SQL Query-3

SQL Functions

SQL has many built-in functions for performing calculations on data.

SQL Aggregate Functions

SQL aggregate functions return a single value, calculated from values in a column.

Function	Description
AVG()	Returns the average value
COUNT()	Returns the number of rows
FIRST()	Returns the first value
LAST()	Returns the last value
MAX()	Returns the largest value
MIN()	Returns the smallest value
ROUND()	Rounds a numeric field to the number of decimals specified
SUM()	Returns the sum

SQL String Functions

Function	Description
CHARINDEX	Searches an expression in a string expression and returns its starting position if found
CONCAT()	
LEFT()	
LEN() / LENGTH()	Returns the length of the value in a text field
LOWER() / LCASE()	Converts character data to lower case
LTRIM()	
SUBSTRING() / MID()	Extract characters from a text field
PATINDEX()	
REPLACE()	
RIGHT()	
RTRIM()	

UPPER() / UCASE()	Converts character data to upper case

SQL Date and Time Data Types and Functions

Function	Description
FORMAT()	Formats how a field is to be displayed
NOW()	Returns the current system date and time

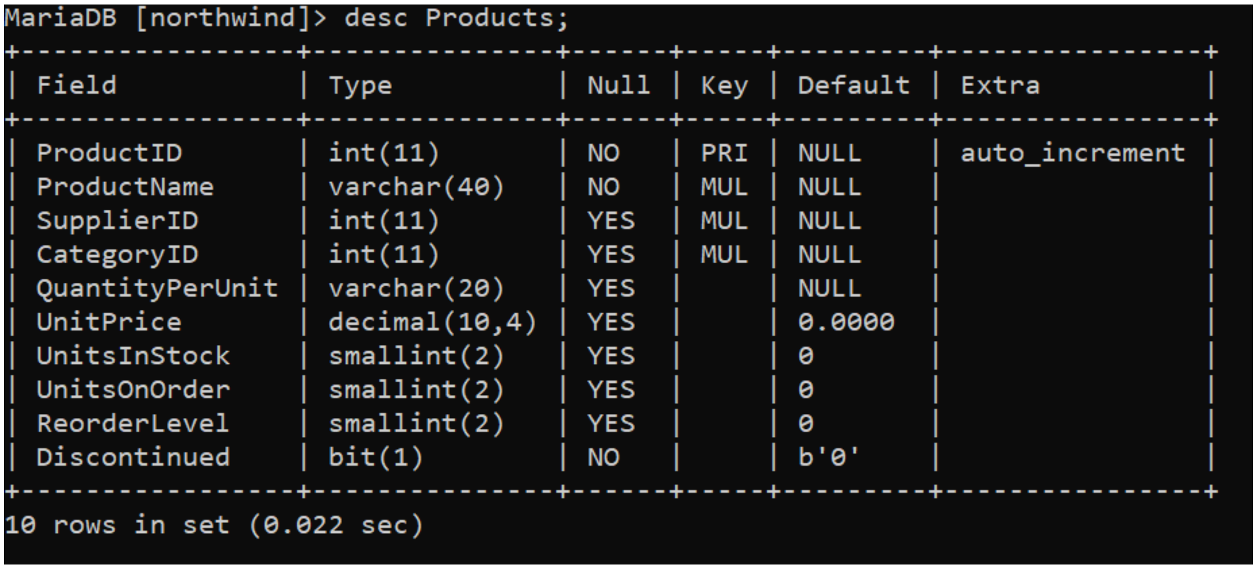
The AVG() Function

The AVG() function returns the average value of a numeric column.

SQL AVG() Syntax

```
SELECT AVG(column_name) FROM table_name
```

Demo Database



```
MariaDB [northwind]> desc Products;
+-----------------+---------------+------+-----+---------+----------------+
| Field           | Type          | Null | Key | Default | Extra          |
+-----------------+---------------+------+-----+---------+----------------+
| ProductID       | int(11)       | NO   | PRI | NULL    | auto_increment |
| ProductName     | varchar(40)   | NO   | MUL | NULL    |                |
| SupplierID      | int(11)       | YES  | MUL | NULL    |                |
| CategoryID      | int(11)       | YES  | MUL | NULL    |                |
| QuantityPerUnit | varchar(20)   | YES  |     | NULL    |                |
| UnitPrice       | decimal(10,4) | YES  |     | 0.0000  |                |
| UnitsInStock    | smallint(2)   | YES  |     | 0       |                |
| UnitsOnOrder    | smallint(2)   | YES  |     | 0       |                |
| ReorderLevel    | smallint(2)   | YES  |     | 0       |                |
| Discontinued    | bit(1)        | NO   |     | b'0'    |                |
+-----------------+---------------+------+-----+---------+----------------+
10 rows in set (0.022 sec)
```

Below is a selection from the "Products" table:

