



Standard Operating Procedures for e-Saude database mask SOP

Jembi Health Systems

e-Saude database mask SOP

Document version: 1.0.0
Platforms versions: mysql 5.6
Written by: Paulo Sérgio
Modified by: Nathan
Approved by: <CDC to approve it>
Last update: Oct 25, 2018

Change Log

Date	Revision	Description	Who
25.10.2018	0.1	- Some basic mysql commands.	Paulo Nathan

Purpose

Obfuscate the patient identification data in an EPTS OpenMRS database.

Scope

The assumption is that the person who is going to go through this steps is a technician and some steps may be skipped, so please fill free to contact Jembi in case you need any other step, This also assume you have test server where you will be running the script provided in this document...

Definition

Responsible Personnel

Requirements/Materials

Hardware Requirements

The absolute **minimum requirements**:

- **CPU:** Intel Core 2 Duo or i3 or equivalent
- **RAM:** 4 GB or 8GB
- **SSD:** 50 GB Free space

You just need a machine with mysql server not openmrs

Software Requirements

Any type of server with mysql installed
Mysql 5.6

Procedure

To scramble the database, you need to log in to the production machine and do a dump of the database being used(The name can vary, ask the person who did the installation) usually will be called 'openmrs', then move this script to a test server and restore it to any name of the database you want and run the mask script. Dump it again from this test machine and you will have a scrambled database.

Prerequisites

You need a test server with mysql installed.

1. Make a dump of a database:

If you already have a dump jump to step 2 if not run this in your production server to create a copy of your database.

```
$ mysqldump -u <username> -p database_name > data_dump.sql
```

- username is the username you can log in to the database with
- database_name is the name of the database that will be exported
- data-dump.sql is the file in the current directory that the output will be saved to

After the execution run this command to check if is legitimate sql dump

```
$ head -n 5 data_dump.sql
```

The result should look something like this

```
-- MySQL dump 10.13  Distrib 5.5.54, for debian-linux-gnu (x86_64)
--
-- Host: localhost    Database: openmrs
--
-- Server version      5.5.54-0ubuntu0.12.04.1
```

Note: In a machine with 8GB of RAM and a db of 886 MB it takes less than a minute to create the dump.

Copy and past this dump into a test server, if you coming from step 3 please stop here (it's a trap - loop ahead).

2. Undump the database:

First log in to the database

```
$ sudo mysql -u root -p
```

Create a database with the name you want (Use suggestive names).

```
$ mysql > CREATE DATABASE new_database;
```

You will see this output confirming that it was created

```
mysql> create database to_mask_sofala;
Query OK, 1 row affected (0.00 sec)
```

Make use of this database you just created

```
$ mysql> use new_database;
```

You will see this output

```
mysql> use to_mask_sofala;
Database changed
```

Make a source of the dump database

```
$ mysql> source </path/to/where/dump/stored/data_dump.sql>;
```

3. Scramble the database

```
$ mysql> source </path/to/where/dump/stored/scramble script received.sql>;
```

If all run smoothly you will see the progress and After this, dump this newly scrambled database, refer to step one if any doubt of how to dump a database.

Funding:

"The PoC project is a result of the partnership between Mozambique and American Government supported by PEPFAR - CDC"



