Sqlsync - library synchronizing databases

Paweł Płuciennik
SILVERCODERS

Sqlsync - library synchronizing databases

by Paweł Płuciennik

Table of Contents

Chapter 1. Introduction

"Sqlsync" library which synchronizes databases has been written in C language. After passing two ODBC aliases it allowes comparing data stored in two SQL databases, or synchronizing them.

For comparing operation input parameters are list of tables names (by default all tables), and for every table optional list of columns (by default all columns). Both databases should contain in given range the same structure. Returned results contain list of tables which took part in comparizon including summary info about number of identical, missing (exist in the first database but not in the second one) or additional (exist in the second database but not in the first one) records. Results for particular table contains also detailed list (with columns names and records data) of records which are different, missing or additional.

Second functionality provided by library is generating SQL queries which will change second database to be identical with first one. Similar to comparizon operation, generating synchronizing queries can be done on level of whole database, specified list of tables or one table. Parameters for this operation are results returned by comparing function and list of tables which user wants to synchronize.

Third functionality provided by library is analogical to the second one but with a difference that generated queries are executed by library, which synchronizes particular tables or whole database.

Included console program allows to use presented library to synchronize databases in a easy way. Library supports SQL servers: PostgreSQL 8.1.x, MySQL 5.0.x, Firebird 1.5.x, Microsoft SQL Server 2000, Microsoft SQL Server 2005, Oracle 10g. Two ODBC aliases doesn't have to point the same type of SQL server.

Chapter 2. Building process

To simplify building library and console application Makefiles has been used. To create library and application it's required to write this command in root source directory

```
make all
```

To identify available options it's needed to write command:

```
make help
```

Available options:
WITH_STATIC - Produce static library and application
WITH_SHARED - Produce shared library and application
WIN32 - Switch needed for cross-platform compilation
WINDOWS - Switch required to compile on Windows platform
WITH_DEBUG - Library will contain debug info and more messages
WITH_DOCUMENTATION - Documentation will be generated
WITH_TRANSLATION - Translation files will be generated
WITH_TESTS - Automated tests will be executed

In order to produce package with library it's required to write:

```
make package
```

In case of building package options WITH_DOCUMENTATION and WITH_TESTS it enables selected elements in package.

```
make package WITH_DOCUMENTATION=YES WITH_TESTS=YES
```

Default settings:

WITH_SHARED=YES

WITH_STATIC=NO
WITH_DOCUMENTATION=NO
WITH_DEBUG=NO
WITH_TESTS=NO

Chapter 3. Library

3.1. Usage

Below there is an example how to use library.

In a first place translation's function has been called - sqlsync_translation_init.
In second step library has been initialized - sqlsync_init.

Then function registered callback function in case of receiving messages from library.
Example function has been presented below (sqlsync_lib_user_handler) -
sqlsync_user_notification_register (sqlsync_lib_user_handler, handler_data);
Messages sqlsync_user_message informs in what state library is.

Next step is connecting to servers.
It's required to pass server ODBC aliast and user name and password.

ret = sqlsync_connect (info, SERVER_SRC, dsn_src, user_src, pass_src);
ret = sqlsync_connect (info, SERVER_DST, dsn_dst, user_dst, pass_dst);

It's possible to ignore case sensitive for tables names and columns.
sqlsync_ignore_case_sensitive(info, SQLSYNC_TRUE);

Main library function is called sqlsync_compare_databases.
It returns number of compared databases.

nr_tables = sqlsync_compare_databases(info, tables, columns,
show_summary, show_queries,
run_queries);

Function takes list of tables and corresponding to them columns.
One value in tables variable must have one vector of columns in columns variable.

Options show_summary, show_queries, run_queries are responsible for showing summary, showing queries and their execu

Last thing is to destroy library instance:
ret = sqlsync_destroy (info);

```
SQLSYNC_RETURN
sqlsync_lib_user_handler (sqlsync_user_message message, sqlsync_user_data * data)
{
 lib_handler_data *handler_data = NULL;

 if (data != NULL)
```

```
 handler_data = (lib_handler_data *) data->user_ptr;

 switch (message)
 {
  case SQLSYNC_USER_ERROR_MESSAGE:
  case SQLSYNC_USER_INFO_MESSAGE:
  {
  //Do whatever is needed to inform user.
  }

  case SQLSYNC_USER_RUN_DIFF_QUERIES:
  case SQLSYNC_USER_RUN_MISS_QUERIES:
  case SQLSYNC_USER_RUN_ADD_QUERIES:
|  {
   //running queries info
   break;
  }
 }
 }
}

//Translation initialization
sqlsync_translation_init ();

//Library initialization
  info = sqlsync_init (&ret);

//Register notification function
sqlsync_user_notification_register (sqlsync_lib_user_handler, handler_data);
SQLSYNC_RETURN sqlsync_user_notification (sqlsync_user_message message,
                                         sqlsync_user_data * data);

//Connecting to databases
ret = sqlsync_connect (info, SERVER_SRC, dsn_src, user_src, pass_src);
ret = sqlsync_connect (info, SERVER_DST, dsn_dst, user_dst, pass_dst);

//Ignoring case sensitive for tables and columns name
sqlsync_ignore_case_sensitive(info, SQLSYNC_TRUE);

//Comparing databases
nr_tables = sqlsync_compare_databases(info, tables, columns,
  show_summary, show_queries,
                                run_queries);

//Closing library
ret = sqlsync_destroy (info);
```

Chapter 4. Console application

4.1. Application description

Purpose of this console program is to use "sqlsync" library for synchronizing database and allowing further processing data i.e. saving synchronizing queries.

