C.Address1, C.City,

  C.State, S.TotalSales

FROM

  Customers C

INNER JOIN

  (SELECT

     CustomerID, SUM(Sales) as TotalSales

   FROM

     Sales

   GROUP BY

     CustomerID) S

ON

  C.CustomerID = S.CustomerID

Hopefully, you will agree that it is a much cleaner **SELECT** statement now, it is more efficient and it logically makes more sense when you look at it.