Red Hat JBoss Data Virtualization

and Hortonworks

Proof of Technology

Table of Contents

Proof of Technology Overview

We will concentrate on three use cases for the Proof of Technology with Hortonworks and Red Hat Jboss Data Virtualization. Each section will go through setting up the environment for the use case and then testing it. Also in the references we have the video links, source code links and supporting collateral links.

Use Case One – Sentiment Analysis and Sales Analysis with Hadoop and MySQL

Use Case Two – Federated Hadoop with Security

Use Case Three– Hadoop Datalake

Use Case One – Sentiment Analysis and Sales Analysis with Hadoop and MySQL

Introduction

This use case is the sentiment analysis and sales analysis with Hadoop and MySQL. It uses one Hortonworks Data Platform VM for the twitter sentiment data and one MySQL database for the sales data. This guide shows how to reproduce the setup and then run the demonstration. The presentation listed in the references section describes the 3 use cases at a high level.

Objective: Determine if sentiment data from the first week of the Iron Man 3 movie is a predictor of sales

Problem: Cannot utilize social data and sentiment analysis with sales management system

Solution: Leverage Jboss Data Virtualization to mashup Sentiment analysis data with ticket and merchandise sales data on MySQL into a single view of the data

We tested on the following two environments:

Environment 1

System 1 Macbook Pro

SquirreL Client, Data Virtualization, VMWare Fusion installed on Mac

HDP Sandbox 2.1 VM

Microsoft Windows VM with Libreoffice and ODBC driver

Environment 2

System 1 Fedora 19

Virtualbox 4.3 with HDP 2

Data Virtualization 6 and JDK 1.7 update 51

MySQL 5.5

System 2 Windows 7

Microsoft Excel 2013 with Powerview

Data Virtualization ODBC Driver

Install Hortonworks Data Platform with Sentiment Data

There are two options to install the HDP with sentiment data. The first is to import the Virtual Box VM with the tweetsbi data already loaded. The second is to use one of the options for the HDP and then load the twitter data in from tweetsbi.csv or the tutorial. For these instructions we are going to concentrate on downloading the HDP Sandbox and importing the tweetsbi data so that you have the most current Sandbox. The sandbox will contain the tweets with sentiment data.

Step 1: Download and install HDP according to the instructions

http://hortonworks.com/products/hortonworks-sandbox/#install

Step 2: Load the sentiment data into the HDP. You can either follow the tutorial below to go through the complete process to create tweetsbi or just load the tweetsbi table from the tweetsbi.csv.

Tutorial: http://hortonworks.com/hadoop-tutorial/how-to-refine-and-visualize-sentiment-data/