4.2. Program description

- --src-dsn Source ODBC alias
- --src-user User name for source ODBC alias
- --src-password Password for source ODBC alias. If password is empty skip this option
- --dst-dsn Destination ODBC alias
- --dst-user User name for destination ODBC alias
- --dst-password Password for destination ODBC alias. If password is empty skip this option
- --silent Program will not show any messages except informations about errors
- --ignore-case-sensitive Ignore table and columns case sensitive
- --tables This option is to pass list of tables and columns to synchronization. In case lack of this option program will compare all available tables.

  --tables table1[column1,column2],table2,table3[column]
  For synchronization table table1 with columns column1 and column2, table2 with all columns and table3 with column will be taken.

- --query-file Runs query from file also --src-dsn parameter is obligatory. Example: --src-dsn my_dsn --query-file /my/query_file.sql
- --show-summary Shows summary with informations about tables (missing, additional, different). Prints also differences between two compared tables
- --show-queries Application displays queries required to synchronize specified tables from databases
- --run-queries Application generates queries required to synchronize specified tables and then executes them

If password for either for source or destination DSN is empty skip respectively --src-password or --dst-password. When neither --show-queries or --run-queries option is used program will use --show-summary by default, in other cases program use selected options

4.3. Usage examples

Comparing whole databases

```
sqlsync --src-dsn sqlsync_mysql --dst-dsn sqlsync_mysql2
  --src-user sqlsync --src-password sqlsync_password
  --dst-user sqlsync_password
```

Comparing selected tables

```
sqlsync --src-dsn sqlsync_mysql --dst-dsn sqlsync_mysql2
         --tables table_char[col1,col2],table_integer,table_var[col]
```

Displaying synchronizing queries for destination database:

```
sqlsync --src-dsn sqlsync_mysql --dst-dsn sqlsync_mysql2 --show-queries
```

Generating and executing synchronizing queries for destination database:

```
sqlsync --src-dsn sqlsync_mysql --dst-dsn sqlsync_mysql2
  --show-queries --run-queries
```

Chapter 5. Automated library tests

5.1. How to run automated tests ?

To run automated tests you can run run_tests.pl which will run all tests from testing directory or run_test.pl with specified test directory i.e. run_test.pl testing/generic

5.2. How to check if test went well?

Script will show information that "Expected results and actual are different see $dir/expected and $dir/results files for more information"

5.3. How to check what output script produced and is expected result ?

Go to directory with test, i.e. testing/generic and check "results" file with "expected"

5.4. What is needed to run test?

Files: script.dat, expected, sql queries.

5.5. How to create test?

Create script.dat and expected file.
Things needed in script

-- src database user
db_src_user sqlsync
-- src database user's password
db_src_password sqlsync_password

-- dst database user
db_dst_user sqlsync
-- dst database user's password
db_dst_password sqlsync_password

```
================================================
================= Test strings ==================
================================================

-- queries from test_mssql_create_strings.sql will be executed on ODBC sqlsync_mssql
cmd_query_run_src sqlsync_mssql  ../common/test_mssql_create_strings.sql
-- queries from test_mssql_create_strings.sql will be executed on ODBC sqlsync_mssql2
cmd_query_run_dst sqlsync_mssql2 ../common/test_mssql_create_strings.sql

-- checking if empty tables are read correctly
cmd_sync_run sqlsync_mssql sqlsync_mssql2 test_strings

-- adding tables into database - running queries from sql files.
cmd_query_run_src sqlsync_mssql  ../common/mssql/test_strings_insert_src.sql
cmd_query_run_dst sqlsync_mssql2 ../common/mssql/test_strings_insert_dst.sql

-- printing tables, sqlsync is executed and tables are compared
cmd_sync_show sqlsync_mssql sqlsync_mssql2 test_strings

-- some comment in results file
cmd_comment
cmd_comment There should be no lines between [1] and [2]
cmd_comment line [1]


-- running queries
cmd_sync_run sqlsync_mssql sqlsync_mssql2 test_strings
-- checking tables if are equal
cmd_sync_show sqlsync_mssql sqlsync_mssql2 test_strings

cmd_comment line [2]
cmd_comment


-- to check some specific table i.e. test2 and two columns, user should write test2[IDKlienta,Imie]
cmd_sync_show sqlsync_postgresql sqlsync_postgresql2 test2[IDKlienta,Imie]

-- running queries with specified tables and columns
cmd_sync_run sqlsync_postgresql sqlsync_postgresql2 test2[IDKlienta,Imie],test2[IDKlienta]


All queries and comments will be in "results" file.

Sqlsync errors will be shown on console.
```