ProductID	ProductName	SupplierID	CategoryID	Unit	Price
1	Chais	1	1	10 boxes x 20 bags	18

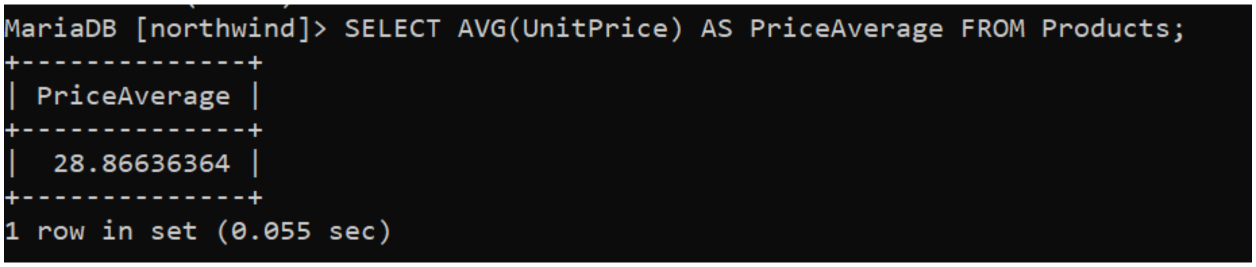
2	Chang	1	1	24 - 12 oz bottles	19
3	Aniseed Syrup	1	2	12 - 550 ml bottles	10
4	Chef Anton's Cajun Seasoning	2	2	48 - 6 oz jars	21.35
5	Chef Anton's Gumbo Mix	2	2	36 boxes	25

SQL AVG() Example

The following SQL statement gets the average value of the "Price" column from the "Products" table:

Example

```
SELECT AVG(UnitPrice) AS PriceAverage FROM Products;
```

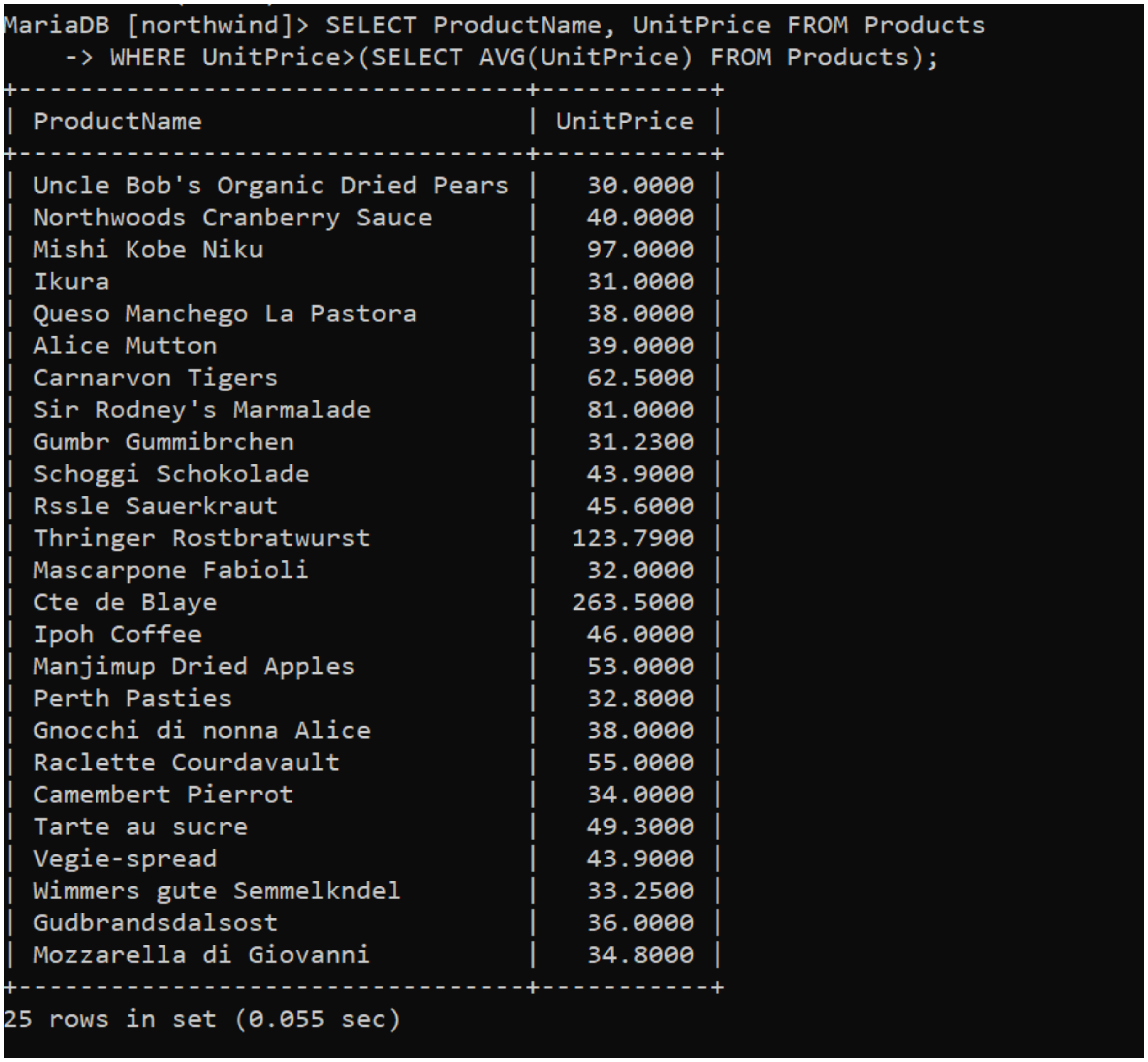


```
MariaDB [northwind]> SELECT AVG(UnitPrice) AS PriceAverage FROM Products;
+--------------+
| PriceAverage |
+--------------+
|  28.86636364 |
+--------------+
1 row in set (0.055 sec)
```

