

Using Relational Database on Linux

“”

Contents

1	Postgresql	1	2.3	Times And Dates	1	
2	The Mysql Relational Database	1	3	Sql	5	
	2.1	Back Ups	1	4	Sqlite	5
	2.2	Tables	1			

This book purports to provide concise and practical recipes for using open-source relational (sql) database systems with the Linux operating system, although the majority of the information should be applicable to any unix-type operating system.

Other databases: swordfish

Postgresql Section 1

www.postgresql.org
the official site

<http://www.cyberciti.biz/faq/psql-fatal-ident-authentication-failed-for-user/>
a good article about authentication problems (i.e. you cant logon)

see <http://bumble.sf.net/books/postgresql/postgresql-book.txt>

The Mysql Relational Database Section 2

Mysql is a very widely-deployed open-source (?) sql database.

Monitor the queries being run by MySQL

```
watch -n 1 mysqladmin --user=<user> --password=<password>  
⇒ processlist
```

2.1 Back Ups

Backup all MySQL Databases to individual files

```
mysql -e 'show databases' | sed -n '2,$p' | xargs -I DB 'mysqldump  
⇒ DB > DB.sql'
```

Convert mysql database from latin1 to utf8

```
mysqldump --add-drop-table -uroot -p "DB_name" | replace  
⇒ CHARSET=latin1 CHARSET=utf8 | iconv -f latin1 -t utf8 | mysql  
⇒ -uroot -p "DB_name"
```

Sync MySQL Servers via secure SSH-tunnel

```
ssh -f -L3307:127.0.0.1:3306 -N -t -x user@host sleep 600 ;  
⇒ mk-table-sync --execute --verbose  
⇒ u=root,p=xxx,h=127.0.0.1,P=3307 u=root,p=xxx,h=localhost
```

2.2 Tables

2.3 Times And Dates

Change date from MM/DD/YYYY to YYYY-MM-DD (mysql like)

```
date -d 09/20/1981 +"%Y-%m-%d"
```

Pulls FTP password out of Plesk database.

```
mysql -uadmin -p'cat /etc/psa/.psa.shadow' -e "use psa; select  
=> accounts.password from accounts INNER JOIN sys_users ON  
=> accounts.id=sys_users.account_id WHERE sys_users.login='xxxx';"
```

Find out if MySQL is up and listening on Linux

```
netstat -tap | grep mysql
```

Passwordless mysql{,dump,admin} via my.cnf file

```
echo -e "[client]\nuser = YOURUSERNAME\npassword = YOURPASSWORD" >  
=> ~/.my.cnf
```

Transfer sqlite3 data to mysql

```
sqlite3 mydb.sqlite3 '.dump' | grep -vE  
=> '^(BEGIN|COMMIT|CREATE|DELETE)"sqlite_sequence"' | sed -r  
=> 's/"([\^"]+)"\/\1\/' | tee mydb.sql | mysql -p mydb
```

Drop all tables from a database, without deleting it

```
MYSQL="mysql -h HOST -u USERNAME -pPASSWORD -D DB_NAME" ; $MYSQL  
=> -Bne "show tables" | awk '{print "set foreign_key_checks=0; drop  
=> table \" $1 \"';}' | $MYSQL unset MYSQL
```

Kill multiple Locked connection by a single user in MYSQL DB

```
for i in `mysqladmin -h x.x.x.x --user=root -pXXXX processlist |  
=> grep <<username>>| grep <<Locked>>| awk {'print $2'}'` do  
=> mysqladmin -h x.x.x.x --user=root -pXXX kill $i; done;
```

Dump mySQL db from Remote Database to Local Database

```
mysqldump --host=[remote host] --user=[remote user]  
=> --password=[remote password] -C db_name | mysql --host=localhost  
=> --user=[local user] --password=[local password] db_name
```

Export a subset of a database

```
mysqldump --where="true LIMIT X" databasename > output.sql
```

Backup all MySQL Databases to individual files

```
for db in $(mysql -e 'show databases' -s --skip-column-names); do  
=> mysqldump $db | gzip >  
=> "/backups/mysqldump-$(hostname)-$db-$(date  
=> +%Y-%m-%d-%H.%M.%S).gz"; done
```

Raw MySQL output to use in pipes

```
mysql DATABASE -N -s -r -e 'SQL COMMAND'
```

