NAME

DBSQLToTextFiles.pl - Export data from MySQL, Oracle or PostgreSQL database into CSV/TSV text files

SYNOPSIS

DBSQLToTextFiles.pl SQLFileName(s) | SQLSelectStatement(s)...

DBSQLToTextFiles.pl [-d, --dbdriver mysql | Oracle | Postgres or Pg] [--dbhost hostname] [--dbname databasename] [--dbpassword password] [--dbusername username] [--exportdatalabels yes | no] [--exportlobs yes | no] [-h, --help] [-m, --mode SQLStatement | SQLFile] [-o, --overwrite] [--outdelim comma | tab | semicolon] [-q, --quote yes | no] [-r, --root rootname] [--replacenullstr string] [-w --workingdir dirname] SQLFileName(s) | SQLSelectStatement(s)...

DESCRIPTION

Export data from MySQL, Oracle or PostgreSQL database into CSV/TSV text files. Based on -m --mode option value, two methods of data selection are availble: in line SQL select statement(s), or SQL file name(s) containing SQL select statement(s). All command line parameters must correspond to similar mode; mixing of parameters for different modes is not supported.

OPTIONS

-d, --dbdriver *mysql* | *Oracle* | *Postgres or Pg*

Database driver name. Possible values: *mysql, Oracle, Postgres or Pg*. Default: *MySQL* or value of environment variable DBI_DRIVER. This script has only been tested with MySQL, Oracle and PostgreSQL drivers.

--dbhost *hostname*

Database host name. Default: *127.0.0.1* for both MySQL, Oracle and PostgreSQL. For remote databases, specify complete remote host domain: *dbhostname.org* or something like it.

--dbname *databasename*

Database name. Default: mysql for MySQL, postgres for PostgreSQL and none for Oracle. For connecting to local/remote Oracle databases, this value can be left undefined assuming --dbhost is correctly specified.

--dbpassword *password*

Database user password. Default: *none* and value of environment variable DBI_PASS is used for connecting to database.

--dbusername *username*

Database user name. Default: *none* and value of environment variable DBI_USER is used for connecting to database.

--exportdatalabels *yes* | *no*

This option is mode specific and controls exporting of column data labels during exportdata mode. Possible values: *yes or no*. Default: *yes*.

--exportlobs *yes* | *no*

This option is mode specific and controls exporting of CLOB/BLOB data columns during exportdata mode. Possible values: *yes or no*. Default: *no*.

-h, --help

Print this help message.

-m, --mode *SQLStatement* | *SQLFile*

Data selection criterion from database. Two different command line parameter methods are available: in line SQL statement(s) specification or file name(s) containing SQL select statement(s). This value determines how command line parameters are processed.

Possible values: *SQLStatement or SQLFile*. Default value: *SQLStatement*

In SQLFile mode, SQL file contains select statements delimited by *;*. And the lines starting with # or - are ignored.

-o, --overwrite

Overwrite existing files.

--outdelim *comma* | *tab* | *semicolon*

Output text file delimiter. Possible values: *comma, tab, or semicolon* Default value: *comma*.

-q, --quote *yes* | *no*

Put quotes around column values in output text file. Possible values: *yes or no*. Default value: *yes*.

-r, --root *rootname*

New file name is generated using the root:<Root><No>.<Ext>. Default new file file names: SQLStatement<No>.<Ext>, or <SQLFileName><StatementNo>.<Ext>. The csv and tsv <Ext> values are used for comma/semicolon, and tab delimited text files respectively.This option is ignored for multiple input parameters.

--replacenullstr *string*

Replace NULL or undefined row values with specified value. Default: *none*

For importing output text files into MySQL database using "load data local infile '<tablename>.tsv' into table <tablename>" command, use *--raplacenullstr "NULL"* in conjunction with *--exportdatalabels no*, *--quote no*, and *--outdelim tab* options: it'll generate files for direct import into MySQL assuming tables already exists.

-w --workingdir *dirname*

Location of working directory. Default: current directory.

EXAMPLES

To export all data in user_info table from a MySQL server running on a local machine using username/password from DBI_USER and DBI_PASS environmental variables, type:

```
% DBSQLToTextFiles.pl -o "select * from user_info"
```

To describe user table in a MySQL server running on a remote machine using explicit username/password and capturing the output into a UserTable.csv file, type:

```
% DBSQLToTextFiles.pl --dbdriver mysql --dbuser <name> --dbpassword
  <pasword> --dbname mysql --dbhost <mysqlhostname.org> -r UserTable
  -m SQLStatement -o "select * from user_info"
```

To describe table all_tables in Oracle running on a remote machine using explicit username/password and capturing the output into a AllTable.tsv file, type:

```
% DBSQLToTextFiles.pl --dbdriver Oracle --dbuser <name> --dbpassword
  <pasword> --dbhost <oraclehostname.com> -r AllTable -m SQLStatement
  --outdelim tab --quote no -o "select * from all_tables"
```

To run all SQL statement in a file sample.sql on a local Oracle host and capturing output in a SampleSQL.csv file, type:

```
% DBSQLToTextFiles.pl --dbdriver Oracle --dbuser <name> --dbpassword
  <pasword> -r SampleSQL -m SQLFile -o sample.sql
```

AUTHOR

Manish Sud <msud@san.rr.com>

SEE ALSO

DBSchemaTablesToTextFiles.pl, DBTablesToTextFiles.pl

COPYRIGHT