

# SQL.days konferenz

Window-Funktionen mit SQL Server

Aufgefrischt von Thomas Hütter



dbWatch



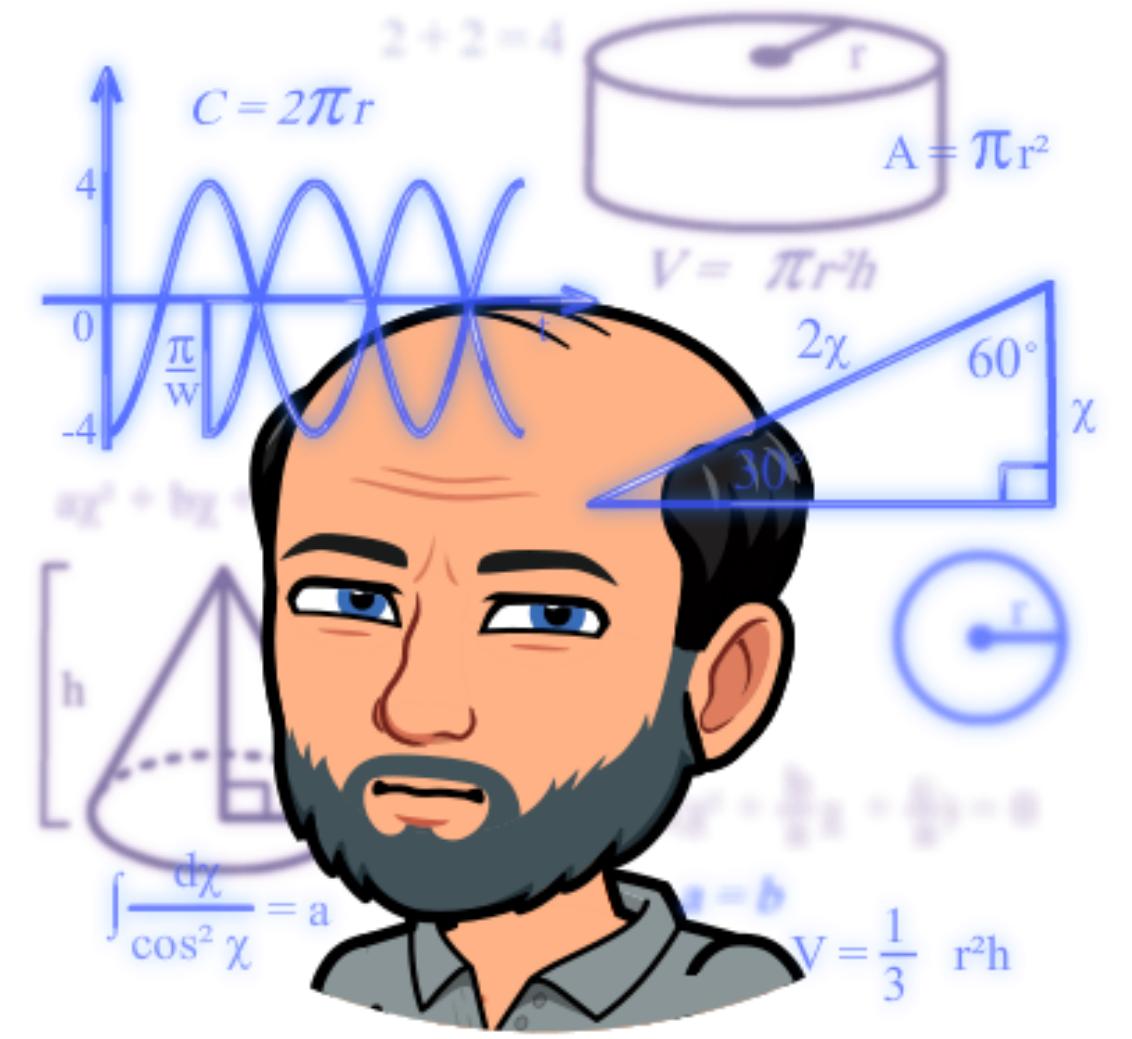
Woodler



ppedv

# Window functions in SQL Server

## - brush up your skills



Thomas Hüttner

# Window functions in SQL Server

Thomas Hütter, Diplom-Betriebswirt

- Developer for ERP apps, SQL scripts and BI stuff
- Worked at consultancies, ISVs, end user companies
- 1995: SQL Server, 2014: R, 2020: Power\*, 2024: Arduino
- Speaker at SQL / data / dev events around Europe



