Subject: Re: Dataminer: faster way to get all records in a IDLdbRecordset table?
Posted by Dick Jackson on Wed, 17 Sep 2003 16:03:50 GMT
View Forum Message <> Reply to Message

Hi Tim,

"Tim Williams" <timothy.williams@nvl.army.mil> wrote in message
news:faf44c99.0309170738.ae75526@posting.google.com...
> I want to put the records in a database table into an IDL table
> widget. For fairly large tables, (> ~1500 rows), it's fairly slow and
> I get "Not responding" in the Task Manager for awhile while I'm
> getting each record. Here's what I'm doing now:
>
> ors=obj_new('IDLdbRecordset', table=tablename)
> status=ors->moveCursor(/first)
> if status eq 1 then begin
> rec=ors->getRecord()
> status=ors->moveCursor(/next)
> while (status eq 1) do begin
> rec=[rec, ors->getRecord()]
> status=ors->moveCursor(/next)
> end while
> endif
>
> Is there a faster way to get all of the records?

With that number of records, I think a lot of time may be in the
innocent-looking array concatenation:

rec=[rec, ors->getRecord()]

At the end, you're taking an array of 1498 records, creating a *new*
array with 1499 records and throwing out the old one. Lots of extra
memory copying here. My faster way is below (this would go within your
'if status eq 1' block, and you'll have to rework the variable names, of
course):

=====

```
;;    Count how many records

nRecords = 0L
WHILE status NE 0 DO BEGIN
  nRecords = nRecords+1
  status = objRS -> MoveCursor(/Next)
ENDWHILE

;;    Get structure from first record, replicate it
```

```
status = objRS -> MoveCursor(/First)
aRecord = objRS -> GetRecord()
result = Replicate(aRecord, nRecords)

FOR recordI=1, nRecords-1 DO IF status NE 0 THEN BEGIN
  status = objRS -> MoveCursor(/Next)
  IF status EQ 0 THEN $              ; Fewer records than when
counted above
    result = result[0:recordI-1] $
  ELSE result[recordI] = objRS -> GetRecord()
ENDIF
```

=====

Hope this helps!

Cheers,
--
-Dick

Dick Jackson / dick@d-jackson.com
D-Jackson Software Consulting / http://www.d-jackson.com
Calgary, Alberta, Canada / +1-403-242-7398 / Fax: 241-7392

Subject: Re: Dataminer: faster way to get all records in a IDLdbRecordset table?
Posted by Olaf Stetzer on Thu, 18 Sep 2003 07:21:19 GMT
View Forum Message <> Reply to Message

Tim Williams schrieb:

```
>  I want to put the records in a database table into an IDL table
>  widget. For fairly large tables, (> ~1500 rows), it's fairly slow and
>  I get "Not responding" in the Task Manager for awhile while I'm
>  getting each record. Here's what I'm doing now:
>
>  ors=obj_new('IDLdbRecordset', table=tablename)
>  status=ors->moveCursor(/first)
>  if status eq 1 then begin
>     rec=ors->getRecord()
>     status=ors->moveCursor(/next)
>     while (status eq 1) do begin
>        rec=[rec, ors->getRecord()]
>        status=ors->moveCursor(/next)
>     end while
>  endif
>
>  Is there a faster way to get all of the records?
```

I am not sure if it is faster but you can try my function
sql_return_array appended to this email! There is even a
minimum of error handling in it.

Olaf

```
; Copyright (c) 2002, Olaf Stetzer, Forschungszentrum Karlsruhe GmbH IMK-3
;
; This is free software; you can redistribute it and/or modify it under the
; terms of the GNU General Public License as published by the Free Software
; Foundation Version 2.
;
; This is distributed in the hope that it will be useful, but WITHOUT
; ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
; FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License
; for more details.
;
; Email bug reports to olaf.stetzer@imk.fzk.de
;
;
; NAME:
;   sql_return_array
;
; PURPOSE:
;   This function returns a struct which contains the result of an
;   SQL-query.
;
;
; CATEGORY:
;   SQL-DATABASE
;
; CALLING SEQUENCE:
;   Result= sql_return_array(objectDB, sqlstr)
;
; INPUTS:
;   objectDB: a database Object for the DB which holds the table with
the desired data
;   sqlstr:   a complete SQL query string
;
; KEYWORD PARAMETERS:
;
;   group    If keyword is set, skip the counting of resulting records. Some
;            queries (like SHOW... and grouping queries) don't work
otherwise.
;
;
; OUTPUTS:
```

```
;   result: array of a struct which contains the data for the desired
fields. The fields
; within the structure have the same names as defined by the variable fiels.
;   These can be substructures depending on the fieldtype in the DB (for
example
;   a datetime-field will be returned as a (sub)structure
ODBC_SQL_TIMESTAMP).
;
;
; EXAMPLE:
;
;
; NOTES:
;
; If you have fields in your DB which are not valid as variable name in
IDL then
; use the AS alias function in the SQL Query to avoid error messages.
See the
; construction of count(*) As COUNT in the test query below.
;
;
; REVISION HISTORY:
;   Written     Olaf Stetzer, March 2002
;
; 21.03.2002 Added test for empty result, now returns 0 if no records
are obtained.
;    Simplified the construction of the resulting array by using count now!
;
;   08.07.2002  Record loop integer n changed to long integer type to
read longer record
;               sets (Martin Schnaiter)
;
;   28.11.2002  Added reform() to the result to avoid doing it in all
calling programs.
;    Added group keyword to reunify the two previous functions.
;-

function sql_return_array, oDB, sqlstr, group=group

; Test if Query returns any records and get the number of records (count):

if not keyword_set(group) then begin

 teststr='SELECT count(*) AS COUNT ' +
strmid(sqlstr,strpos(strupcase(sqlstr),' FROM '))

 oRS = obj_new('IDLDBRecordset',oDB ,SQL=teststr)
```

```
 status = oRS->MoveCursor(/FIRST)
 record = oRS->GetRecord()
 obj_destroy, oRS

 count=record.count

 if count eq 0 then begin

  print, 'Query returns no records: ', sqlstr
  return, 0

 endif

 ; Create the recordset-object:

 oRS = obj_new('IDLDBRecordset',oDB ,SQL=sqlstr)

endif else begin

 oRS = obj_new('IDLDBRecordset',oDB ,SQL=sqlstr)

 ; Count records "manually":

 status = oRS->MoveCursor(/FIRST)

 count=1
 last=0
 while last eq 0 do begin
  if (oRS->MoveCursor(/NEXT) eq 1) then count=count+1 else last=1
 endwhile

endelse

; Construct a well-dimensioned array for the result:

status = oRS->MoveCursor(/FIRST)
record = oRS->GetRecord()
record=replicate(record,count)

; Fill the array with the records:

for n=0l,count-1 do begin
 record[n]=oRS->GetRecord()
 status = oRS->MoveCursor(/NEXT)
endfor

obj_destroy, oRS
```

```
return, reform(record)
end
```

Subject: Re: Dataminer: faster way to get all records in a IDLdbRecordset table?
Posted by timothy.williams on Mon, 22 Sep 2003 12:41:01 GMT
View Forum Message <> Reply to Message

Thanks to both for your help. The problem was that I was growing my array by 1 each iteration. I should have seen that myself. Once I created the array first, then filled it out, things went MUCH faster.