**1. SQL REVERSE()**

A Reverses SQL Function is a string and returns the result! GOODBYE > EYBDOOG

The Syntax for Reverse SQL Function is: REVERSE ( string\_expression )

*Example*:

SELECT FirstName, REVERSE(FirstName) AS Reverse

FROM Person.Person

WHERE BusinessEntityID < 5

ORDER BY FirstName;

GO

Result:

FirstName Reverse

-------------- --------------

Ken neK

Rob boR

Roberto otreboR

Terri irreT

**2. SQL TIME()**

The Time SQL Function can extract the time value from a date/time expression. If the expression

is not a time or a date/time, this function will return "00:00:00" and if it is NULL the

function will also return NULL.

The Syntax of SQL Time Function is: TIME(expression)

*Example*: SELECT TIME("2015-08-16 12:15:");

**3. SQL TRIM()**

Trim function removes leading and trailing spaces from a string - you have to include

a string parameter in the function. It also is used to remove a specified prefix or suffix from a string.

The Syntax for SQL Trim Function is: TRIM( [ [LOCATION] [remstr] FROM ] str)

*Example*: SELECT TRIM(" NO FIGHTING ALLOWED ") AS TrimmedString;

**4. SQL FORMAT ()**

The SQL Format Function returns a value formatted with the specified format. It also takes a number and another that will specify how many decimal places this

function will return and then returns the result as a string.

The syntax for SQL Format Function is: FORMAT(NUMBER, NumberOfDecimals)

*Example*: SELECT FORMAT(987456.2145, 2);

**5. SQL UCASE()**

The SQL Function UCASE converts a specified string to upper-case. UCASE can also use the UPPER()

function.

The Syntax for SQL UCASE Function is: SELECT UCASE(col\_name) FROM tbl\_name;

*Example*: SELECT UCASE("Mix Martial Arts is Fun!");

SELECT UCASE(CustomerName) AS UppercaseCustomerName FROM Customers;

**6. SQL REPLACE()**

The SQL Replace Function replaces all occurrences of a specified string. You must be careful because it is case-sensitive.

The Syntax for SQL Replace Function is: REPLACE( string1, string\_to\_replace, replacement\_string )

*Example*: SELECT REPLACE('This is Replace Tutorial', 'T', 'R')

**7. SQL GREATEST()**

This function returns the greatest value in a list of expressions.

The syntax for SQL Greatest Function is: GREATEST( expr1 [, expr2, ... expr\_n] )

*Example*: SELECT GREATEST(12, 13, 165, 13, 1654);

SELECT GREATEST(Salary) AS "WEALTHIEST" FROM Salaries;

**8. SQL NOW()**

The NOW SQL Function returns the current date and time. “The NOW function is useful when you need to display the current date and time on a worksheet or calculate a value based on the current date and time, and have that value updated each time you open the worksheet.” Referenced here: <https://msdn.microsoft.com/en-us/query-bi/dax/now-function-dax>. There are no parameters or arguments on a NOW Function.

The syntax for SQL NOW Function is: NOW();

*Example*: SELECT (NOW() - INTERVAL 1 DAY)

**9. SQL LEAST()**

The SQL LEAST Function returns the least value in a list of expressions.

The syntax for LEAST Function is: LEAST(expression1, expr2, expr3, ...)

*Example*: SELECT LEAST(3, 6, 9, 12, 15);

**10. SQL CHAR\_LENGTH()**

The SQL CHAR Function is one of the String Function, which returns the length of the user specified string measured in characters. This function will consider the multi byte character as the single character.

The syntax for SQL CHAR Function is: CHAR\_LENGTH( string )

*Example*: SELECT CHAR\_LENGTH("StoreName") AS StorePrices FROM StoreValue;

/\* I accidentally researched on SQL subject matter that was not SQL ***FUNCTIONS*** and later realized that I needed to redo my research, so instead of deleting it, I left it here. \*/

There are many different types of SQL and the ten that I am going to write about are as followed: SQL Union, SQL Exists, SQL Comments, SQL Null, SQL Stored Procedures, SQL Select Distinct, SQL Order By, SQL Select Top, SQL Group By, and SQL Aliases.

**The SQL Union** clause combines the results of two SQL queries into a single table of matching rows. In order to unite, the two data queries must result in the same number of columns and be compatible data types. Unless Union All Is used any duplicate records are automatically removed. Union is useful in database applications where tables aren’t perfectly normalized.

**The SQL Exists** clause is a logical operator that evaluates a subquery for existence of records. If the subquery has one or more records Exists returns True. If the subquery has no records, Exists returns false. Since Exists return a logical value, it is common to use Exists in the WHERE clause, which expects logical expressions. It also can make some queries faster.

**The SQL Comments** are used to explain sections of SQL statements, or to prevent execution of SQL statements. SQL supports single line comments and multiple line comments. Single line comments start with - - (w/o the space) symbols, while multi line comments start with /\* and end with \*/. SQL Comments make your application easier to read and understand because you can describe the purpose of the statement within your application by adding comments and/or hints.

**The SQL Null** indicates that no data value exists in a database. Null is a lack of value and is not the same meaning as [0]. Null value is a reference that means is it not pointing to any object. The Null value does not indicate why a value is absent – it only marks the places that don’t have a data value.

**The SQL Stored Procedures** is a group of T-SQL (Transact SQL) statements. They can be triggered with a short “Execute” statement. Common functions of stored procedures are modifying, deleting and retrieving data. They typically only accept one or more input parameters. A Single Stored Procedure can be used by several people using different input data. Also, if there is a situation where you write the same query over and over again, you can save that specific query as a stored procedure and call it just by its name.

**The SQL Select Distinct** statement returns only different values. Sometimes you need to list the Distinct values, because a column has duplicate values. The Select Distinct clause is key to removing duplicate rows.

**THE SQL Order By** is used for sorting the result-set that is created using the SELECT statement. When using this statement, you can also sort the result-set by one or more columns in an ascending or descending order.

**The SQL Select Top** clause is needed to describe the amount of data records to bring back. This has huge tables with a lot of data which makes this clause extremely helpful. Select Top is not supported by all of the databases.

**The SQL Aliases** are used when you need to rename headings of a column or table temporarily. Aliases are also used when you need to make column names more readable. An alias only exists for the duration of the query.

**QL Group By** statement is used with aggregate functions, such as MAX, MIN, COUNT, AVG, and SUM, to group the result -set by one or more columns. SQL Group By allows you to combine or group rows returned from a SQL statement and perform aggregate functions on them. It can group by multiple columns.

**References:**

<https://www.w3schools.com/sql/sql_null_values.asp>

<https://www.w3resource.com/sql/joins/perform-an-inner-join.phpperform>

<https://www.tutorialspoint.com/sql/sql-inner-joins.htm>

<http://www.sqltutorial.org/sql-left-join/>

<https://msdn.microsoft.com/en-us/query-bi/dax/now-function-dax>.