

SQL SELECT LAST

SQL (**Structured Query Language**), is the most popular standard language to work on databases. It is a domain-specific language that is mostly used to perform tons of operations which include creating a database, storing data in the form of tables, modifying, extract and lot more. There are different versions of SQL like MYSQL, PostgreSQL, Oracle, SQL lite, etc.

SQL became the norm of **American National Standards Institute (ANSI)** in the year 1986 and the next year in 1987 it became the norm of the **International Organization for Standardization (ISO)**. Today is the world of the Internet and technology. We are surrounded by tons of data. So to store this data securely and to manage them we need a proper database and to manage this database we need a language which is SQL.

In this article, we are going to look at how to get the **last entry** of any record in a Table. We are going to discuss **four different ways** to extract the last entry of any given table in a database.

ID	Age	Student Name	Sex
1	22	Harry	Male
2	23	Vishal	Male
3	20	Snehal	Female
4	25	Ram	Male
5	24	Hina	Female

1. Creating a Database

```
CREATE DATABASE [database_name];
```

2. Creating a Table

```
CREATE TABLE [table_name] (  
  [col_1] [type] [col_1] [constraint],  
  [col_2] [type] [col_2] [constraint]  
  .....  
);
```

col: Column name

TYPE: Data type whether an integer, variable character, etc.

col_constraint: Constraints in SQL like PRIMARY KEY, NOT NULL, UNIQUE, REFERENCES, etc.

3. Inserting Into A Table

```
INSERT INTO [table_name]
VALUES (val_1, val_2, val_3, .....);

val: Values in particular column.
```

4. View The Table

```
SELECT * FROM [table_name]
```

Method 1: Using MS Access

We can use the command FIRST() to extract the first entry of a particular column and LAST() to extract the last entry of a particular column in a Table. For more information visit [First\(\) and Last\(\)Function in MS Access](#).

Basic Syntax

```
LAST (expression)
```

For example, say we want to extract the last student name from the table “Student Information”

```
SELECT LAST(Student_name) AS Stud_Name
FROM StudentInformation;
```

Output



Name

As we can see, the last student name “Hina” is extracted using the above query. But it is to be noted that SELECT LAST or LAST(expression) is only supported in MS Access. This statement doesn’t support MYSQL, Oracle, PostgreSQL, etc. There are alternate ways as discussed below to perform the above operation in other versions of SQL like MYSQL, Oracle, PostgreSQL, etc.

Method 2: By Sorting The Data

We can use the **ORDER BY** statement and **LIMIT** clause to extract the last data. The basic idea is to sort the table in descending order and then we will limit the number of rows to 1. In this way, we will get the output as the last row of the table. And then we can select the entry which we want to retrieve.

MYSQL syntax

```
SELECT col_name(s) FROM Table_Name
ORDER BY appr_col_name DESC
LIMIT 1;
```

col_name(s): The name of the column(s).

appr_col_name: Appropriate column name to perform ORDER BY.

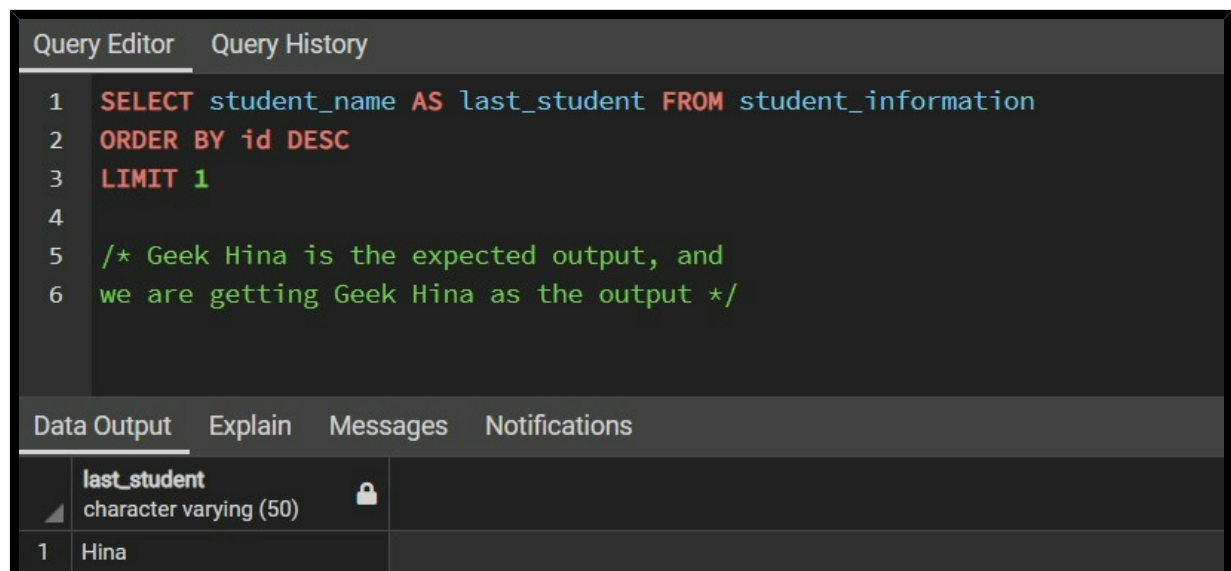
Oracle syntax

```
SELECT col_name(s) FROM Table_Name
ORDER BY appr_col_name DESC
WHERE ROWNUM <= 1;
```

col_name(s): The name of the column(s).

appr_col_name: Appropriate column name to perform ORDER BY.

