



SQL Standard

SQL Fetch

7 months ago • by John Otieno

When we are working in an SQL database, we may come across instances where we need to fetch a specific subset of rows from a given table. This allows us to limit the resources needed to fetch the values from the table.

Luckily, in SQL, we have access to the LIMIT clause which allows us to control the number of rows returned within a given result set. This feature is very useful when we are dealing with a large data set and do not actually need the entire rows but instead a subset of it. This can be either to get the layout of the data or presentation.

However, did you know that LIMIT is not part of Standard SQL definition? Although it is widely adopted by most databases, it is not part of the SQL standard. That part is taken by the FETCH clause.

In this tutorial, we will learn how we can use and work with the FETCH clause in SQL databases to specify the number of rows that we wish to retrieve from the database.

SQL FETCH Clause

In SQL:2008, the OFFSET FETCH clause is introduced which serves a function that is akin to the LIMIT clause.

With the OFFSET FETCH clause, you can skip the initial N rows within a result set before commencing the retrieval of any rows.

The syntax is as follows:

```
OFFSET offset_rows { ROW | ROWS }  
FETCH { FIRST | NEXT } [ fetch_rows ] { ROW | ROWS } ONLY
```

In the given syntax:

- a. The ROW and ROWS/FIRST and NEXT serve as synonyms which allow for interchangeable usage.
- b. The OFFSET_ROWS represents an integer number which must be zero or a positive value. If the “offset_row” exceeds the total number of rows in the result set, the query returns no rows.
- c. The FETCH_ROWS specifies an integer number that indicates the number of rows to retrieve, with its value being one or greater.

NOTE: The OFFSET FETCH clause is supported by various database engines such as Oracle, PostgreSQL, Microsoft SQL Server, etc. Nevertheless, it's worth noting that each of these database engines implements the OFFSET FETCH clause with some variations.

Example:

Let us setup a basic table that can help us demonstrate how to use the OFFSET FETCH clause. The “create” and “insert” statements are as follows:

```
CREATE TABLE employees (  
    employee_id INT AUTO_INCREMENT PRIMARY KEY,  
    first_name VARCHAR(50),  
    last_name VARCHAR(50),  
    department VARCHAR(50)  
);  
INSERT INTO employees (first_name, last_name, department) VALUES  
    ('Alice', 'Smith', 'Human Resources'),  
    ('Bob', 'Johnson', 'Marketing'),  
    ('Charlie', 'Wilson', 'Finance'),  
    ('David', 'Brown', 'Sales'),  
    ('Eva', 'Davis', 'Engineering');
```

Once we have the table setup, we can proceed and learn how to use this clause.

The following example shows how to use the OFFSET FETCH clause to skip the first three rows:

```
select  
    *  
from  
    employees offset 3 rows fetch next 1 rows only;
```

This should skip the first three rows from the table and fetch the fourth one only.

Conclusion

In this tutorial, we learned how to use the OFFSET FETCH clause in SQL to limit and skip the number of rows from a “select” statement. It is similar to the LIMIT clause but it is part of the Standard SQL definitions.