
Subject: Re: IDL and sqlite

Posted by [Brian Larsen](#) on Mon, 10 Nov 2008 12:49:41 GMT

[View Forum Message](#) <> [Reply to Message](#)

I have played with this some but I found it a little awkward and not that fast. However some more thought may fix these issues. I attached the code that I used and then decided not to go any further, more for lack of time than technical reasons. This was all done on this idl:

```
IDL> print, !version
```

```
{ i386 darwin unix Mac OS X 6.4.1 Sep 25 2007    32    64}
```

The files I used all looked like this:

L	Phi
4.66432	1.65347
4.76471	1.68353
4.76471	1.71359
4.76471	1.74366
4.76471	1.77372
4.66432	1.80378

and on for a long way in lots of files.

```
:: .run create_ppb_sqlite
```

```
database_fname = 'ppb_database.sql'
```

```
:: create the table, error will throw if it already exists and thats
```

```
:: probably ok
```

```
spawn, 'sqlite3 ' + database_fname + " ' CREATE TABLE file(" + $  
    "file_id INTEGER PRIMARY KEY, " + $  
    "filename TEXT, " + $  
    "euv_date_arr integer[5], " + $  
    "jd real" + $  
    ");"
```

```
:: create a table for the data in the file
```

```
spawn, 'sqlite3 ' + database_fname + " ' CREATE TABLE data(" + $  
    "id INTEGER PRIMARY KEY, " + $  
    "file_id INTEGER, " + $  
    "l REAL, " + $  
    "phi REAL, " + $  
    "mlt REAL" + $
```

```
");"
```

```
;; read in the ppb files and feed their names into the database
progressBar = Obj_New("PROGRESSBAR")
progressBar->Start
progressBar2 = Obj_New("PROGRESSBAR")
progressBar2->Start

files = file_search('*.ppb.*')
FOR i = 0L, n_elements(files)-1 DO BEGIN
  spawn, 'sqlite3 ' + database_fname + " ' INSERT into file values("
+ $
  "NULL," + $
  ""+files[i]+"" + "," + $
  trim((euv_date2arr(files[i]))[0]) + "," + $
  trim(euv_date2jd(files[i]), '(d14.6))+ $
  "');"

  dat = read_ppa(files[i], /compress, /mlt)
  ind = where(finite(dat.l) EQ 1, n_ind)
  FOR j = 0L, n_ind-1 DO BEGIN
    spawn, 'sqlite3 ' + database_fname + " ' INSERT into data
values(" + $
    "NULL," + $
    "NULL," + $
    trim(dat.l[ind[j]]) + "," + $
    trim(dat.phi[ind[j]]) + "," + $
    trim(dat.mlt[ind[j]]) + $
    "');"
    IF j MOD 5 EQ 0 AND j NE 0 OR j EQ n_ind-1 THEN BEGIN
      IF i MOD 2 EQ 0 THEN progressBar2 -> SetProperty,
Color='red' ELSE progressBar2 -> SetProperty, Color='black'
      progressBar2->Update, float(j+1)/n_ind*100.
      cancelled = progressBar2->CheckCancel()
      IF cancelled THEN BREAK
    ENDIF
  ENDFOR

  IF cancelled THEN break
  progressBar->Update, float(i+1)/n_elements(files)*100.
  cancelled = progressBar->CheckCancel()
  IF cancelled THEN break

ENDFOR
```

progressBar->Destroy
Obj_Destroy, progressBar

progressBar2->Destroy
Obj_Destroy, progressBar2

END

Cheers,

Brian

Brian Larsen
Boston University
Center for Space Physics
<http://people.bu.edu/balarsen/Home/IDL>

Subject: Re: IDL and sqlite
Posted by [Anonymous](#) on Tue, 11 Nov 2008 16:55:12 GMT
[View Forum Message](#) <> [Reply to Message](#)

Originally posted by: Demitri

Brian,

You will get a good boost in performance if you batch all of the inserts into a single transaction - you're paying a big penalty otherwise. That's just the sqlite3 side - there's also the penalty of spawn creating a new process for every row of data you have, and that's adding up. But fixing these things is simple.

