

Order of Execution in SQL



Vikram Singh

Assistant Manager - Content

Updated on Jun 14, 2024 14:20 IST

*An SQL query comprises of various clauses like **SELECT**, **FROM**, **WHERE**, **GROUPBY**, **HAVING**, and **ORDERBY** clauses. Each clause has a specific role in the query. In this article, we will discuss the importance and implication of order of execution in SQL.*

Order of Execution in SQL



SQL, or **Structured Query Language**, is a standard language for managing data (**create/read/update/delete**) in a relational database. Using SQL, you can create a new table (or database) and insert, update, retrieve, or delete data from the existing table.

But you must follow an order while writing any SQL query to fetch or insert data from the table. The order of Execution in SQL is similar to the mathematical operation BODMAS.

In this article, we will learn the importance and implication of order of execution (or



Disclaimer: This PDF is auto-generated based on the information available on Shiksha as on 15-Jun-2024.

SQL order of operation) with the help of an example. Understanding the order of operation allows you to write more efficient and accurate queries. It also helps to predict the outcome of queries, troubleshoot issues, and optimize performance.

Must Check: [SQL Online Course and Certification](#)

Must Check: [Database and SQL Online Course and Certifications](#)

So, let’s begin the article.

Defining Order of Execution in SQL

An SQL query comprises of various clauses like SELECT , FROM, WHERE, GROUPBY, HAVING, and ORDERBY clauses. Each clause has a specific role in the query. Let’s understand each of them briefly.

When you write any query, your query is processed in the following steps:

- Getting Data (FROM/JOIN)
- Row Filter (WHERE)
- Grouping (GROUP BY)
- Group Filter (HAVING)
- Return Expression (SELECT)
- Order & Paging (ORDER BY & LIMIT/OFFSET)

Clause	Function
FROM / JOIN	When you write any query, SQL starts by identifying the tables for the data retrieval and how they are connected.
WHERE	It acts as a filter; it filters the record based on the conditions specified by the users.



GROUP BY	The filtered data is grouped based on the specified condition.
HAVING	It is similar to the WHERE clause but applied after grouping the data.
SELECT	The clause selects the columns to be included in the final result.
DISTINCT	Remove the duplicate rows from the result. Once you apply this clause, you are only left with distinct records.
ORDER BY	It sorts (increasing/decreasing/A->Z/Z->A) the results based on the specified condition.
LIMIT / OFFSET	It determines the number of records to return and from where to start.

Until now, you clearly understand the theoretical aspect of the order of execution in SQL. Now, let's take an example to get a better understanding of the concept.

Let's consider a simple dataset with two tables: **Customers and Orders**.

- The Customers table has 5 columns: **customer_id, first_name, last_name, age, country**.
- Orders Table has 4 columns: **order_id, item, amount, customer_id**

Customers Table

customer_id	first_name	last_name	age	country
1	John	Doe	31	USA
2	Robert	Luna	22	USA
3	David	Robinson	22	UK



customer_id	first_name	last_name	age	country
4	John	Reinhardt	25	UK
5	Betty	Doe	28	UAE

Orders Table

order_id	item	amount	customer_id
1	Keyboard	400	4
2	Mouse	300	4
3	Monitor	12000	3
4	Keyboard	400	1
5	Mousepad	250	2

Problem Statement: Find the amount spent by each customer belonging to the USA.

Copy code

```
SELECT Customers.first_name, Customers.last_name, SUM(Orders.Amount) as
Amount
FROM Customers
JOIN Orders ON Customers.customer_id = Orders.customer_id
WHERE Customers.country = 'USA'
GROUP BY Customers.first_name, Customers.last_name
ORDER BY Amount DESC;
```

Output



first_name	last_name	Amount
John	Doe	400
Robert	Luna	250

Explanation

- **FROM** and **JOIN**: We start by identifying the '**Customers**' and '**Orders**' tables and joining them on '**customer_id**'.
- **WHERE**: It will filter the record to include only those where '**country**' = '**USA**'.
- **GROUP BY**: Group the remaining entries (after filtering by WHERE clause) by '**first_name**' and '**last_name**'.
- **SELECT**: SELECT the '**first_name**', '**last_name**', and the sum of '**Amount**' for each group.
- **ORDER BY**: Finally, the result is sorted by '**Amount**' in descending order.

Now, let's take an example and reshuffle the order of execution in sql.

Case-1: Let you want to filter the record based on the 'Amount' using the WHERE clause.



[Copy code](#)

```
SELECT Customers.first_name, Customers.last_name, SUM(Orders.Amount) as  
Amount  
FROM Customers  
JOIN Orders ON Customers.customer_id = Orders.customer_id  
WHERE Orders.Amount > 300  
GROUP BY Customers.first_name, Customers.last_name  
ORDER BY Amount DESC;
```

Output

first_name	last_name	Amount
David	Robinson	12000
John	Doe	400
John	Reinhardt	400

Case-2: Filter the record based on the 'Amount' using the HAVING clause.