The following SQL statement selects the "ProductName" and "Price" records that have an above average price:

Example

```
SELECT ProductName, Price FROM Products
WHERE Price>(SELECT AVG(Price) FROM Products);
```



```
MariaDB [northwind]> SELECT ProductName, UnitPrice FROM Products
    -> WHERE UnitPrice>(SELECT AVG(UnitPrice) FROM Products);
+----------------------------------+-----------+
| ProductName                      | UnitPrice |
+----------------------------------+-----------+
| Uncle Bob's Organic Dried Pears  |   30.0000 |
| Northwoods Cranberry Sauce       |   40.0000 |
| Mishi Kobe Niku                  |   97.0000 |
| Ikura                            |   31.0000 |
| Queso Manchego La Pastora        |   38.0000 |
| Alice Mutton                     |   39.0000 |
| Carnarvon Tigers                 |   62.5000 |
| Sir Rodney's Marmalade           |   81.0000 |
| Gumbr Gummibrchen                |   31.2300 |
| Schoggi Schokolade               |   43.9000 |
| Rssle Sauerkraut                 |   45.6000 |
| Thringer Rostbratwurst           |  123.7900 |
| Mascarpone Fabioli               |   32.0000 |
| Cte de Blaye                     |  263.5000 |
| Ipoh Coffee                      |   46.0000 |
| Manjimup Dried Apples            |   53.0000 |
| Perth Pasties                    |   32.8000 |
| Gnocchi di nonna Alice           |   38.0000 |
| Raclette Courdavault             |   55.0000 |
| Camembert Pierrot                |   34.0000 |
| Tarte au sucre                   |   49.3000 |
| Vegie-spread                     |   43.9000 |
| Wimmers gute Semmelkndel         |   33.2500 |
| Gudbrandsdalsost                 |   36.0000 |
| Mozzarella di Giovanni           |   34.8000 |
+----------------------------------+-----------+
25 rows in set (0.055 sec)
```

SQL COUNT() Function

The COUNT() function returns the number of rows that matches a specified criteria.

SQL COUNT(column_name) Syntax

The COUNT(column_name) function returns the number of values (NULL values will not be counted) of the specified column:

```
SELECT COUNT(column_name) FROM table_name;
```

SQL COUNT(*) Syntax

The COUNT(*) function returns the number of records in a table:

```
SELECT COUNT(*) FROM table_name;
```

SQL COUNT(DISTINCT column_name) Syntax

The COUNT(DISTINCT column_name) function returns the number of distinct values of the specified column:

```
SELECT COUNT(DISTINCT column_name) FROM table_name;
```

Note: COUNT(DISTINCT) works with ORACLE and Microsoft SQL Server, but not with Microsoft Access.

Demo Database

Below is a selection from the "Orders" table:

OrderID	CustomerID	EmployeeID	OrderDate	ShipperID
10265	7	2	1996-07-25	1
10266	87	3	1996-07-26	3
10267	25	4	1996-07-29	1

SQL COUNT(column_name) Example

The following SQL statement counts the number of orders from "CustomerID"=7 from the "Orders" table:

Example

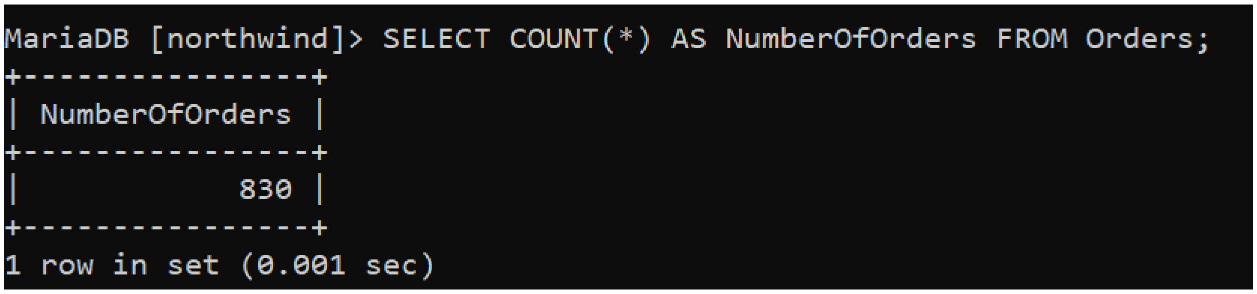
```
SELECT COUNT(CustomerID) AS OrdersFromCustomerID7 FROM Orders
WHERE CustomerID=7;
```

