CONTENTS

Particulars

1. INTRODUCTION

1.1 FUNCTIONS

A function is a set of SQL statements that perform a specific task. Functions foster code reusability. If you have to repeatedly write large SQL scripts to perform the same task, you can create a function that performs that task. Next time instead of rewriting the SQL, you can simply call that function. A function accepts inputs in the form of parameters and returns a value. SQL Server comes with a set of built-in functions that perform a variety of tasks.

FUNCTION CATEGORY	DESCRIPTION
Configuration functions	Return information about the current configuration
Conversion functions	Support data type casting and converting
Cursor functions	Return information about cursors
Date and Time Data	Perform operations on a date and time input
Types and Functions values	Values and return string, numeric, or date and time
Logical Functions	Perform logical operations
Mathematical Functions	Performs calculations based on input values provided
Metadata Functions objects	Return information about the database and database objects
Security Functions	Return information about users and roles
String Functions	Performs operations on a string input value and return a string or numeric value
System Functions	Performs operations and return information about values,objects,and settings in an instance of SQL server

2. AGGREGATE FUNCTIONS

An aggregate function performs a calculation on a set of values, and returns a single value. Except for COUNT(*), aggregate functions ignore null values. Aggregate functions are often used with the GROUP BY clause of the SELECT statement.

All aggregate functions are deterministic. In other words, aggregate functions return the same value each time that they are called, when called with a specific set of input values. See Deterministic and Nondeterministic Functions for more information about function determinism. The OVER clause may follow all aggregate functions, except the STRING_AGG, GROUPING or GROUPING_ID functions.

VARIOUS AGGREGATE FUNCIONS

1. Count() : Returns total number of records
2. Sum() : Sum all Non Null values of Column
3. Avg() : avg returns average of records
4. Min() : Minimum value in the selected column
5. Max() : Maximum value in the selected column

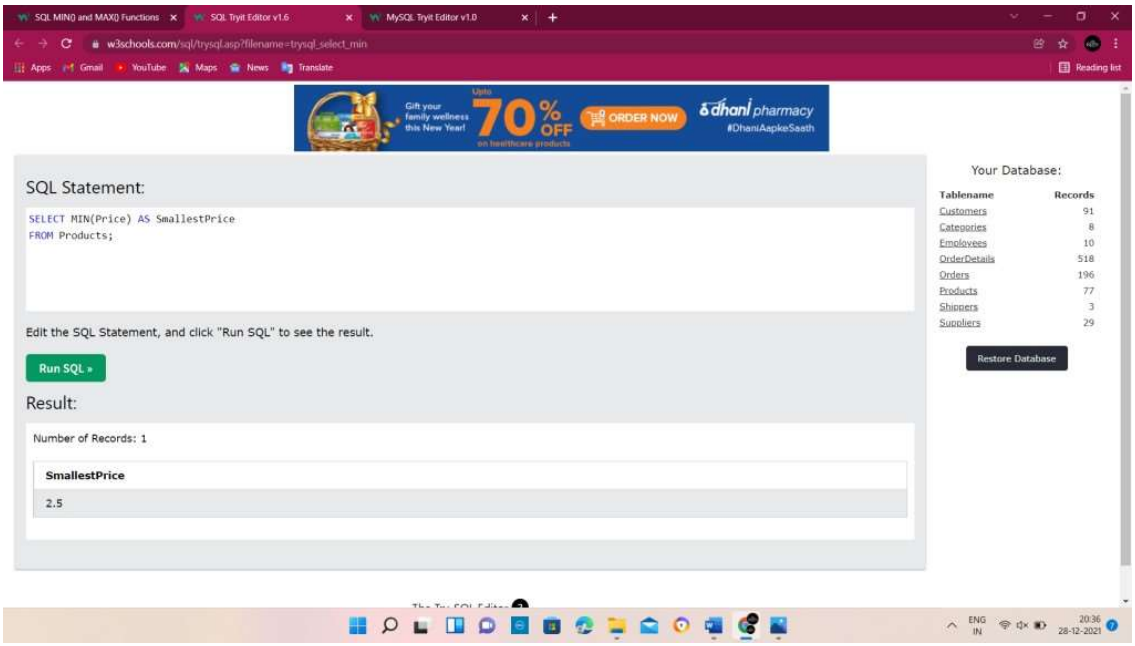

Fig . 7.1

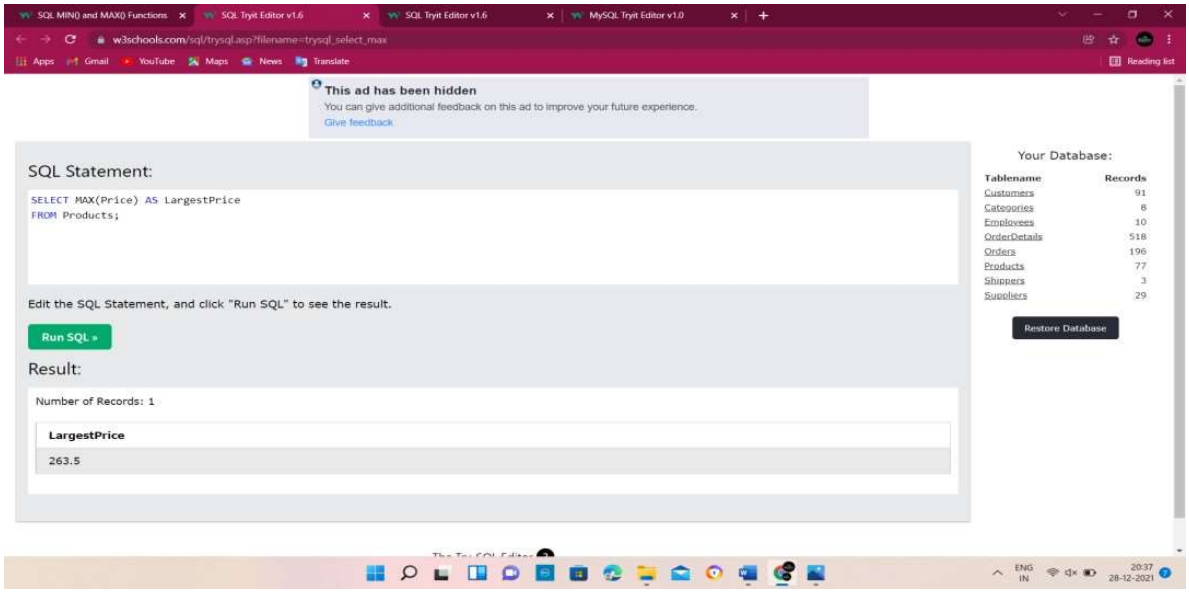
SQL MIN() and MAX() Functions
SQL Tryit Editor v1.6
SQL Tryit Editor v1.6
MySQL Tryit Editor v1.0
w3schools.com/sql/trysql.asp?filename=trysql_select_max
Apps
Gmail
YouTube
Maps
News
Translate
Reading list
This ad has been hidden
You can give additional feedback on this ad to improve your future experience.
Give feedback
SQL Statement:
SELECT MAX(Price) AS LargestPrice
FROM Products;
Edit the SQL Statement, and click "Run SQL" to see the result.
Run SQL »
Result:
Number of Records: 1
LargestPrice
263.5
Your Database:
Tablename
Records
Customers 91
Categories 8
Employees 10
OrderDetails 518
Orders 196
Products 77
Shippers 3
Suppliers 29
Restore Database
ENG
IN
20:37
28-12-2021