[Copy code](#)

```
SELECT Customers.first_name, Customers.last_name, SUM(Orders.Amount) as  
Amount  
FROM Customers  
JOIN Orders ON Customers.customer_id = Orders.customer_id  
GROUP BY Customers.first_name, Customers.last_name  
HAVING Amount > 300  
ORDER BY Amount DESC;
```

Output

first_name	last_name	Amount
David	Robinson	12000
John	Reinhardt	700
John	Doe	400

Now, let's see what happened in both cases:

Since the WHERE clause is processed before the SELECT clause in the Order of Execution. So, in the first case, SQL won't recognize the Amount and will give the error.

It just filters out the record of the customer who purchased orders greater than 300.

However, the best way to filter the aggregate function is to use the HAVING clause. Since the HAVING clause is processed after the GROUP BY clause. So, in the second case, the HAVING clause filters the group to include only those where the total Amount is greater than 300.

Tips for Writing Efficient SQL Queries



Disclaimer: This PDF is auto-generated based on the information available on Shiksha as on 15-Jun-2024.

- The first thing you must know while writing the SQL queries is the correct order of SQL query execution.
 - Since a lot of people think SQL processes queries from top to bottom as they have written.
 - But SQL processes queries in the order: FROM, JOIN, WHERE, GROUP BY, HAVING, SELECT, DISTINCT, ORDER BY, and finally, LIMIT/OFFSET.
- One of the common mistakes is using aliases defined in the SELECT clause within the WHERE clause.
 - Because SQL processes the WHERE clause before the SELECT clause.
- Use the HAVING clause if you need to filter your query based on the result of an aggregate function.
- While joining multiple tables, start with the smallest table or the table that allows you to filter out the most data early on.

Conclusion

An SQL query comprises of various clauses like SELECT, FROM, WHERE, GROUPBY, HAVING, and ORDERBY clauses. Each clause has a specific role in the query.

In this article, we have learned the importance and implication of order of execution (or SQL order of operation) with the help of an example. Understanding the order of operation allows you to write more efficient and accurate queries. It also helps to predict the outcome of queries, troubleshoot issues, and optimize performance.

Hope you will like the article.

Keep Learning!!

Keep Sharing!!



Related Reads



What is the Difference Between SQL and MySQL?

SQL is query programming language to manage RDBMS while MySQL is RDBMS that uses SQL.

In this article, we will discuss the key differences between SQL (Structured Query Language) and...[read more](#)



Difference between SQL and NoSQL

This article designed for beginners will teach you difference between SQL and NoSQL. The tutorial includes concepts, such as what is a SQL, what is NoSQL, and more.



Difference Between DELETE and TRUNCATE

The DELETE command in SQL deletes specific rows from a table based on a specific condition using the WHERE clause, whereas the TRUNCATE command deletes all the rows from the...[read more](#)



How to Use TRUNCATE Command in SQL?

SQL TRUNCATE TABLE command is used to remove the record from the table. Confused, the SQL DELETE statement also removes the record from the table. So, why use another command...[read more](#)



Delete Statement in SQL

Delete statement is very important statement in SQL. This article covers delete statement with condition and without condition(with proper syntax and examples)



SQL ALTER TABLE : ADD, DROP, MODIFY, RENAME

ALTER TABLE in SQL is used to change the structure of the existing table. In this article, we will briefly discuss how to add, modify, drop, rename columns,



constraints, and...[read more](#)



How to use GROUP BY in SQL?

The GROUP BY clause in sql is used to group the rows that have identical values by one or more columns. In this article, we will briefly discuss how GROUP...[read more](#)



How to use DROP command in SQL?

In SQL, the DROP command is used to delete a database, table, index, or view. To use the DROP command, you must be very careful because it permanently deletes the...[read more](#)



SQL RIGHT JOIN – Example and Syntax

Right Join in the sql returns all the rows from the right table and the matching entries from both tables. In this article, we will discuss how to use right...[read more](#)



Master SQL in No Time: The Ultimate Tutorial for... Beginners!

Forget years of struggle; conquer SQL in weeks! This beginner-friendly guide cuts through the jargon and outlines a crystal-clear path to SQL mastery. No prior coding experience? No problem! Dive...[read more](#)



FOREIGN KEY IN SQL

Foreign key concept is very important concept in SQL. This article explains creation, addition and deletion of foreign key. This article explains foreign key in SQL. It also explains the...[read more](#)





Cross Join in SQL

SQL Cross Join combines each row of one table with each row of another table and the number of rows in the result set is the product of number of...[read more](#)

FAQs

What is the full form of SQL?



What is the order of execution in SQL?



What is the correct order of execution for a given query?



What order does ORDER BY query in SQL?