SQL COUNT(*) Example

The following SQL statement counts the total number of orders in the "Orders" table:

Example

```
SELECT COUNT(*) AS NumberOfOrders FROM Orders;
```



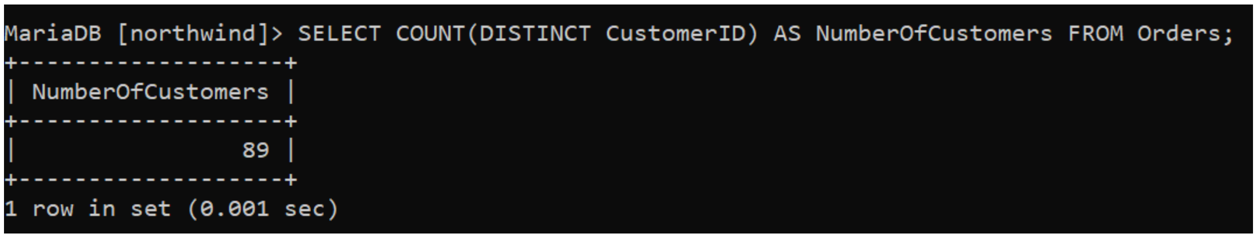
```
MariaDB [northwind]> SELECT COUNT(*) AS NumberOfOrders FROM Orders;
+----------------+
| NumberOfOrders |
+----------------+
|            830 |
+----------------+
1 row in set (0.001 sec)
```

SQL COUNT(DISTINCT column_name) Example

The following SQL statement counts the number of unique customers in the "Orders" table:

Example

```
SELECT COUNT(DISTINCT CustomerID) AS NumberOfCustomers FROM Orders;
```



```
MariaDB [northwind]> SELECT COUNT(DISTINCT CustomerID) AS NumberOfCustomers FROM Orders;
+-------------------+
| NumberOfCustomers |
+-------------------+
|                89 |
+-------------------+
1 row in set (0.001 sec)
```

The FIRST() Function

The FIRST() function returns the first value of the selected column.

SQL FIRST() Syntax

```
SELECT FIRST(column_name) FROM table_name;
```

Note: The FIRST() function is only supported in MS Access.

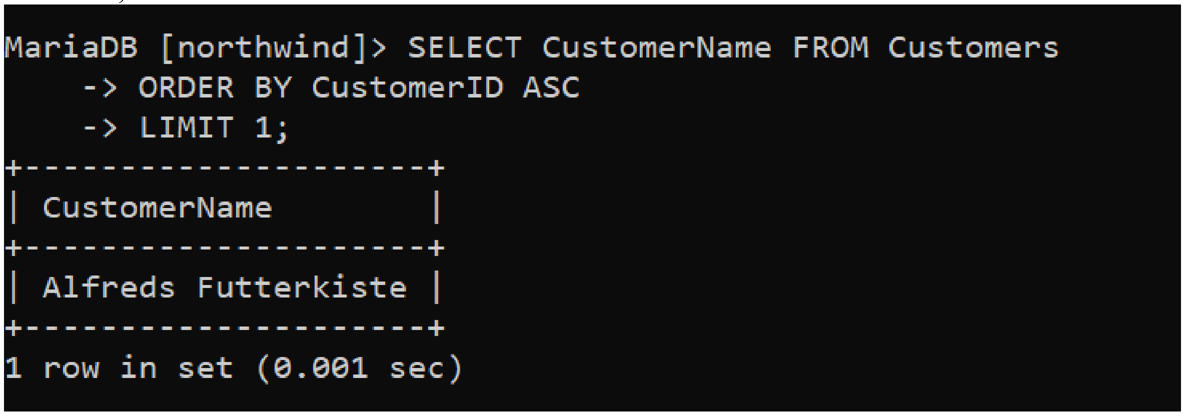
SQL FIRST() Workaround in MySQL

MySQL Syntax

```
SELECT column_name FROM table_name
ORDER BY column_name ASC
LIMIT 1;
```

Example

```
SELECT CustomerName FROM Customers
ORDER BY CustomerID ASC
LIMIT 1;
```



```
MariaDB [northwind]> SELECT CustomerName FROM Customers
    -> ORDER BY CustomerID ASC
    -> LIMIT 1;
+---------------------+
| CustomerName        |
+---------------------+
| Alfreds Futterkiste |
+---------------------+
1 row in set (0.001 sec)
```

Demo Database

Below is a selection from the "Customers" table:

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden

SQL FIRST() Example

The following SQL statement selects the first value of the "CustomerName" column from the "Customers" table:

Example

```
SELECT FIRST(CustomerName) AS FirstCustomer FROM Customers;
```

The LAST() Function

The LAST() function returns the last value of the selected column.