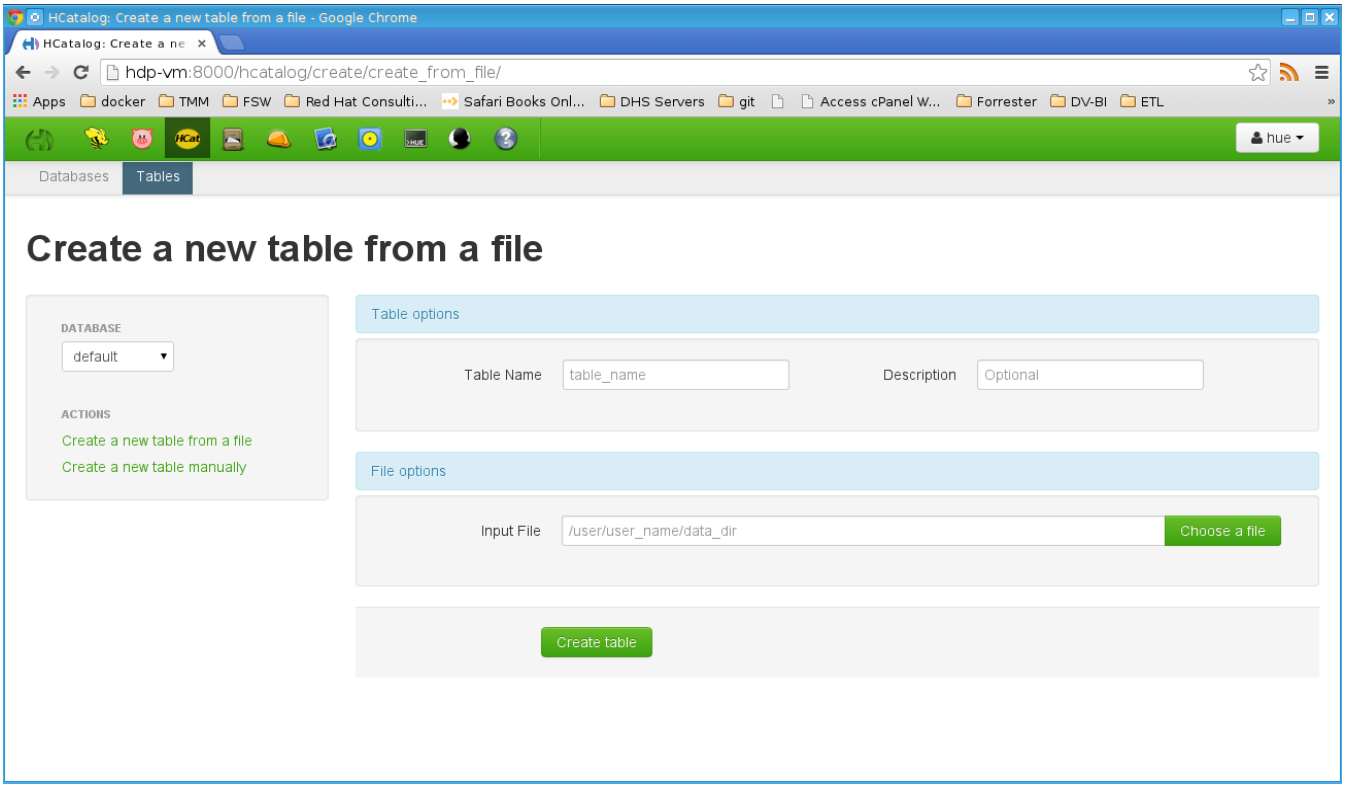
To load the data from tweetsbi.csv download the csv.

tweetsbi.csv: https://drive.google.com/file/d/0B5kKwcd4kOq9a3c5R2ZlRXZOdlU/edit?usp=sharing

Next create the tweetsbi table from a file. Make sure the HDP Sandbox is running. Browsing to

http://hdp-vm:8000/hcatalog/create/create_from_file

where hdp-vm is the IP of the sandbox or add hdp-vm to the host file with the IP. Next enter tweetsbi as the tablename, choosing the tweetsbi.csv and then clicking create table. Make sure the delimeter is changed to a comma but keep the other defaults.



Step 3: In Ambari using admin/admin verify some configuration by Ambari > Services > Configs > Advanced. This should be done to avoid having to issue grants when doing imports.