 X @DerFredo <https://twitter.com/DerFredo>

 <https://bsky.app/profile/derfredo.bsky.social>

 [de.linkedin.com/in/derfredo](https://de.linkedin.com/in/derfredo)



{ } NDC  
Conferences



# Agenda

- Introduction: window functions
- Aggregate functions
- Window frames
- Ranking functions
- Statistical/analytical functions
- Offset functions
- Specials
- Round-up, resources, learning

# Introduction: window functions

- „In SQL, a window function or analytic function is a function which uses values from one or multiple rows to return a value for each row.  
(This contrasts with an aggregate function, which returns a single value for multiple rows.)  
Window functions have an OVER clause; any function without an OVER clause is not a window function, but rather an aggregate or single-row (scalar) function.“ \*1)
- „A window function in SQL is a type of function that allows us to perform calculations across a specific set of rows related to the current row. These calculations happen within a defined window of data, and they are particularly useful for aggregates, rankings, and cumulative totals without altering the dataset.“ \*2)
- A window function applies additional filtering to a query, which can be defined by partitioning, ordering, and framing clauses.  
Contrary to a WHERE clause, this works on each row individually.

\*1: [https://en.wikipedia.org/wiki/Window\\_function\\_\(SQL\)](https://en.wikipedia.org/wiki/Window_function_(SQL)), \*2: <https://www.geeksforgeeks.org/window-functions-in-sql>

# Introduction: window functions

- **General form:**

```
SELECT <column_1>, <column_2>,  
<window_function> OVER ( PARTITION BY <...> | ORDER BY <...> | <window_frame>) <column_alias>  
FROM <table_name> ...
```

- **<window\_frame>:**

Rows | Range [ | Groups] BETWEEN <lower\_bound> AND <upper\_bound> [<window frame exclusion>]  
Unbounded preceding | N preceding | Current row | N following | Unbounded following

- **Named window definition (SQL Server 2022):**

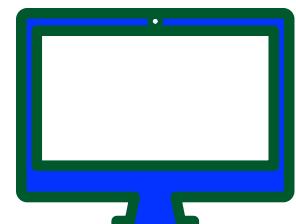
```
SELECT <column_1>, <column_2>,  
<window_function_1>() OVER <window_name> <column_alias_1>  
<window_function_2>() OVER <window_name> <column_alias_2>  
FROM <table_name> ...  
WINDOW <window_name> AS (PARTITION BY <...> | ORDER BY <...> | <window_frame>)
```

# Aggregate functions

The usual suspects:

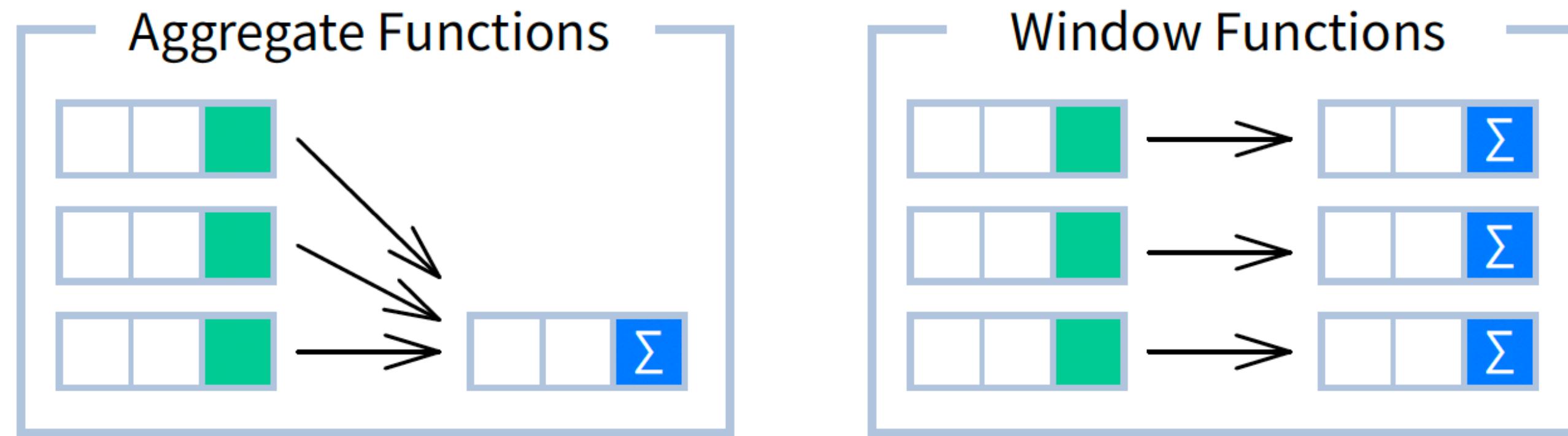
- AVG() Average value
- COUNT(), COUNT\_BIG() Count of values
- MIN(), MAX() Minimum / Maximum of value
- SUM(), PRODUCT() Sum / product of values
- and also VAR(), VARP(), STDEV(), STDEVP(), CHECKSUM\_AGG(int)