SQL LAST() Syntax

```
SELECT LAST(column_name) FROM table_name;
```

Note: The LAST() function is only supported in MS Access.

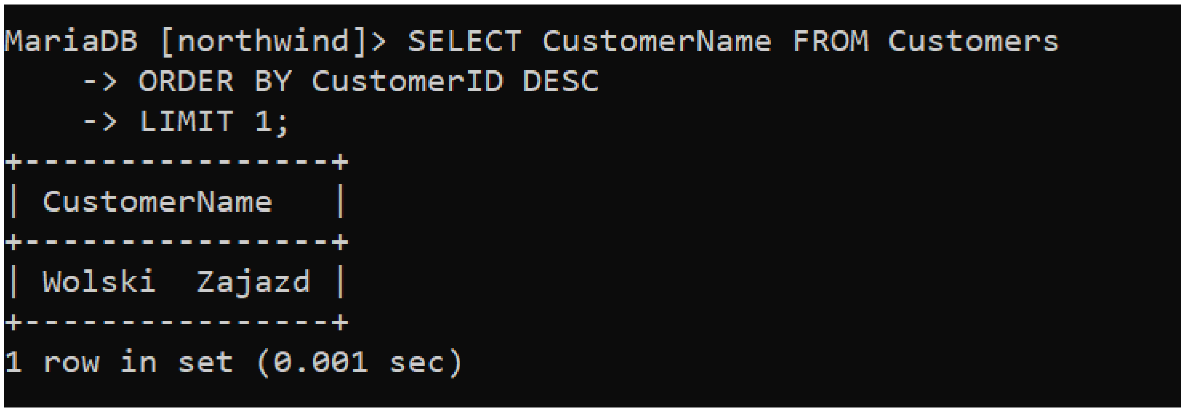
SQL LAST() Workaround in MySQL

MySQL Syntax

```
SELECT column_name FROM table_name
ORDER BY column_name DESC
LIMIT 1;
```

Example

```
SELECT CustomerName FROM Customers
ORDER BY CustomerID DESC
LIMIT 1;
```


```
MariaDB [northwind]> SELECT CustomerName FROM Customers
    -> ORDER BY CustomerID DESC
    -> LIMIT 1;
+-----------------+
| CustomerName    |
+-----------------+
| Wolski  Zajazd |
+-----------------+
1 row in set (0.001 sec)
```

Demo Database

In this tutorial we will use the well-known Northwind sample database.

Below is a selection from the "Customers" table:

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden

SQL LAST() Example

The following SQL statement selects the last value of the "CustomerName" column from the "Customers" table:

Example

```
SELECT LAST(CustomerName) AS LastCustomer FROM Customers;
```

The MAX() Function

The MAX() function returns the largest value of the selected column.

SQL MAX() Syntax

```
SELECT MAX(column_name) FROM table_name;
```

Demo Database

In this tutorial we will use the well-known Northwind sample database.

Below is a selection from the "Products" table:

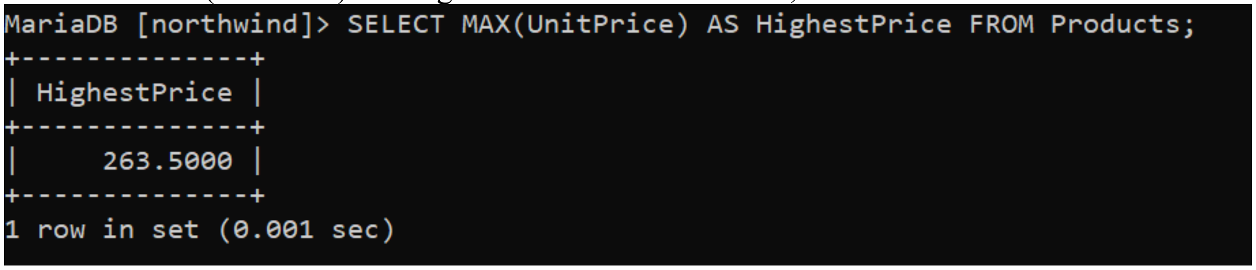
ProductID	ProductName	SupplierID	CategoryID	Unit	Price
1	Chais	1	1	10 boxes x 20 bags	18
2	Chang	1	1	24 - 12 oz bottles	19
3	Aniseed Syrup	1	2	12 - 550 ml bottles	10
4	Chef Anton's Cajun Seasoning	2	2	48 - 6 oz jars	21.35
5	Chef Anton's Gumbo Mix	2	2	36 boxes	25

SQL MAX() Example

The following SQL statement gets the largest value of the "Price" column from the "Products" table:

Example

```
SELECT MAX(UnitPrice) AS HighestPrice FROM Products;
```



```
MariaDB [northwind]> SELECT MAX(UnitPrice) AS HighestPrice FROM Products;
+--------------+
| HighestPrice |
+--------------+
|     263.5000 |
+--------------+
1 row in set (0.001 sec)
```

The MIN() Function

The MIN() function returns the smallest value of the selected column.