* hive.security.authorization.enabled, change from true to false

* hive.server2.enable.doAs, change from true to false

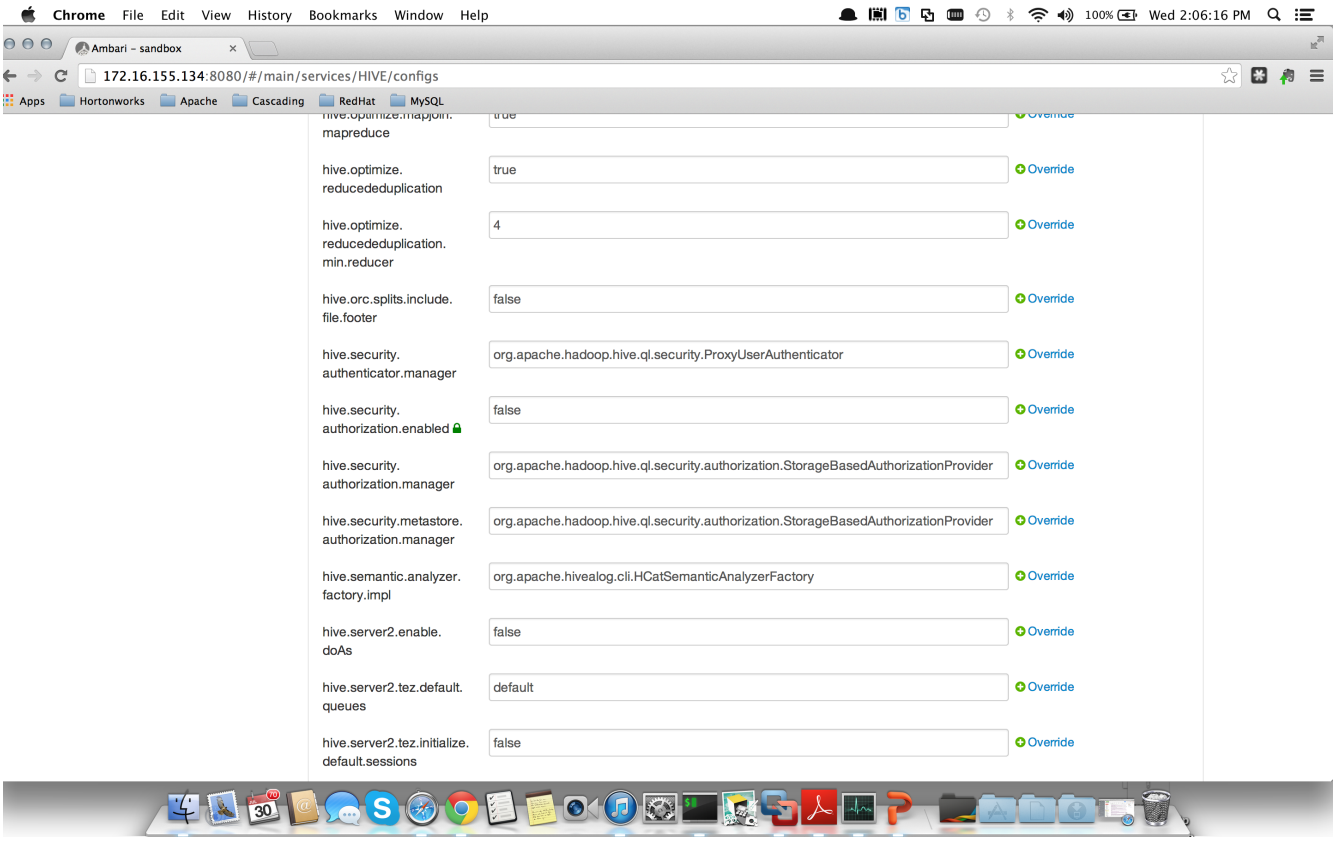
Optionally, and recommended for performance, change this option:

* hive.execution.engine, change from mr to tez

Restart All Hive Services

NOTE: Should you skip step 3 or have permission errors when access the tweetsbi table through hive then issue the query below through http://hdp-vm:8000/beeswax/

grant select on table tweetsbi to user hdfs;



Install MySQL with Sales Data

MySQL will contain the sales data for the iron man movie. MySQL can be installed locally but we will use MySQL on the HDP Sandbox.

Step 1: Grant privileges for the local machine for DV and SquirreL client to connect to the MySQL instance on the sandbox. Log into sandbox as root, type in “mysql” and then enter the following command. You can use % meaning all remote access or you can use the host/local OS IP.

mysql> use mysql

mysql> GRANT ALL ON *.* to admin@'%' IDENTIFIED BY 'admin';

mysql> FLUSH PRIVILEGES;

Step 2: Download the sql script and then scp <sqlscript> root@<sandbox_ip>

https://drive.google.com/file/d/0B5kKwcd4kOq9UWJvSDU5Q1NOcjg/edit?usp=sharing

Step 3: Run the sql script to create the database, table and data

mysql < sales-create-table-and-data.sql

Step 4: Test user and check table by signing on

mysql -u admin -p'admin' hadoopworld

Step 5: Run the SQL command

SELECT * FROM sales;

Install SQuirreL Client to test the database connections

NOTE: SquirreL Client is running on the local/host to connect to DV and the HDP VM.