work as ‘normal’ aggregates, but also in conjunction with Window functions.



# Aggregate functions

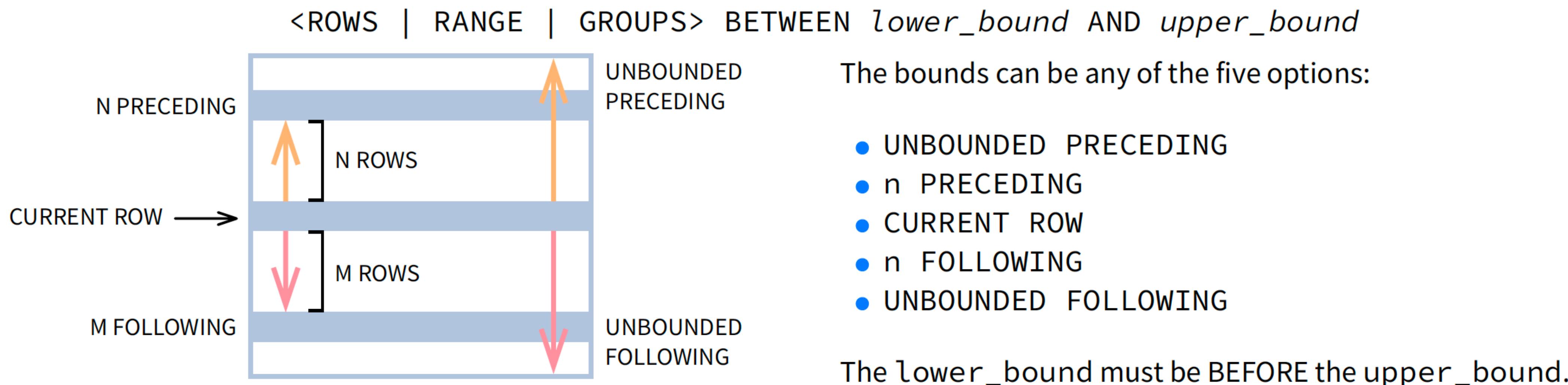
## Grouped aggregates vs Window aggregates



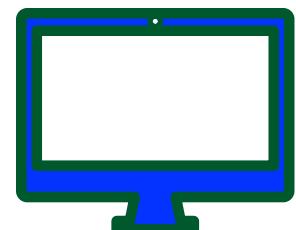
Aggregating on a grouped level vs  
aggregating on criteria relative to each individual row.

# Window frames

- Introduce an additional level of filtering, relative to the current row, applicable to aggregate functions and `First_value`, `Last_value`, `[Nth_value]`



! The default behaviour with an optional ORDER BY is:  
RANGE BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW



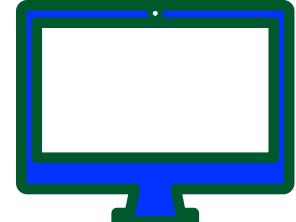
Visualization by: <https://learnsql.com/blog/sql-window-functions-cheat-sheet/>

