Subject: Date emulation for Sqlite
Posted by mirek on Tue, 12 Sep 2006 15:37:58 GMT
View Forum Message <> Reply to Message

Sqlite does not support DATE type, which is a bit trouble with U++, as we cannot use natural Date <- column conversions.

I have this idea of Date emulation in Sqlite:

Store Date d as

(d - Date(1970, 1, 1)) * 1.0e300.

This way we would know that any double stored in database bigger than 1.0e300 is Date. Comparisons would work as well, the only downsize is that "number of days between" would be a bit high and that you would not be able to store doubles bigger than 1.0e300...

Should I do that? Any better solutions?

Mirek

Subject: Re: Date emulation for Sqlite
Posted by Ulti on Tue, 12 Sep 2006 16:20:04 GMT
View Forum Message <> Reply to Message

maybe this can help,I don't match datetime type yet.
http://www.sqlite.org/cvstrac/wiki?p=DateAndTimeFunctions

Subject: Re: Date emulation for Sqlite
Posted by zsolt on Tue, 12 Sep 2006 16:28:47 GMT
View Forum Message <> Reply to Message

My problem with this special double is that it would break two things:
1. Internal sqlite date operations as Ulti pointed to that
2. the readability of the result of a simple sql select.

I don't know well the logic of sch files yet, but wouldn't it be possible to use them to get the type of a column?

Subject: Re: Date emulation for Sqlite
Posted by unodgs on Tue, 12 Sep 2006 18:07:40 GMT

The only problem with sqlite is when you select "stringed dates" into value and you want that value to be date type.
Now I use ScanDate which seems to work well.

```
Value dt = ScanDate(String(SQL[DT])); //DT column that holds date

Date ScanDate(String &s)
{
 if(s.IsEmpty())
  return Null;
 int year  = atoi(s.Left(4));
 int month = atoi(s.Mid(5, 2));
 int day   = atoi(s.Right(2));

 return Date(year, month, day);
}
```

But I agree that is troublesome and your chage is wanted.

Subject: Re: Date emulation for Sqlite
Posted by mirek on Tue, 12 Sep 2006 18:33:51 GMT

zsolt wrote on Tue, 12 September 2006 12:28My problem with this special double is that it would break two things:
1. Internal sqlite date operations as Ulti pointed to that
2. the readability of the result of a simple sql select.

I don't know well the logic of sch files yet, but wouldn't it be possible to use them to get the type of a column?

1. can be to large degree solved by SqlExp. In fact, it is "timestamp" type in disguise...

2. OK, this has not come to my mind, that is valid concern. OTOH, it affects non-U++ tools only. In U++, you would be able to get correct dates only.

What about to be a little crazy and encode Date(2006, 9, 12) as 0.20060912e300 ?

Anyway, any other ideas how to store Date into Sqlite?

The only other one that I have is to store it as string with prefix - that however would mean to introduce some trouble to regular strings as well.

Well, I guess I owe the reason why that is so important. The thing to solve is that you cannot now with Sqlite associate EditDate with Sqlite column. Especially troublesome with SqlCtrls or SqlArray.

Mirek

Subject: Re: Date emulation for Sqlite
Posted by zsolt on Tue, 12 Sep 2006 18:57:18 GMT
View Forum Message <> Reply to Message

Quote:What about to be a little crazy and encode Date(2006, 9, 12) as 0.20060912e300 ?
It is much more readale

BTW, what about time?

Subject: Re: Date emulation for Sqlite
Posted by mirek on Tue, 12 Sep 2006 19:01:00 GMT
View Forum Message <> Reply to Message

Time would be nice as well, but is not nearly as important. I am using 2 DATE columns per table average, however I never used any time column in my career.

Mirek

Subject: Re: Date emulation for Sqlite
Posted by zsolt on Thu, 14 Sep 2006 09:02:57 GMT
View Forum Message <> Reply to Message

FYI: Using sqlite, it is possible, to use any typename in a create table statement. For example, this code is valid SQL:
create table test (id integer primary key, some_date_field date)
And of course, you can get this "custom" type, querying the type of the field.

Subject: Re: Date emulation for Sqlite
Posted by zsolt on Thu, 14 Sep 2006 09:26:39 GMT
View Forum Message <> Reply to Message

You can use these two functions to get the declared type of a column:
const char *sqlite3_column_decltype(sqlite3_stmt *, int i);
const void *sqlite3_column_decltype16(sqlite3_stmt*,int);

Subject: Re: Date emulation for Sqlite
Posted by mirek on Fri, 15 Sep 2006 12:18:34 GMT
View Forum Message <> Reply to Message

Well, looks like an excellent option, unfortunately it seems like it does not work for

select max(mydate) from table

(or any other expression).

Subject: Re: Date emulation for Sqlite
Posted by fabio on Sun, 26 Nov 2006 17:07:41 GMT
View Forum Message <> Reply to Message

EditDate not work with SqlCtrl and Sqlite, but you can make new ctrl like SqlEditDate which work with date like "yyyy-mm-dd" in sqlite table. The only requirement is SQL.GetDialect().

You can change method for support more formats like yyyymmdd (string like number) and others.

```
 //---------------------------------------------------------
//Add this class definition in  SqlCtrl.h

class SqlEditDate : public EditDate
{
public:
    virtual void  SetData(const Value& data);
};

//-------------------------------------------------
// Add this method in SqlCtrl.cpp

void SqlEditDate::SetData(const Value& data)
{
 Value valdata = data;
 if (SQL.GetDialect()==SQLITE3){
    switch(data.GetType()){
    case STRING_V:
    case WSTRING_V:
        valdata =  Date(atoi(data.ToString().Mid(0,4)),atoi(data.ToString().Mid
(5,2)),atoi(data.ToString().Mid(8,2)));
 break;
    }
  }
  SetText((WString)convert->Format(valdata));
}
//-------------------------------------------------
```

Fabio

File Attachments

1) SqlEditDate.txt, downloaded 743 times

Subject: Re: Date emulation for Sqlite
Posted by zsolt on Sun, 26 Nov 2006 17:24:47 GMT
View Forum Message <> Reply to Message

Quote:EditDate not work with SqlCtrl and Sqlite, but you can make new ctrl like SqlEditDate which work with date like "yyyy-mm-dd" in sqlite table.

This is very good idea.
But I think, the main problem is, that it doesn't resolve the Value problem. The Value class is typed internally, and the problem is, that in the core of SQL classes you can not set Value instances correctly, because SQLite doesn't report date types in metadata of query results.
