Window Functions in SQL

Last Updated : 20 Nov, 2024

SQL window functions are an essential tool for **database management** and **data analysis**. They allow users to perform **advanced calculations** across a specified set of rows known as a "**window**" while retaining the individual data rows.

Unlike **traditional aggregate functions** that return a single result for an entire group, window functions provide calculations across different partitions or **groups of data**, enabling **complex analysis** without altering the **original dataset**.

In this article, we will cover everything we need to know about **SQL window functions**, including the **OVER clause**, **partitioning**, **ordering**, and **common use cases**.

What is a Window Function in SQL?

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.

The OVER clause is key to defining this window. It partitions the data into different sets (using the **PARTITION BY** clause) and orders them (using the **ORDER BY** clause). These windows enable functions like **SUM()**, **AVG()**, **ROW_NUMBER()**, **RANK()**, and **DENSE_RANK()** to be applied in a sophisticated manner.

```
 OVER([PARTITION BY column_name1] [ORDER BY column_name3]) AS
new_column
FROM table_name;
```

Key Terms

- **window_function**= any aggregate or ranking function
- **column_name1**= column to be selected
- **column_name2**= column on which window function is to be applied
- **column_name3**= column on whose basis partition of rows is to be done
- **new_column**= Name of new column
- **table_name**= Name of table

Types of Window Functions in SQL

SQL window functions can be categorized into two primary types: **aggregate window functions** and **ranking window functions**. These two types serve different purposes but share a common ability to perform calculations over a defined **set of rows** while retaining the original data.

1. Aggregate Window Function

These functions allow us to perform aggregate operations across a set of rows within the defined window, while still retaining the detail-level data.

Common Aggregate Window Functions include:

- **SUM()**
- **AVG()**
- **COUNT()**
- **MAX()**
- **MIN()**

Example of Aggregate Window Function:

Consider an employee table with the following columns: **`Name`**, **`Age`**, **`Department`**, and **`Salary`**.

Name	Age	Department	Salary
Ramesh	20	Finance	50, 000
Deep	25	Sales	30, 000
Suresh	22	Finance	50000
Ram	28	Finance	20, 000
Pradeep	22	Sales	20, 000

Query:

```
SELECT Name, Age, Department, Salary,
 AVG(Salary) OVER( PARTITION BY Department) AS Avg_Salary
 FROM employee
```

Output

Name	Age	Department	Salary	Avg_Salary
Ramesh	20	Finance	50,000	40,000
Suresh	22	Finance	50,000	40,000
Ram	28	Finance	20,000	40,000
Deep	25	Sales	30,000	25,000
Pradeep	22	Sales	20,000	25,000

Explanation:

As shown, the **average salary** within each department remains constant for all rows in the department.

2. Ranking Window Functions

These functions provide rankings of rows within a partition based on specific criteria. Common ranking functions include:

- **RANK()** – As the name suggests, the **rank function** assigns rank to all the rows within every partition. Rank is assigned such that **rank 1**

same rank. For the **next rank** after two same rank values, **one rank value** will be skipped. For instance, if two rows share rank 1, the next row gets rank 3, not 2.

- **DENSE_RANK()** – It assigns rank to each row within partition. Just like rank function first row is assigned rank 1 and rows having same value have same rank. The difference between RANK() and DENSE_RANK() is that in DENSE_RANK(), for the next rank after two same rank, consecutive integer is used, no rank is skipped.

- **ROW_NUMBER()** – ROW_NUMBER() gives each row a **unique number**. It numbers rows from one to the total rows. The rows are put into **groups** based on their values. Each group is called a **partition**. In each partition, rows get numbers one after another. No two rows have the same number in a partition.

Example of Ranking Window Functions

To calculate the **rank**, **dense rank**, and **row number** of employees within each department based on salary, use the following query:

Query:

```
SELECT
    ROW_NUMBER() OVER (PARTITION BY Department ORDER BY Salary
DESC) AS emp_row_no,
```

```
    Name,
    Department,
    Salary,
    RANK() OVER(PARTITION BY Department ORDER BY Salary DESC) AS
emp_rank,
    DENSE_RANK() OVER(PARTITION BY Department ORDER BY Salary
DESC) AS emp_dense_rank
FROM
    employee;
```

Output

emp_row_no	Name	Department	Salary	emp_rank	emp_d
1	Ramesh	Finance	50, 000	1	
2	Suresh	Finance	50, 000	1	
3	Ram	Finance	20, 000	3	
1	Deep	Sales	30, 000	1	
2	Pradeep	Sales	20, 000	2	

Explanation:

So, we can see that as mentioned in the definition of **ROW_NUMBER()** the row numbers are **consecutive integers** within each partition. Also, we can see difference between **rank** and dense rank that in dense rank there is **no gap** between **rank values** while there is gap in rank values after **repeated rank**.

