**Note:** This file is created in assumption that you already have Linux installed. If this isn’t the case, please follow this link for installation instructions.

**Installing MySQL on RHEL Steps:**

**PRE-INSTALLATION CONFIGURATION**

1. Ensure the both <you> and `sudo` user has appropriate privileges (i.e. rwx) on the following files:

* /usr/bin/sudo (--x)
* /etc/sudoers (rwx)
* /usr/bin/nohup (--x)
* /usr/bin/uptime (--x)
* /usr/bin/which (--x)
* /usr/bin/stat (--x)
* /bin/bash (rwx)

Also, the Root Admin should edit this file `/etc/sudoers` and add you (DBA) as a super user. The following lines of command will do this:

Username ALL=(ALL:ALL) ALL

env\_keep +="HOME"

Set PATH to the appropriate *root’s sub*directory. Here, edit the “.path” file and add this line:

set path=(/export/data/hdd/software/canvas/bin \

    $bin\_path $glue\_path . )

1. Adding the appropriate RHEL Repository (YUM or RPM) as follow:

   a. Download the repository (i.e. YUM, APT, etc) for us, YUM is what we need.

   b. Copy download to remote server (i.e Nightingale)

   c. Once copied, run the following commands:

shell> sudo rpm -Uvh /whole\_download\_path/download-package-name.rpm

1. **OPTIONAL**-Select a new (usually older version) or leave the release data as default (often latest release)

shell> yum repolist all | grep mysql (checks the release)

  shell> sudo yum-config-manager --disable mysql80-community (disables latest version)

shell> sudo yum-config-manager --enable mysql57-community (enables older version)

Enable subrepository for one release series at any time. When subrepositories for more than one release series are enabled, the latest series will be used by Yum

shell> yum repolist enabled | grep mysql (verifies release version)

**NOTE**: Before you begin this installation or any other installation, ensure your PATH variable is set to the root level directories. Otherwise, your installation will end up in your HOME directory.

**INSTALLATION**

1. a. Download, copy MySQL DBMS package (YUM for example) remotely to Nightingale where it can be installed.

NOTE: It’s strongly recommended that you download the .tar file. This way, you wouldn’t be missing any necessary packages. Run the following command to unzip the .tar file:

shell> tar –C /expected\_etracted\_files\_location\_directory/ -xvf /Location\_of\_the\_tar\_file

Then, run the following Linux command to begin installation:

shell> sudo yum install /whole\_download\_path/mysql-community-serve

**Starting the MySQL Server**

1. After installation, run the following scripts to start-up the MySQL DB

shell> sudo systemctl start mysqld.service

shell> sudo service mysqld start (for versions earlier than RHEL 7 only)

shell> sudo systemctl status mysqld.service (Check DB status with this command)

shell> sudo service mysqld status (for versions earlier than RHEL 7 only)

1. *MySQL Server Initialization (as of MySQL 5.7):*

At the initial start up of the server, the following happens, given that the data directory of the server is empty:

The server is initialized.

An SSL certificate and key files are generated in the data directory.

The [validate\_password plugin](https://dev.mysql.com/doc/refman/8.0/en/validate-password.html" \t "_top) is installed and enabled.

A superuser account 'root'@'localhost' is created. A password for the superuser is set and stored in the error log file. To reveal it, use the following command:

shell> sudo grep 'temporary password' /var/log/mysqld.log

Change the root password as soon as possible by logging in with the generated, temporary password and set a custom password for the superuser account:

shell> mysql -uroot -p

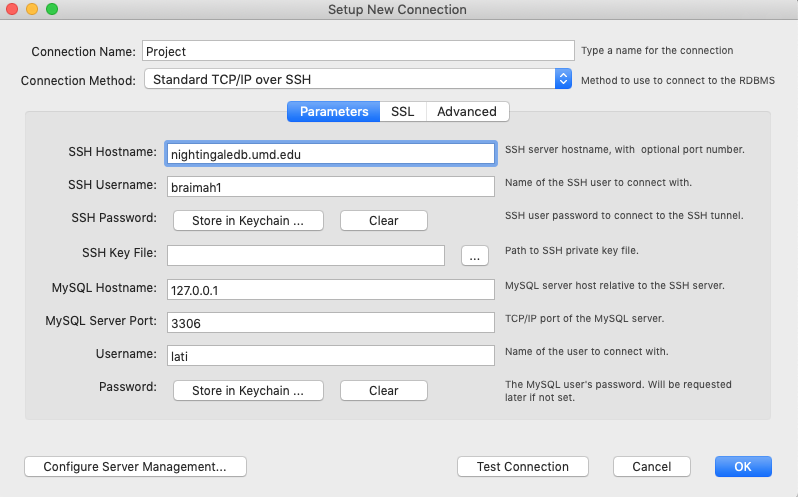
mysql> ALTER USER 'root'@'localhost' IDENTIFIED BY 'MyNewPass4!';

**Note:** MySQL's [validate\_password](https://dev.mysql.com/doc/refman/8.0/en/validate-password.html" \t "_top) plugin is installed by default. This will require that passwords contain at least one upper case letter, one lower case letter, one digit, and one special character, and that the total password length is at least 8 characters.