Fig. 8.1

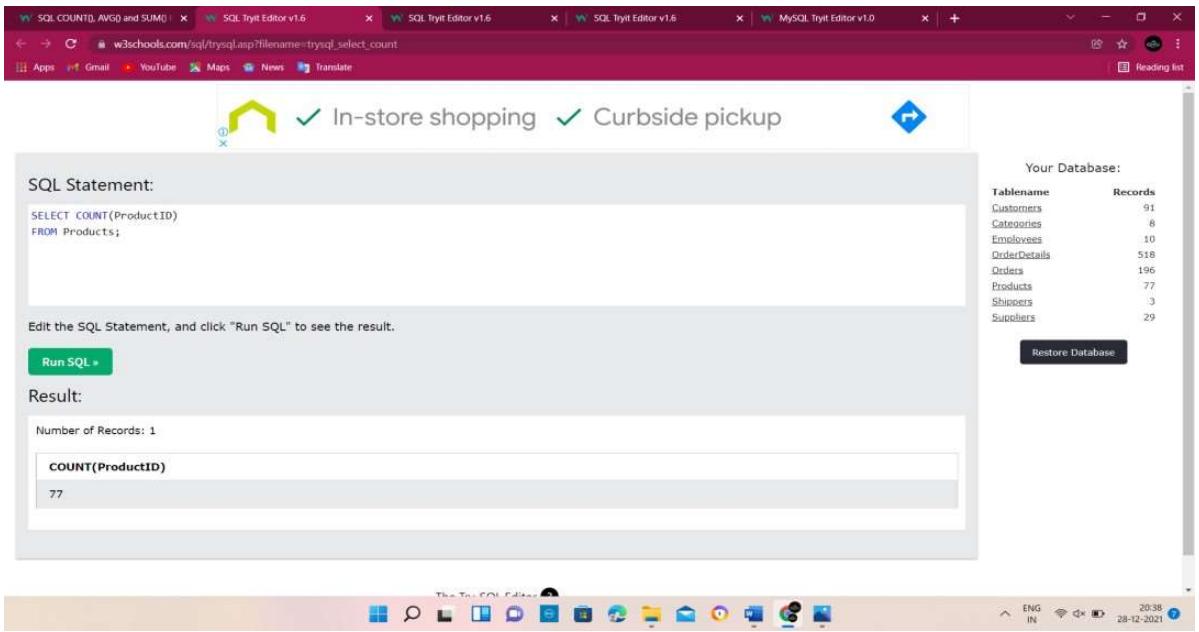
SQL COUNT(), AVG() and SUM()
SQL Tryit Editor v1.6
SQL Tryit Editor v1.6
SQL Tryit Editor v1.6
MySQL Tryit Editor v1.0
w3schools.com/sql/trysql.asp?filename=trysql_select_count
Apps
Gmail
YouTube
Maps
News
Translate
Reading list
In-store shopping
Curbside pickup
SQL Statement:
SELECT COUNT(ProductID)
FROM Products;
Edit the SQL Statement, and click "Run SQL" to see the result.
Run SQL »
Result:
Number of Records: 1
COUNT(ProductID)
77
Your Database:
Tablename
Records
Customers 91
Categories 8
Employees 10
OrderDetails 518
Orders 196
Products 77
Shippers 3
Suppliers 29
Restore Database
ENG
IN
20:38
28-12-2021

Fig. 8.2

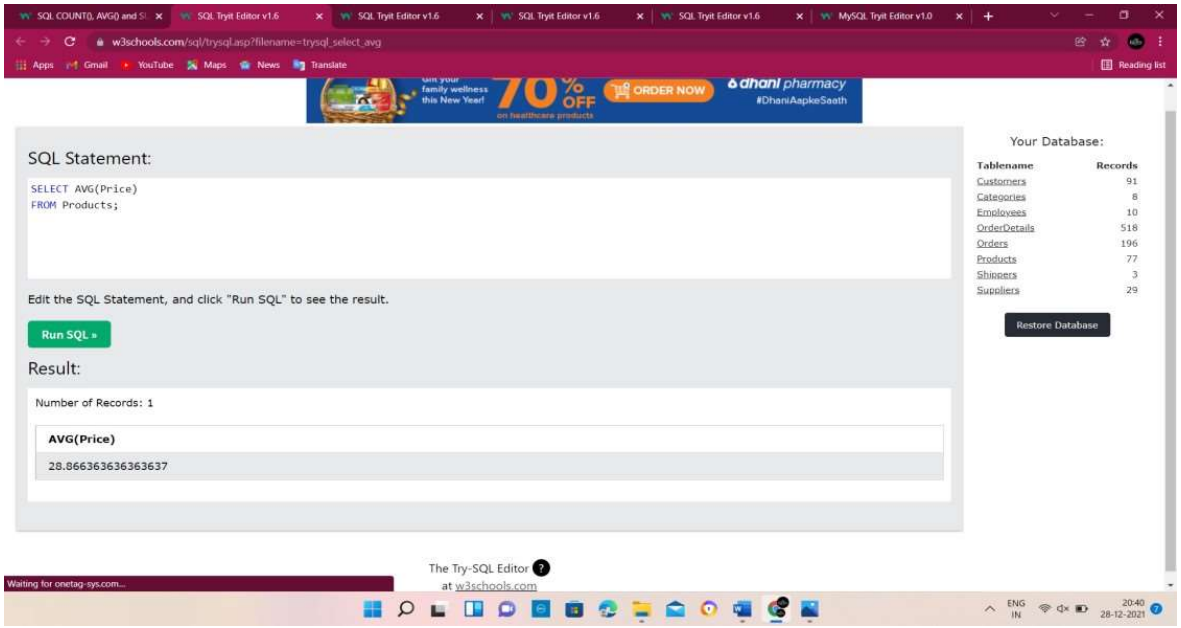
w3schools.com/sql/trysql.asp?filename=trysql_select_avg
SQL Statement:
SELECT AVG(Price)
FROM Products;
Edit the SQL Statement, and click "Run SQL" to see the result.
Run SQL »
Result:
Number of Records: 1
AVG(Price)
28.866363636363637
Your Database:
Tablename Records
Customers 91
Categories 8
Employees 10
OrderDetails 518
Orders 196
Products 77
Shippers 3
Suppliers 29
Restore Database
The Try-SQL Editor
at w3schools.com