Conclusion

SQL **window functions** are a crucial feature for advanced data analysis and provide flexibility when working with partitioned data. By mastering the **OVER clause**, PARTITION BY, and **ORDER BY**, we can perform complex calculations like **aggregate calculations**, **ranking**, and **cumulative totals** while preserving the row-level data.

Using these **window functions SQL features**, we can perform advanced data analysis tasks with ease. The combination of window functions with **ORDER BY** and **PARTITION BY** provides a flexible approach for data manipulation across **different types of datasets**.

FAQs

What is a window function in SQL?

A window function in SQL performs a calculation across a set of table rows related to the current row within a specified window. Unlike regular aggregate functions, window functions allow you to retain individual rows while performing the calculation.

What is the window function OVER in SQL?

The `OVER` clause in SQL defines the window or range of rows the window function should operate on. It is used with functions like `ROW_NUMBER()`, `RANK()`, or `SUM()` to calculate values across a partition of the result set.

What is the window function SQL LAST?

The `LAST_VALUE()` function in SQL is a window function that returns the last value in a specified window of rows. It allows you to retrieve the last value within a partition, ordered according to the given sorting criteria.

"This course is very well structured and easy to learn. Anyone with zero experience of data science, python or ML can learn from this. This course makes things so easy that anybody can learn on their own. It's helping me a lot. Thanks for creating such a great course."- **Ayushi Jain | Placed at Microsoft**

Now's your chance to unlock high-earning job opportunities as a Data Scientist! Join our **Complete Machine Learning & Data Science Program** and get a 360-degree learning experience mentored by industry experts.

Get hands on practice with **40+ Industry Projects, regular doubt solving sessions**, and much more. Register for the Program today!

Comment More info

Next Article

Pivot and Unpivot in SQL

Window functions in SQL

Similar Reads

SQL Server Window Functions ROWS vs. RANGE

We will cover the important concept of the MS SQL Server which is the difference between Row and Range. The confusion between the row an...

5 min read

How to Find the Length of a Series Using Window Functions in SQL

In data analysis, understanding the sequential patterns within datasets is crucial for informed decision-making. SQL's window functions offer a...

5 min read

Find the Length of a Series Using Window Functions in SQL Server

Window functions in SQL Server are a powerful tool for performing calculations across a series of rows related to the current row. Unlike...

5 min read

Window Functions in PL/SQL

In Oracle PL/SQL, analyzing and managing complex data relationships often involves performing calculations across sets of rows. This is where...

7 min read

Difference between Structured Query Language (SQL) and Transact-...

Structured Query Language (SQL): Structured Query Language (SQL) has a specific design motive for defining, accessing and changement of data. ...

2 min read

Configure SQL Jobs in SQL Server using T-SQL

In this article, we will learn how to configure SQL jobs in SQL Server using T-SQL. Also, we will discuss the parameters of SQL jobs in SQL...

7 min read

MySQL Window Functions

MySQL Window Functions are advanced SQL capabilities that enable expensive calculations across sets of rows related to the current row....

6 min read

Postgre Window Functions

PostgreSQL is an advanced relational database management system, popular for its ability to handle both SQL (structured) and JSON (non-...

6 min read

Functions in MySQL | Numeric, String and Date Time Functions in...

In MySQL, functions play a crucial role in performing various operations on data, such as calculations, string manipulations, and date handling. Thes...

4 min read

SQL | Advanced Functions

SQL (Structured Query Language) offers a wide range of advanced functions that allow you to perform complex calculations, transformation...

