

DECFLOAT (Firebird 4.0)

Source: https://github.com/FirebirdSQL/firebird/blob/master/doc/sql.extensions/README.data_types

Function

DB2-compliant numeric type. **DECFLOAT** precisely (unlike **FLOAT** or **DOUBLE PRECISION** that provide binary approximation) stores decimal values being therefore ideal choice for business applications. Firebird according to IEEE standard has both 16- and 34-digit decimal float encodings. All intermediate calculations are performed with 34-digit values.

Author Alex Peshkoff peshkoff@mail.ru

Syntax rules

```
DECFLOAT  
DECFLOAT(16)  
DECFLOAT(34)
```

Storage

64-bit / 128-bit, format according to IEEE 754 Decimal64/Decimal128

Example(s)

1. `DECLARE VARIABLE VAR1 DECFLOAT(34);`
2. `CREATE TABLE TABLE1 (FIELD1 DECFLOAT(16));`

Note(s)

1. If no precision has been specified in the type declaration, the precision is 34.

2. A number of standard functions can be used with **DECFLOAT** datatype. It is:

```
ABS, CEILING, EXP, FLOOR, LN, LOG, LOG10, POWER, SIGN, SQRT.
```

Aggregate functions **SUM**, **AVG**, **MAX** and **MIN** also work with **DECFLOAT** data. All statistics aggregates (like but not limited to **STDDEV** or **CORR**) work with **DECFLOAT** data.

3. Firebird supports four functions, specially designed to support **DECFLOAT** data:

- **COMPARE_DECFLOAT** - compares two **DECFLOAT** values to be equal, different or unordered. Returns **SMALLINT** value which can be as follows:

```
0 - values are equal  
1 - first value is less than second  
2 - first value is greater than second  
3 - values unordered (i.e. one or both is NAN / SNAN)
```

Unlike comparison operators ('<', '=', '>', etc.) comparison is exact - i.e. `COMPARE_DECFLOAT(2.17, 2.170)` returns 2, not 0.

- **NORMALIZE_DECFLOAT** - has single `DECFLOAT` argument returned in it's simplest form. That means that for any nonzero value trailing zero are removed with appropriate correction of an exponent. For example `NORMALIZE_DECFLOAT(12.00)` returns 12 and `NORMALIZE_DECFLOAT(120)` returns 1.2E+2.
- **QUANTIZE** - has two `DECFLOAT` arguments. The returned value is first argument scaled using second value as a pattern. For example `QUANTIZE(1234, 9.999)` returns 1234.000.
- **TOTALORDER** - compares two `DECFLOAT` values including any special value. The comparison is exact. Returns `SMALLINT` value which can be as follows:

1. 1 - first value is less than second

0 - values are equal

1 - first value is greater than second

`DECFLOAT` values are ordered as follows:

1. $\text{nan} < -\text{snan} < -\text{inf} < -0.1 < -0.10 < -0 < 0 < 0.10 < 0.1 < \text{inf} < \text{snan} < \text{nan}$

4. Firebird supports new session control operator `SET DECFLOAT`. It has following forms:

- **SET DECFLOAT ROUND <mode>** - controls rounding mode used in operations with `DECFLOAT` values. Valid modes are: `CEILING` (towards +infinity), `UP` (away from 0), `HALF_UP` (to nearest, if equidistant - up), `HALF_EVEN` (to nearest, if equidistant - ensure last digit in the result to be even), `HALF_DOWN` (to nearest, if equidistant - down), `DOWN` (towards 0), `FLOOR` (towards -infinity), `REROUND` (up if digit to be rounded is 0 or 5, down in other cases). `HALF_UP` rounding is used by default.
- **SET DECFLOAT TRAPS TO <comma-separated traps list - may be empty>** - controls which exceptional conditions cause a trap. Valid traps are: `Division_by_zero`, `Inexact`, `Invalid_operation`, `Overflow` and `Underflow`. By default traps are set to: `Division_by_zero`, `Invalid_operation`, `Overflow`.
- **SET DECFLOAT BIND <bind-type>** - controls how are `DECFLOAT` values represented in outer world (i.e. in messages or in `XSQLDA`). Valid binding types are: `NATIVE` (use IEEE754 binary representation), `CHAR/CHARACTER` (use ASCII string), `DOUBLE PRECISION` (use 8-byte FP representation - same as used for `DOUBLE PRECISION` fields) or `BIGINT` with possible comma-separated `SCALE` clause (i.e. '`BIGINT, 3`'). Various bindings are useful if one plans to use `DECFLOAT` values with some old client not supporting native format. One can choose between strings (ideal precision, but poor support for further processing), floating point values (ideal support for further processing but poor precision) or scaled integers (good support for further processing and required precision but range of values is very limited). When using in a tool like generic purpose GUI client choice of `CHAR` binding is OK in most cases. By default `NATIVE` binding is used.

5. The length of `DECFLOAT` literals are limited to 1024 characters. For longer values, you will need to use the scientific notation. For example, the `0.0<1020 zeroes>11` cannot be used as a literal, instead you can use the equivalent in scientific notation: `1.1E-1022`. Similarly `10<1022 zeroes>0` can be presented as `1.0E1024`.

From:
<http://ibexpert.com/docu/> - **IBExpert**

Permanent link:
<http://ibexpert.com/docu/doku.php?id=01-documentation:01-13-miscellaneous:glossary:decfloat>

Last update: **2023/08/14 15:55**