Step 1: Download and install SquirreL client to test the databases

http://squirrel-sql.sourceforge.net/

Step 2: Install the jdbc drivers by downloading them from the below and placing them in the SquirreL client lib folder

Hive (hive0jdbc-0.11.0.jar):
https://drive.google.com/file/d/0B5kKwcd4kOq9MElfam9yaGw2Z1E/edit?usp=sharing

MySQL (mysql-connector-java-5.1.25-bin.jar):
https://drive.google.com/file/d/0B5kKwcd4kOq9UC1keS1Id3Fsamc/edit?usp=sharing

Teiid (teiid-8.4.1-redhat-2-jdbc.jar):
https://drive.google.com/file/d/0B5kKwcd4kOq9U1pQQjVLMlVUSEk/edit?usp=sharing

Setup the Drivers for according to the below which should give a blue checkmark beside the driver

Hive:

Class Name: org.apache.hive.jdbc.HiveDriver

Example URL: jdbc:hive2://localhost:10000/default

Mysql:

Class Name: com.mysql.jdbc.Driver

Example URL: jdbc:mysql://hostname:3306/dbname

Unififed View (Teiid):

Class Name: org.teiid.jdbc.TeiidDriver

Example URL: jdbc:teiid:theVDB@mm://localhost:31000

Step 3: Setup the database connections according to the below

Hive:

Name: Hive-HDP

Driver: Apache Hive

URL: jdbc:hive2://<sandbox-IP>:10000/default

Username: hdfs

Password: empty

Mysql:

Name: MySQL

Driver: MySQL

URL: jdbc:mysql://<sandbox-IP>:3306/hadoopworld

User: admin

Password: admin

Unififed View (Teiid):

Name: SalesSentimentCountry

Driver: Teiid JDBC Driver

URL: jdbc:teiid:HiveTestVDB@mm://localhost:31000

User: user

Password: user

Step 4: Connect to the databases and select the tables then the content tab to see the database data

A. HDP-VM preview tweetsbi which will take some time for the data to be returned on the first query

B. MYSQL on HDP-VM preview sales which should be quick

C. Sentiment preview which will preview the unified view (NOTE: will not work until after DV has been installed and running)

Install Data Virtualization with Virtual Database

Step 1: Clone the Simplified Data Virtualization template.

https://github.com/kpeeples/simplified-dv-template.git

Step 2: Download Data Virtualization to the distros folder from

http://www.jboss.org/products/datavirt/download/

Step 3: Run the install-run.sh script to install the DV server

Step 4: Stop the server

Step 5: Create the Hive Module by creating the org/apache/hadoop/hive directory under the DV_ROOT/modules/system/layers/base folder and then unzipping the org-apache-hive.zip files to the new folder

https://drive.google.com/file/d/0B5kKwcd4kOq9RkdjdXRKZmxlTUk/edit?usp=sharing

Step 6: Deploy the MySQL JDBC driver from the support folder by going to localhost:8080 and clilcking on the admin console with the username admin and password redhat1!. Then click on manage deployments and click add

Step 7: Setup a new datasource called MySQLSalesModel with the MySQL driver according to

```
<datasource jndi-name="java:/MysqlDatasource" pool-name="MysqlDatasource" enabled="true">
    <connection-url>jdbc:mysql://<sandbox_IP>:3306/hadoopworld</connection-url>
    <driver>mysql-connector-java-5.1.25-bin.jar</driver>
    <security>
        <user-name>admin</user-name>
        <password>admin</password>
    </security>
</datasource>
```

Step 8: Setup a new datasource called HiveConnection with the Hive driver according to

```
<datasource jndi-name="java:/HiveConnection" pool-name="HiveConnection" enabled="true">
    <connection-url>jdbc:hive2://<sandbox_IP>:10000/default</connection-url>
    <driver>hive</driver>
    <security>
        <user-name>hdfs</user-name>
        <password>admin</password>
    </security>
</datasource>
```

