Subject: Fields of the same name in diff tables
Posted by jeremy_c on Mon, 01 Jun 2009 00:43:21 GMT
View Forum Message <> Reply to Message

I am confused about how to do this:

```
TABLE_(HIVE_ACCESSORY)
   INT_ (ID) PRIMARY_KEY AUTO_INCREMENT
   STRING_ (NAME, 128) NOT_NULL
END_TABLE

TABLE_(HIVE)
   INT_ (ID) PRIMARY_KEY AUTO_INCREMENT
   STRING_ (NAME, 128) NOT NULL
   BLOB_ (DESCRIPTION)
END_TABLE
```

This results in compilation errors of duplicate ID and NAME values. I saw elsewhere in the forum that you should use INT_ on first use and INT on subsequent (of the same name). However, when I do that my HIVE table has only the field DESCRIPTION. It does not contain ID or NAME.

Any thoughts on how to accomplish my task? Do I have to create unique field names for all table entries?

Jeremy

Subject: Re: Fields of the same name in diff tables
Posted by jeremy_c on Mon, 01 Jun 2009 03:27:10 GMT
View Forum Message <> Reply to Message

Hm, It seems to be only when I have more attributes, such as:

```
TABLE_(ABC)
   STRING_ (NAME, 80) NOT_NULL UNIQUE
END_TABLE

TABLE_(DEF)
   STRING (NAME, 80) NOT_NULL UNIQUE
END_TABLE
```

DEF will not have the NAME column. If I remove the NOT_NULL UNIQUE from each, then both will have the NAME column.

Jeremy

---

Subject: Re: Fields of the same name in diff tables
Posted by mirek on Mon, 01 Jun 2009 06:41:12 GMT
View Forum Message <> Reply to Message

jeremy_c wrote on Sun, 31 May 2009 23:27Hm, It seems to be only when I have more attributes, such as:

```
TABLE_(ABC)
   STRING_ (NAME, 80) NOT_NULL UNIQUE
END_TABLE

TABLE_(DEF)
   STRING (NAME, 80) NOT_NULL UNIQUE
END_TABLE
```

DEF will not have the NAME column. If I remove the NOT_NULL UNIQUE from each, then both will have the NAME column.

Jeremy

Well, this rather looks like problem in generated script.

What DB is it?

Put SetTrace for your session and then check the .log file (Alt+L in theide).

Mirek

---

Subject: Re: Fields of the same name in diff tables
Posted by jeremy_c on Mon, 01 Jun 2009 14:23:09 GMT
View Forum Message <> Reply to Message

It is Sqlite3. Here is the line causing the problem:

```
   STRING_ (NAME, 80) NOT_NULL UNIQUE
```

Here is the log line that shows what is going on behind the scenes:

ERROR Cannot add a NOT NULL column with default value NULL(0): Preparing: alter table SUPPLIER add
NAME text not null

Basically, I do not want it to be NULL and I want no default. I would like it to be required to be supplied by the application/user.

Jeremy

---

Subject: Re: Fields of the same name in diff tables
Posted by mirek on Mon, 01 Jun 2009 15:34:16 GMT
View Forum Message <> Reply to Message

I guess addind SQLDEFAULT would help here.

Mirek

---

Subject: Re: Fields of the same name in diff tables
Posted by jeremy_c on Mon, 01 Jun 2009 18:34:11 GMT
View Forum Message <> Reply to Message

What would I default it to if I don't want a default? Simply what I want (in SQL) is:

```
CREATE TABLE ABC (
   NAME varchar(80) NOT NULL
);
```

Jeremy

---

Subject: Re: Fields of the same name in diff tables
Posted by mirek on Mon, 01 Jun 2009 21:09:06 GMT
View Forum Message <> Reply to Message

jeremy_c wrote on Mon, 01 June 2009 14:34What would I default it to if I don't want a default? Simply what I want (in SQL) is:

```
CREATE TABLE ABC (
   NAME varchar(80) NOT NULL
);
```

Jeremy

I am not sure, I believe that the real reason is that choosen DB does not allow that.. You can easily test by issuing the command manually.

Mirek

---

Subject: Re: Fields of the same name in diff tables
Posted by jeremy_c on Mon, 01 Jun 2009 21:17:29 GMT
View Forum Message <> Reply to Message

SQLite, PostgreSQL and MySQL all support the syntax given. What it does not support is:

```
CREATE TABLE ABC (
   NAME VARCHAR(80) NOT NULL DEFAULT NULL
);
```

Which seems to be what

```
TABLE_(ABC)
  STRING_  (NAME, 80) NOT_NULL
END_TABLE
```

is trying to do.

