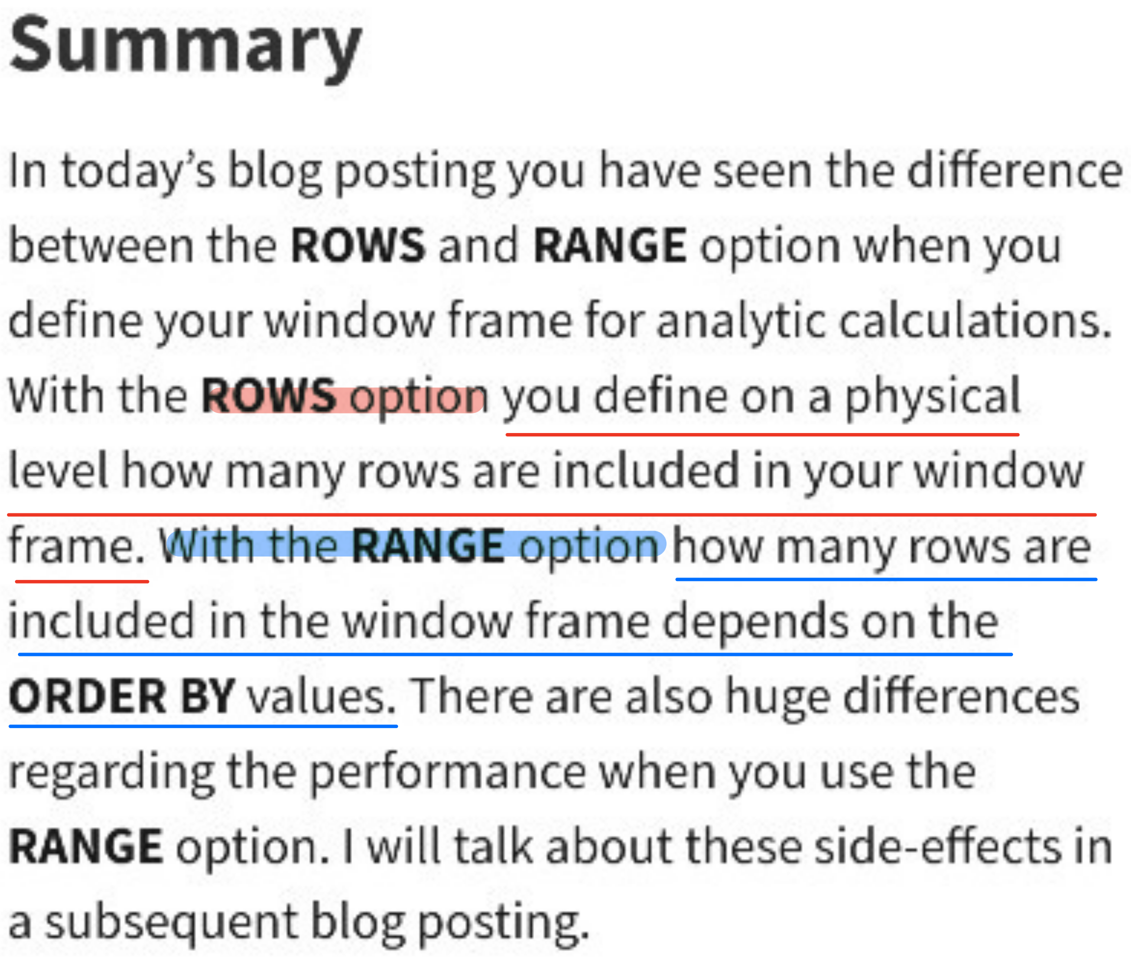
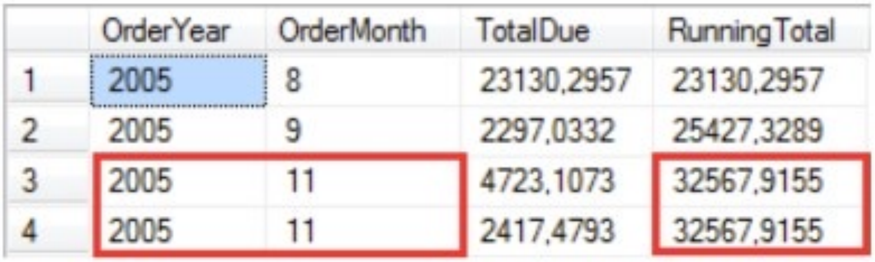


Summary

In today's blog posting you have seen the difference between the **ROWS** and **RANGE** option when you define your window frame for analytic calculations. With the **ROWS** option you define on a physical level how many rows are included in your window frame. With the **RANGE** option how many rows are included in the window frame depends on the **ORDER BY** values. There are also huge differences regarding the performance when you use the **RANGE** option. I will talk about these side-effects in a subsequent blog posting.



	OrderYear	OrderMonth	TotalDue	RunningTotal
1	2005	8	23130,2957	23130,2957
2	2005	9	2297,0332	25427,3289
3	2005	11	4723,1073	32567,9155
4	2005	11	2417,4793	32567,9155

Let's try to understand why the **RANGE** option gives you here a different result than the **ROWS** option. With the **ROWS** option you define a fixed number of rows preceding and following the current row. Which rows you see here through your window frame depends on the **ORDER BY** clause of the window frame. You can also say that you define your window frame on a physical level.

Things change when you use the **RANGE** option. The **RANGE** option includes all the rows within the window frame that have the same **ORDER BY** values as the current row. As you can see from the previous picture, for the 2 records of November 2005 you get the same sum, because both rows have the same **ORDER BY** values (November 2005). With the **RANGE** option you define your window frame on a logical level. If more rows have the same **ORDER BY** value, your window frame consists of more rows than when you use the **ROWS** option.