Subject: Re: WHERE problems (longish)
Posted by David Fanning on Tue, 22 Jul 2003 15:47:14 GMT
View Forum Message <> Reply to Message

Benjamin Panter writes:

> The values which have -1 certainly exist - and were generated in exactly
> the same way as the others. I've put the array online if anyone fancies
> looking at it - http://www.roe.ac.uk/~bdp/where_problem.idl
>
> Am I being stupid again? What is special about 2980,3000 and 3020??

There is nothing special about *those* numbers, but those
are not the numbers you are using in your WHERE statement.
You are using 2980.0, 3000.0, and 3020.0. While there isn't
a big difference between integers and floats to you, there
is a HUGE difference to a computer. Better read these articles:

http://www.dfanning.com/math_tips/sky_is_falling.html
http://www.dfanning.com/math_tips/razoredge.html

Cheers,

David

--
David W. Fanning, Ph.D.
Fanning Software Consulting, Inc.
Phone: 970-221-0438, E-mail: david@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: WHERE problems (longish)
Posted by Justin[2] on Tue, 22 Jul 2003 15:49:21 GMT
View Forum Message <> Reply to Message

This could possibly be a rounding problem with floating points. Try
using integers instead of floats for your wavelengths:

print, where(3000 eq ROUND(reform(dust_lookup[*,0])))

or look for values between 2999.9 and 3000.1 say...

Justin

Benjamin Panter <See@the_end.not> wrote in
news:3F1D5706.7050901@the_end.not:

> print, where(2900. eq reform(dust_lookup[*,0]))
> which works absolutly fine for most values: unfortunatly not for all:

---

Subject: Re: WHERE problems (longish)
Posted by Benjamin Panter on Tue, 22 Jul 2003 15:59:50 GMT
View Forum Message <> Reply to Message

David Fanning wrote:
> Benjamin Panter writes:
>
>
>> The values which have -1 certainly exist - and were generated in exactly
>> the same way as the others. I've put the array online if anyone fancies
>> looking at it - http://www.roe.ac.uk/~bdp/where_problem.idl
>>
>> Am I being stupid again? What is special about 2980,3000 and 3020??
>
>
> There is nothing special about *those* numbers, but those
> are not the numbers you are using in your WHERE statement.
> You are using 2980.0, 3000.0, and 3020.0. While there isn't
> a big difference between integers and floats to you, there
> is a HUGE difference to a computer. Better read these articles:
>
> http://www.dfanning.com/math_tips/sky_is_falling.html
> http://www.dfanning.com/math_tips/razoredge.html

Yeup, thanks for that David - I'm still a bit confused though, as the values I give in the test program (3000.) are floats and the values in the look up table are also floats! I nievely thought that would avoid the razors and ints. Is it just by luck that where found the majority and lost those three?

After reading your pages I've come up with a solution though... if I replace the where lines with

print, where(abs(3000. - reform(dust_lookup[*,0])) lt 0.1)

I come out with the right answer.

thanks for you help,

Ben

--
Ben Panter, Edinburgh
My name (no spaces)@bigfoot which is a com.

---

Subject: Re: WHERE problems (longish)
Posted by David Fanning on Tue, 22 Jul 2003 16:07:21 GMT
View Forum Message <> Reply to Message

Benjamin Panter writes:

> Yeup, thanks for that David - I'm still a bit confused though, as the
> values I give in the test program (3000.) are floats and the values in
> the look up table are also floats! I nievely thought that would avoid
> the razors and ints. Is it just by luck that where found the majority
> and lost those three?
>
> After reading your pages I've come up with a solution though... if I
> replace the where lines with
>
> print, where(abs(3000. - reform(dust_lookup[*,0])) lt 0.1)
>
> I come out with the right answer.

And that would be the right approach. With floats
you can't (generally) ask if one float *equals*
another. You can ask if one float is within some
very tiny delta of another. That's what you ended
up doing here, and it works spectacularly. :-)

Cheers,

David
--
David W. Fanning, Ph.D.
Fanning Software Consulting, Inc.
Phone: 970-221-0438, E-mail: david@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

---

Subject: Re: WHERE problems (longish)
Posted by Paul Van Delst[1] on Tue, 22 Jul 2003 16:45:54 GMT
View Forum Message <> Reply to Message

David Fanning wrote:

>
> Benjamin Panter writes:
>
>> The values which have -1 certainly exist - and were generated in exactly
>> the same way as the others. I've put the array online if anyone fancies
>> looking at it - http://www.roe.ac.uk/~bdp/where_problem.idl
>>
>> Am I being stupid again? What is special about 2980,3000 and 3020??
>
> There is nothing special about *those* numbers, but those
> are not the numbers you are using in your WHERE statement.
> You are using 2980.0, 3000.0, and 3020.0. While there isn't
> a big difference between integers and floats to you, there
> is a HUGE difference to a computer. Better read these articles:
>
>  http://www.dfanning.com/math_tips/sky_is_falling.html
>  http://www.dfanning.com/math_tips/razoredge.html

Hmm. Reading that razor's edge article made me dig out a little f90 program I wrote to determine the real number spacing.

Given some number, A, one can do:

```
exponent=ceil(alog(abs(A))/alog(2.0d))
radix=2.0d  ; (machar(/double)).ibeta ??
epsilon=(machar(/double)).eps
spacing=epsilon * (radix^(exponent-1))
```

So for, say, A = 1.234568d+16

```
EXPONENT       LONG     =          54
SPACING        DOUBLE   =       2.0000000
```

For A = 1.234568d+01

```
EXPONENT       LONG     =           4
SPACING        DOUBLE   =    1.7763568e-15
```

For A = 1.234568d-01

```
EXPONENT       LONG     =          -3
SPACING        DOUBLE   =    1.3877788e-17
```

and for A = 1.234568d-16

```
EXPONENT       LONG     =         -52
SPACING        DOUBLE   =    2.4651903e-32
```

which agree with the outputs of my f90 code using the intrinsic functions EXPONENT and SPACING. The problem is, of course, what to do when A = 0.0. I would just use spacing = epsilon/2.0. I think that, in this case, doing something like

```
two=2.0d
radix=two
IF ( A .eq. 0.0 ) THEN $
  spacing = epsilon/two $
ELSE BEGIN
  exponent=ceil(alog(abs(A))/alog(two))
  spacing=epsilon * (radix^(exponent-1))
ENDELSE
```

where you are counting on the equality check .EQ. *not* to work for numbers that aren't represented as exactly zero (since exact zero can be represented). But I'm not sure. You may also need a check to limit the calculated exponent to the range allowed (the minexp and maxexp fields of the output from machar).

ANYway.... back to work....

paulv

--
Paul van Delst
CIMSS @ NOAA/NCEP/EMC
Ph: (301)763-8000 x7748
Fax:(301)763-8545

---