



# Reporting Functions & Operators

## Functions

### List of Built-in Field Functions

| Function       | Description  |
|----------------|--|
| Average        | Return the average of all values within the field.             |
| Count          | Return the quantity of all values within the field.            |
| Count Distinct | Return the count unique number of all values within the field. |
| Maximum        | Return the maximum value of the field.                         |
| Minimum        | Return the minimum value of the field.                         |
| Sum            | Return the sum of all values within the field.                 |
| Sum Distinct   | Return the sum unique number of all values within the field.   |

### List of Built-in Calculated Field Functions

| Function  | Description  | Result Data Types                 | Examples |
|---|--|-----------------------------------|----------|
| <b>AVG</b><br><code>AVG(expression)</code><br>Numeric, Money                          | Returns the average of the values in a group. Null values are ignored. | Numeric, Money                    |          |
| <b>COUNT</b><br><code>COUNT(expression)</code><br>Any data type except Image and Lob. | Returns the number of items in a group.                                | Numeric.                          |          |
| <b>MAX</b><br><code>MAX(expression)</code>  | Returns the maximum  | The same data type as expression. |          |

# MARQUIS

| Function  | Description   | Result Data Types                 | Examples  |
|---|---|-----------------------------------|---|
| Any data type except Image and Lob.   | value in a group.   |                                   |   |
| <b>MIN</b><br><code>MIN(expression)</code><br>Any data type except Image and Lob. | Returns the minimum value in a group.   | The same data type as expression. |   |
| <b>SUM</b><br><code>SUM(expression)</code><br>Numeric, Money.                     | Returns the sum of all the values in a group. Null values are ignored.                    | The same data type as expression. |   |
| <b>LEN</b><br><code>LEN(expression)</code><br>Text.                               | Returns the number of characters of the given text expression, excluding trailing blanks. | Numeric.                          |   |
| <b>ROUND</b><br><code>ROUND(expression)</code><br>Numeric, Money.                 | Returns the expression rounded to the specified length or precision.                      | The same data type as expression. |   |
| <b>CONCAT</b><br><code>CONCAT(expression, expression [, ...])</code><br>Text.     | Returns the concatenation of all the parameters in that exact order.                      | Text.                             | <code>CONCAT('ab', 'cd', [SHIP COUNTRY])</code> |
| <b>GETDATE</b><br><code>GETDATE()</code><br>N/A.                                  | Returns the current system date and time.   | Datetime.                         |   |

# MARQUIS

| Function  | Description   | Result Data Types                       | Examples  |
|---|---|---|---|
| <p><b>DATEADD</b></p> <pre>DATEADD(datepart, number, date)</pre> <p><b>datepart:</b> the part of the date. (See table <a href="#">List of Dateparts and Abbreviations</a>)</p> <p><b>number:</b> the value used to increment datepart.</p> <p><b>date:</b> an expression that returns a datetime value.</p> | <p>Returns a new datetime value based on adding an interval to the specified date.</p>                  | <p>Datetime.</p>                        | <pre>DATEADD(day, 3, [DueDate])</pre>             |
| <p><b>DATEDIFF</b></p> <pre>DATEDIFF(datepart, startdate, enddate)</pre> <p><b>datepart:</b> the part of the date. (See table <a href="#">List of Dateparts and Abbreviations</a>)</p> <p><b>startdate, enddate:</b> expressions that return datetime values.</p>   | <p>Returns the number of date and time boundaries crossed between two specified dates.</p>              | <p>Numeric.</p>                         | <pre>DATEDIFF(day, [OrderDate], [ShipDate])</pre> |
| <p><b>DATEPART</b></p> <pre>DATEPART(datepart, date)</pre> <p><b>datepart:</b> the part of the date. (See table <a href="#">List of Dateparts and Abbreviations</a>)</p> <p><b>date:</b> an expression that returns a datetime value.</p>   | <p>Returns a number representing the specified datepart of the specified date.</p>                      | <p>Numeric.</p>                         |   |
| <p><b>CONVERT</b></p> <pre>CONVERT(data_type, expression)</pre> <p><b>data_type:</b> any data type.</p>   | <p>Explicitly converts an expression of one data type to another, similar to <code>CAST..AS</code>.</p> | <p>The same data type as data_type.</p> |   |

