**Question: How window functions are different from Aggregate functions? Any similarities between them?**

Window functions are similar to the aggregation done in the GROUP BY clause. However, rows are not grouped into a single row, each row retains their separate identity. That is, a window function may return a single value for each row. Here’s a good visualization of what I mean by that.

Diagram

Description automatically generated

**WINDOW VS GROUP BY**

Let’s say we have some salary data and we want to find to create a column that gives us the average salary for each job title.

Table

Description automatically generated

Table

Description automatically generated

# Why use Window Functions?

1. **One major advantage of window functions is that it allows you to work with both aggregate and non-aggregate values all at once because the rows are not collapsed together.**
2. Window functions are also simple to use and read. That is, they can reduce the complexity of your queries, which makes it easier to maintain down the road.
3. In addition, they can help with performance issues. For example, you can use a window function instead of having to do a self-join or cross-join.

**This is important because based off of this logical order, window functions are allowed in SELECT and ORDER BY, but they are not allowed in FROM, WHERE, GROUP BY, or HAVING clauses.**

**ORDER OF OPERATION: number 6**

Text

Description automatically generated with medium confidence

**SYNTAX**

# Window Function Syntax

Here’s what the generic syntax looks like for a window function in the SELECT clause.

A screenshot of a computer

Description automatically generated with low confidence

Image by Author

There’s a lot of words here, so let’s look at some definitions:

* **window\_function**is the name of the window function we want to use; for example, sum, avg, or row\_number (we’ll learn more about these later)
* **expression**is the name of the column that we want the window function operated on. This may not be necessary depending on what window\_function is used
* **OVER** is just to signify that this is a window function
* **PARTITION BY**divides the rows into partitions so we can specify which rows to use to compute the window function
* **partition\_list**is the name of the column(s) we want to partition by
* **ORDER BY**is used so that we can order the rows within each partition. This is optional and does not have to be specified
* **order\_list**is the name of the column(s) we want to order by
* **ROWS** can be used if we want to further limit the rows within our partition. This is optional and usually not used
* **frame\_clause** defines how much to offset from our current row

**LIST OF WINDOW FUNCTIONS**

Diagram

Description automatically generated