Step 9: Deploy the HiveTestVDB by following the same process as step 6

https://drive.google.com/file/d/0B5kKwcd4kOq9Z2FJYXF5VFlrdnM/edit?usp=sharing

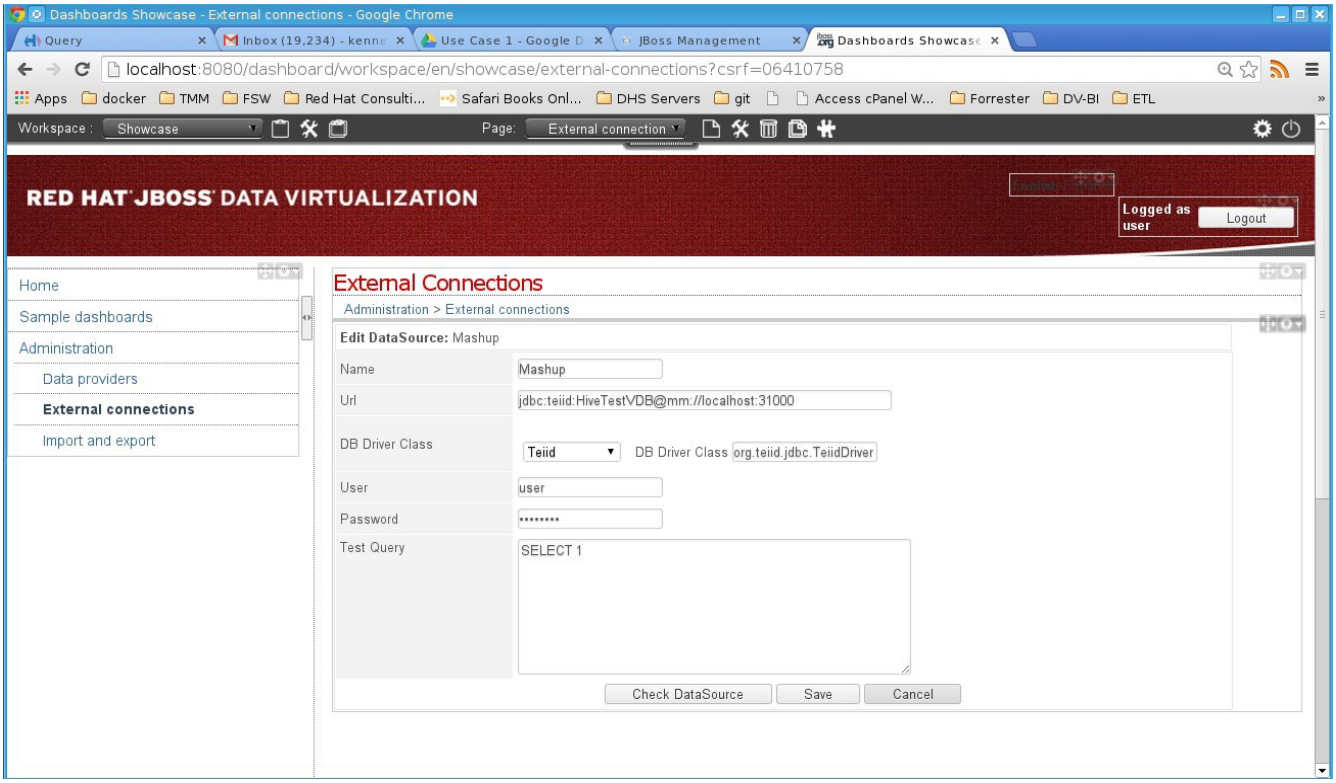
Step 10: Start the server by running standalone.sh -b 0.0.0.0 from the DV_ROOT/bin