Jeremy

---

Subject: Re: Fields of the same name in diff tables
Posted by mirek on Mon, 01 Jun 2009 21:30:35 GMT
View Forum Message <> Reply to Message

jeremy_c wrote on Mon, 01 June 2009 17:17SQLite, PostgreSQL and MySQL all support the syntax given. What it does not support is:

```
CREATE TABLE ABC (
   NAME VARCHAR(80) NOT NULL DEFAULT NULL
);
```

Which seems to be what

```
TABLE_(ABC)
  STRING_  (NAME, 80) NOT_NULL
END_TABLE
```

is trying to do.

Jeremy

OK, I will check Sqlite3 schema dialect.

Mirek

---

Subject: Re: Fields of the same name in diff tables
Posted by mirek on Mon, 01 Jun 2009 21:36:08 GMT
View Forum Message <> Reply to Message

Well, the code generated seems to be:

```
create table ABCTEST (  NAME text not null )
```

and sqlite3 eats it without problems. Something else is going on...

Mirek

---

Subject: Re: Fields of the same name in diff tables
Posted by jeremy_c on Mon, 01 Jun 2009 21:58:21 GMT
View Forum Message <> Reply to Message

Ok, here is the full definition:

```
TABLE_(SUPPLIER)
 INT_    (ID) PRIMARY_KEY AUTO_INCREMENT
 STRING_ (NAME, 80) NOT_NULL UNIQUE
 STRING_ (ADDRESS1, 128)
 STRING_ (ADDRESS2, 128)
```

```
 STRING_ (CITY, 128)
 STRING_ (STATE, 80)
 STRING_ (POSTAL_CODE, 24)
 STRING_ (PRIMARY_PHONE, 35)
 STRING_ (SECONDARY_PHONE, 35)
 STRING_ (EMAIL, 128)
 STRING_ (WEB_ADDRESS, 500)
 BLOB_   (DESCRIPTION)
END_TABLE
```

Here is the resulting SQL:

```
-- From S_BeekPRO.sql
create table SUPPLIER (
 ID integer primary key autoincrement,
 NAME text not null,
 ADDRESS1 text,
 ADDRESS2 text,
 CITY text,
 STATE text,
 POSTAL_CODE text,
 PRIMARY_PHONE text,
 SECONDARY_PHONE text,
 EMAIL text,
 WEB_ADDRESS text,
 DESCRIPTION blob
);

-- From A_BeekPRO.sql
create unique index IDX_SUPPLIER$NAME on SUPPLIER(NAME);
```

Here is the output from sqlite3 session:

```
C:\Develop\Tools\upp\out\MSC9.Debug.Debug_full.Gui>sqlite3 BeekPRO.db3
SQLite version 3.6.7
Enter ".help" for instructions
Enter SQL statements terminated with a ";"
sqlite> .schema SUPPLIER
CREATE TABLE SUPPLIER (  ID integer primary key autoincrement ,
ADDRESS1 text, ADDRESS2 text, CITY text, STATE text, POSTAL_CODE
text, PRIMARY_PHONE text, SECONDARY_PHONE text, EMAIL text,
WEB_ADDRESS text, DESCRIPTION blob);
```

Here is the output from the log:

* C:\develop\tools\upp\out\MSC9.Debug.Debug_full.Gui\BeekPRO.exe 01.06.2009 17:52:23, user: jeremy

ERROR Cannot add a NOT NULL column with default value NULL(0): Preparing: alter table SUPPLIER add  NAME text not null
ERROR table SUPPLIER has no column named NAME(0): Preparing: create unique index IDX_SUPPLIER$NAME on SUPPLIER(NAME)

Jeremy

---

Subject: Re: Fields of the same name in diff tables
Posted by andrei_natanael on Tue, 02 Jun 2009 05:36:55 GMT
View Forum Message <> Reply to Message

Hi, I remember a (short)discussion I had with Mirek some months ago about using sch files only for mapping DB fields in C++ code and let the programmer to create the DB structure in plain SQL because differences between SQL dialect implemented by different SQL "engines" in different ways... Now I think it's oblivious that sch MACROS doesn't cover good enough creation of tables(because difference between implementation).
It's simply to let the programmer to create tables with a custom SQL query, and let sch files be used only for identify(mapping) columns name in C++ code.
I think you may use a custom sql query(compatible with SQLite) to create tables in database.