Explanation of system and MySQL error codes

```
perror NUMBER
```

Get table column names from an MySQL-database in comma-separated

```
mysql -u<user> -p<password> -s -e 'DESCRIBE <table>' <database> |  
=> tail
```

Determine configure options used for MySQL binary builds

```
grep CONFIG $(which mysqlbug)
```

Show database sql schema from Remote or Local database

```
mysqldump -u<dbusername> -p<dbpassword> <databasename> --no-data  
⇒ --tables
```

Command line to optimize all table from a mysql database

```
mysql -u uname dbname -e "show tables" | grep -v Tables_in | grep -v  
⇒ "+" | gawk '{print "optimize table " $1 ";"}' | mysql -u uname  
⇒ dbname
```

Parallel mysql dump restore

```
find -print0 | xargs -0 -n 1 -P 4 -I {} sh -c "zcat '{}'" | mysql  
⇒ nix"
```

Count the number of queries to a MySQL server

```
echo "SHOW PROCESSLIST\G" | mysql -u root -p | grep "Info:" | awk  
⇒ -F":" '{count[$NF]++}END{for(i in count){printf("%d: %s\n",  
⇒ count[i], i)}}' | sort -n
```

Export MySQL query as .csv file

```
echo "SELECT * FROM table;" | mysql -u root -p${MYSQLROOTPW}  
⇒ databasename | sed 's/\t/" /g;s/^/" /;s/$/" /;s/\n//g' >  
⇒ outfile.csv
```

Create an SSH tunnel for accessing your remote MySQL database with

```
ssh -CNL 3306:localhost:3306 user@site.com
```

Backup all MySQL Databases to individual files

```
for I in $(echo "show databases;" | mysql | grep -v Database); do  
⇒ mysqldump $I > "$I.sql"; done
```

Monitor server load as well as running MySQL processes

```
watch -n 1 uptime\;mysqladmin --user=<user> --password=<password>  
⇒ --verbose processlist
```

Create a mysql database from the command line

```
mysqladmin -u username -p create dbname
```

Display lines with a given string

```
look mysql /etc/group
```

Script para hacer un copia d ela base de datos mysql

```
FECHA=$(date +"%F") FINAL="$FECHA.sql.gz" mysqldump -h localhost -u  
⇒ user --password="pass" --opt jdiaz61_lupajuridica | gzip >  
⇒ /home/jdiaz61/www/backup/$FINAL
```

Checks all MySQL tables

```
myisamchk /path/to/mysql/files/*.MYI
```

Command line to drop all table from a databse

```
mysql -u uname dbname -e "show tables" | grep -v Tables_in | grep -v  
⇒ "+" | gawk '{print "drop table " $1 ";"}' | mysql -u uname  
⇒ dbname
```

Backup a remote database to your local filesystem

```
ssh user@host 'mysqldump dbname | gzip' >
⇒ /path/to/backups/db-backup-'date +%Y-%m-%d'.sql.gz
```

Load multiple sql script in mysql

```
cat schema.sql data.sql test_data.sql | mysql -u user
⇒ --password=pass dbname
```

Generate CHECK TABLE statements for all MySQL database tables on a

```
DD='cat /etc/my.cnf | sed "s/#.*//g;" | grep datadir | tr '=' ' ' |
⇒ gawk '{print $2;}' && ( cd $DD ; find . -mindepth 2 | grep -v
⇒ db\.opt | sed 's/\.\\///g; s/\\.\\.\\.\\$/g; s/\\//./;' | sort | uniq |
⇒ tr '/' ' .' | gawk '{print "CHECK TABLE",""$1"","";"}' )
```

Watch mysql processlist on a remote host

```
watch -n 0.5 ssh [user]@[host] mysqladmin -u [mysql_user]
⇒ -p[password] processlist | tee -a /to/a/file
```

Top like mysql monitor

```
mytop --prompt
```

Show mysql process ids

```
mysql -s -e "show processlist" | awk '{print $1}'
```

Get a mysqldump with a timestamp in the filename and gzip it all

```
mysqldump [options] | gzip ->mysqldump-$(date +%Y-%m-%d-%H.%M.%S).gz
```

Convert all MySQL tables and fields to UTF8

```
mysql --database=dbname -B -N -e "SHOW TABLES" | awk '{print "ALTER
⇒ TABLE", $1, "CONVERT TO CHARACTER SET utf8 COLLATE
⇒ utf8_general_ci;"}' | mysql --database=dbname &
```

