Subject: Any way to generate sch from database?
Posted by alendar on Mon, 05 Apr 2010 07:16:19 GMT
View Forum Message <> Reply to Message

Hi,

I searched the forums for any examples of generating a simple SCH file from a PostgreSQL connection, but no luck. Is there such a beast? I hesitate to write a script if there's one out there already (Lazy Programmer Syndrome)

Thanks,
Jeff

Subject: Re: Any way to generate sch from database?
Posted by alendar on Mon, 05 Apr 2010 08:28:08 GMT
View Forum Message <> Reply to Message

I hacked a script together, but it poops out if there's a column the same name as a table.

```
drop sequence if exists foo;
create sequence foo;
drop table if exists x ;
create temporary table x as
select table_name, column_name, udata_type, data_type, character_maximum_length,
nextval('foo') as i
from (select upper(table_name) as table_name, upper(column_name) as column_name
,case data_type
    when 'integer' then 'INT'
    when 'character' then 'STRING'
    when 'character varying' then 'STRING'
    when 'double precision' then 'DOUBLE'
    when 'text' then 'STRING'
    when 'timestamp without time zone' then 'TIMESTAMP'
    when 'timestamp with time zone' then 'TIMESTAMP'
    when 'array' then 'TIMESTAMP' -- Bug
    else data_type
    end as Udata_type
    , data_type as data_type
    , character_maximum_length
from information_schema.columns where table_schema = 'mydb'
and data_type <> 'tsvector'
order by 1,2) as rr;

select line from (
select distinct table_name, '1' as ord1, 'TABLE_(' || table_name || ')' as line from x
union all
select table_name, '2' as ord1, '   ' || Udata_type || case when i = (select min(xb.i) from x xb where
```

```
xb.column_name = x.column_name) then '_' else '' end
|| '   (' || column_name ||
  case when udata_type = 'STRING' then ', ' || coalesce(cast(
 case when data_type = 'text' then 1024 else character_maximum_length end as character
varying), '255') else '' end || ')'
from x
union all
select distinct table_name, '3' as ord1, 'END_TABLE ' as line from x
order by 1,2,3
) as rr
```