Fig. 9.1

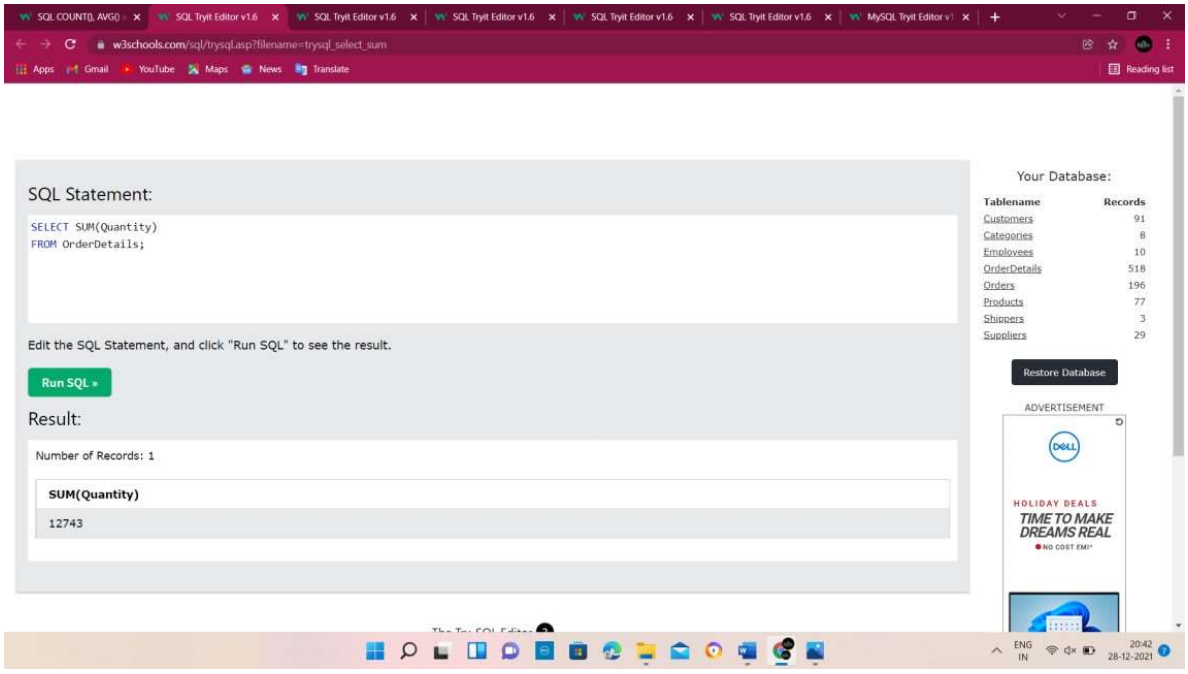
w3schools.com/sql/trysql.asp?filename=trysql_select_sum
SQL Statement:
SELECT SUM(Quantity)
FROM OrderDetails;
Edit the SQL Statement, and click "Run SQL" to see the result.
Run SQL »
Result:
Number of Records: 1
SUM(Quantity)
12743
Your Database:
Tablename Records
Customers 91
Categories 8
Employees 10
OrderDetails 518
Orders 196
Products 77
Shippers 3
Suppliers 29
Restore Database
ADVERTISEMENT
HOLIDAY DEALS
TIME TO MAKE DREAMS REAL

Fig. 9.2

3. NUMBER FUNCTIONS

Numeric Functions are used to perform operations on numbers and return numbers.

Following are the numeric functions defined in SQL:

1. ABS(): It returns the absolute value of a number.

Syntax: SELECT ABS(-243.5);

Output: 243.5

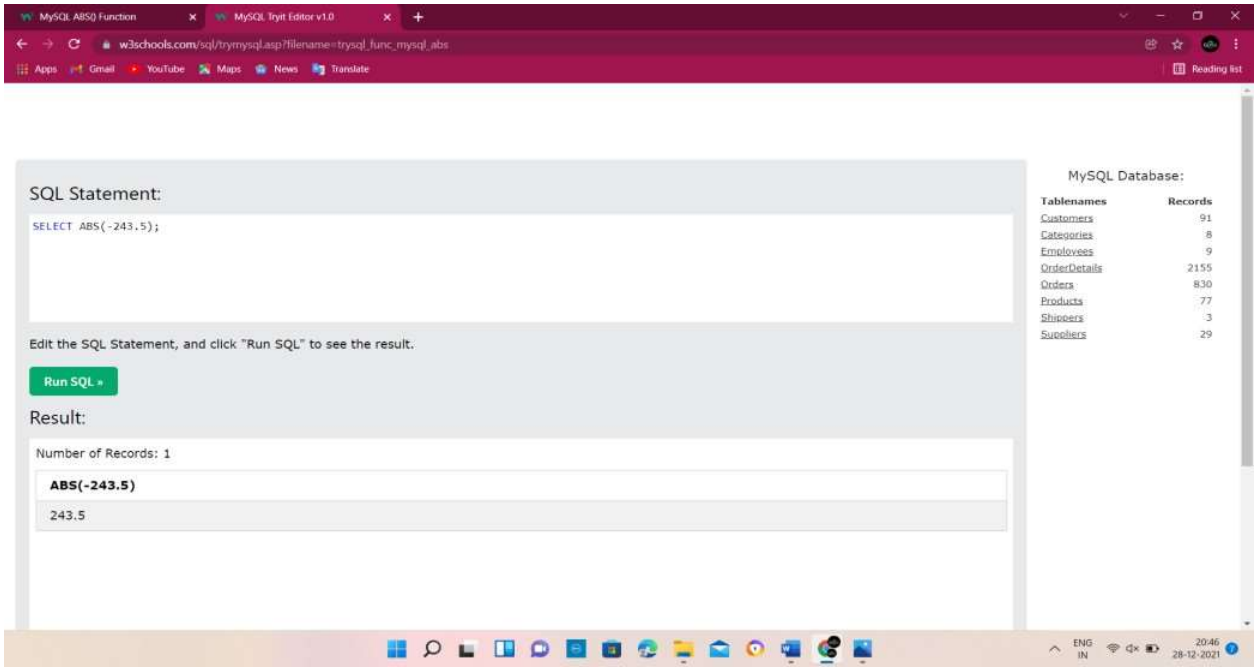

Fig. 10.1

2. ACOS(): It returns the cosine of a number.

Syntax: SELECT ACOS(0.25);

Output: 1.318116071652818

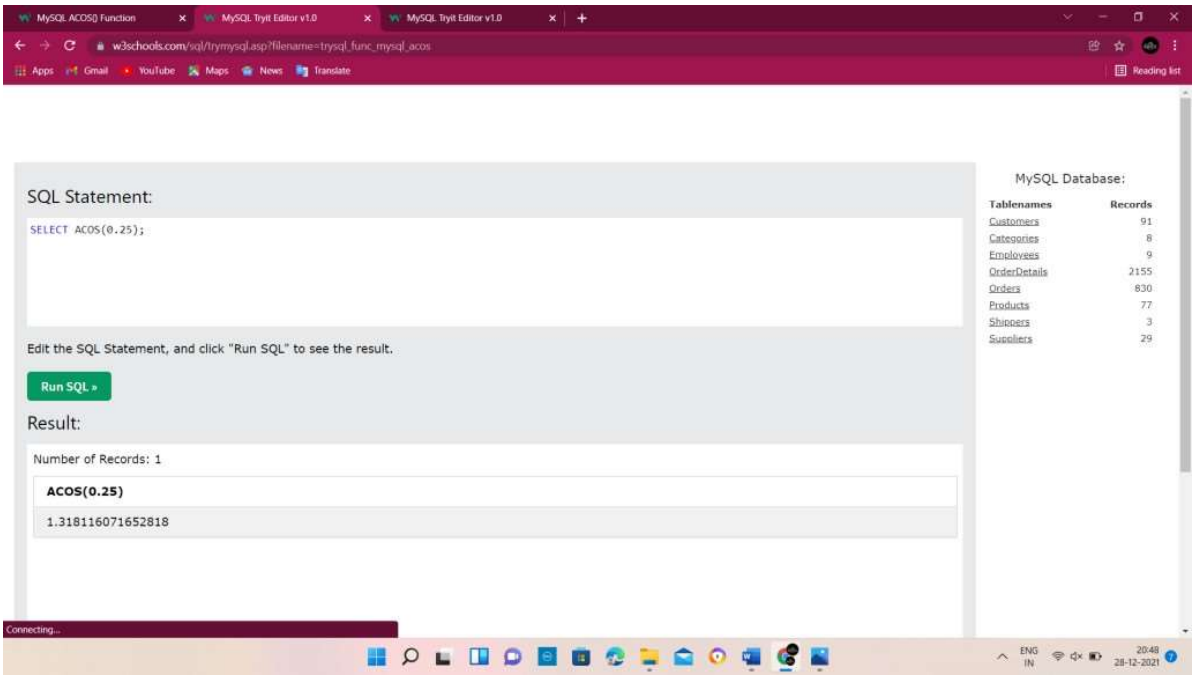

Fig. 11.1

3. ASIN(): It returns the arc sine of a number.

Syntax: SELECT ASIN(0.25);

