

RE: DBI-1.48 bind_param_inout produces ORA-01461 colliding with a CLOB

Source: <http://coding.derkeiler.com/Archive/Perl/perl.dbi.users/2005-07/msg00110.html>

- *From:* catharine_drozdowski@xxxxxxxxxx (Catharine Drozdowski)
 - *Date:* Wed, 13 Jul 2005 17:28:55 -0700
-

Steve,

There is an Oracle sql function TO_LOB which can be used to convert the input to a lob You might also check out the Oracle 9i Application developer documentation which has an excellent chapter on working with OBs.... You can find that documentation at otn.oracle.com... you may be asked to create an account, but it is well worth it.... Not sure if you have the ability to do so, but upgrading to 9206 would be a good idea. Check out the bugs fixed in 9205 and 9206 against the error you are getting....

catharine

-----Original Message-----

From: Steven Lembark [<mailto:lembark@xxxxxxxxxx>]

Sent: Wed 7/13/2005 10:00 AM

To: dbi-users@xxxxxxxxxx

Subject: DBI-1.48 bind_param_inout produces ORA-01461 colliding with a CLOB

Summary

=====

perl-5.8.7, DBI-1.48, Oracle-9.2.0.4.

This is probably an Oracle question in the long run: How to frame a query so that bind_param_inout can pass strings as varchar's and have them inserted into a clob. It'd be nicer if there were a way to handle this in Perl however...

In theory I could write something to query the data dictionary for LOB fields and adjust the binding type used with bind_param_inout; in reality I have a short time to finish this and need a quicker fix.

The default handling of varchar bumping into a CLOB w/in Oracle is probably what is killing me... This code has worked for any number of tables, so the CLOB issue is probably what's biting me.

RE: DBI-1.48 bind_param_inout produces ORA-01461 colliding with a CLOB

Any help would be appreciated.

=====
Detail
=====

Google on qw(perl DBI bind_param_inout ORA-01461)
in various combinations gets me nowhere. The POD
for bind_param_inout references bind_parm, which
notes that:

Perl only has string and number scalar data types. All
database types that aren't numbers are bound as
strings and must be in a format the database will
understand except where the bind_param() TYPE
attribute specifies a type that implies a particular
format. For example, given:

```
$sth->bind_param(1, $value, SQL_DATETIME);
```

the driver should expect \$value to be in the ODBC
standard SQL_DATETIME format, which is 'YYYY-MM-DD
HH:MM:SS'. Similarly for SQL_DATE, SQL_TIME etc.

As an alternative to specifying the data type in the
"bind_param" call, you can let the driver pass the
value as the default type ("VARCHAR"). You can then
use an SQL function to convert the type within the
statement. For example:

```
INSERT INTO price(code, price) VALUES (?, CONVERT(MONEY,?))
```

The "CONVERT" function used here is just an example.
The actual function and syntax will vary between dif-
ferent databases and is non-portable.

I cannot find a reference to a "convert varchar
to clob" function for Oracle (but, then, my
doc's on that are limited).

The default handling of varchar bumping into a
CLOB w/in Oracle is probably what is killing me...
This code has worked for any number of tables,
so the CLOB issue is probably what's biting me.

Aside: Searching the POD for "Placeholders and
Bind Values" gets me plenty of referencing lines
but no title with that name.

RE: DBI-1.48 bind_param_inout produces ORA-01461 colliding with a CLOB

RE: DBI-1.48 bind_param_inout produces ORA-01461 colliding with a CLOB

The code that blows up is a generic query handler.
The code here returns a closure bound for a
specific query. In this case, for an
insert ... returning query:

...

```
my @bound = ( 1..$total );

my @copy = ( 0..$countz{insert}-1 );
my @return = ( $countz{insert} .. $#bound );

$sth->bind_param_inout( $_+1, \@bound[$_], 4096 )
for( 0 .. $#bound );
sub
{
  @bound[ @copy ] = @_[ @copy ];

  # caller gets back an anon-array
  # of the return values (a la
  # fetchrow_arrayref) or undef if
  # the call fails.

  eval
  {
    $sth->execute;

    [ @bound[@return] ]
  }
}
```

Which works for tables without CLOB's.

Inserting into this table using the SQL below gives
me an oracle error:

```
SQL> desc entry_comment;
Name Null? Type
```

```
-----
ENTRY_COMMENT_ID NOT NULL NUMBER(10)
COMMENTS CLOB
ENTRY_ID NUMBER(10)
CREATOR VARCHAR2(16)
CREATE_DATE DATE
UPDATOR VARCHAR2(16)
UPDATE_DATE DATE
```

DBD::Oracle::st execute failed: ORA-01461: can bind a LONG value only for

RE: DBI-1.48 bind_param_inout produces ORA-01461 colliding with a CLOB

RE: DBI-1.48 bind_param_inout produces ORA-01461 colliding with a CLOB

insert into a LONG column (DBD ERROR: OCISstmtExecute) [for Statement

"insert into entry_comment

(

entry_comment_id,

comments,

creator,

create_date,

updater,

update_date,

entry_id

)

values

(

entry_comment_seq.nextval, -- entry_comment_id,

?, -- comments,

?, -- creator,

?, -- create_date,

?, -- updater,

?, -- update_date,

(-- entry_id

select entry_id

from entry

where entry_accno = ?