SQL MIN() Syntax

SELECT MIN(column_name) FROM table_name;

Demo Database

Below is a selection from the "Products" table:

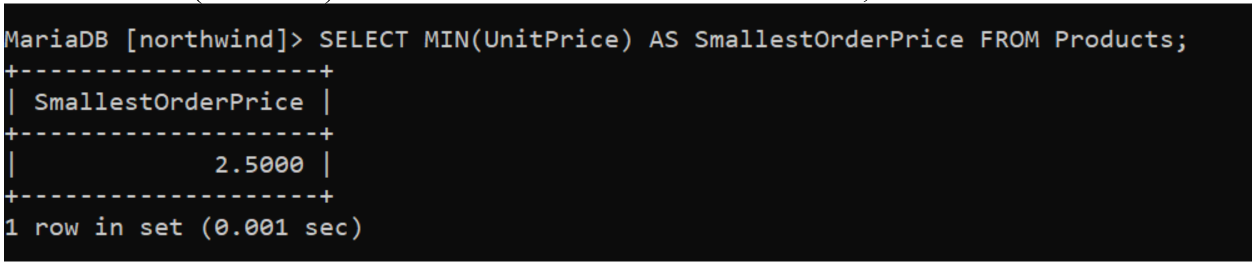
ProductID	ProductName	SupplierID	CategoryID	Unit	Price
1	Chais	1	1	10 boxes x 20 bags	18
2	Chang	1	1	24 - 12 oz bottles	19
3	Aniseed Syrup	1	2	12 - 550 ml bottles	10
4	Chef Anton's Cajun Seasoning	2	2	48 - 6 oz jars	21.35
5	Chef Anton's Gumbo Mix	2	2	36 boxes	25

SQL MIN() Example

The following SQL statement gets the smallest value of the "Price" column from the "Products" table:

Example

SELECT MIN(UnitPrice) AS SmallestOrderPrice FROM Products;



```
MariaDB [northwind]> SELECT MIN(UnitPrice) AS SmallestOrderPrice FROM Products;
+--------------------+
| SmallestOrderPrice |
+--------------------+
|             2.5000 |
+--------------------+
1 row in set (0.001 sec)
```

The SUM() Function

The SUM() function returns the total sum of a numeric column.

SQL SUM() Syntax

SELECT SUM(column_name) FROM table_name;

Demo Database

Below is a selection from the "OrderDetails" table:

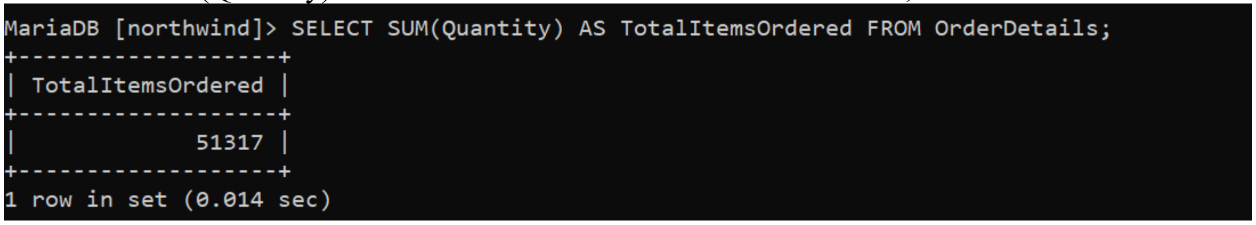
OrderDetailID	OrderID	ProductID	Quantity
1	10248	11	12
2	10248	42	10
3	10248	72	5
4	10249	14	9
5	10249	51	40

SQL SUM() Example

The following SQL statement finds the sum of all the "Quantity" fields for the "OrderDetails" table:

Example

```
SELECT SUM(Quantity) AS TotalItemsOrdered FROM OrderDetails;
```



```
MariaDB [northwind]> SELECT SUM(Quantity) AS TotalItemsOrdered FROM OrderDetails;
+-------------------+
| TotalItemsOrdered |
+-------------------+
|             51317 |
+-------------------+
1 row in set (0.014 sec)
```

The GROUP BY Statement

The GROUP BY statement is used in conjunction with the aggregate functions to group the result-set by one or more columns.