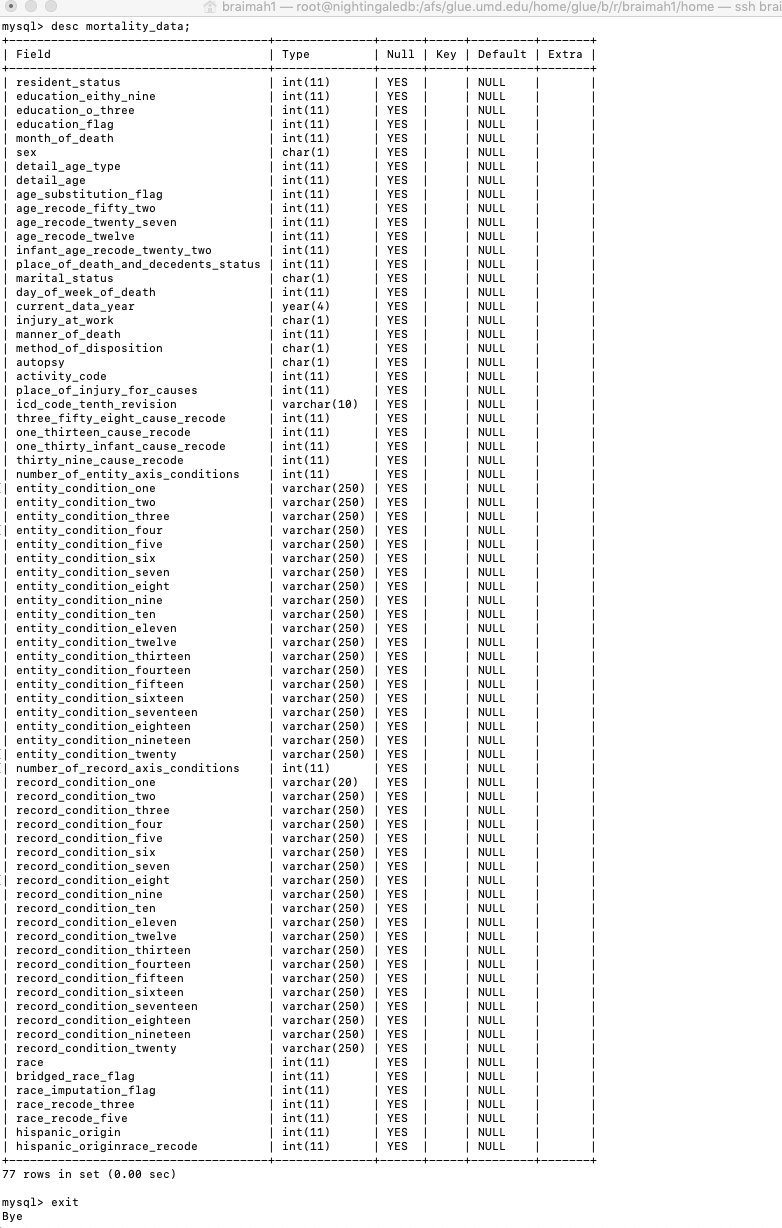
## Installing Additional MySQL Products and Components (i.e. workbench)

shell> sudo yum install /whole\_download\_path/mysql-community-workbench

1. Once installed, launch the application and create a connection by clicking on the “+” sign and filling out the form as shown below:

