# SQL | LIMIT Clause



SQL limit clause is very useful in some scenarios where we really need the data in some sorted manner suppose if there are a large number of tuples satisfying the query conditions, it might be resourceful to view only a handful of them at a time.

**Important points to remember:**

- The LIMIT clause is used to set an upper limit on the number of tuples returned by SQL.
- It is important to note that this clause is not supported by all SQL versions.
- The LIMIT clause can also be specified using the SQL 2008 OFFSET/FETCH FIRST clauses.
- The limit/offset expressions must be a non-negative integer.

**Example:**

Let's assume that we have one sample table name Student and to understand better we will see some query about limit clause.Say we have a relationship, Student.



**Student Table:**

```
CREATE TABLE student (
  id INT PRIMARY KEY,
  name VARCHAR(50),
  age INT
);

INSERT INTO student (id, name, age)
VALUES (1, 'Shubham Thakur', 18),
       (2, 'Aman Chopra', 19),
```

```
(3, 'Bhavika uppala', 20),
(4,'Anshi Shrivastava',22);
```

**Output:**

| id | name | age |
|---|---|---|
| 1 | Shubham Thakur | 18 |
| 2 | Aman Chopra | 19 |
| 3 | Bhavika uppala | 20 |
| 4 | Anshi Shrivastava | 22 |

**Queries:**

```
SELECT *
FROM student
LIMIT 3;
```

**Output:**

| id | name | age |
|---|---|---|
| 1 | Shubham Thakur | 18 |
| 2 | Aman Chopra | 19 |
| 3 | Bhavika uppala | 20 |

Other Query to check with ORDER BY Clause:

```
SELECT *
FROM Student
ORDER BY Grade DESC
LIMIT 3;
```

**Output:**

| id | name | age |
|---|---|---|
| 4 | Anshi Shrivastava | 22 |
| 3 | Bhavika uppala | 20 |
| 2 | Aman Chopra | 19 |

The LIMIT operator can be used in situations such as the above, where we need to find the top 3 students in a class and do not want to use any conditional statements.

## Using LIMIT along with OFFSET

LIMIT x OFFSET y simply means skip the first y entries and then return the next x entries. OFFSET can only be used with the ORDER BY clause. It cannot be used on its own. OFFSET value must be greater than or equal to zero. It cannot be negative, else returns an error.

**Queries:**

```
SELECT *
FROM Student
ORDER BY ROLLNO LIMIT 5 OFFSET 2;

or

SELECT *
FROM Student
ORDER BY ROLLNO LIMIT 2,5; # it skips the
first 2 values and then return the next 5 entries
```

The first query and second query return the same results. In the second query, limit is followed by two values. LIMIT X, Y The first value X is the offset value (skips X number of entries) and the second value Y is the limit (it returns the next Y number of entries).

**Output:**

| id | name | age |
|---|---|---|
| 3 | Bhavika uppala | 20 |
| 4 | Anshi Shrivastava | 22 |

## SQL LIMIT to Get the nth Highest or Lowest Value

Now we will look for LIMIT use in finding highest or lowest value we need to retrieve the rows with the nth highest or lowest value. In that situation, we can use the subsequent MySQL LIMIT clause to obtain the desired outcome.

**Syntax:**

*SELECT column_list*

*FROM table_name*

*ORDER BY expression*

*LIMIT n-1, 1;*

**Query:**

```
SELECT age FROM Student
ORDER BY age  LIMIT 2, 1;
```

**Output:**

| age |
| --- |
| 20 |

## The Limit in MySQL with Where

The WHERE clause can also be used with MySQL Limit. It produces the rows that matched the condition after checking the specified condition in the table.

**Query:**

```
SELECT age
FROM Student
WHERE id<4
ORDER BY age
LIMIT 2, 1;
```

**Output:**

| age |
| --- |
| 20 |

## Restrictions on the LIMIT clause

The LIMIT clause's limitations. The following situations do not allow the LIMIT clause to be used:

- With regard to defining a view
- The use of nested SELECT statements
- Except for subqueries with table expressions specified in the FROM clause.
- Embedded SELECT statements are used as expressions in a singleton SELECT (where max = 1) within an SPL routine where embedded SELECT statements are used as expressions.

This article is contributed by **Anannya Uberoi**. If you like GeeksforGeeks and would like to contribute, you can also write an article using write.geeksforgeeks.org or mail your article to review-team@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please write comments if you find anything incorrect, or if you want to share more information about the topic discussed above.