Analytic Window Functions

A practical look at using analytic functions

Olympia Area SqlServer User Group

By Gary Melhaff

March 15, 2017

About me

- Over 30 years in IT
- Certifications & experience in Oracle, Teradata and Sql Server
- Can be blamed for designing around dozen datamarts and data warehouses since late 1990s
- Worked for DSHS and DOT in the 1980s, then DNR until 1996, then Weyerhaeuser, Freeinternet.com, consulting gigs, then Washington Mutual Bank for around 7 years
- Last 7 years as Data Architect at World Vision
- Product reviews at https://www.itcentralstation.com for SSIS, Wherescape RED, MDS and Melissa Data Quality

Environment

- Data warehouse environment - 100% Microsoft
- Sources: Microsoft Dynamics CRM Online, Oracle Enterprise Business Suite, Blackbaud CRM (hosted), flat files from multiple sources, Master Data Services (MDS), Adobe Campaign, Oracle NetSuite (new)
- MelissaData Matchup for SSIS, Attunity OLEDB for Oracle, Kingswaysoft CRM Adapter for SSIS, Visual Studio Online (TFS), BiXpress (for SSIS monitoring&notification), SSIS MultipleHash
- Master Data Services
- SSAS tabular (2016 SP1)
- BI tools: SSRS, PowerBI and Excel
- VMWare and SSD San

Why you should care

Analytic Window Functions are *extremely* powerful

Require very little coding and are easy to use once you learn the basic concepts

Workarounds would be relatively complex and less efficient

What are we talking about?

Any calculation that is defined by an "OVER()" clause…GROUP BY is not required

They give you access to rows *outside* of the immediate row you are on

In other words you can do things like…

- access aggregate values along with detail rows
- access row values from other rows besides the current row

Examples of practical use

- Record De-Duplication (picking rows that share same key values)
- Assigning effective date ranges

Analytic and Ranking Window Functions

- Lead/Lag
- First_Value/Last_Value
- Row_Number
- Rank
- Dense_Rank
- Ntile
- Cume_Dist
- Percent_Rank
- Percentile_Disc
- Percentile_Cont

Functions most often
used in ETL operations

Aggregate window functions

- MIN, MAX, AVG, SUM, COUNT, COUNT_BIG
- CHECKSUM_AGG
- STDEV, STDEVP, VAR, VARP

Concepts – The Window

It is set with the “partition by” clause

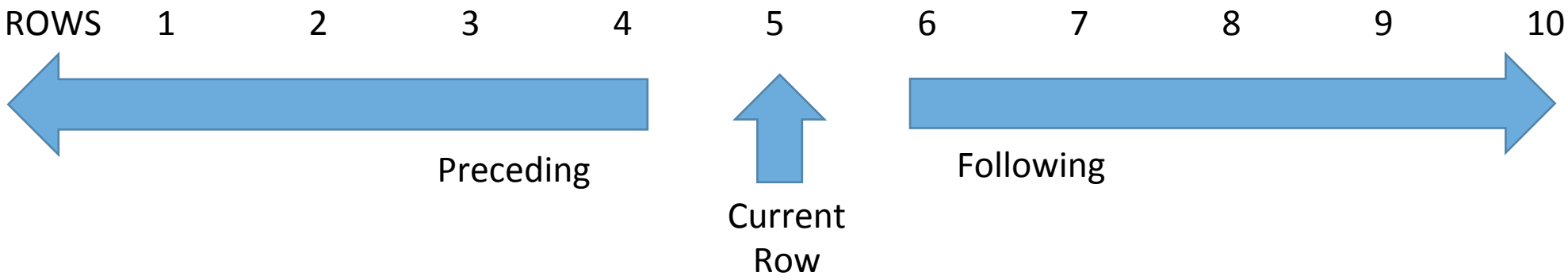
- It’s the set of rows that share the same column(s)
- Partition clause is optional – if you don’t use it then *all* rows are the window size

Concepts – The Window Frame

Frame set by keywords shown below

- Defines the subset of rows within the window that the function will utilize
- *Careful* - Defaults to start of window up to current row!
- You set it using ROW or RANGE syntax

Rows in perspective of the window frame



"Unbounded preceding" sets the frame to the *start* of the window.

"Unbounded following" sets the frame to the *end* of the window.

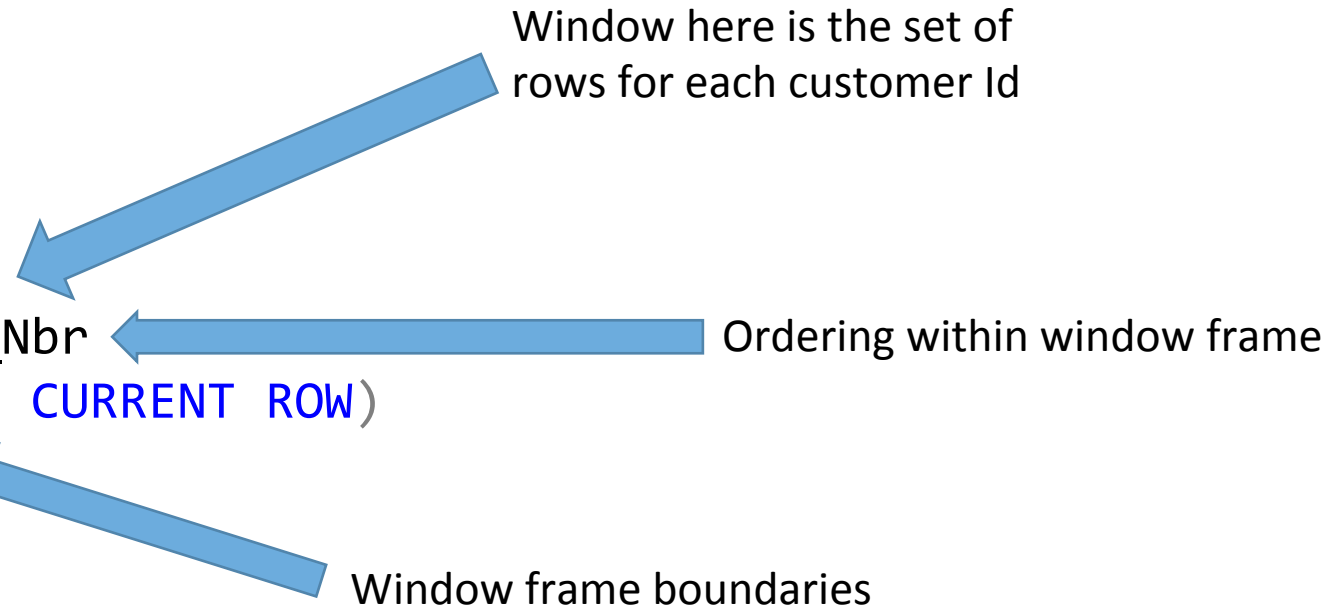
You can also specify an offset of rows preceding or following.

Caveats or Restrictions

- Cannot use one of these within a “where” or “having” clause!
- Certain functions will go to the beginning or end of a set of rows but not in-between (eg. first/last)
- Some options such as window frame size are not available for all functions. For example row_number() only supports the default window frame size.

Example Use Case

```
Rolling11Months_Amt =
    SUM(ISNULL(Total_Amt,0))
  OVER (PARTITION BY Customer_Dim_Id
        ORDER BY Calendar_Year_Month_Nbr
        ROWS BETWEEN 11 PRECEDING AND CURRENT ROW)
```



** SSIS executes this in a dataflow computing rolling total by customer by month for 150 million records across over 8 million customers in just over 16 minutes on 12 vproc server*

Demos