SQL GROUP BY Syntax

```
SELECT column_name, aggregate_function(column_name)
FROM table_name
WHERE column_name operator value
GROUP BY column_name;
```

Demo Database

Below is a selection from the "Orders" table:

OrderID	CustomerID	EmployeeID	OrderDate	ShipperID
10248	90	5	1996-07-04	3
10249	81	6	1996-07-05	1
10250	34	4	1996-07-08	2

And a selection from the "Shippers" table:

ShipperID	ShipperName
1	Speedy Express
2	United Package
3	Federal Shipping

And a selection from the "Employees" table:

EmployeeID	LastName	FirstName	BirthDate	Photo	Notes
1	Davolio	Nancy	1968-12-08	EmpID1.pic	Education includes a BA....
2	Fuller	Andrew	1952-02-19	EmpID2.pic	Andrew received his BTS....
3	Leverling	Janet	1963-08-30	EmpID3.pic	Janet has a BS degree....

SQL GROUP BY Example

Now we want to find the number of orders sent by each shipper.

The following SQL statement counts as orders grouped by shippers:

Example

```
SELECT Shippers.CompanyName,COUNT(Orders.OrderID) AS NumberOfOrders FROM
Orders
LEFT JOIN Shippers
ON Orders.ShipperID=Shippers.ShipperID
GROUP BY CompanyName;
```

GROUP BY More Than One Column

We can also use the GROUP BY statement on more than one column, like this:

Example

```
SELECT Shippers.ShipperName, Employees.LastName,
COUNT(Orders.OrderID) AS NumberOfOrders
FROM ((Orders
INNER JOIN Shippers
ON Orders.ShipperID=Shippers.ShipperID)
INNER JOIN Employees
ON Orders.EmployeeID=Employees.EmployeeID)
GROUP BY ShipperName,LastName;
```

The HAVING Clause

The HAVING clause was added to SQL because the WHERE keyword could not be used with aggregate functions.

SQL HAVING Syntax

```
SELECT column_name, aggregate_function(column_name)
FROM table_name
WHERE column_name operator value
GROUP BY column_name
HAVING aggregate_function(column_name) operator value;
```

Demo Database

Below is a selection from the "Orders" table:

OrderID	CustomerID	EmployeeID	OrderDate	ShipperID
10248	90	5	1996-07-04	3
10249	81	6	1996-07-05	1
10250	34	4	1996-07-08	2

And a selection from the "Employees" table:

EmployeeID	LastName	FirstName	BirthDate	Photo	Notes
1	Davolio	Nancy	1968-12-08	EmpID1.pic	Education includes a BA....
2	Fuller	Andrew	1952-02-19	EmpID2.pic	Andrew received his BTS....
3	Leverling	Janet	1963-08-30	EmpID3.pic	Janet has a BS degree....

SQL HAVING Example

Now we want to find if any of the employees has registered more than 10 orders.

We use the following SQL statement:

Example

```
SELECT Employees.LastName, COUNT(Orders.OrderID) AS NumberOfOrders FROM (Orders
INNER JOIN Employees
ON Orders.EmployeeID=Employees.EmployeeID)
GROUP BY LastName
HAVING COUNT(Orders.OrderID) > 10;
```

Now we want to find if the employees "Davolio" or "Fuller" have registered more than 25 orders.

We add an ordinary WHERE clause to the SQL statement:

Example

```
SELECT Employees.LastName, COUNT(Orders.OrderID) AS NumberOfOrders FROM Orders
INNER JOIN Employees
ON Orders.EmployeeID=Employees.EmployeeID
WHERE LastName='Davolio' OR LastName='Fuller'
GROUP BY LastName
HAVING COUNT(Orders.OrderID) > 25;
```

The UCASE() Function

The UCASE() function converts the value of a field to uppercase.

SQL UCASE() Syntax

```
SELECT UCASE(column_name) FROM table_name;
```

Syntax for SQL Server

```
SELECT UPPER(column_name) FROM table_name;
```

Demo Database

Below is a selection from the "Customers" table:

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country

1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden

SQL UCASE() Example

The following SQL statement selects the "CustomerName" and "City" columns from the "Customers" table, and converts the "CustomerName" column to uppercase:

Example

```
SELECT UCASE(CustomerName) AS Customer, City
FROM Customers;
```

The LCASE() Function

The LCASE() function converts the value of a field to lowercase.

SQL LCASE() Syntax

```
SELECT LCASE(column_name) FROM table_name;
```

Syntax for SQL Server

```
SELECT LOWER(column_name) FROM table_name;
```

Demo Database

Below is a selection from the "Customers" table:

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country

1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden

SQL LCASE() Example

The following SQL statement selects the "CustomerName" and "City" columns from the "Customers" table, and converts the "CustomerName" column to lowercase:

Example

```
SELECT LCASE(CustomerName) AS Customer, City
FROM Customers;
```

The MID() Function

The MID() function is used to extract characters from a text field.