Output: 0.25268025514207865

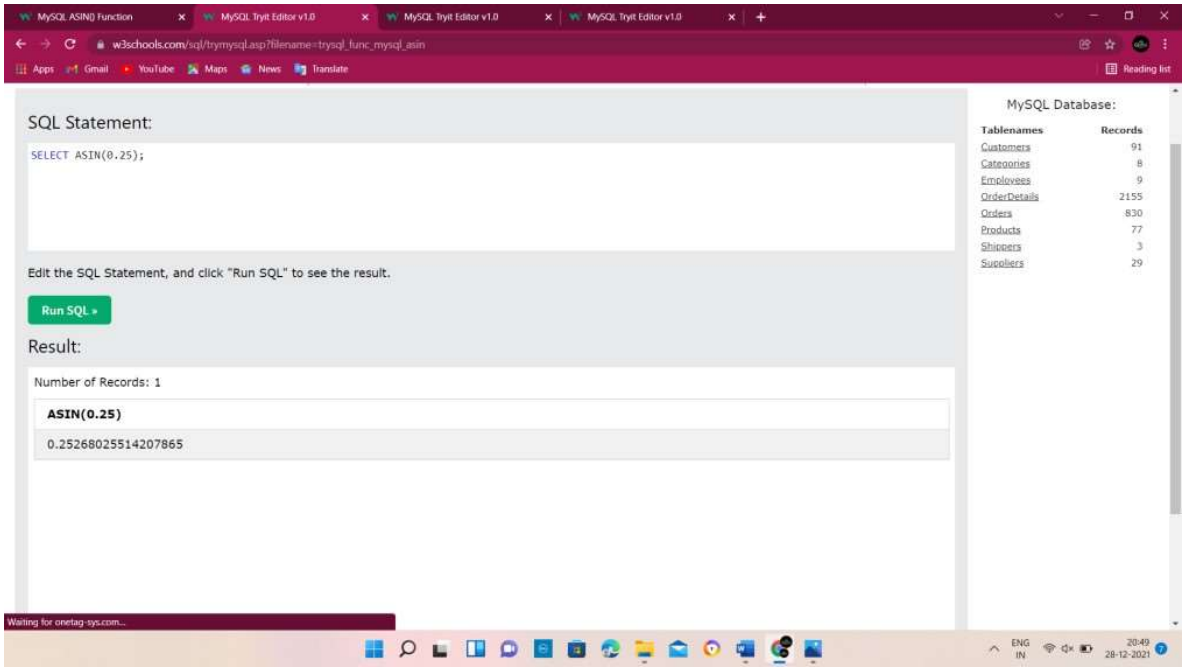

Fig. 11.2

4. CEIL(): It returns the smallest integer value that is greater than or equal to a number.

Syntax: SELECT CEIL(25.75);

Output: 26

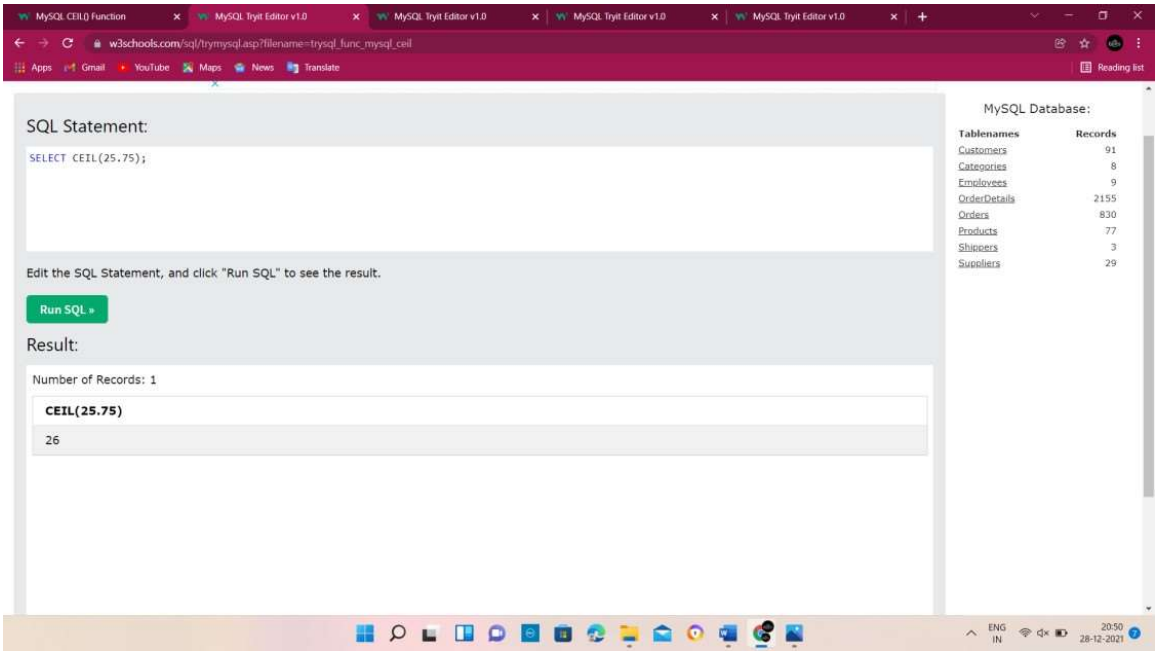

Fig. 12.1

5. COS(): It returns the cosine of a number.

Syntax: SELECT COS(30);

Output: 0.15425144988758405

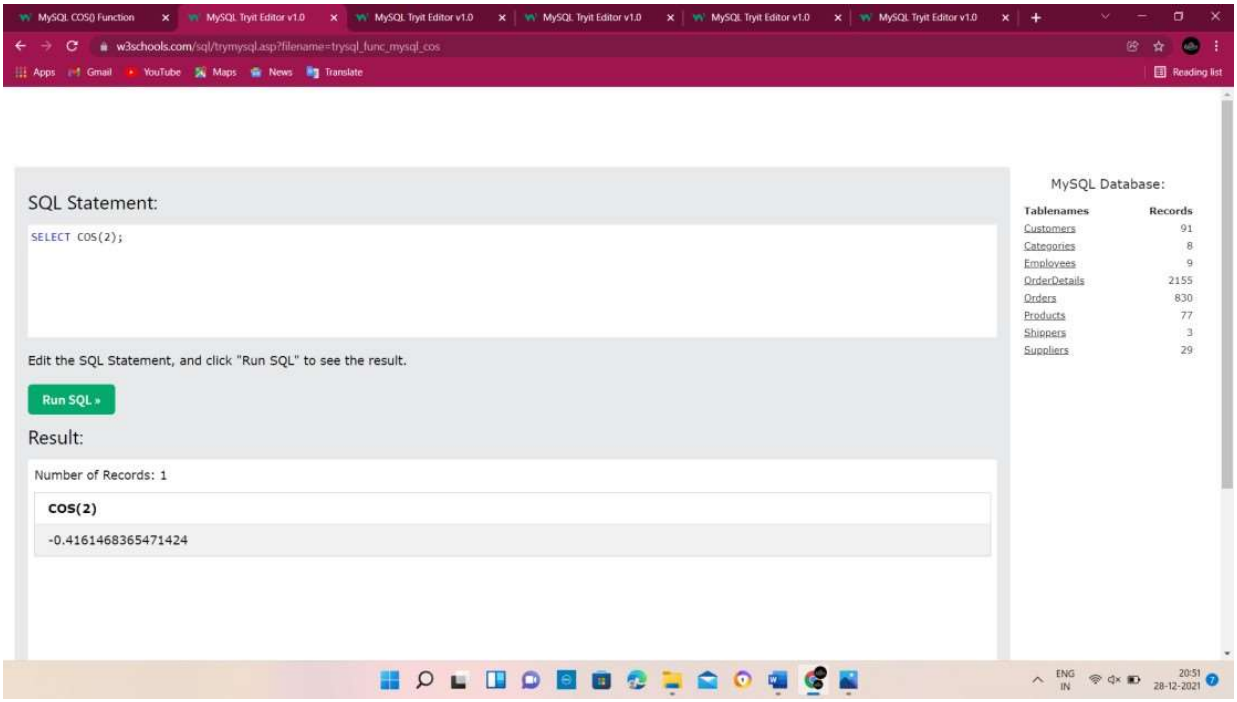

Fig. 13.1

6. COT(): It returns the cotangent of a number.

syntax: SELECT COT(6);

