

MySQL Backup

It is good practice to back up your server regularly. If you are using MySQL, the mysqldump command makes it very easy.

The typical command to back up your entire database is:

```
mysqldump -u root -p -e --single-transaction -q -r"/home/ben/backup.sql" openmrs
```

- -uroot = use username of root
- -p = ask for password
- -e = extended inserts (makes the dump 10x smaller)
- --single-transaction = run this in a single transaction (to prevent corruption)
- -q = quick
- -r = file location to store
- openmrs = name of your database

If you want to automate it, you can provide the username and password for mysql in the command:

```
mysqldump -u root -pPA$$W0rd -e --single-transaction -q -r"/home/ben/backup.sql" openmrs
```

If you want to exclude a few of the extremely large tables, you can do that too

```
mysqldump -u root -p -e --single-transaction -q -r"/home/ben/backup.sql" --ignore-table=hl7_in_archive --ignore-table=formentry_archive openmrs
```

If you are on a linux variant, you can insert the date automatically into the backup sql filename:

```
mysqldump -uroot -p -e --single-transaction -q -r"/home/ben/backup`date +%Y%m%d`.sql" openmrs
```

If you are on windows, you can insert the date automatically too, its just a little uglier:

```
mysqldump -u root -p -e --single-transaction -q -r"C:\backup%date:~10,4%%date:~4,2%%date:~7,2%" openmrs
```

After creating the backup .sql file, you can zip it and store it for later use.

Restoring From A Backup File

The now get the data out of an archived sql backup file and into a new database its merely three simple steps:

1. Create your new database
 - create database openmrsbackuprestoration default character set utf8;
2. Tell mysql to work against that new database
 - use openmrsbackuprestoration
3. Run the sql file
 - source backup.sql

Now wait for 1, 20, or 30 minutes and you'll have all your data back.