SQL MID() Syntax

```
SELECT MID(column_name,start,length) AS some_name FROM table_name;
```

Parameter	Description
column_name	Required. The field to extract characters from
start	Required. Specifies the starting position (starts at 1)
length	Optional. The number of characters to return. If omitted, the MID() function returns the rest of the text

Note: The equivalent function for SQL Server is SUBSTRING():

```
SELECT SUBSTRING(column_name,start,length) AS some_name FROM table_name;
```

Note: The equivalent function for Oracle is SUBSTR():

```
SELECT SUBSTR(column_name,start,length) AS some_name FROM table_name;
```

Demo Database

Below is a selection from the "Customers" table:

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden

SQL MID() Example

The following SQL statement selects the first four characters from the "City" column from the "Customers" table:

Example

```
SELECT MID(City,1,4) AS ShortCity
FROM Customers;
```

The LEN() Function

The LEN() function returns the length of the value in a text field.

SQL LEN() Syntax

```
SELECT LEN(column_name) FROM table_name;
```

Syntax for Oracle

```
SELECT LENGTH(column_name) FROM table_name;
```

Demo Database

Below is a selection from the "Customers" table:

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden

SQL LEN() Example

The following SQL statement selects the "CustomerName" and the length of the values in the "Address" column from the "Customers" table:

Example

```
SELECT CustomerName,LEN(Address) as LengthOfAddress
FROM Customers;
```

The ROUND() Function

The ROUND() function is used to round a numeric field to the number of decimals specified.

Note: Many database systems do rounding differently than you might expect. When rounding a number with a fractional part to an integer, our school teachers told us to round .1 through .4 DOWN to the next lower integer, and .5 through .9 UP to the next higher integer. But if all the digits 1 through 9 are equally likely, this introduces a slight bias towards infinity, since we always round .5 up. Many database systems have adopted the IEEE 754 standard for arithmetic operations, according to which the default rounding behavior is "round half to even." In this scheme, .5 is rounded to the nearest even integer. So, both 11.5 and 12.5 would be rounded to 12.

SQL ROUND() Syntax

```
SELECT ROUND(column_name,decimals) FROM table_name;
```

Parameter	Description
column_name	Required. The field to round.
decimals	Required. Specifies the number of decimals to be returned.

Demo Database

Below is a selection from the "Products" table:

ProductID	ProductName	SupplierID	CategoryID	Unit	Price
1	Chais	1	1	10 boxes x 20 bags	18
2	Chang	1	1	24 - 12 oz bottles	19
3	Aniseed Syrup	1	2	12 - 550 ml bottles	10
4	Chef Anton's Cajun Seasoning	2	2	48 - 6 oz jars	21.35
5	Chef Anton's Gumbo Mix	2	2	36 boxes	25

SQL ROUND() Example

The following SQL statement selects the product name and rounds the price in the "Products" table:

Example

```
SELECT ProductName, ROUND(Price,0) AS RoundedPrice
FROM Products;
```

The NOW() Function

The NOW() function returns the current system date and time.

SQL NOW() Syntax

```
SELECT NOW() FROM table_name;
```

Demo Database

Below is a selection from the "Products" table:

ProductID	ProductName	SupplierID	CategoryID	Unit	Price
1	Chais	1	1	10 boxes x 20 bags	18
2	Chang	1	1	24 - 12 oz bottles	19
3	Aniseed Syrup	1	2	12 - 550 ml bottles	10
4	Chef Anton's Cajun Seasoning	2	2	48 - 6 oz jars	21.35
5	Chef Anton's Gumbo Mix	2	2	36 boxes	25

SQL NOW() Example

The following SQL statement selects the product name, and price for today from the "Products" table:

Example

```
SELECT ProductName, Price, Now() AS PerDate
FROM Products;
```

The FORMAT() Function

The FORMAT() function is used to format how a field is to be displayed.

SQL FORMAT() Syntax

```
SELECT FORMAT(column_name,format) FROM table_name;
```

Parameter	Description
column_name	Required. The field to be formatted.
format	Required. Specifies the format.

Demo Database

Below is a selection from the "Products" table:

ProductID	ProductName	SupplierID	CategoryID	Unit	Price
1	Chais	1	1	10 boxes x 20 bags	18
2	Chang	1	1	24 - 12 oz bottles	19
3	Aniseed Syrup	1	2	12 - 550 ml bottles	10
4	Chef Anton's Cajun Seasoning	2	2	48 - 6 oz jars	21.35
5	Chef Anton's Gumbo Mix	2	2	36 boxes	25

SQL FORMAT() Example

The following SQL statement selects the product name, and price for today (formatted like YYYY-MM-DD) from the "Products" table:

Example

```
SELECT ProductName, Price, FORMAT(Now(),'YYYY-MM-DD') AS PerDate
FROM Products;
```