MYSQL DATABASE EXPORT - BACKUP METHODS

The simplest way of exporting a table data into a text file is using SELECT...INTO OUTFILE statement that exports a query result directly into a file on the server host.

Exporting Data with the SELECT ... INTO OUTFILE Statement:

The syntax for this statement combines a regular SELECT with INTO OUTFILE *filename* at the end. The default output format is the same as for LOAD DATA, so the following statement exports the tutorials_tbl table into /tmp/tutorials.txt as a tab-delimited, linefeed-terminated file:

```
mysql> SELECT * FROM tutorials_tbl
    -> INTO OUTFILE '/tmp/tutorials.txt';
```

You can change the output format using options to indicate how to quote and delimit columns and records. To export the tutorial_tbl table in CSV format with CRLF-terminated lines, use this statement:

```
mysql> SELECT * FROM passwd INTO OUTFILE '/tmp/tutorials.txt'
    -> FIELDS TERMINATED BY ',' ENCLOSED BY '"'
    -> LINES TERMINATED BY '\r\n';
```

The **SELECT ... INTO OUTFILE** has the following properties:

- The output file is created directly by the MySQL server, so the filename should indicate where you want the file to be written on the server host. There is no LOCAL version of the statement analogous to the LOCAL version of LOAD DATA.
- You must have the MySQL FILE privilege to execute the SELECT ... INTO statement.
- The output file must not already exist. This prevents MySQL from clobbering files that may be important.
- You should have a login account on the server host or some way to retrieve the file from that host. Otherwise, SELECT ... INTO OUTFILE likely will be of no value to you.
- Under UNIX, the file is created world readable and is owned by the MySQL server. This means that although you'll be able to read the file, you may not be able to delete it.

Exporting Tables as Raw Data:

The ***mysqldump*** program is used to copy or back up tables and databases. It can write table output either as a raw datafile or as a set of INSERT statements that recreate the records in the table.

To dump a table as a datafile, you must specify a --tab option that indicates the directory, where you want the MySQL server to write the file.

For example, to dump the tutorials_tbl table from the TUTORIALS database to a file in the /tmp directory, use a command like this:

```
$ mysqldump -u root -p --no-create-info \
            --tab=/tmp TUTORIALS tutorials_tbl
password ******
```

Exporting Table Contents or Definitions in SQL Format:

To export a table in SQL format to a file, use a command like this:

```
$ mysqldump -u root -p TUTORIALS tutorials_tbl > dump.txt
password ******
```

This will a create file having content as follows:

```
-- MySQL dump 8.23
--
-- Host: localhost    Database: TUTORIALS
---------------------------------------------------------
-- Server version       3.23.58

--
-- Table structure for table `tutorials_tbl`
--

CREATE TABLE tutorials_tbl (
  tutorial_id int(11) NOT NULL auto_increment,
  tutorial_title varchar(100) NOT NULL default '',
  tutorial_author varchar(40) NOT NULL default '',
  submission_date date default NULL,
  PRIMARY KEY  (tutorial_id),
  UNIQUE KEY AUTHOR_INDEX (tutorial_author)
) TYPE=MyISAM;

--
-- Dumping data for table `tutorials_tbl`
--

INSERT INTO tutorials_tbl
       VALUES (1,'Learn PHP','John Poul','2007-05-24');
INSERT INTO tutorials_tbl
       VALUES (2,'Learn MySQL','Abdul S','2007-05-24');
INSERT INTO tutorials_tbl
       VALUES (3,'JAVA Tutorial','Sanjay','2007-05-06');
```

To dump multiple tables, name them all following the database name argument. To dump an entire database, don't name any tables after the database as follows:

```
$ mysqldump -u root -p TUTORIALS > database_dump.txt
password ******
```

To back up all the databases available on your host, use the following:

```
$ mysqldump -u root -p --all-databases > database_dump.txt
password ******
```

The --all-databases option is available as of MySQL 3.23.12.

This method can be used to implement a database backup strategy.

Copying Tables or Databases to Another Host:

If you want to copy tables or databases from one MySQL server to another, then use mysqldump with database name and table name.

Run the following command at source host. This will dump complete database into dump.txt file:

```
$ mysqldump -u root -p database_name table_name > dump.txt
password *****
```

You can copy complete database without using a particular table name as explained above.

Now, ftp dump.txt file on another host and use the following command. Before running this command, make sure you have created database_name on destination server.

```
$ mysql -u root -p database_name < dump.txt
password *****
```

Another way to accomplish this without using an intermediary file is to send the output of mysqldump directly over the network to the remote MySQL server. If you can connect to both servers from the host where source database resides, use this command (Make sure you have access on both the servers):

```
$ mysqldump -u root -p database_name \
       | mysql -h other-host.com database_name
```

The mysqldump half of the command connects to the local server and writes the dump output to the pipe. The remaining mysql half of the command connects to the remote MySQL server on other-host.com. It reads the pipe for input and sends each statement to the other-host.com server.