EXAMPLE - Rolling Functions

This example describes how to use the rolling computational functions:

- `ROLLINGSUM` - computes a rolling sum from a window of rows before and after the current row. See *ROLLINGSUM Function*.
- `ROLLINGAVERAGE` - computes a rolling average from a window of rows before and after the current row. See *ROLLINGAVERAGE Function*.
- `ROWNUMBER` - computes the row number for each row, as determined by the ordering column. See *ROWNUMBER Function*.

The following dataset contains sales data over the final quarter of the year.

Source:

Date	Sales
10/2/16	200
10/9/16	500
10/16/16	350
10/23/16	400
10/30/16	190
11/6/16	550
11/13/16	610
11/20/16	480
11/27/16	660
12/4/16	690
12/11/16	810
12/18/16	950
12/25/16	1020
1/1/17	680

Transform:

First, you want to maintain the row information as a separate column. Since data is ordered already by the `Date` column, you can use the following:

```
window value:ROWNUMBER() order:Date
```

Rename this column to `rowId` for week of quarter.

Now, you want to extract month and week information from the `Date` values. Deriving the month value:

```
derive type:single value:MONTH(Date) as:'Month'
```

Deriving the quarter value:

```
derive type:single value:(1 + FLOOR(((month-1)/3))) as:'QTR'
```

Deriving the week-of-quarter value:

```
window value:ROWNUMBER() order:Date group:QTR
```

Rename this column `WOQ` (week of quarter).

Deriving the week-of-month value:

```
window value:ROWNUMBER() group:Month order:Date
```

Rename this column `WOM` (week of month).

Now, you perform your rolling computations. Compute the running total of sales using the following:

```
window value: ROLLINGSUM(Sales, -1, 0) order: Date group:QTR
```

The `-1` parameter is used in the above computation to gather the rolling sum of all rows of data from the current one to the first one. Note that the use of the `QTR` column for grouping, which moves the value for the `01/01/2017` into its own computational bucket. This may or may not be preferred.

Rename this column `QTD` (quarter to-date). Now, generate a similar column to compute the rolling average of weekly sales for the quarter:

```
window value: ROUND(ROLLINGAVERAGE(Sales, -1, 0)) order: Date group:QTR
```

Since the `ROLLINGAVERAGE` function can compute fractional values, it is wrapped in the `ROUND` function for neatness. Rename this column `avgWeekByQuarter`.

Results:

When the unnecessary columns are dropped and some reordering is applied, your dataset should look like the following:

Date	WOQ	Sales	QTD	avgWeekByQuarter
10/2/16	1	200	200	200
10/9/16	2	500	700	350
10/16/16	3	350	1050	350
10/23/16	4	400	1450	363
10/30/16	5	190	1640	328
11/6/16	6	550	2190	365
11/13/16	7	610	2800	400
11/20/16	8	480	3280	410
11/27/16	9	660	3940	438
12/4/16	10	690	4630	463
12/11/16	11	810	5440	495
12/18/16	12	950	6390	533
12/25/16	13	1020	7410	570
1/1/17	1	680	680	680