One option would be:

- * create a new temporary file
- * extract the values from your data file
- * write each INSERT statement to the temporary file
- * perform a single SPAWN passing the temp file to sqlite3, e.g.

spawn, 'sqlite3 ' + database_name + ' < ' + temp_filename

In the case that you don't need anything IDL specific to extract the data (or one is better with Perl/Python/etc. coding), a second option

would be to write a small external script that will parse the given data file, create the database, and insert the values (again, best in a batch). Then IDL can simply spawn that. That's up to the user.

Cheers,

Demitri

On 2008-11-10 07:49:41 -0500, Brian Larsen <balarsen@gmail.com> said:

> I have played with this some but I found it a little awkward and not
> that fast. However some more thought may fix these issues. I
> attached the code that I used and then decided not to go any further,
> more for lack of time than technical reasons. This was all done on
> this idl:

```
> IDL> print, !version  
> { i386 darwin unix Mac OS X 6.4.1 Sep 25 2007    32    64}
```

> The files I used all looked like this:

```
>      L      Phi  
> 4.66432  1.65347  
> 4.76471  1.68353  
> 4.76471  1.71359  
> 4.76471  1.74366  
> 4.76471  1.77372  
> 4.66432  1.80378
```

> and on for a long way in lots of files.

```
> ;; .run create_ppb_sqlite
```

```
> database_fname = 'ppb_database.sql'
```

```
> ;; create the table, error will throw if it already exists and thats
```

```
> ;; probably ok
```

```
> spawn, 'sqlite3 ' + database_fname + " ' CREATE TABLE file(" + $
```

```
>   "file_id INTEGER PRIMARY KEY, " + $
```

```
>   "filename TEXT," + $
```

```
>   "euv_date_arr integer[5]," + $
```

```
>   "jd real" + $
```

```
>   ");"
```

```

>
> ;; create a table for the data in the file
> spawn, 'sqlite3 ' + database_fname + " ' CREATE TABLE data(" + $
>   "id INTEGER PRIMARY KEY, " + $
>   "file_id INTEGER," + $
>   "l REAL," + $
>   "phi REAL," + $
>   "mlt REAL" + $
>   ");"
>
>
> ;; read in the ppb files and feed their names into the database
> progressBar = Obj_New("PROGRESSBAR")
> progressBar->Start
> progressBar2 = Obj_New("PROGRESSBAR")
> progressBar2->Start
>
> files = file_search('*.ppb.*')
> FOR i = 0L, n_elements(files)-1 DO BEGIN
>   spawn, 'sqlite3 ' + database_fname + " ' INSERT into file values("
> + $
>   "NULL," + $
>   ""+files[i]+"" + "," + $
>   trim((euv_date2arr(files[i]))[0]) + "," + $
>   trim(euv_date2jd(files[i]), '(d14.6)')+ $
>   ");"
>
>   dat = read_ppa(files[i], /compress, /mlt)
>   ind = where(finite(dat.l) EQ 1, n_ind)
>   FOR j = 0L, n_ind-1 DO BEGIN
>     spawn, 'sqlite3 ' + database_fname + " ' INSERT into data
> values(" + $
>     "NULL," + $
>     "NULL," + $
>     trim(dat.l[ind[j]]) + "," + $
>     trim(dat.phi[ind[j]]) + "," + $
>     trim(dat.mlt[ind[j]]) + $
>     ");"
>     IF j MOD 5 EQ 0 AND j NE 0 OR j EQ n_ind-1 THEN BEGIN
>       IF i MOD 2 EQ 0 THEN progressBar2 -> SetProperty,
> Color='red' ELSE progressBar2 -> SetProperty, Color='black'
>       progressBar2->Update, float(j+1)/n_ind*100.
>       cancelled = progressBar2->CheckCancel()
>       IF cancelled THEN BREAK
>     ENDIF
>   ENDFOR
>
>   IF cancelled THEN break

```

```
> progressBar->Update, float(i+1)/n_elements(files)*100.  
> cancelled = progressBar->CheckCancel()  
> IF cancelled THEN break  
>  
>  
>  
> ENDFOR  
>  
>  
> progressBar->Destroy  
> Obj_Destroy, progressBar  
>  
> progressBar2->Destroy  
> Obj_Destroy, progressBar2  
>  
>  
> END  
>  
>  
>  
> Cheers,  
>  
> Brian  
>  
> -----  
> Brian Larsen  
> Boston University  
> Center for Space Physics  
> http://people.bu.edu/balarsen/Home/IDL
```