Test the unified view

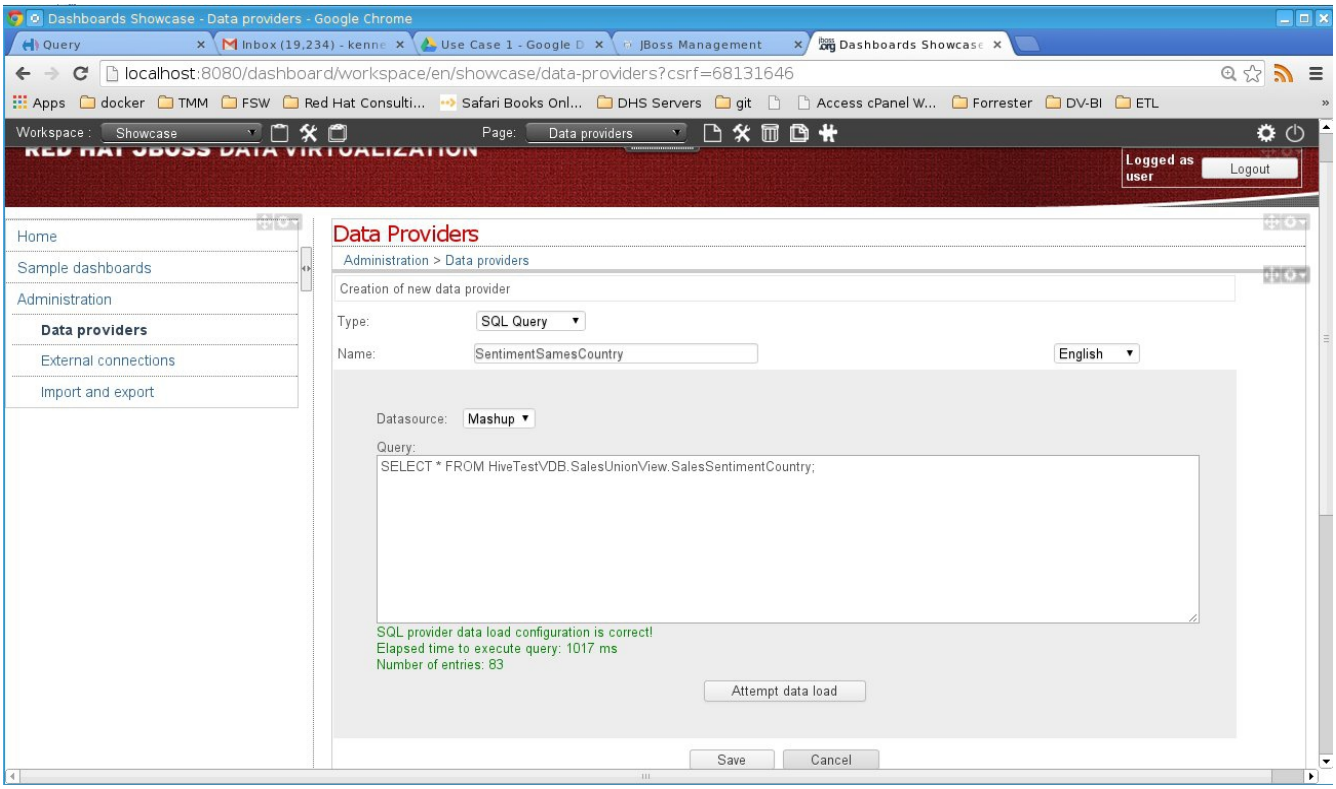
Data Virtualization Dashboard

Step 1: Test the Data Virtualization Dashboard. While the DV server is running browse to localhost:8080/dashboard

Step 2: Create the Teiid external connections



Step 3: Create the data provider



Step 4: Create the workspace

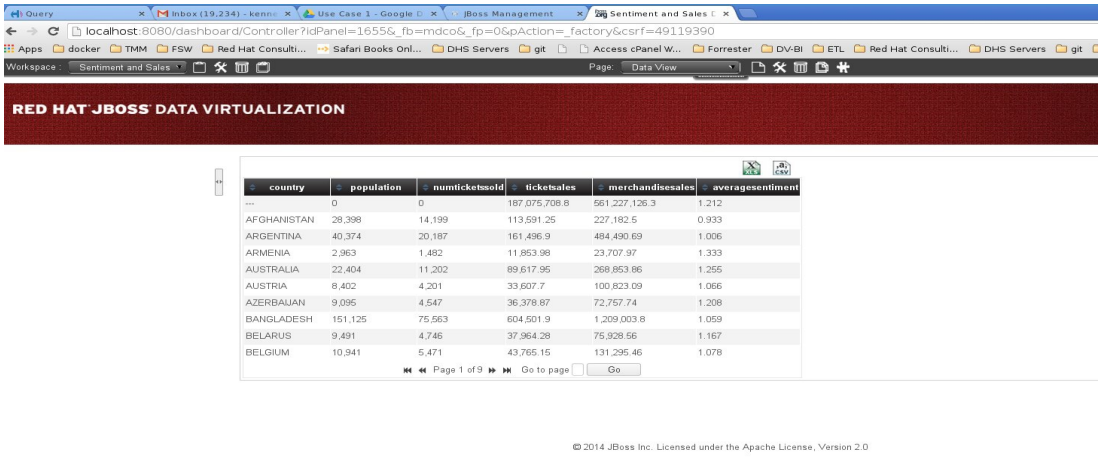
Step 5: Create a new page after selecting the newly created workspace