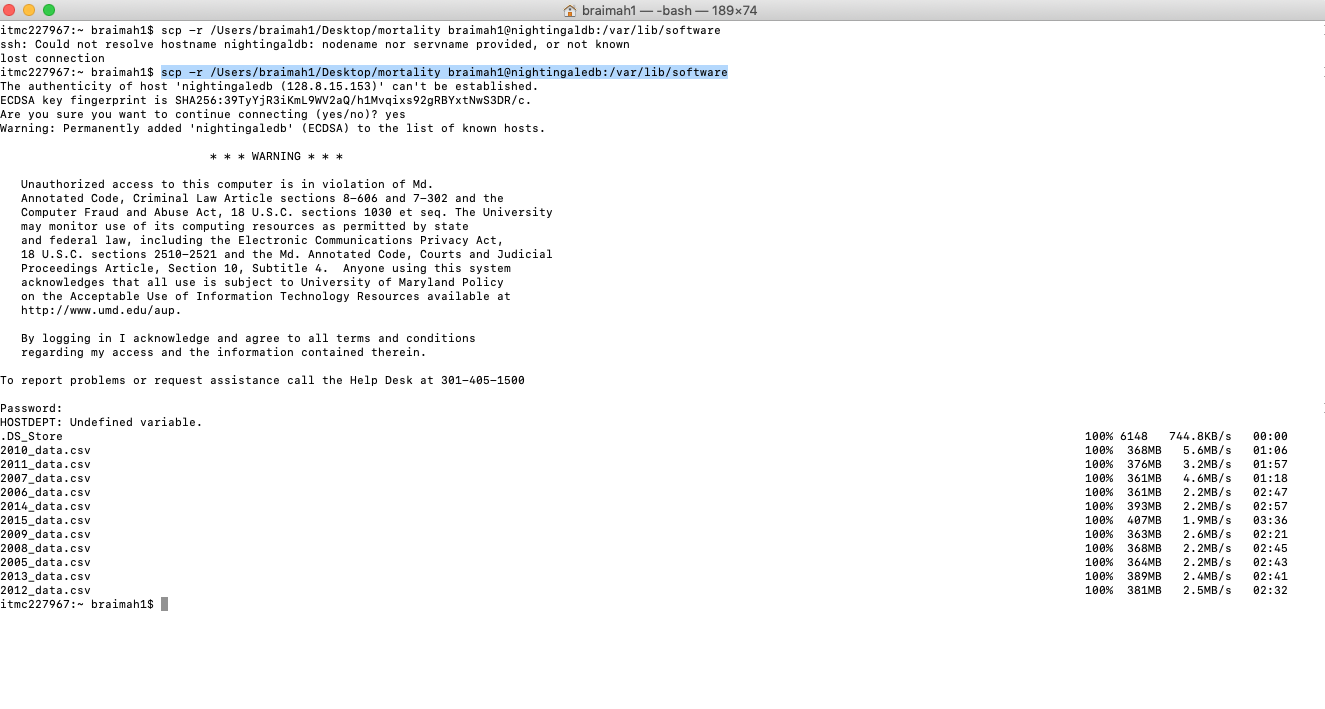


1. Create a Database called ‘pyspark’.
2. Create a Table called “Mortality\_Data” to be used to merge our dataset from 2005 to 2015. (See SQL file for the create table query).