Output: -3.436353004180128

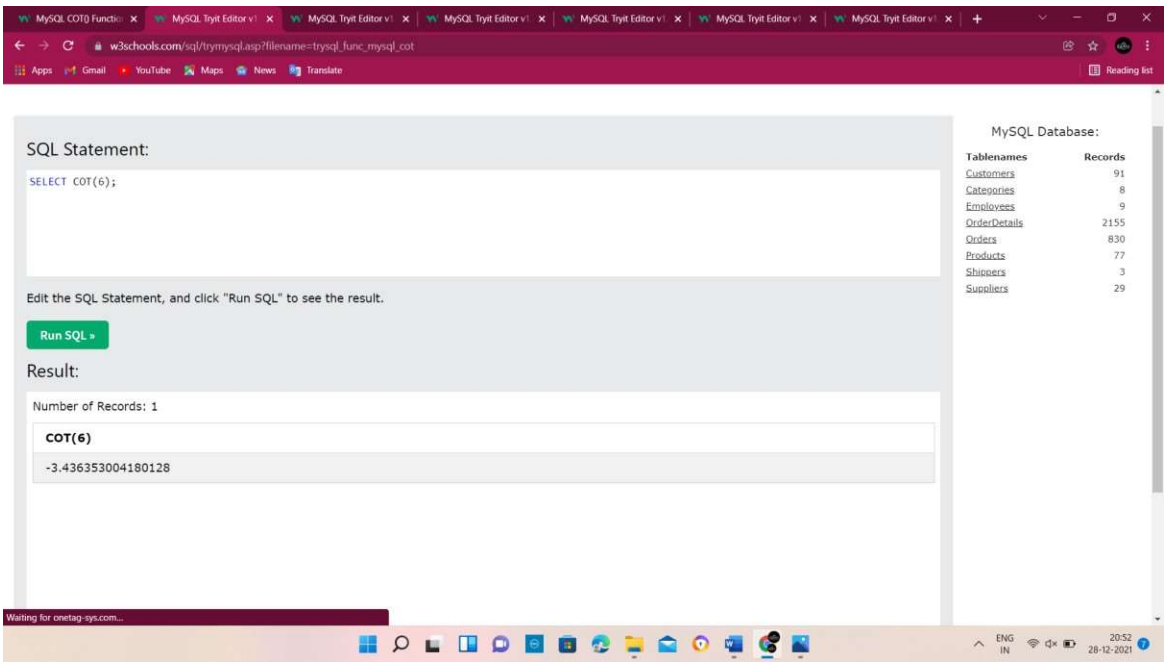

Fig. 13.1

4. CHARACTER FUNCTION

Character functions accept character inputs and can return either characters or number values as output. SQL provides a number of different character data types which includes
– CHAR, VARCHAR, VARCHAR2, LONG, RAW, and LONG RAW.

SQL provides a rich set of character functions that allow you to get information about strings and modify the contents of those strings in multiple ways. Character functions are of the following two types:

1. Case-Manipulative Functions (LOWER, UPPER and INITCAP)
2. Character-Manipulative Functions (CONCAT, LENGTH, SUBSTR, INSTR, LPAD, RPAD, TRIM and REPLACE)

4.1 Case manipulative functions :-

LOWER : This function converts alpha character values to lowercase. LOWER will actually return a fixed-length string if the incoming string is fixed-length. LOWER will not change any characters in the string that are not letters, since case is irrelevant for numbers and special characters, such as the dollar sign ($) or modulus (%).

Syntax: LOWER
Input: SELECT LOWER('GEEKSFORGEEKS') FROM DUAL;
Output: geeksforgeeks

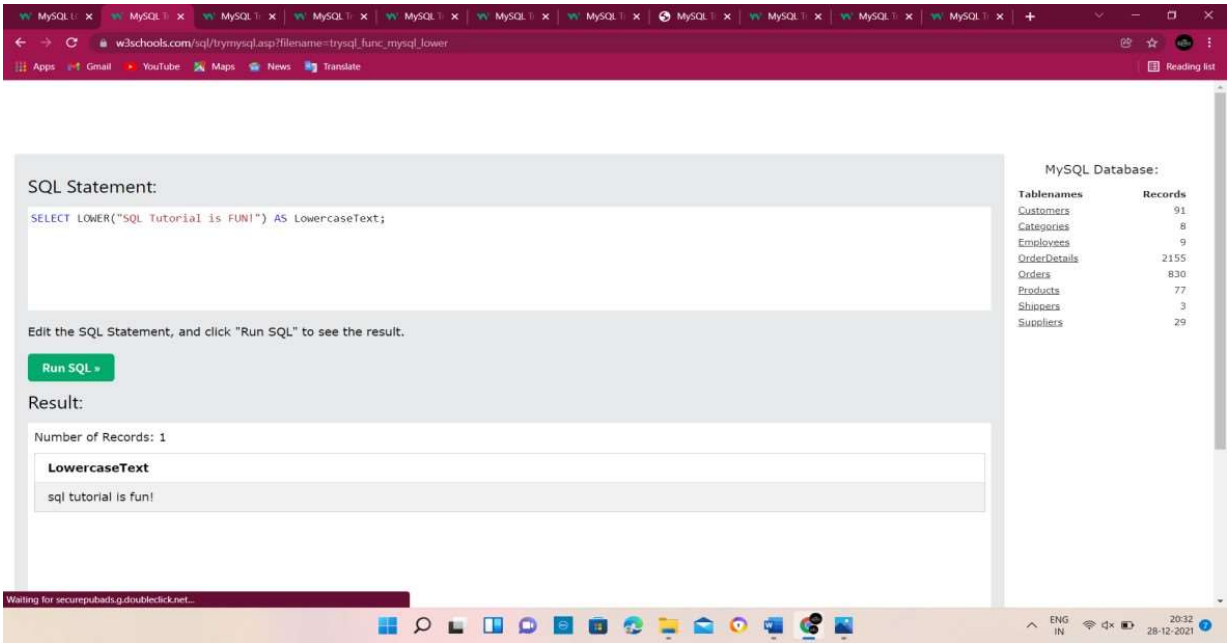

Fig. 14.1

UPPER : This function converts alpha character values to uppercase. Also UPPER function too, will actually return a fixed-length string if the incoming string is fixed-length. UPPER will not change any characters in the string that are not letters, since case is irrelevant for numbers and special characters, such as the dollar sign ($) or modulus (%).