---

Subject: Re: Fields of the same name in diff tables
Posted by mr_ped on Tue, 02 Jun 2009 08:33:16 GMT
View Forum Message <> Reply to Message

While that sounds as good workaround, having working schema in C++ is very interesting too, because you will keep your versioning of DB definitions together with code versioning (if you use SVN/etc.).
If you do all your modifications directly to DB in some admin tool, you have no history documentation. (You can fight with it by doing export of DB schema and commit that file to SVN, but it feels like additional hassle to me, I prefer simple things, so having full DB generated from the info in C++ is IMHO nice feature).
(right now I have zero C++ DB projects, all my projects using some SQL are in php, but the wish for having the full schema generated trough source does apply there too)

---

Subject: Re: Fields of the same name in diff tables
Posted by jeremy_c on Tue, 02 Jun 2009 12:27:01 GMT
View Forum Message <> Reply to Message

I like the idea of Ultimate++ generating table structures for me, I just think it needs a little more work and a little more documentation. Now, I am a database programmer by trade, so when it comes to an enterprise application, I doubt that UPP can ever keep up with what can be accomplished using the database, for instance, all sorts of trigger functions and other server side programming. For desktop applications w/simple databases, however, it's great.

Now, about managing SQL independent. What I do is I always have a directory structure such as:

project/docs
project/src
project/sql
project/sql/0001.sql
project/sql/0001_r.sql
project/sql/0002.sql
project/sql/0002_r.sql

0001.sql is the first version of my database schema. 0002.sql is the create/alter statements necessary to bring 0001.sql (current at the time) to my new version. So, if I were starting a new project, I could psql thedb < 0001.sql ... psql thedb < 0002.sql, etc... Now, 0001_r.sql and 0002_r.sql are "Revert" scripts which will undo the changes. I don't always make those but it's a good idea.

Jeremy

---

Subject: Re: Fields of the same name in diff tables
Posted by mirek on Tue, 02 Jun 2009 12:47:19 GMT
View Forum Message <> Reply to Message

Well, this works:

Quote:
```
TABLE_(SUPPLIER)
 INT_    (ID) PRIMARY_KEY AUTO_INCREMENT
 STRING_ (NAME, 80) NOT_NULL SQLDEFAULT("")
 STRING_ (ADDRESS1, 128)
 STRING_ (ADDRESS2, 128)
 STRING_ (CITY, 128)
 STRING_ (STATE, 80)
 STRING_ (POSTAL_CODE, 24)
 STRING_ (PRIMARY_PHONE, 35)
```

```
 STRING_ (SECONDARY_PHONE, 35)
 STRING_ (EMAIL, 128)
 STRING_ (WEB_ADDRESS, 500)
 BLOB_   (DESCRIPTION)
END_TABLE
```

Hard to say what it dislikes.. I guess this is rather question to the sqlite3 forum...

Mirek

---

Subject: Re: Fields of the same name in diff tables
Posted by mirek on Tue, 02 Jun 2009 12:50:20 GMT
View Forum Message <> Reply to Message

jeremy_c wrote on Tue, 02 June 2009 08:27
Now, I am a database programmer by trade, so when it comes to an enterprise application, I doubt that UPP can ever keep up with what can be accomplished using the database, for instance, all sorts of trigger functions and other server side programming.

Obviously, triggers/views etc.. are beyond what you can achieve in .sch.

I usually put them into regular .sql files into the project, then use "DataFile" to load them and perform them when code starts in debug mode (SqlPerformScript).

Recent ide even has .sql highlighting

Mirek

---

Subject: Re: Fields of the same name in diff tables
Posted by mirek on Tue, 02 Jun 2009 12:52:11 GMT
View Forum Message <> Reply to Message

andrei_natanael wrote on Tue, 02 June 2009 01:36Hi, I remember a (short)discussion I had with Mirek some months ago about using sch files only for mapping DB fields in C++ code and let the programmer to create the DB structure in plain SQL because differences between SQL dialect implemented by different SQL "engines" in different ways... Now I think it's oblivious that sch MACROS doesn't cover good enough creation of tables(because difference between implementation).

It really is not that bad and it works fine. You just have to use it for what it is intended...

And, in fact, it is able to hide many DB engine differences for you. (YOu just cannot expect all of

them to be hidden I guess

Mirek

Subject: Re: Fields of the same name in diff tables
Posted by lectus on Thu, 20 Dec 2012 18:20:51 GMT
View Forum Message <> Reply to Message

Hi!

Sorry for ressurrecting this old thread.

But I noticed sch files really have trouble creating tables in Sqlite.

I had to add a create table command manually before everything. Before that I was getting an error.

Is that the expected behavior or am I missing something?

Subject: Re: Fields of the same name in diff tables
Posted by mirek on Sun, 23 Dec 2012 09:17:09 GMT
View Forum Message <> Reply to Message

lectus wrote on Thu, 20 December 2012 13:20Hi!

Sorry for ressurrecting this old thread.

But I noticed sch files really have trouble creating tables in Sqlite.

I had to add a create table command manually before everything. Before that I was getting an error.

Is that the expected behavior or am I missing something?

Can you post a simple testcase, please?

Mirek