Windowing Functions

Overview

Understand windowing functions as a way to write complex queries

Use windowing functions to calculate

- Moving averages
- Percentage contribution
- Percentile for a value

Implement queries for all of these

What are Window Functions?

Window Functions

A suite of functions which are syntactic sugar for complex queries

Window Functions

Make complex operations simple without needing many intermediate calculations

Window Functions

Reduces the need for intermediate tables to store temporary data

Window Functions

What were the top selling N products in last week’s sale?

Window = one week

Operation = ranking product sales

Window Functions

What revenue percentile did this supplier fall into for this quarter?

Window = one quarter

Operation = percentiles on revenues

Window Functions

What were the top selling N products in last week's sale?

What revenue percentile did this supplier fall into for this quarter?

Can be expressed in a single query

An Example of a Window Function

Running Total of Revenue

ID	Store	Product	Date	Revenue
o1	Seattle	Bananas	2017-01-01	7
o2	Kent	Apples	2017-01-02	20
o3	Bellevue	Flowers	2017-01-02	10
o4	Redmond	Meat	2017-01-03	40
o5	Seattle	Potatoes	2017-01-04	9
o6	Bellevue	Bread	2017-01-04	5
o7	Redmond	Bread	2017-01-05	5
o8	Issaquah	Onion	2017-01-05	4

Orders in a grocery store

Running Total of Revenue

ID	Store	Product	Date	Revenue
o1	Seattle	Bananas	2017-01-01	7
o2	Kent	Apples	2017-01-02	20
o3	Bellevue	Flowers	2017-01-02	10
o4	Redmond	Meat	2017-01-03	40
o5	Seattle	Potatoes	2017-01-04	9
o6	Bellevue	Bread	2017-01-04	5
o7	Redmond	Bread	2017-01-05	5
o8	Issaquah	Onion	2017-01-05	4

Sorted by order ID

Running Total of Revenue

ID	Store	Product	Date	Revenue
o1	Seattle	Bananas	2017-01-01	7
o2	Kent	Apples	2017-01-02	20
o3	Bellevue	Flowers	2017-01-02	10
o4	Redmond	Meat	2017-01-03	40
o5	Seattle	Potatoes	2017-01-04	9
o6	Bellevue	Bread	2017-01-04	5
o7	Redmond	Bread	2017-01-05	5
o8	Issaquah	Onion	2017-01-05	4

Running Total
7
27
37
77
86
91
96
10

Running Total of Revenue

ID	Store	Product	Date	Revenue
o1	Seattle	Bananas	2017-01-01	**7**
o2	Kent	Apples	2017-01-02	20
o3	Bellevue	Flowers	2017-01-02	10
o4	Redmond	Meat	2017-01-03	40
o5	Seattle	Potatoes	2017-01-04	9
o6	Bellevue	Bread	2017-01-04	5
o7	Redmond	Bread	2017-01-05	5
o8	Issaquah	Onion	2017-01-05	4

Running Total
7
27
37
77
86
91
96
10

Running Total of Revenue

ID	Store	Product	Date	Revenue
o1	Seattle	Bananas	2017-01-01	**7**
o2	Kent	Apples	2017-01-02	**20**
o3	Bellevue	Flowers	2017-01-02	10
o4	Redmond	Meat	2017-01-03	40
o5	Seattle	Potatoes	2017-01-04	9
o6	Bellevue	Bread	2017-01-04	5
o7	Redmond	Bread	2017-01-05	5
o8	Issaquah	Onion	2017-01-05	4

Running Total
7
27
37
77
86
91
96
10

Running Total of Revenue

ID	Store	Product	Date	Revenue
o1	Seattle	Bananas	2017-01-01	**7**
o2	Kent	Apples	2017-01-02	**20**
o3	Bellevue	Flowers	2017-01-02	**10**
o4	Redmond	Meat	2017-01-03	40
o5	Seattle	Potatoes	2017-01-04	9
o6	Bellevue	Bread	2017-01-04	5
o7	Redmond	Bread	2017-01-05	5
o8	Issaquah	Onion	2017-01-05	4

Running Total
7
27
37
77
86
91
96
10

Running Total of Revenue

ID	Store	Product	Date	Revenue	Running Total
o1	Seattle	Bananas	2017-01-01	**7**	**7**
o2	Kent	Apples	2017-01-02	**20**	**27**
o3	Bellevue	Flowers	2017-01-02	**10**	**37**
o4	Redmond	Meat	2017-01-03	**40**	**77**
o5	Seattle	Potatoes	2017-01-04	9	86
o6	Bellevue	Bread	2017-01-04	5	91
o7	Redmond	Bread	2017-01-05	5	96
o8	Issaquah	Onion	2017-01-05	4	10