Syntax: UPPER
Input: SELECT UPPER('geeksforgeeks') FROM DUAL;
Output: GEEKSFORGEEKS

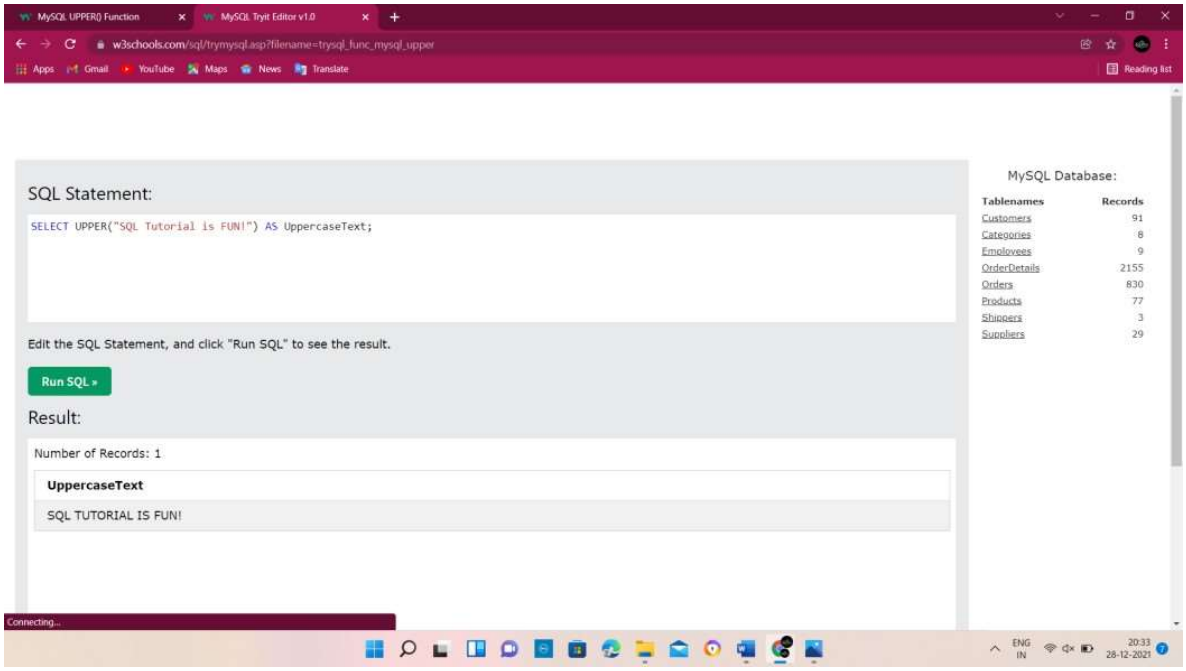

Fig.15.1

4.2 Character-Manipulative Functions

CONCAT : This function always appends (concatenates) string2 to the end of string1. If either of the string is NULL, CONCAT function returns the non-NULL argument. If both strings are NULL, CONCAT returns NULL.

Syntax: CONCAT('String1', 'String2')
Input: SELECT CONCAT('computer' ,'science') FROM DUAL;
Output: computerscience

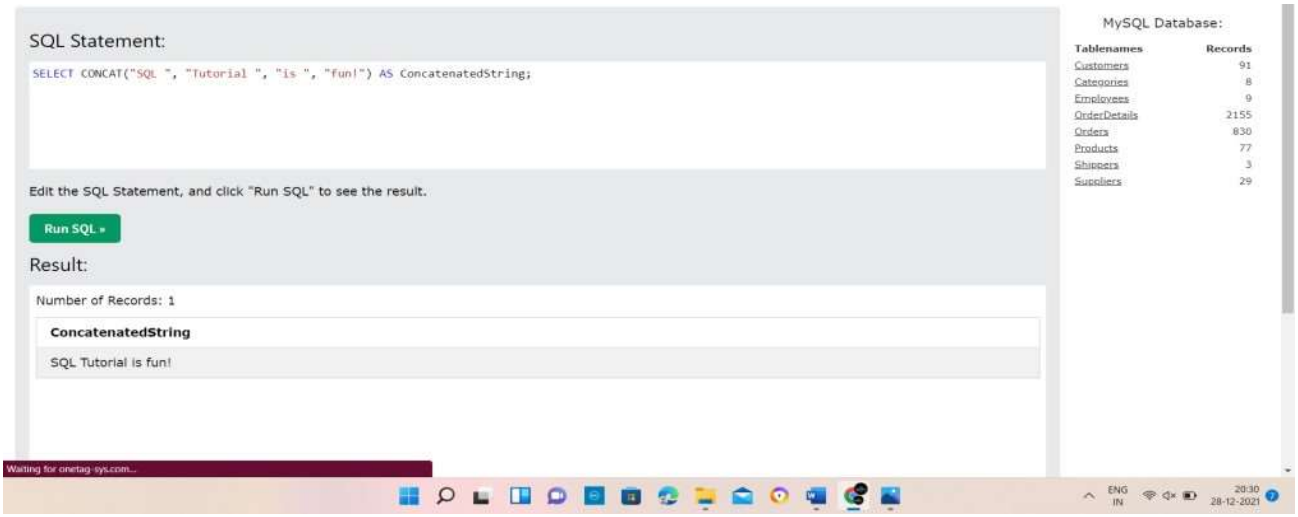

Fig. 16.1

LENGTH : This function returns the length of the input string. If the input string is NULL, then LENGTH function returns NULL and not Zero. Also, if the input string contains extra spaces at the start, or in between or at the end of the string, then the LENGTH function includes the extra spaces too and returns the complete length of the string.

Syntax: LENGTH (Column | Expression)
Input: SELECT LENGTH('Learning Is Fun') FROM DUAL;
Output: 15

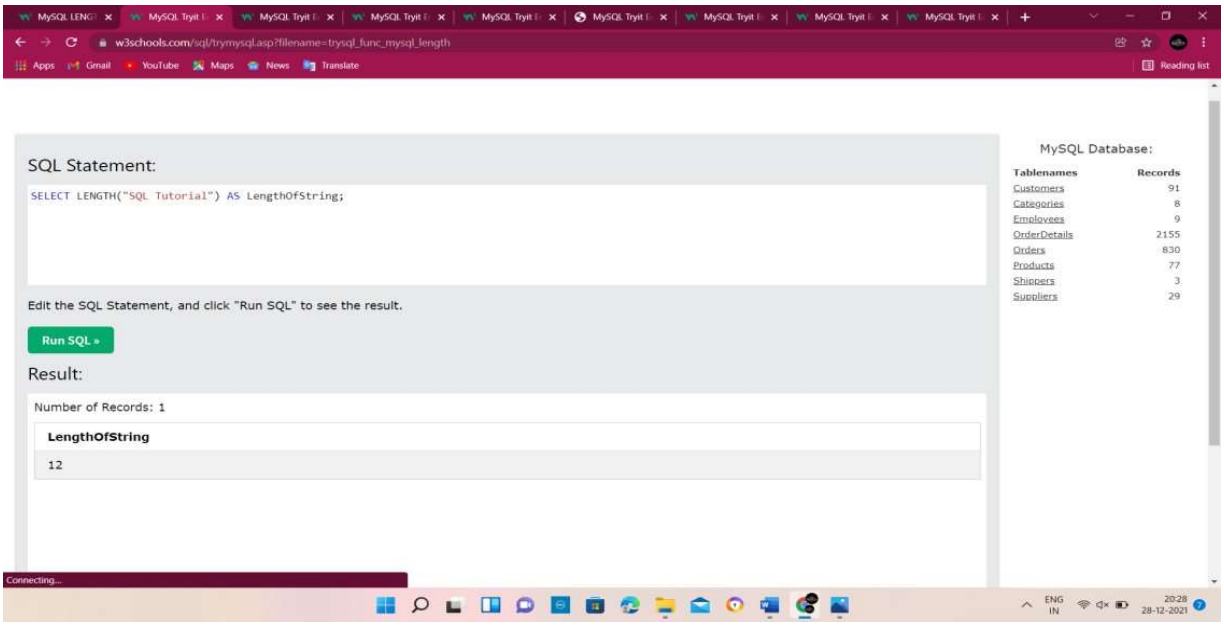

Fig.17.1

5. CONVERSION FUNCTIONS

In some cases, the Server uses data of one type where it expects data of a different data type. This can happen when the Server can automatically convert the data to the expected data type. This data type conversion can be done implicitly by the Server, or explicitly by the user.

