Subject: Re: PROPOSAL: Access to S_* Structure of TABLE crash Application.
Posted by Sender Ghost on Sat, 19 May 2012 03:47:47 GMT
View Forum Message <> Reply to Message

sergeynikitin wrote on Sat, 19 May 2012 03:28Sad!! Very sad!
It turns out SQLite driver works with huge errors.

I confess, I have experimented on native sqlite3 tools.

From this it follows that it is the implementation of U++ provides such terrible mistakes?

I think, no. After you wrote this, I tried to use SQLite command-line shell with following results:

```
sqlite3.exe
SQLite version 3.7.12 2012-05-14 01:41:23
Enter ".help" for instructions
Enter SQL statements terminated with a ";"
sqlite> .read S_SQL_Sqlite3.sql

sqlite> .header on
sqlite> .mode column
sqlite> insert into WORKER(ID, NAME, LASTNAME, PLANT_ID) values (0, 'Joe', 'Smith', 0);
sqlite> insert into WORKER(ID, NAME, LASTNAME, PLANT_ID) values (1, 'Mike', 'Smith', 0);
sqlite> insert into WORKER(ID, NAME, LASTNAME, PLANT_ID) values (2, 'Jon', 'Goober', 1);
sqlite> insert into PLANT(ID, NAME, ADDRESS) values (0, 'First Plant', 'First st.');
sqlite> insert into PLANT(ID, NAME, ADDRESS) values (1, 'Second Plant', 'Second st.');
sqlite> select WORKER.NAME, WORKER.LASTNAME, PLANT.NAME, PLANT.ADDRESS from
WORKER left outer join PLANT on WORKER.PLANT_ID = PLANT.ID;
NAME        LASTNAME    NAME        ADDRESS
----------  ----------  ----------  ----------
Joe         Smith       First Plant  First st.
Mike        Smith       First Plant  First st.
Jon         Goober      Second Plan  Second st.
sqlite> select WORKER.NAME, WORKER.LASTNAME, PLANT.NAME PLANT_NAME,
PLANT.ADDRESS from WORKER left outer join PLANT on WORKER.PLANT_ID = PLANT.ID;
NAME        LASTNAME    PLANT_NAME   ADDRESS
----------  ----------  ----------  ----------
Joe         Smith       First Plant  First st.
Mike        Smith       First Plant  First st.
Jon         Goober      Second Plan  Second st.
sqlite> .quit
```

Contents of "S_SQL_Sqlite3.sql" file, which generated U++ executable to create tables:

```
create table WORKER (
 ID integer primary key,
 NAME text,
 LASTNAME text,
```

```
 PLANT_ID integer
);

create table PLANT (
 ID integer primary key,
 NAME text,
 ADDRESS text
);
```

The U++ wrapper returns the same column names in my case.
sergeynikitin wrote on Sat, 19 May 2012 03:28Interesting, how things are with other database engines?
I don't have other database engines to check, for now.
sergeynikitin wrote on Sat, 19 May 2012 03:28Well, what is the salvation of the drowning - a handwork of drowning.....
No need to sinking :)
There are about three methods to solve such issue already:
- Accessing result set through indexes.
- Using "AS" clause for clashing column names.
- Using unique column names across database.

Edit:
I tested the same queries for MySQL database through U++ wrapper and MySQL Workbench.
And it returns the same column names for two mentioned queries:

NAME LASTNAME NAME ADDRESS
NAME LASTNAME PLANT_NAME ADDRESS

The modified version of SQL_MySql reference you could find in the attachment.

File Attachments

1) SQL_MySql.zip, downloaded 359 times