Real vs. Floating Point vs. Money

Asked 16 years, 1 month ago   Modified 6 months ago   Viewed 28k times

▲

21

▼

Why is it that when I save a value of say 40.54 in SQL Server to a column of type Real, that it returns a value that is more like 40.53999878999 instead of 40.54? I've seen this a few times but have never figured out quite why it happens. Has anyone else experienced this issue, and if so, what causes it?

sql-server   types

Share

Improve this question

Follow

edited Jun 10 at 17:45

asked Nov 7, 2008 at 19:40

Middletone
4,270 ● 12 ● 56 ● 74

1   This: panix.com/~arnow/brooklyn_college/diatribe.html
    – Moshe Aug 29, 2013 at 16:11
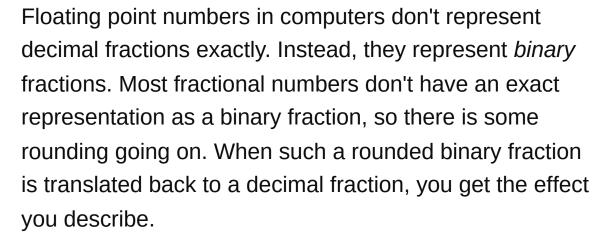
4 Answers

Sorted by: Highest score (default) ⇕

▲

46

▼

Have a look at [What Every Computer Scientist Should Know About Floating Point Arithmetic](#).

Floating point numbers in computers don't represent decimal fractions exactly. Instead, they represent *binary* fractions. Most fractional numbers don't have an exact representation as a binary fraction, so there is some rounding going on. When such a rounded binary fraction is translated back to a decimal fraction, you get the effect you describe.

For storing money values, SQL databases normally provide a DECIMAL type that stores exact decimal digits. This format is slightly less efficient for computers to deal with, but it is quite useful when you want to avoid decimal rounding errors.

Share   Improve this answer

Follow

3   Actually, it's *significantly* slower for computers to deal with (around ten times, due primarily to the lack of hardware support), but the slowness doesn't matter for most applications since numeric operations are nowhere near the bottleneck. – crazy2be Aug 6, 2011 at 2:45

3   Floating point numbers don't represent **some** decimal fractions exactly. Real numbers in real life also doesn't

represent *some* fractions exactly. `1/2` , for example, can be represented exactly on both systems. – axiac Sep 13, 2016 at 12:09

Floating point numbers use binary fractions, and they don't correspond exactly to decimal fractions.

For money, it's better to either store number of cents as integer, or use a decimal number type. For example, Decimal(8,2) stores 8 digits including 2 decimals (xxxxxx.xx), i.e. to cent precision.

Share Improve this answer

Follow

edited Sep 13, 2016 at 12:06

jumxozizi
649 ● 10 ● 23

answered Nov 7, 2008 at 19:42

Javier
62.4k ● 9 ● 81 ● 126

I'd love to have both of you get the points for answering well. Instead of storing the cents we are using themoney value. Do you think that this is also a good practice? – Middletone Nov 7, 2008 at 19:48

Storing the cents is bad news. Financial institutions often use fractional cents in calculations, and sometimes need to store them as well. I was on a project once where this came up after the cents-storing app was deployed. Very ugly. – MusiGenesis Nov 7, 2008 at 20:08

1   Just a note, isn't Decimal(8,2) actually xxxxxx.xx instead of xxxxxxxx.xx ? 8 in the definition refers to the total number of

digits both before and after the decimal point.
– Giannis Paraskevopoulos May 28, 2016 at 13:00

In a nutshell, it's for pretty much the same reason that one-third cannot be exactly expressed in decimal. Have a look at David Goldberg's classic paper "What Every Computer Scientist Should Know About Floating-Point Arithmetic" for details.

7

Share Improve this answer

Follow

edited May 15 at 14:23

Wolf
10.2k ● 8 ● 66 ● 112

answered Nov 7, 2008 at 19:44

Sherm Pendley
13.6k ● 3 ● 47 ● 57

I fixed the dead link to the moved content in docs.oracle.com/cd/E19957-01/806-3568/ncg_goldberg.html (see also the archive at web.archive.org/web/20090212082339/http://docs.sun.com/source/...) – Wolf May 15 at 14:22 ✎

To add a clarification, a floating point numbers stored in a computer behaves as described by other posts here, because as described, it is stored in binary format. This means that unless it's value (both the mantissa and exponent components of the value) are powers of two, and cannot be represented exactly.

1

Some systems, on the other hand store fractional numbers in decimal (SQL Server Decimal, and Numeric data types, and Oracle Number datatype for example,) and then their internal representation is, therefore, exact for any number that is a power of 10. But then numbers that are not powers of 10 cannot be represented exactly.

Share Improve this answer

Follow

answered Nov 7, 2008 at 20:27

Charles Bretana

146k ● 23 ● 155 ● 218