# MARQUIS

| Function   | Description   | Result Data Types                        | Examples |
|--|---|--|----------|
| <p><b>expression:</b> any expression.</p>  |   |  |          |
| <p><b>CAST..AS</b></p> <p><code>CAST(expression AS data_type)</code></p> <p><b>data_type:</b> any data type.</p> <p><b>expression:</b> any expression.</p>                 | <p>Explicitly converts an expression of one data type to another, similar to <code>CONVERT</code>.</p>  | <p>The same data type as data_type.</p>  |          |
| <p><b>ISNULL</b></p> <p><code>ISNULL(check_expression, replacement_expression)</code></p> <p><b>check_expression</b> and <b>replacement_expression:</b> any data type.</p> | <p>Returns the value of check_expression if it is not NULL; otherwise, returns the value of replacement_expression.</p>   | <p>The same data type as expression.</p> |          |
| <p><b>BETWEEN..AND</b></p> <p><code>test_expression [NOT] BETWEEN begin_expression AND end_expression</code></p> <p>Any data type except Image and Lob.</p>                | <p>Returns TRUE if the value of test_expression is greater than or equal to the value of begin_expression and less than or equal to the value of end_expression, otherwise returns FALSE.</p> | <p>Boolean.</p>                          |          |
| <p><b>AND</b></p> <p><code>boolean_expression AND boolean_expression</code></p> <p>Boolean.</p>  | <p>Returns TRUE when both expressions are TRUE, otherwise returns FALSE.</p>  | <p>Boolean.</p>                          |          |

# MARQUIS

| Function  | Description   | Result Data Types  | Examples |
|---|---|--|----------|
| <p><b>OR</b></p> <pre>boolean_expression AND boolean_expression</pre> <p>Boolean.</p>   | <p>Returns TRUE when either expression is TRUE, otherwise returns FALSE.</p>  | <p>Boolean.</p>  |          |
| <p><b>DISTINCT</b></p> <pre>DISTINCT (expression) or DISTINCT expression</pre> <p>Any data type except Image and Lob.</p>   | <p>Returns unique values.</p>   | <p>The same data type as expression.</p>   |          |
| <p><b>IFF</b></p> <pre>IFF (boolean_expression, true_expression[, false_expression])</pre> <p><b>boolean_expression:</b><br/>Boolean.</p> <p><b>true_expression, false_expression:</b> any data type except Image and Lob.</p>                                | <p>Returns the value of true_expression when boolean_expression is TRUE, otherwise returns the value of false_expression.</p> | <p>The highest precedence data type from data types of true_expression and false_expression.</p> |          |
| <p><b>IF..THEN..ELSE..END</b></p> <pre>IF (boolean_expression) THEN (true_expression) [ELSE (false_expression)] END</pre> <p><b>boolean_expression:</b><br/>Boolean.</p> <p><b>true_expression, false_expression:</b> any data type except Image and Lob.</p> | <p>Returns the value of true_expression when boolean_expression is TRUE, otherwise returns the value of false_expression.</p> | <p>The highest precedence data type from data types of true_expression and false_expression.</p> |          |
| <p><b>CASE..WHEN..THEN..ELSE..END</b></p>   | <p>Returns the value of</p>   | <p>The highest precedence</p>  |          |

# MARQUIS

| Function   | Description  | Result Data Types  | Examples |
|--|--|--|----------|
| <pre>CASE (input_expression) WHEN (when_expression) THEN (result_expression) [...n] [ELSE (else_result_expression)] END</pre> <p>Any data type except Image and Lob.</p> | <p>result_expression on matching the first when_expression with the value equal to input_expression, otherwise return the value of else_result_expression.</p> | <p>data type from data types of all result_expressions and else_result_expression.</p> |          |
| <p><b>RUNNINGSUM</b></p> <pre>RUNNINGSUM(expression)</pre> <p>Numeric, Money.</p>  | <p>Returns the sum of all the values of expression from the first row up to the current row.</p>   | <p>The same data type as expression.</p>   |          |
| <p><b>RUNNINGAVG</b></p> <pre>RUNNINGAVG(expression)</pre> <p>Numeric, Money.</p>  | <p>Returns the average of all the values of expression from the first row up to the current row.</p>   | <p>The same data type as expression.</p>   |          |
| <p><b>RUNNINGCOUNT</b></p> <pre>RUNNINGCOUNT(expression)</pre> <p>Any data type except Image and Lob.</p>  | <p>Returns the number of unique values of expression from the first row up to the current row.</p>   | <p>Numeric.</p>  |          |

## List of Date parts and Abbreviations

| Datepart    | Abbreviations   |
|-------------|-----------------|
| <b>year</b> | <b>yy, yyyy</b> |

# MARQUIS

|             |        |
|-------------|--------|
| quarter     | qq, q  |
| month       | mm, m  |
| dayofyear   | dy, y  |
| day         | dd, d  |
| week        | ww, wk |
| weekday     | dw     |
| hour        | hh     |
| minute      | mi, n  |
| second      | ss, s  |
| millisecond | ms     |

## Operators

### List of Operators

| Operator   | Description   | Result Data Types | Examples   |
|--|---|-------------------|--|
| <p><b>+ (Text Concatenation)</b></p> <pre>expression + expression</pre> <p>Text + Text</p> | <p>Concatenates two text values in that exact order.</p> <p>Not supported by Oracle and PostgreSQL. In cases where Oracle or PostgreSQL raises an error, please use the <code>CONCAT</code> function instead.</p> | Text              | <p>To return employee's full name from <code>[FirstName]</code> and <code>[LastName]</code> columns:</p> <pre>[FirstName]<br/>+ ' ' +<br/>[LastName]</pre> |