Spits out:

```
TABLE_(ARTISTS_BLOCKED)
  STRING_   (ADDINGUSERNAME, 255)
  STRING_   (ARTISTNAME, 255)
  STRING_   (COMMENTS, 255)
  STRING_   (REASON, 255)
  TIMESTAMP_   (WHENADDED)
END_TABLE
TABLE_(EXTRACTFOLDERS)
  STRING_   (DRIVENAME, 1024)
  STRING_   (FOLDERNAME, 255)
  STRING_   (MACHINENAME, 255)
END_TABLE
TABLE_(FOLDERS_001)
  STRING   (COMMENTS, 1024)
  STRING   (DRIVENAME, 255)
  STRING   (FOLDERNAME, 255)
  STRING   (MACHINENAME, 255)
  STRING_   (ISFROZEN, 1)
  TIMESTAMP_   (WHENLASTCHANGED)
END_TABLE
TABLE_(GENRES_001)
  STRING_   (GENRENAME, 255)
  STRING_   (GENRES_TBL, 255)
END_TABLE
TABLE_(GENRES_BLOCKED)
  STRING   (ADDINGUSERNAME, 255)
  STRING   (COMMENTS, 255)
  STRING   (GENRENAME, 255)
  STRING   (REASON, 255)
  TIMESTAMP   (WHENADDED)
END_TABLE
TABLE_(LIST_SONGS_001)
  DOUBLE_   (PROCESSORDER)
  INT_   (LISTID)
```

```
  INT_   (SONGID)
  STRING   (ARTISTNAME, 255)
  STRING   (COMMENTS, 255)
  STRING   (DRIVENAME, 255)
  STRING   (FOLDERNAME, 255)
  STRING   (MACHINENAME, 255)
  STRING_   (ALBUMNAME, 255)
  STRING_   (FILENAME, 255)
  STRING_   (FILEPATH, 255)
  STRING_   (ISUNCPATH, 1)
  STRING_   (ISURL, 1)
  STRING_   (LISTNAME, 255)
  STRING_   (LISTS_TBL, 255)
  STRING_   (LISTSONGS_TBL, 255)
  STRING_   (LOOKTODB_TBL, 255)
  STRING_   (LOOKTOLISTS_TBL, 255)
  STRING_   (LOOKTOSONGS_TBL, 255)
  STRING_   (MACHINES_TBL, 255)
  STRING_   (NETWORKNAME, 255)
  STRING_   (NETWORKS_TBL, 255)
  STRING_   (SONGS_TBL, 255)
  STRING_   (TITLE, 255)
  STRING_   (USERNAME, 255)
  STRING_   (USERS_TBL, 255)
  TIMESTAMP   (WHENADDED)
END_TABLE
TABLE_(LISTS_001)
  INT   (LISTID)
  STRING   (COMMENTS, 255)
  STRING   (LISTNAME, 255)
  STRING   (LISTS_TBL, 255)
  STRING   (MACHINENAME, 255)
  STRING   (MACHINES_TBL, 255)
  STRING   (NETWORKNAME, 255)
  STRING   (NETWORKS_TBL, 255)
  STRING   (USERNAME, 255)
  STRING   (USERS_TBL, 255)
  STRING_   (ISJAMMIT, 1)
  TIMESTAMP_   (WHENCREATED)
END_TABLE
TABLE_(MACHINES_001)
  STRING   (COMMENTS, 255)
  STRING   (MACHINENAME, 255)
  STRING   (MACHINES_TBL, 255)
  STRING   (NETWORKNAME, 255)
  STRING   (NETWORKS_TBL, 255)
  STRING_   (PRIMARYMACHINEUSERNAME, 255)
END_TABLE
```

```
TABLE_(NETWORKS_001)
  STRING   (COMMENTS, 255)
  STRING   (NETWORKNAME, 255)
  STRING   (NETWORKS_TBL, 255)
END_TABLE
TABLE_(REASONS_001)
  INT_   (REASONID)
  STRING   (REASON, 255)
  STRING_   (FORACTION, 255)
  STRING_   (REASONS_TBL, 255)
END_TABLE
TABLE_(SCRIPTS_001)
  STRING   (COMMENTS, 1024)
  STRING_   (ISREADYFORREGULARUSE, 1)
  STRING_   (SCRIPT, 1024)
  STRING_   (SCRIPTNAME, 80)
  STRING_   (SCRIPTTYPE, 20)
  TIMESTAMP   (WHENADDED)
END_TABLE
TABLE_(SEQUENCES_001)
  INT_   (NEXTID)
  STRING_   (TABLENAME, 255)
END_TABLE
TABLE_(SONGS_001)
  INT   (SONGID)
  INT_   (BITRATE)
  INT_   (CHANNELS)
  INT_   (FILESIZE)
  INT_   (FINISHCT)
  INT_   (LENGTHMS)
  INT_   (ORIGINALFILESIZE)
  INT_   (ORIGINALLENGTHMS)
  INT_   (PLAYMS)
  INT_   (RATING)
  INT_   (REPEATCT)
  INT_   (ROLLINCT)
  INT_   (ROLLINDEPTHSUM)
  INT_   (SAMPLERATE)
  INT_   (SELECTCT)
  INT_   (SKIPCT)
  INT_   (SKIPINMS)
  INT_   (SKIPMS)
  INT_   (STARTCT)
  INT_   (TRACKNO)
  INT_   (TRIMOFFENDMS)
  STRING   (ALBUMNAME, 255)
  STRING   (ARTISTNAME, 255)
  STRING   (COMMENTS, 255)
```

```
  STRING   (DRIVENAME, 255)
  STRING   (FILENAME, 255)
  STRING   (FILEPATH, 255)
  STRING   (FOLDERNAME, 255)
  STRING   (GENRENAME, 255)
  STRING   (ISUNCPATH, 1)
  STRING   (ISURL, 1)
  STRING   (MACHINENAME, 255)