# Ranking functions

- `ROW_NUMBER()`
  - returns the sequential number of each row within a partition
- `RANK()`
  - similar to `ROW_NUMBER()`, but returns the same value for ties, allows gaps
- `DENSE_RANK()`
  - returns the same rank value for ties, without gaps
- `NTILE(n)`
  - evenly distributes the rows into the specified number of groups,  
returns the group number for each row; i.e. `NTILE(4)` -> quartiles



all of them require `OVER(ORDER BY ...)`



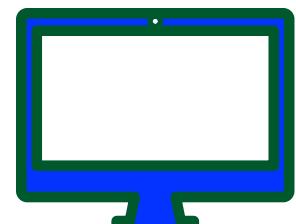
# Statistical/analytical functions

## Rank distribution:

- **PERCENT\_RANK()**  
calculates the relative rank of a row within a partition
- **CUME\_DIST()**  
returns the cumulative distribution of a row within a partition

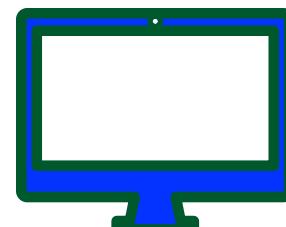
## Inverse distribution:

- **PERCENTILE\_CONT**  
calculates a percentile based on continuous distribution
- **PERCENTILE\_DISC**  
returns the percentile based on discrete distribution



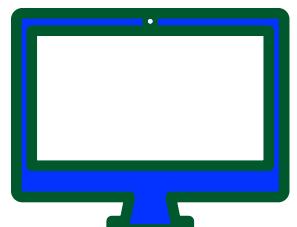
# Offset functions

- `LAG(expr, offset, default)`      value from row before current row
- `LEAD(expr, offset, default)`      value from row after current row
- `FIRST_VALUE()`      by order within the partition (not the smallest value)
- `LAST_VALUE()`      by order within the partition (not the highest value)
- `[NTH_VALUE()]`      not implemented in SQL Server



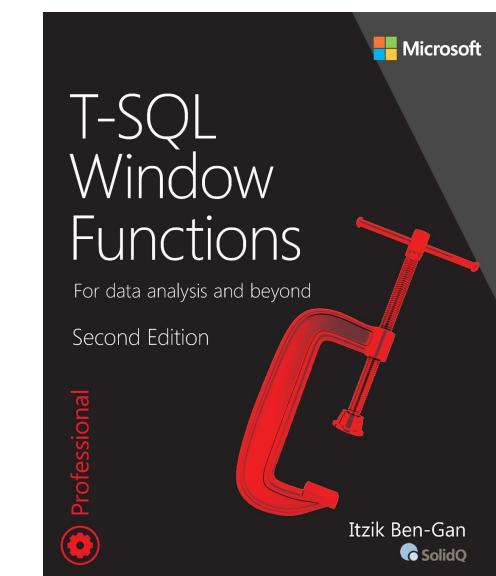
# Specials

- Ordered set functions
  - STRING\_AGG() - in SQL-standard called LISTAGG
  - [Hypothetical set functions] grouped
  - [Inverse distribution functions] grouped
- [Row-pattern recognition] Totally not implemented in SQL Server 😞
- First and last record in a partition
- Gaps and islands



# Resources:

- MS Learn: <https://learn.microsoft.com/en-us/training/modules/write-queries-that-use-window-functions/>  
<https://learn.microsoft.com/en-us/sql/t-sql/queries/select-over-clause-transact-sql/>  
<https://learn.microsoft.com/en-us/sql/t-sql/queries/select-window-transact-sql/>
- LearnSQL (incl. cheat sheet): <https://learnsql.com/blog/sql-window-functions-guide/>
- AW2022 download: <https://learn.microsoft.com/en-us/sql/samples/adventureworks-install-configure>
- AdventureWorks diagram: <https://www.dbdiagrams.com/online-diagrams/adventureworks/>
- Itzik Ben-Gan: „T-SQL Window Functions - For data analysis and beyond“, ISBN 978-0135861448



# Window functions in SQL Server

Thank you for your time and interest & keep in touch:

-  X @DerFredo <https://twitter.com/DerFredo>
-  <https://bsky.app/profile/derfredo.bsky.social>
-  [de.linkedin.com/in/derfredo](https://de.linkedin.com/in/derfredo)



This file and the demo scripts can be found at:

<https://bit.ly/DerFredoErding2025>



Vielen Dank!  
Ich freue mich auf  
Feedback!

Scan  
mich!