Determine configure options used for MySQL binary builds

```
cat 'whereis mysqlbug | awk '{print $2}'' | grep 'CONFIGURE_LINE='
```

Execute all SQL files in a directory

```
cat *.sql | mysql <db_name>
```

Changes standard mysql client output to 'less'.

```
echo -e "[mysql]\npager=less -niSFX" >> ~/.my.cnf
```

Create MySQL-Dump, copy db to other Server and upload the db.

```
mysqldump -uUserName -pPassword tudb | ssh root@rootsvr.com "mysql
⇒ -uUserName -pPassword -h mysql.rootsvr.com YourDBName"
```

Get a MySQL DB dump from a remote machine

```
ssh user@host "mysqldump -h localhost -u mysqluser -pP@$$W3rD
⇒ databasename | gzip -cf" | gunzip -c > database.sql
```

Mysql DB size

```
mysql -u root -pPasswort -e 'select
⇒ table_schema,round(sum(data_length+index_length)/1024/1024,4)
⇒ from information_schema.tables group by table_schema;'
```

Copy a MySQL Database to a new Server via SSH with one command

```
mysqldump --add-drop-table --extended-insert --force
⇒ --log-error=error.log -uUSER -pPASS OLD_DB_NAME | ssh -C
⇒ user@newhost "mysql -uUSER -pPASS NEW_DB_NAME"
```

Pulls email password out of Plesk database for given email

```
mysql -uadmin -p'cat /etc/psa/.psa.shadow' -e "use psa; select
⇒ accounts.password FROM accounts JOIN mail ON
⇒ accounts.id=mail.account_id WHERE mail.mail_name='webmaster';"
```

Backup all MySQL Databases to individual files

```
for I in $(mysql -e 'show databases' -s --skip-column-names); do
⇒ mysqldump $I | gzip > "$I.sql.gz"; done
```

Create Encrypted WordPress MySQL Backup without any DB details,

```
eval $(sed -e
⇒ "s/^d[^D]*DB_\([NUPH]\).*',\[^']*'\([^']**\)'.*/_\1='\2';/" -e
⇒ "/^_/_!d" wp-config.php) && mysqldump --opt --add-drop-table
⇒ -u$_U -p$_P -h$_H $_N | gpg -er AskApache >'date
⇒ +%m%d%y-%H%M.$_N.sqls'
```

Get column names in MySQL

```
mysql -u <user> --password=<password> -e "SHOW COLUMNS FROM <table>"
⇒ <database> | awk '{print $1}' | tr "\n" "," | sed 's/,,$//g'
```

Backup local MySQL database into a folder and removes older then

```
mysqldump -uUSERNAME -pPASSWORD database | gzip >
⇒ /path/to/db/files/db-backup-'date +%Y-%m-%d'.sql.gz ;find
⇒ /path/to/db/files/* -mtime +5 -exec rm {} \;
```

Positions the mysql slave at a specific master position

```
slave start; SELECT MASTER_POS_WAIT('master.000088', '8145654');
⇒ slave stop;
```

Analyze, check, auto-repair and optimize Mysql Database

```
mysqlcheck -a --auto-repair -c -o -uroot -p [DB]
```

Section 3

Sql

Create SQL-statements from textfile with awk

```
for each in `cut -d " " -f 1 inputfile.txt`; do echo "select * from
⇒ table where id = \"$each\";"; done
```

Section 4

Sqlite

Sqlite is a very small and compact implementation of a relational database which is designed for restricted and embedded environments, such as mobile phones, or small applications which want to save their data somewhere.

Speed up launch of liferea

```
sqlite3 ~/.liferea_1.4/liferea.db 'VACUUM;'
```

Extract your list of blocked images hosts from Firefox database

```
sqlite3 -noheader -list
⇒ ~/.mozilla/firefox/<your_profile>/permissions.sqlite "select
⇒ host from moz_hosts where type='image' and permission=2"
```

Create sqlite db and store image

```
sqlite3 img.db "create table imgs (id INTEGER PRIMARY KEY, img
⇒ BLOB); insert into imgs (img) values ('\$(base64 -w0
⇒ /tmp/Q.jpg)\"); select img from imgs where id=1;" | base64 -d
⇒ -w0 > /tmp/W.jpg
```