



Mastering SQL: Aggregation Functions & GROUP BY

Welcome to Chapter 2 of our SQL journey! Today, we'll dive into the powerful world of aggregation functions and the crucial GROUP BY clause. These tools are fundamental for summarizing and analyzing data in your databases.

Introduction to Aggregation Functions

Aggregation functions perform calculations on a set of rows and return a single summary value. They are essential for gaining insights from large datasets, allowing us to answer questions like "how many?" or "what's the average?"

Our Sample Data: The Article Table

Throughout this presentation, we will be using the following Article table for our SQL examples:

| | | | | |
|---|----------------------|---|------------|-----|
| 1 | Introduction SQL | 1 | 2024-01-10 | 120 |
| 2 | Fonctions SQL | 1 | 2024-01-15 | 90 |
| 3 | JOIN en SQL | 2 | 2024-02-01 | 150 |
| 4 | GROUP BY expliqué | 1 | 2024-02-10 | 200 |
| 5 | Base de données | 3 | 2024-02-20 | 70 |
| 6 | HAVING en SQL | 3 | 2024-02-25 | 110 |

COUNT(): Counting Rows



Explanation

The `COUNT()` function is used to count the number of rows that match a specified criterion. When used with an asterisk (*), it counts all rows in the table or group, including duplicates and NULL values.



SQL Application

```
SELECT COUNT(*) AS total_articles  
FROM Article;
```



SQL Screen Result

6

MAX(): Finding the Maximum Value

Explanation

The MAX() function retrieves the highest value within a specified column. This is particularly useful for identifying the latest date, largest number, or alphabetically last string in a dataset.

SQL Application

```
SELECT MAX(date_pub) AS  
derniere_publication FROM  
Article;
```

SQL Screen Result

```
2024-02-25
```

AVG(): Calculating the Average

Explanation

The `AVG()` function computes the average (mean) of the numeric values in a column. It ignores NULL values by default, providing an accurate average of the available data.

SQL Application

```
SELECT AVG(nb_vues) AS  
moyenne_vues FROM Article;
```

SQL Screen Result

123.33

MIN(): Identifying the Minimum Value

Explanation

The MIN() function returns the smallest value in a selected column. Similar to MAX(), it can be used with numeric, string, or date data types to find the earliest, smallest, or alphabetically first value.

SQL Application

```
SELECT MIN(nb_vues) AS  
moins_vues FROM Article;
```

SQL Screen Result

70

SUM(): Aggregating Numeric Values

Explanation

The `SUM()` function calculates the total sum of all non-NULL numeric values in a specified column. This is useful for getting a grand total, like the total number of views or sales.



SQL Application

```
SELECT SUM(nb_vues) AS  
total_vues FROM Article;
```



SQL Screen Result

740

GROUP BY: Grouping Your Data

1

Explanation

The `GROUP BY` clause is used to arrange identical data into groups. It's often used with aggregation functions to perform calculations on each group rather than on the entire dataset.

2

SQL Application

```
SELECT id_utilisateur, COUNT(*)  
AS nb_articles FROM Article  
GROUP BY id_utilisateur;
```

3

SQL Screen Result

| | |
|---|---|
| 1 | 3 |
| 2 | 1 |
| 3 | 2 |

HAVING: Filtering Groups



Explanation

The **HAVING** clause is used to filter results based on aggregate functions. Unlike **WHERE**, which filters individual rows, **HAVING** applies conditions to groups created by **GROUP BY**.



SQL Application

```
SELECT id_utilisateur, COUNT(*) AS  
nb_articles FROM Article GROUP BY  
id_utilisateur HAVING COUNT(*) >=  
2;
```



SQL Screen Result

| | |
|---|---|
| 1 | 3 |
| 3 | 2 |

Chapter 2 Summary: Key Takeaways

Aggregation Functions

- COUNT(): Total items
- SUM(): Total value
- AVG(): Average value
- MIN(): Smallest value
- MAX(): Largest value



GROUP BY

Organizes rows into groups based on specified column(s) for aggregate calculations.

HAVING

Filters groups based on conditions applied to aggregate function results.

By mastering these concepts, you can perform powerful data analysis and extraction, transforming raw data into meaningful insights. Keep practicing!