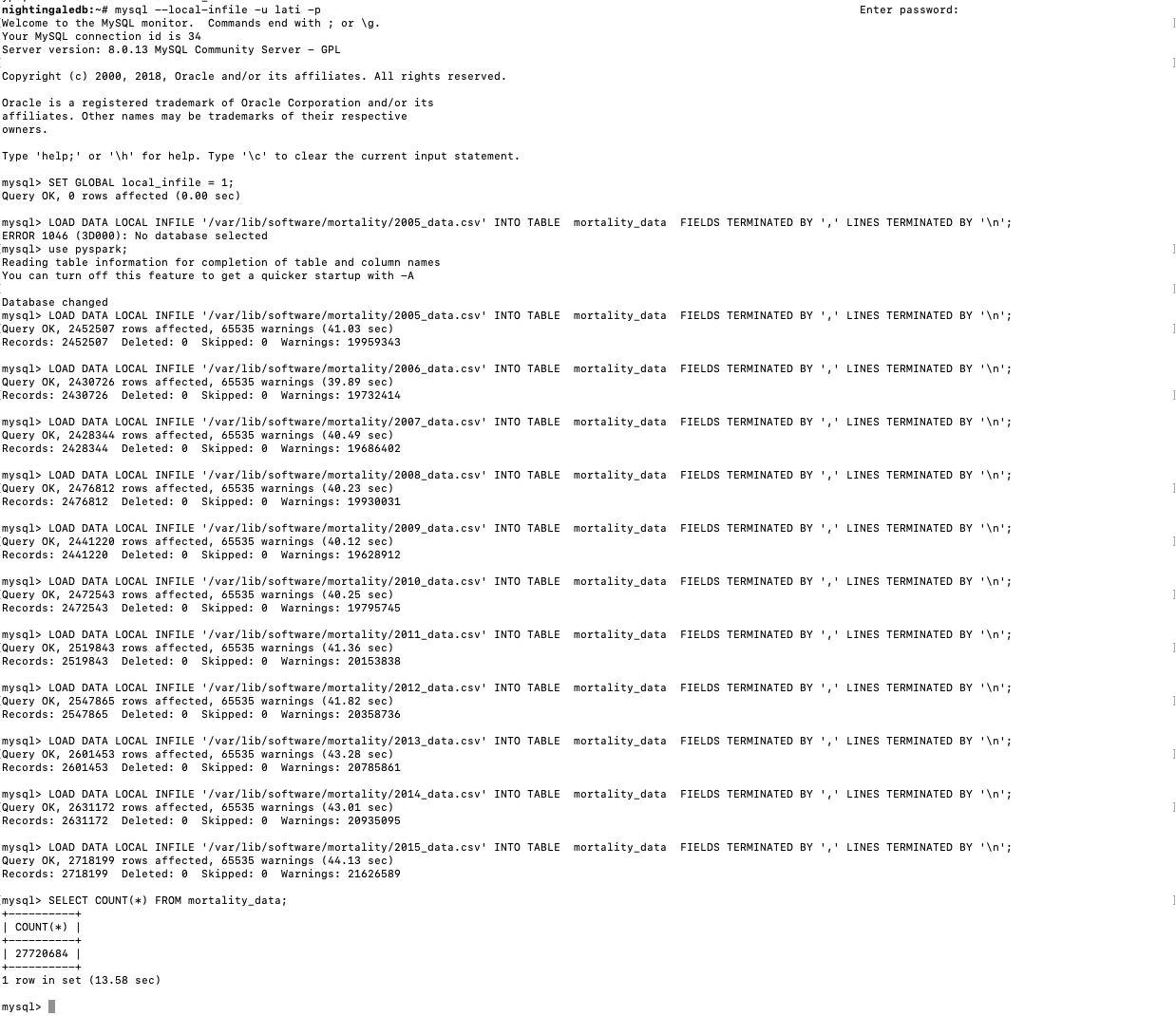


1. Download and Load the mortality data into the new table. See SQL file for the load query. In this case, we are using a virtual machine on a MAC book, so we downloaded and used SSH ‘scp’ command to copy the data over to the Linux server as follow:

$ scp -r /Users/braimah1/Desktop/mortality braimah1@nightingaledb:/var/lib/software



Screenshots of data upload showing total number of records (27,720,684) after complete upload:



## Installing and Setting-up Apache Spark Standalone cluster on RHEL 7

## Requirements

## Java (JRE) 7+

## Python 2.6 or higher

## Step 1: Before Installation Run the following commands:

$ sudo yum update –- This, updates the yum libraries

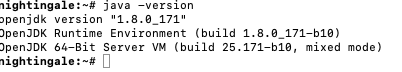
$ sudo systemclt stop firewall-cmd

-- Stops the firewall to make all the ports accessible (i.e. for master and slave clusters to communicate effectively).

$ sudo setenforce 0 -- disable SELinux for the session i.e. until next reboot – to permanently disable it set SELINUX=disabled in /etc/selinux/config file

**Step 2: Verify that Java is installed:**

$ sudo java -version



Otherwise, follow this link for instructions on [how to install Java](http://devopspy.com/linux/install-java-8-on-centos-rhel-7/).

**Step 3 Install Apache Spark:**

Create a directory for your spark files:

$ mkdir apacheSpark

$ cd apacheSpark

Download Spark:

$ sudo wget <http://apache.claz.org/spark/spark-2.4.0/spark-2.4.0-bin-hadoop2.7.tgz>



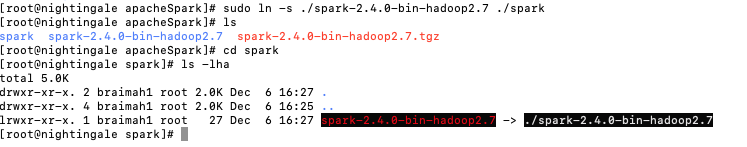
Untar Spark download

$ sudo tar -xzf spark-2.4.0-bin-hadoop2.7.tgz

Create a Symlink: This enables us to use multiple Spark versions by just linking to the one we need at a particular time. First create a directory called ‘spark’, then symlink to it

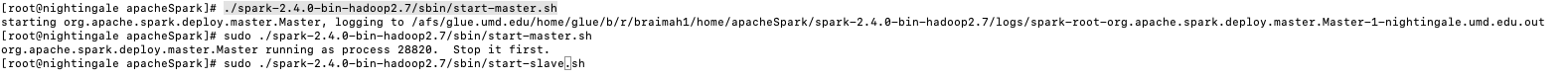
$ mkdir spark

$ sudo ln -s ./spark-2.4.0-bin-hadoop2.7 ./spark



## Step 4 Launch standalone Spark cluster

$ ./spark-2.4.0-bin-hadoop2.7/sbin/start-master.sh



## Step 5 Install and Launch PySpark:

## $ spark-2.4.0-bin-hadoop2.7/bin/pyspark

## 

## Spark Plot: <https://plot.ly/python/apache-spark/>

## NOTE: Logging in to PySpark or Scala (spark-sell), from apacheSpark Dir:

## $ spark-2.4.0-bin-hadoop2.7/bin/pyspark

## $ spark-2.4.0-bin-hadoop2.7/bin/pyspark --packages mysql:mysql-connector-java:5.1.38

## $ spark-2.4.0-bin-hadoop2.7/bin/spark-shell