
Subject: How can .sch express a many-to-many relationship ?

Posted by [jfranks](#) on Thu, 19 Jan 2017 22:48:07 GMT

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

Fellow developers,

I've been tasked with implementing a schema using SQLite + Upp.

Sql is not my strong suit, so I used a tool that converted an ERD --> DDL.

As I discovered, UPP has it's own .sch language representation that has benefits and DDL is derived from.

However, I could not find examples of a many-to-many relationship being expressed in .sch and am somewhat over-loaded with all the tasks ahead of me.

I figured that I could just convert by hand the DDL and produce a representative .sch -- and this worked except I could not figure out the many-to-many expressions needed in .sch

The following is a minimal DDL example that I cannot figure out how to do in .sch.

Can someone give me some pointers on this?

Thank you.

-- Jeff Franks

----- Example DDL for a many-to-many relationship -----

--

-- multiple user accounts w/ the possibility of multiple roles

--

```
CREATE TABLE USER_ACCOUNT (  
  ID      INTEGER PRIMARY KEY,  
  NAME    varchar(255));
```

```
CREATE TABLE ROLE (  
  ID      INTEGER PRIMARY KEY,  
  ASPECT  integer(2),  
  DESCRIPTION varchar(255));
```

```
CREATE TABLE USER_ROLE (  
  USER_ACCOUNT_ID integer(10),  
  ROLE_ID          integer(10),  
  PRIMARY KEY(USER_ACCOUNT_ID, ROLE_ID),  
  FOREIGN KEY(USER_ACCOUNT_ID) REFERENCES USER_ACCOUNT(ID),  
  FOREIGN KEY(ROLE_ID) REFERENCES ROLE(ID));
```

Subject: Re: How can .sch express a many-to-many relationship ?

Posted by [mirek](#) on Tue, 07 Mar 2017 12:03:55 GMT

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

There might be errors, doing it from the head:

```
TABLE_(USER_ACCOUNT)
  ISERIAL (ID) PRIMARY_KEY
  STRING_ (NAME, 255)
END_TABLE
```

```
TABLE_(ROLE)
  ISERIAL (ID) PRIMARY_KEY
  INT_ (ASPECT)
  STRING_ (DESCRIPTION, 255)
END_TABLE
```

```
TABLE_(USER_ROLE)
  INT_ (USER_ACCOUNT_ID) REFERENCES(USER_ACCOUNT)
  INT_ (ROLE_ID) REFERENCES(ROLE)
  DUAL_PRIMARY_KEY (USER_ACCOUNT_ID, ROLE_ID)
END_TABLE
```

(I suspect it is too late for you to help, but putting it here for future reference...)