  STRING   (MACHINES_TBL, 255)
  STRING   (NETWORKNAME, 255)
  STRING   (NETWORKS_TBL, 255)
  STRING   (SONGS_TBL, 255)
  STRING   (TITLE, 255)
  STRING_   (COMPOSERNAME, 255)
  STRING_   (FILEID, 128)
  STRING_   (ISBLOCKED, 1)
  STRING_   (ISKEEPER, 1)
  STRING_   (LANGUAGE, 45)
  STRING_   (LISTSSONGISON_COLL, 255)
  STRING_   (LYRICS, 1024)
  STRING_   (ORIGINALFILENAME, 255)
  STRING_   (ORIGINALTITLE, 255)
  STRING_   (PUBLISHERNAME, 255)
  STRING_   (RELEASEYEAR, 255)
  TIMESTAMP   (WHENADDED)
  TIMESTAMP_   (WHENFILECREATED)
  TIMESTAMP_   (WHENFILEMODIFIED)
END_TABLE
TABLE_(SONGS_REMOVED)
  INT   (BITRATE)
  INT   (CHANNELS)
  INT   (FILESIZE)
  INT   (LENGTHMS)
  INT   (ORIGINALFILESIZE)
  INT   (ORIGINALLENGTHMS)
  INT   (RATING)
  INT   (SAMPLERATE)
  INT   (SONGID)
  STRING   (ALBUMNAME, 255)
  STRING   (ARTISTNAME, 255)
  STRING   (COMMENTS, 1024)
  STRING   (DRIVENAME, 255)
  STRING   (FILEID, 128)
  STRING   (FILEPATH, 255)
  STRING   (FOLDERNAME, 255)
  STRING   (ISBLOCKED, 1)
  STRING   (ISUNCPATH, 1)
  STRING   (ISURL, 1)
```

```
  STRING   (LANGUAGE, 45)
  STRING   (LYRICS, 1024)
  STRING   (MACHINENAME, 255)
  STRING   (MACHINES_TBL, 255)
  STRING   (NETWORKNAME, 255)
  STRING   (NETWORKS_TBL, 255)
  STRING   (ORIGINALFILENAME, 255)
  STRING   (RELEASEYEAR, 255)
  STRING   (SONGS_TBL, 255)
  STRING   (TITLE, 255)
  STRING_   (ISPERMITTEDBACK, 1)
  STRING_   (REMOVALREASONID, 255)
  STRING_   (REMOVINGUSERNAME, 255)
  TIMESTAMP   (WHENADDED)
  TIMESTAMP_   (WHENREMOVEDFROMDB)
  TIMESTAMP_   (WHENREMOVEDFROMMACHINE)
END_TABLE
TABLE_(SONGS_WORKQUEUE_001)
  INT   (SONGID)
  INT_   (WORKTYPEID)
  STRING   (LOOKTOSONGS_TBL, 255)
  STRING   (SONGS_TBL, 255)
  STRING_   (ISBEINGWORKEDON, 1)
  STRING_   (ISCOMPLETE, 1)
  STRING_   (ISERROR, 1)
  STRING_   (OUTCOME, 255)
  TIMESTAMP_   (WHENCOMPLETED)
END_TABLE
TABLE_(USERS_001)
  INT_   (DEFAULTLISTID)
  INT_   (USERID)
  STRING   (COMMENTS, 1024)
  STRING   (USERNAME, 255)
  STRING   (USERS_TBL, 255)
  STRING_   (DEFAULTLISTS_TBL, 255)
  STRING_   (DEFAULTMACHINENAME, 255)
  STRING_   (DEFAULTMACHINES_TBL, 255)
  STRING_   (DEFAULTSONGS_TBL, 255)
  STRING_   (KEEPERSONGS_TBL, 255)
  STRING_   (RAWSONGS_TBL, 255)
  STRING_   (REMOVESONGS_TBL, 255)
  STRING_   (USERALIASES_COLL, 255)
END_TABLE
TABLE_(WORKTYPE)
  INT   (WORKTYPEID)
  STRING_   (WORKTYPENAME, 255)
END_TABLE
```

Probably a U++ code method would be better.

Subject: Re: Any way to generate sch from database?
Posted by mirek on Tue, 06 Apr 2010 13:00:43 GMT
View Forum Message <> Reply to Message

Sorry for late reply. These functions is what are you looking for:

```
String ExportSch(SqlSession& session, const String& database);
String ExportIds(SqlSession& session, const String& database);

#ifndef NOAPPSQL
String ExportSch(const String& database);
String ExportIds(const String& database);
#endif
```

Here is example exporting schema from MySql:

```
#include <MySql/MySql.h>

using namespace Upp;

CONSOLE_APP_MAIN
{
 MySqlSession session;
 // substitute your 'username' and 'password' here:
 if(!session.Connect("root", "heslo", "test", "10.0.0.19")) {
  printf("Can't connect with MySql\n");
  return;
 }
 Sql sql(session);
 sql.Execute("use test");
 sql.Execute("show tables");
 while(sql.Fetch())
  Cout() << (String)sql[0] << '\n';
 SaveFile("u:/lego.sch", ExportSch(session, "test"));
 SaveFile("u:/legoid", ExportIds(session, "test"));
}
```

Note that no column attributes (contraints, keys, indicies) are exported.

Mirek