2 min read

Article Tags : Databases Experiences Internship SQL +1 More

GeeksforGeeks
Sanchhaya Education Private Limited

Corporate & Communications Address:- A-143, 7th Floor, Sovereign Corporate Tower, Sector- 136, Noida, Uttar Pradesh (201305) | Registered Address:- K 061, Tower K, Gulshan Vivante Apartment, Sector 137, Noida, Gautam Buddh Nagar, Uttar Pradesh, 201305

GET IT ON Google Play Download on the App Store

Company Explore

About Us
Legal
Careers
In Media
Contact Us
Advertise with us
GFG Corporate Solution
Placement Training Program

Job-A-Thon Hiring Challenge
Hack-A-Thon
GfG Weekly Contest
Offline Classes (Delhi/NCR)
DSA in JAVA/C++
Master System Design
Master CP
GeeksforGeeks Videos
Geeks Community

Languages

Python
Java
C++
PHP
GoLang
SQL
R Language
Android Tutorial

DSA

Data Structures
Algorithms
DSA for Beginners
Basic DSA Problems
DSA Roadmap
DSA Interview Questions
Competitive Programming

Data Science & ML

Data Science With Python
Data Science For Beginner
Machine Learning
ML Maths
Data Visualisation
Pandas
NumPy
NLP
Deep Learning

Web Technologies

HTML
CSS
JavaScript
TypeScript
ReactJS
NextJS
NodeJs
Bootstrap
Tailwind CSS

Python Tutorial

Python Programming Examples
Django Tutorial
Python Projects
Python Tkinter
Web Scraping
OpenCV Tutorial
Python Interview Question

Computer Science

GATE CS Notes
Operating Systems
Computer Network
Database Management System
Software Engineering
Digital Logic Design
Engineering Maths

DevOps

Git
AWS
Docker
Kubernetes
Azure
GCP
DevOps Roadmap

System Design

High Level Design
Low Level Design
UML Diagrams
Interview Guide
Design Patterns
OOAD
System Design Bootcamp
Interview Questions

School Subjects

Mathematics
Physics
Chemistry
Biology
Social Science
English Grammar

Commerce

Accountancy
Business Studies
Economics
Management
HR Management
Finance
Income Tax

Databases

SQL
MYSQL
PostgreSQL
PL/SQL
MongoDB

Preparation Corner

Company-Wise Recruitment Process
Resume Templates
Aptitude Preparation
Puzzles
Company-Wise Preparation
Companies
Colleges

Competitive Exams

JEE Advanced
UGC NET
UPSC
SSC CGL
SBI PO
SBI Clerk
IBPS PO
IBPS Clerk

More Tutorials

Software Development
Software Testing
Product Management
Project Management
Linux
Excel
All Cheat Sheets
Recent Articles

Free Online Tools

Typing Test
Image Editor
Code Formatters
Code Converters
Currency Converter
Random Number Generator
Random Password Generator

Write & Earn

Write an Article
Improve an Article
Pick Topics to Write
Share your Experiences
Internships

DSA/Placements

DSA - Self Paced Course

DSA in JavaScript - Self Paced Course

DSA in Python - Self Paced

C Programming Course Online - Learn C with Data Structures

Complete Interview Preparation

Master Competitive Programming

Core CS Subject for Interview Preparation

Mastering System Design: LLD to HLD

Tech Interview 101 - From DSA to System Design [LIVE]

DSA to Development [HYBRID]

Placement Preparation Crash Course [LIVE]

Development/Testing

JavaScript Full Course

React JS Course

React Native Course

Django Web Development Course

Complete Bootstrap Course

Full Stack Development - [LIVE]

JAVA Backend Development - [LIVE]

Complete Software Testing Course [LIVE]

Android Mastery with Kotlin [LIVE]

Machine Learning/Data Science

Complete Machine Learning & Data Science Program - [LIVE]

Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]

Data Science Training Program - [LIVE]

Mastering Generative AI and ChatGPT

Data Science Course with IBM Certification

Programming Languages

C Programming with Data Structures

C++ Programming Course

Java Programming Course

Python Full Course

Clouds/Devops

DevOps Engineering

AWS Solutions Architect Certification

Salesforce Certified Administrator Course

GATE

GATE CS & IT Test Series - 2025

GATE DA Test Series 2025

GATE CS & IT Course - 2025

GATE DA Course 2025

@GeeksforGeeks, Sanchhaya Education Private Limited, All rights reserved