# MARQUIS

| Operator   | Description   | Result Data Types                                | Examples   |
|--|---|--|--|
| <p><b>+ Add</b></p> <p><code>expression + expression</code></p> <p>Numeric/Money +<br/>Numeric/Money</p> <p>Datetime + Numeric</p> <p>Numeric + Datetime</p> | <p>Adds two numbers or two currency amount, or adds a number of days to a date.</p>             | <p>The data type with the higher precedence.</p> | <p>Two and a half days from hire date:</p> <p><code>[HireDate] + 2.5</code></p>                            |
| <p><b>- (Subtract)</b></p> <p><code>expression - expression</code></p> <p>Numeric/Money -<br/>Numeric/Money</p> <p>Datetime - Numeric</p>                    | <p>Subtracts two numbers or two currency amount, or subtracts a number of days from a date.</p> | <p>The data type with the higher precedence.</p> | <p>The number of products in stock after 10 have been taken out:</p> <p><code>UnitsInStock - 10</code></p> |
| <p><b>* (Multiply)</b></p> <p><code>expression * expression</code></p> <p>Numeric/Money *<br/>Numeric/Money</p>  | <p>Multiplies two numbers, or a currency amount with a number.</p>                              | <p>Number</p>                                    | <p>Price after 15% discount:</p> <p><code>UnitPrice * 0.85</code></p>                                      |
| <p><b>/ (Divide)</b></p> <p><code>expression / expression</code></p> <p>Numeric/Money /<br/>Numeric/Money</p>  | <p>Divides one number or currency amount by another.</p>  | <p>Number</p>                                    | <p>Price after 50% discount:</p> <p><code>UnitPrice / 2</code></p>   |
| <p><b>= (Equals)</b></p> <p><code>expression = expression</code></p>   | <p>Compares the equality in value of two expressions.</p>                                       | <p>Boolean</p>                                   | <p>Employees named 'John':</p>   |

# MARQUIS

| Operator   | Description   | Result Data Types | Examples  |
|--|---|-------------------|---|
| <p>Any data type except Image and Lob.</p>   | <p>The expression of the lower precedence data type is converted to the higher precedence data type in case data types are different.</p>   |                   | <pre>[FirstName] = 'John'</pre>   |
| <p><b>&gt; (Greater Than)</b></p> <pre>expression &gt; expression</pre> <p>Any data type except Image and Lob.</p> | <p>Compares if the value of the first expression is higher than the second expression. The expression of the lower precedence data type is converted to the higher precedence data type in case data types are different.</p> | <p>Boolean</p>    | <p>Products with more than 10 units in stock:</p> <pre>UnitsInStock &gt; 10</pre> |
| <p><b>&lt; (Less Than)</b></p> <pre>expression &lt; expression</pre> <p>Any data type except Image and Lob.</p>    | <p>Compares if the value of the first expression is lower than the second expression. The expression of the lower precedence data type is converted to the higher precedence data type in case data types are different.</p>  | <p>Boolean</p>    | <p>Products with less than 3 units in stock:</p> <pre>UnitsInStock &lt; 3</pre>   |
| <p><b>&gt;= (Greater Than or Equal To)</b></p>   | <p>Compares if the value of the first expression is higher</p>  | <p>Boolean</p>    |   |

# MARQUIS

| Operator   | Description   | Result Data Types | Examples  |
|--|---|-------------------|---|
| <p><code>expression &gt;= expression</code></p> <p>Any data type except Image and Lob.</p>   | <p>than or equal to the second expression. The expression of the lower precedence data type is converted to the higher precedence data type in case data types are different.</p>   |                   |   |
| <p><b>&lt;= (Less Than or Equal To)</b></p> <p><code>expression &lt;= expression</code></p> <p>Any data type except Image and Lob.</p> | <p>Compares if the value of the first expression is lower than or equal to the second expression. The expression of the lower precedence data type is converted to the higher precedence data type in case data types are different.</p>            | <p>Boolean</p>    |   |
| <p><b>&lt;&gt; (Not Equal To)</b></p> <p><code>expression &lt;&gt; expression</code></p> <p>Any data type except Image and Lob.</p>    | <p>Compares if the value of the first expression is not equal to the second expression. The expression of the lower precedence data type is converted to the higher precedence data type in case data types are different. types are different.</p> | <p>Boolean</p>    | <p>Products still available in stock:</p> <p><code>UnitsInStock &lt;&gt; 0</code></p> |



## Attention with Possibly NULL Fields

Special attention when checking if a field has value: the equal operator cannot be used with `NULL`, `IS NULL` and `IS NOT NULL` should be used instead.

To check if there is a photo for an employee in Northwind database: `[Photo] IS NOT NULL`

Similarly, when using other comparison operators (e.g. "Greater Than >") on a field that possibly has `NULL` value: the `NULL` case must be tested separately.

To check if for employees with hire date later than 2010, including unknown hire date: `[HireDate] IS NULL OR [HireDate] > '01-01-2010'`