Output



The screenshot shows a database interface with a 'Query Editor' and a 'Query History' tab. The query editor contains the following SQL code:

```
1 SELECT student_name AS last_student FROM student_information
2 ORDER BY id DESC
3 LIMIT 1
4
5 /* Geek Hina is the expected output, and
6 we are getting Geek Hina as the output */
```

Below the query editor, there are tabs for 'Data Output', 'Explain', 'Messages', and 'Notifications'. The 'Data Output' tab is active, showing a table with one column, 'last_student', and one row, 'Hina'.

	last_student
1	Hina

Last Student Name

It is important to note that in order to perform sorting, **the column needs to be chosen properly**. For example, if we choose “ORDER BY Student_name DESC” then it will alphabetically sort the table on the basis of names. So, the row containing “Vishal” will come at the top, but the row having “Vishal” as entry is not the last row of the table. Also, we can’t use the column “Age” to perform ORDER BY as shown below:

Query Editor		Query History	
1	SELECT student_name AS last_student FROM student_information		
2	ORDER BY age DESC		
3	LIMIT 1		
4			
5	/* Geek Hina should have been the output, but here order by		
6	will sort one the basis of age in descending order. So, we get		
7	Ram having maximum age = 25 as output */		
Data Output		Explain	Messages
	last_student character varying (50)		
1	Ram		

Ram is not the Last Student Name

Query Editor		Query History	
1	SELECT student_name AS last_student FROM student_information		
2	ORDER BY student_name DESC		
3	LIMIT 1		
4			
5	/* Geek Hina should have been the output, but here order by		
6	will sort alphabetically from Z to A, so, we get		
7	Vishal as output */		
Data Output		Explain	Messages
	last_student character varying (50)		
1	Vishal		

Vishal Is not the Last Student Name

Hence, it is mandatory to use the **column ID** or any column which is unique and sequentially increasing with every record in the table.

Method 3: By Using Subquery And AGGREGATE MAX()

Subquery is nothing but a query inside another query that maintains a parent-child relationship. The inner query will execute first followed by the outer query. Here, in this method, the basic idea is to get the maximum ID using aggregate function **MAX** and then select the student name associated with that maximum ID. In this way, we can extract the last student's name from the table.

```
SELECT col_name(s) FROM Table_Name
WHERE appr_col_name = (
    SELECT MAX(appr_col_name)
    FROM Table_Name
);
```

col_name(s): The name of the column(s).
appr_col_name: Appropriate column name. For example ID.

Output

The screenshot shows a SQL Query Editor with two tabs: 'Query Editor' and 'Query History'. The 'Query Editor' tab is active, displaying a SQL query across six lines. Below the query editor, there are four tabs: 'Data Output', 'Explain', 'Messages', and 'Notifications'. The 'Data Output' tab is active, showing a table with one column, 'last_student', which is a character varying (50) type. The table contains one row with the value 'Hina'.

```
1 SELECT student_name AS last_student FROM
2 student_information WHERE
3 id=(SELECT MAX(id) FROM student_information)
4
5 /* Geek Hina is the expected output, and
6 we are getting Geek Hina as the output */
```

	last_student character varying (50)
1	Hina

Last Student Name

Method 4: By Comparative Or Relative Approach

In this method also, we are going to use a subquery. The basic idea is to filter out the rows and to check that no such row exists having higher ID values than the row we are going to extract. This will help in getting the row with maximum ID and hence we can retrieve the last information. It is a complex query and is an iterative approach in which we are going to use the NOT EXISTS statement. This method will take more time to execute if there are more records in the table.

```
SELECT col_name(s) FROM Table_Name t1
WHERE NOT EXISTS (
    SELECT * FROM Table_Name t2
    WHERE t2.appr_col_name > t1.appr_col_name
);
```

col_name(s): The name of the column(s).
appr_col_name: Appropriate column name. For example ID.

Output

Query EditorQuery History

1SELECT student_name AS last_student FROM

2student_information S1 WHERE NOT EXISTS

3(SELECT * FROM student_information S2

4WHERE S2.id>S1.id)

5

6/* Geek Hina is the expected output, and

7we are getting Geek Hina as the output */

Data OutputExplainMessagesNotifications

last_student

character varying (50)

1Hina

Last Student Name