)

)

returning

entry_comment_id into ?" with ParamValues: :p5='2005.06.16-09:16:41',

:p3='2005.06.16-00:00:00', :p6='G00462083', :p1='Activins are dimeric

growth and differentiation factors which belong to the transforming growth

factor-beta (TGF-beta) superfamily of structurally related signaling

proteins. Activins signal through a heteromeric complex of receptor serine

kinases which include at least two type I (I and IB) and two type II (II

and IIB) receptors. These receptors are all transmembrane proteins,

composed of a ...', :p7='327011', :p4=undef, :p2='LocusLink'] at

/sandbox/lembark/lib/Plugin/Language/DML/DBI.pm line 209.

at /sandbox/lembark/lib/Plugin/Language/DML/DBI.pm line 209

eval {...} called at /sandbox/lembark/lib/Plugin/Language/DML/DBI.pm line

208

Plugin::Language::DML::DBI::__ANON__[/sandbox/lembark/lib/Plugin/Language/DML/DBI.pm:213]

called at /sandbox/lembark/lib/Plugin/Language/DML.pm line 146

Plugin::Language::DML::__ANON__[/sandbox/lembark/lib/Plugin/Language/DML.pm:160]('dest=HASH(0x861a30c

'Activins are dimeric growth and differentiation factors which...',

'LocusLink', '2005.06.16-00:00:00', 'undef', '2005.06.16-09:16:41',

'G00462083') called at

/sandbox/lembark/lib/Plugin/Language/DML/MultiToken.pm line 171

eval {...} called at /sandbox/lembark/lib/Plugin/Language/DML/MultiToken.pm

line 171

Plugin::Language::DML::MultiToken::__ANON__[/sandbox/lembark/lib/Plugin/Language/DML/MultiToken.pm:174]

called at /sandbox/lembark/lib/Plugin/Language/DML.pm line 146

Plugin::Language::DML::__ANON__[/sandbox/lembark/lib/Plugin/Language/DML.pm:160]('dest=HASH(0x861a30c

'Activins are dimeric growth and differentiation factors which...',

RE: DBI-1.48 bind_param_inout produces ORA-01461 colliding with a CLOB

RE: DBI-1.48 bind_param_inout produces ORA-01461 colliding with a CLOB

'LocusLink', '2005.06.16-00:00:00', 'undef', '2005.06.16-09:16:41',
'G00462083') called at
/sandbox/lembark/lib/Plugin/Language/DML/MultiToken.pm line 93
Plugin::Language::DML::MultiToken::__ANON__[/sandbox/lembark/lib/Plugin/Language/DML/MultiToken.pm:101
called at /sandbox/lembark/lib/Plugin/Language/DML.pm line 146
Plugin::Language::DML::__ANON__[/sandbox/lembark/lib/Plugin/Language/DML.pm:160]('ARRAY(0x8ad2380)',
'ARRAY(0x8ad1e70)', 'ARRAY(0x8b004dc)', 'ARRAY(0x8b00590)',
'ARRAY(0x8b000e0)', 'ARRAY(0x8b00188)', 'ARRAY(0x8b00230)',
'ARRAY(0x8b126e4)', 'ARRAY(0x8b1263c)', ...) called at copy-ccm-curation
line 656
dest::entry_comment('dest=HASH(0x861a30c)', 'ARRAY(0x8b12c48)') called at
copy-ccm-curation line 131
eval {...} called at copy-ccm-curation line 131

—
Steven Lembark 85-09 90th Street
Workhorse Computing Woodhaven, NY 11421
lembark@xxxxxxxxxxxx 1 888 359 3508

• ***Follow-Ups:***

- ◆ ***RE: DBI-1.48 bind_param_inout produces ORA-01461 colliding with a CLOB***
◇ *From:* Steven Lembark

- Prev by Date: ***ORA -03113 Error with perl 5.8.4 whne using certain NLS charsets***
- Next by Date: ***Trouble running Perl:DBI from PHP Script***
- Previous by thread: ***RE: DBI-1.48 bind_param_inout produces ORA-01461 colliding with a CLOB***
- Next by thread: ***RE: DBI-1.48 bind_param_inout produces ORA-01461 colliding with a CLOB***
- Index(es):
 - ◆ ***Date***
 - ◆ ***Thread***