

sql over clause

The `OVER` clause in SQL is a powerful tool used with window functions to compute values over a set of rows, known as a window^{[1] [2] [3]}. It determines the partitioning and ordering of a rowset before the window function is applied^[2]. Unlike `GROUP BY`, which aggregates the entire result set, the `OVER` clause provides control over the window's start and end for each row^[1].

Syntax

The basic syntax of the `OVER` clause is^[1]:

```
SELECT column_name, aggregate_function(column_name) OVER (  
    [PARTITION BY clause]  
    [ORDER BY clause]  
    [ROWS or RANGE clause]  
)  
FROM table_name;
```

- **PARTITION BY:** Divides the result set into partitions^{[1] [2]}.
- **ORDER BY:** Arranges the rows in each partition^{[1] [2]}.
- **ROWS/RANGE:** Defines the window frame^{[1] [2]}.

Key Uses and Benefits

1. **Aggregated Values:** It allows you to compute aggregated values such as moving averages, cumulative aggregates, running totals, or top N per group results^[2].
2. **Complex Data Analysis:** The `OVER` clause is useful when you need to perform complex data analysis tasks^[1].
3. **Efficiency:** Using the `OVER` clause can be more efficient than using subqueries^[2].
4. **Window Definition:** The `OVER()` clause lets you define windows for window functions and the order in which a given window function is applied^[4].

Example

Consider an `Employees` table with columns like `Id`, `Name`, `Department`, and `Salary`. You can compute the total salary in each department using the `OVER` clause^[1]:

```
SELECT Department, Salary, SUM(Salary) OVER (PARTITION BY Department) AS Total  
FROM Employees;
```

This will return the department, salary, and the total salary for each department without grouping the result set^[1].

Functions Used with the OVER Clause

Many functions can be used with the OVER clause, each with specific requirements or allowances for its sub-clauses^[5]:

- ROW_NUMBER()
- RANK()
- DENSE_RANK()
- Aggregate functions like SUM(), AVG(), MIN(), MAX(), and COUNT()^[2]
- NTILE()
- LAG()
- LEAD()
- FIRST_VALUE()
- LAST_VALUE()
- PERCENT_RANK()
- CUME_DIST()

The specific sub-clauses allowed or required depend on the function being used^[5].

**

1. <https://reintech.io/blog/sql-over-clause-tutorial>
2. <https://learn.microsoft.com/en-us/sql/t-sql/queries/select-over-clause-transact-sql?view=sql-server-ver16>
3. <https://www.baeldung.com/sql/over>
4. <https://learnsql.com/blog/what-is-sql-over-clause/>
5. <https://www.sqlservercentral.com/articles/understanding-the-over-clause>