Step 6: Create a new panel after selecting the newly created page

Step 7: Select the key performance indicator and drag and drop to the panel

Step 8: Select the created data provider

Step 9: Select the Data Table to see the data in a spreadsheet

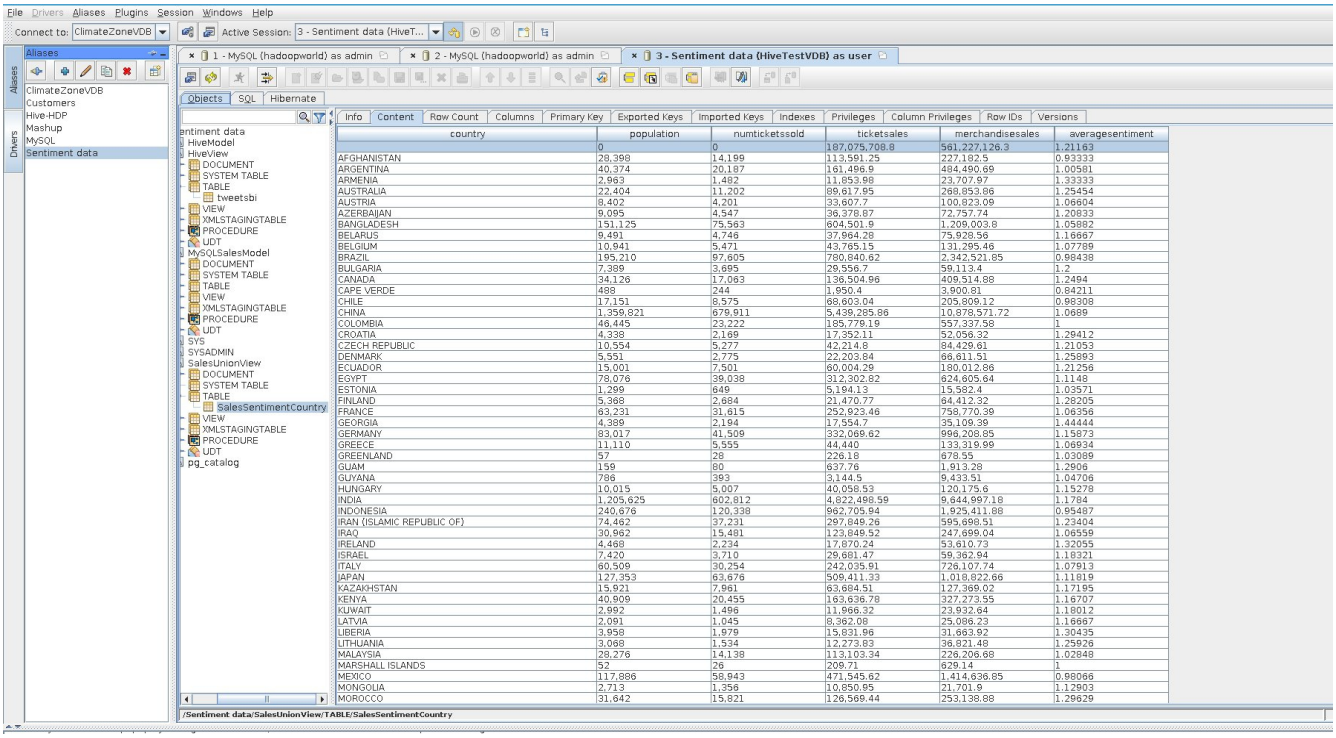


country	population	numticketssold	ticketsales	merchandisesales	averagesentiment
---	0	0	187,075,708.8	561,227,126.3	1.212
AFGHANISTAN	28,398	14,199	113,591.25	227,182.5	0.933
ARGENTINA	40,374	20,187	161,496.9	484,490.69	1.006
ARMENIA	2,963	1,482	11,853.98	23,707.97	1.333
AUSTRALIA	22,404	11,202	89,617.95	268,853.86	1.255
AUSTRIA	8,402	4,201	33,607.7	100,823.09	1.066
AZERBAIJAN	9,095	4,547	36,378.87	72,757.74	1.208
BANGLADESH	151,125	75,563	604,501.9	1,209,003.8	1.059
BELARUS	9,491	4,746	37,964.28	75,928.56	1.167
BELGIUM	10,941	5,471	43,765.15	131,295.46	1.078

