Relational DB & SQL - C11

Other Functions & Expressions

To do: Go through the activity to the end

ROUND() and ISNULL() Function

```
ROUND(numeric_expression , length [ ,function ])
```

Returns a numeric value, rounded to the specified length or precision.

`ROUND()` function takes three arguments:

numeric_expression: Is an expression of the exact numeric or approximate numeric data type category, except for the bit data type.

length: Is the precision to which numeric_expression is to be rounded. length must be an expression of type tinyint, smallint, or int. When length is a positive number, numeric_expression is rounded to the number of decimal positions specified by length. When length is a negative number, numeric_expression is rounded on the left side of the decimal point, as specified by length.

function: Is the type of operation to perform. function must be tinyint, smallint, or int. When function is omitted or has a value of 0 (default), numeric_expression is rounded. When a value other than 0 is specified, numeric_expression is truncated.

Examples:

ROUND returns a rounded numeric_expression, regardless of data type, when length is a negative number.

query:

```
1  SELECT ROUND(565.49, -1) AS col;
```

result:

```
1  col
2  ------
3  570.00
```

query:

```
1  SELECT ROUND(565.49, -2) AS col;
```

result:

```
col
------
600.00
```

The following example shows two expressions that demonstrate by using ROUND the last digit is always an estimate.

query:

```
SELECT ROUND(123.9994, 3) AS col1, ROUND(123.9995, 3) AS col2;
```

result:

```
col1      col2
--------  --------
123.9990  124.0000
```

The following example shows rounding and approximations.

query:

```
SELECT ROUND(123.4545, 2) col1, ROUND(123.45, -2) AS col2;
```

result:

```
col1      col2
--------  ------
123.4500  100.00
```

The following example uses two SELECT statements to demonstrate the difference between rounding and truncation. The first statement rounds the result. The second statement truncates the result.

query:

```
SELECT ROUND(150.75, 0) AS col1, ROUND(150.75, 0, 1) AS col2;
```

return:

```
col1      col2
------  ------
151.00  150.00
```

`ISNULL(check expression, replacement value)`

Replaces `NULL` with the specified replacement value.

`ISNULL()` function takes two arguments:

check expression: Is the expression to be checked for `NULL`. *check expression* can be of any type.

replacement value: Is the expression to be returned if *check expression* is NULL. *replacement value* must be of a type that is implicitly convertible to the type of *check expression*.

Examples:

query:

```
SELECT ISNULL(NULL, 'Not null yet.') AS col;
```

result:

```
col
-------------
Not null yet.
```

query:

```
SELECT ISNULL(1, 2) AS col;
```

result:

```
col
---
1
```

If you work on the table values, you can write column names replace "1" and "2" in the above query.

For example, you want to calculate the average list price. But the column has null values. If you want to fill the null values with a specific value, you can use `ISNULL()` function.

Previous Next