Running Total of Revenue

ID	Store	Product	Date	Revenue
o1	Seattle	Bananas	2017-01-01	**7**
o2	Kent	Apples	2017-01-02	**20**
o3	Bellevue	Flowers	2017-01-02	**10**
o4	Redmond	Meat	2017-01-03	**40**
o5	Seattle	Potatoes	2017-01-04	**9**
o6	Bellevue	Bread	2017-01-04	5
o7	Redmond	Bread	2017-01-05	5
o8	Issaquah	Onion	2017-01-05	4

Running Total
7
27
37
77
86
91
96
10

For each row:
Calculate sum over all rows till the current row

Running Total of Revenue

ID	Store	Product	Date	Revenue
o1	Seattle	Bananas	2017-01-01	**7**
o2	Kent	Apples	2017-01-02	**20**
o3	Bellevue	Flowers	2017-01-02	**10**
o4	Redmond	Meat	2017-01-03	**40**
o5	Seattle	Potatoes	2017-01-04	**9**
o6	Bellevue	Bread	2017-01-04	5
o7	Redmond	Bread	2017-01-05	5
o8	Issaquah	Onion	2017-01-05	4

Running Total
7
27
37
77
86
91
96
10

Operation = sum

Running Total of Revenue

ID	Store	Product	Date	Revenue
o1	Seattle	Bananas	2017-01-01	**7**
o2	Kent	Apples	2017-01-02	**20**
o3	Bellevue	Flowers	2017-01-02	**10**
o4	Redmond	Meat	2017-01-03	**40**
o5	Seattle	Potatoes	2017-01-04	**9**
o6	Bellevue	Bread	2017-01-04	5
o7	Redmond	Bread	2017-01-05	5
o8	Issaquah	Onion	2017-01-05	4

Running Total
7
27
37
77
86
91
96
10

Window is all rows from the top to the current row

Demo

Calculate aggregations over a window of records using window functions

- running total

- running average

Window Function with a Reset

Per Day Revenue Totals

ID	Store	Product	Date	Revenue
o1	Seattle	Bananas	2017-01-11	7
o2	Kent	Apples	2017-01-11	20
o3	Bellevue	Flowers	2017-01-11	10
o4	Redmond	Meat	2017-01-12	40
o5	Seattle	Potatoes	2017-01-12	9
o6	Bellevue	Bread	2017-01-12	5
o7	Redmond	Bread	2017-01-13	5
o8	Issaquah	Onion	2017-01-13	4

A revenue running total for each day

Per Day Revenue Totals

ID	Store	Product	Date	Revenue
o1	Seattle	Bananas	2017-01-11	7
o2	Kent	Apples	2017-01-11	20
o3	Bellevue	Flowers	2017-01-11	10
o4	Redmond	Meat	2017-01-12	40
o5	Seattle	Potatoes	2017-01-12	9
o6	Bellevue	Bread	2017-01-12	5
o7	Redmond	Bread	2017-01-13	5
o8	Issaquah	Onion	2017-01-13	4

Reset to 0 when a new day begins

Per Day Revenue Totals

ID	Store	Product	Date	Revenue
o1	Seattle	Bananas	2017-01-11	7
o2	Kent	Apples	2017-01-11	20
o3	Bellevue	Flowers	2017-01-11	10
o4	Redmond	Meat	2017-01-12	40
o5	Seattle	Potatoes	2017-01-12	9
o6	Bellevue	Bread	2017-01-12	5
o7	Redmond	Bread	2017-01-13	5
o8	Issaquah	Onion	2017-01-13	4

Operate on blocks of one day

Demo

Calculate aggregations over a window on blocks of records

- running total

- running count

Moving Averages using a Window Function

Moving Averages

ID	Store	Product	Date	Revenue
o1	Seattle	Bananas	2017-01-11	7
o2	Kent	Apples	2017-01-11	20
o3	Bellevue	Flowers	2017-01-11	10
o4	Redmond	Meat	2017-01-12	40
o5	Seattle	Potatoes	2017-01-12	9
o6	Bellevue	Bread	2017-01-12	5
o7	Redmond	Bread	2017-01-13	5
o8	Issaquah	Onion	2017-01-13	4