Implicit Data-Type Conversion :

In this type of conversion the data is converted from one type to another implicitly (by itself/automatically).

From	To
VARCHAR2 or CHAR	NUMBER
VARCHAR2 or CHAR	DATE
DATE	VARCHAR2
NUMBER	VARCHAR2

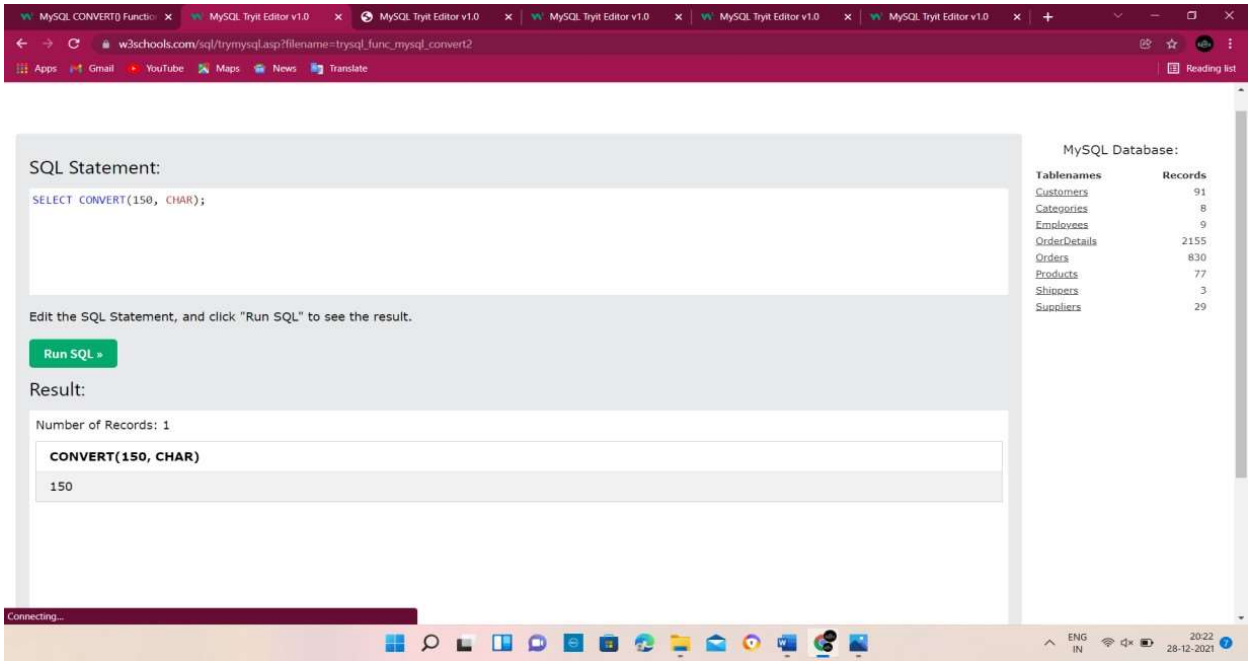

Fig. 18.1

5.1 Explicit Data-Type Conversion :

TO_CHAR Function :

TO_CHAR function is used to typecast a numeric or date input to character type with a format model (optional).

SYNTAX : TO_CHAR(number1, [format], [nls_parameter]) Using the TO_CHAR Function with Dates :

SYNTAX :TO_CHAR(date, 'format_model')

The format model:

Must be enclosed in single quotation marks and is case sensitive

Can include any valid date format element

Has an from element to remove padded blanks or suppress leading zeros

Is separated from the date value by a comma

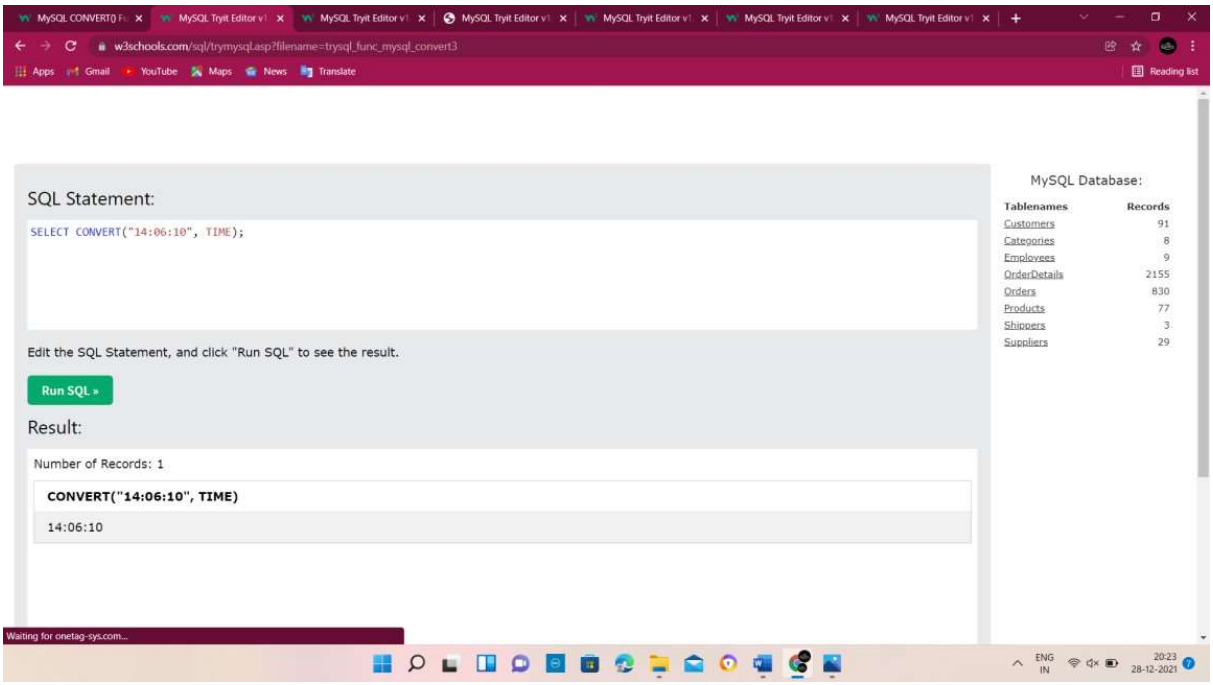

Fig.20.2

Elements of the Date Format Model :

YYYY	Full year in Numbers
YEAR	Year spelled out
MM	Two digit value for month
MONTH	Full name of the month
MON	Three Letter abbreviation of the month
DY	Three letter abbreviation of the day of the week
DAY	Full Name of the of the week
DD	Numeric day of the month

6. DATE FUNCTION

The following table lists the most important built-in date functions in SQL Server:

Function	Description
GETDATE()	Returns the current date and time
DATEPART()	Returns a single part of adate/time
DATEADD()	Adds or subtracts a specified time interval from a date
DATEDIFF()	Returns the time between two dates
CONVERT()	Displays date/time data in different formats data types for storing a date or a date/time value in the database:

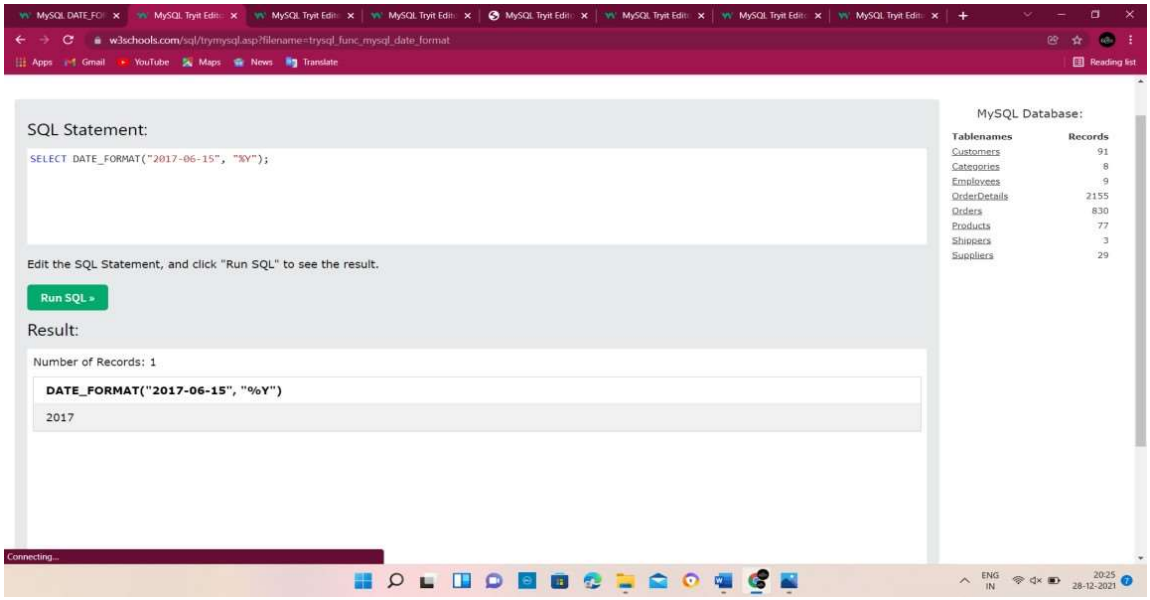

Fig. 20.1

DATE - format YYYY-MM-DD

DATETIME - format: YYYY-MM-DD HH:MI:SS

SMALLDATETIME - format: YYYY-MM-DD HH:MI:SS

TIMESTAMP - format: a unique number

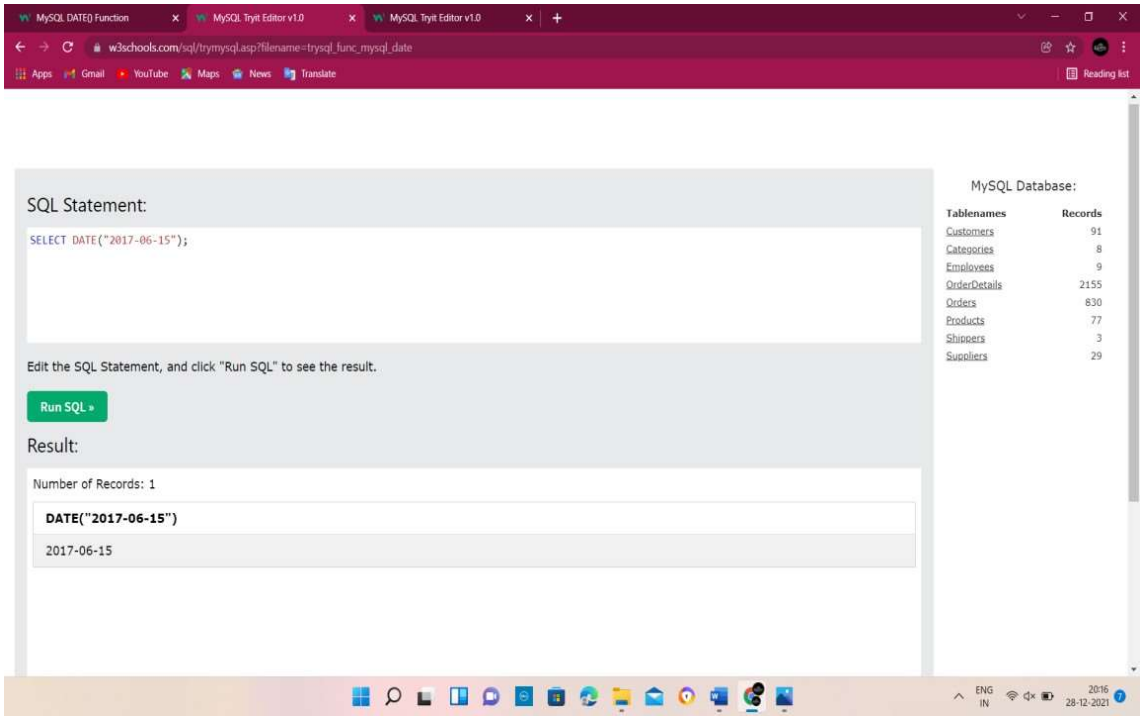

Fig. 21.1

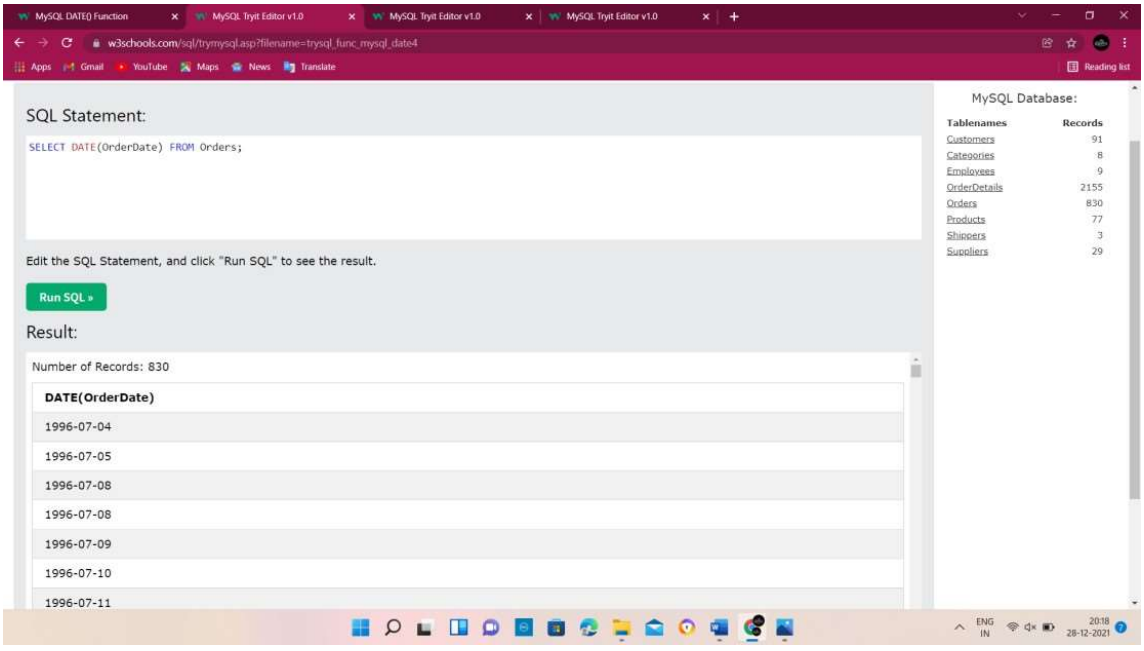

Fig. 21.2

EXAMPLE QUERY:

SELECT * FROM Orders WHERE OrderDate='2008-11-11'

7. CONCLUSION

7.1
Functions

This module discussed SQL functions and explanations of how they can be used in your queries. There are many, many other functions that you can use in your reports and queries. No other function will get any more complicated than those we have covered in this module. That's important to understand. You will simply pass parameters to the functions and get results.