Calculate the average for the last 3 items sold

Moving Averages

ID	Store	Product	Date	Revenue
o1	Seattle	Bananas	2017-01-11	7
o2	Kent	Apples	2017-01-11	20
o3	**Bellevue**	**Flowers**	**2017-01-11**	**10**
o4	Redmond	Meat	2017-01-12	40
o5	Seattle	Potatoes	2017-01-12	9
o6	Bellevue	Bread	2017-01-12	5
o7	Redmond	Bread	2017-01-13	5
o8	Issaquah	Onion	2017-01-13	4

3 previous rows from the current row

Moving Averages

ID	Store	Product	Date	Revenue
o1	Seattle	Bananas	2017-01-11	7
o2	Kent	Apples	2017-01-11	20
o3	Bellevue	Flowers	2017-01-11	10
o4	**Redmond**	**Meat**	**2017-01-12**	**40**
o5	Seattle	Potatoes	2017-01-12	9
o6	Bellevue	Bread	2017-01-12	5
o7	Redmond	Bread	2017-01-13	5
o8	Issaquah	Onion	2017-01-13	4

3 previous rows from the current row

Moving Averages

ID	Store	Product	Date	Revenue
o1	Seattle	Bananas	2017-01-11	7
o2	Kent	Apples	2017-01-11	20
o3	Bellevue	Flowers	2017-01-11	10
o4	Redmond	Meat	2017-01-12	40
o5	**Seattle**	**Potatoes**	**2017-01-12**	**9**
o6	Bellevue	Bread	2017-01-12	5
o7	Redmond	Bread	2017-01-13	5
o8	Issaquah	Onion	2017-01-13	4

3 previous rows from the current row

Demo

Calculate aggregations over a window of a specific size

Each Order as a Percentage of the Total

Percentage of the Total per Day

ID	Store	Product	Date	Revenue
o1	Seattle	Bananas	2017-01-11	7
o2	Kent	Apples	2017-01-11	20
o3	Bellevue	Flowers	2017-01-11	10
o4	Redmond	Meat	2017-01-12	40
o5	Seattle	Potatoes	2017-01-12	9
o6	Bellevue	Bread	2017-01-12	5
o7	Redmond	Bread	2017-01-13	5
o8	Issaquah	Onion	2017-01-13	4

Orders for one day, total revenue = 37

Percentage of the Total per Day

ID	Store	Product	Date	Revenue
o1	Seattle	Bananas	2017-01-11	7
o2	Kent	Apples	2017-01-11	20
o3	Bellevue	Flowers	2017-01-11	10
o4	Redmond	Meat	2017-01-12	40
o5	Seattle	Potatoes	2017-01-12	9
o6	Bellevue	Bread	2017-01-12	5
o7	Redmond	Bread	2017-01-13	5
o8	Issaquah	Onion	2017-01-13	4

% contribution of o1 = 7/37 * 100 = 18.9%

Percentage of the Total per Day

ID	Store	Product	Date	Revenue
o1	Seattle	Bananas	2017-01-11	7
o2	Kent	Apples	2017-01-11	20
o3	Bellevue	Flowers	2017-01-11	10
o4	Redmond	Meat	2017-01-12	40
o5	Seattle	Potatoes	2017-01-12	9
o6	Bellevue	Bread	2017-01-12	5
o7	Redmond	Bread	2017-01-13	5
o8	Issaquah	Onion	2017-01-13	4

% contribution of o2 = 20 /37 * 100 = 54%

Percentage of the Total per Day

ID	Store	Product	Date	Revenue
o1	Seattle	Bananas	2017-01-11	7
o2	Kent	Apples	2017-01-11	20
o3	Bellevue	Flowers	2017-01-11	10
o4	Redmond	Meat	2017-01-12	40
o5	Seattle	Potatoes	2017-01-12	9
o6	Bellevue	Bread	2017-01-12	5
o7	Redmond	Bread	2017-01-13	5
o8	Issaquah	Onion	2017-01-13	4

% contribution of o3 = 10 /37 * 100 = 27%

Percentage of the Total per Day

$$\% \text{ contribution} = \frac{\text{Revenue} \times 100}{\text{Total Revenue}}$$

Percentage of the Total per Day

$$\% \text{ contribution} = \frac{\text{Revenue x 100}}{\text{sum(revenue) over (partition by day)}}$$

Demo

Calculate the percentage contribution of each order to a day’s revenue

Demo

Use the row_number() and rank() window function and see the difference between the two

Demo

Use the ntile() window function to see which percentile each record falls into

Summary

Understood windowing functions as a way to write complex queries

Got hands on practice with windowing functions to calculate

- Moving averages
- Percentage contribution
- Percentile for a value