Page 1 of 9 Go to page Go

SquirreL Client

Step 1 Preview the content for the SalesSentimentCountry Table with the datasource that was setup above



country	population	numticketssold	ticketsales	merchandisesales	averagesentiment
	0	0	187,075,708.8	561,227,126.3	1.21163
AFGHANISTAN	28,398	14,199	113,591.25	227,182.5	0.93333
ARGENTINA	40,374	20,187	161,496.9	484,490.69	1.00581
ARMENIA	2,963	1,482	11,853.98	23,707.97	1.33333
AUSTRALIA	22,404	11,202	89,617.95	268,853.86	1.25454
AUSTRIA	8,402	4,201	33,607.7	100,823.09	1.06604
AZERBAIJAN	9,095	4,547	36,378.87	72,757.74	1.20833
BANGLADESH	151,125	75,563	604,501.9	1,209,003.8	1.05882
BELARUS	9,491	4,746	37,964.28	75,928.56	1.16667
BELGIUM	10,941	5,471	43,765.15	131,295.46	1.07789
BRAZIL	195,210	97,605	780,840.62	2,342,521.85	0.98438
BULGARIA	7,389	3,695	29,556.7	59,113.4	1.2
CANADA	34,126	17,063	136,504.96	409,514.88	1.2494
CAPE VERDE	488	244	1,950.4	3,900.81	0.84211
CHILE	17,151	8,575	68,603.04	205,809.12	0.98308
CHINA	1,359,821	679,911	5,438,285.86	10,878,571.72	1.0989
COLOMBIA	46,445	23,222	185,779.19	557,337.58	1
CROATIA	4,338	2,169	17,352.11	52,056.32	1.29412
CZECH REPUBLIC	10,554	5,277	42,214.8	84,429.61	1.21053
DENMARK	5,551	2,775	22,203.64	66,611.51	1.25893
ECUADOR	15,001	7,501	60,004.29	180,012.86	1.21256
EGYPT	78,076	39,038	312,302.82	624,605.64	1.1148
ESTONIA	1,299	649	5,194.13	15,582.4	1.03571
FINLAND	5,368	2,684	21,470.77	64,412.32	1.28205
FRANCE	63,231	31,615	252,923.46	758,770.39	1.06356
GEORGIA	4,389	2,194	17,554.7	35,109.39	1.44444
GERMANY	83,017	41,509	332,069.62	996,208.85	1.15873
GREECE	11,110	5,555	44,440	133,319.99	1.06934
GREENLAND	57	28	226.18	678.55	1.03089
GUAM	159	80	637.76	1,913.28	1.2906
GUYANA	786	393	3,144.5	9,433.51	1.04706
HUNGARY	10,015	5,007	40,058.53	120,175.6	1.15278
INDIA	1,205,625	602,812	4,822,498.59	9,644,997.18	1.1784
INDONESIA	240,676	120,338	962,705.94	1,925,411.88	0.95487
IRAN (ISLAMIC REPUBLIC OF)	74,462	37,231	297,849.26	595,698.51	1.23404
IRAQ	30,962	15,481	123,849.52	247,699.04	1.06559
IRELAND	4,468	2,234	17,870.24	53,610.73	1.32055
ISRAEL	7,420	3,710	29,681.47	59,362.94	1.18321
ITALY	60,509	30,254	242,035.91	726,107.74	1.07913
JAPAN	127,353	63,676	509,411.33	1,018,822.66	1.11819
KAZAKHSTAN	15,921	7,961	63,684.51	127,369.02	1.17195
KENYA	40,909	20,455	163,636.78	327,273.55	1.16707
KUWAIT	2,992	1,496	11,966.32	23,932.64	1.18012
LATVIA	2,091	1,045	8,362.08	25,086.23	1.19867
LIBERIA	3,958	1,979	15,831.96	31,663.92	1.30435
LITHUANIA	3,068	1,534	12,273.83	36,821.48	1.25926
MALAYSIA	28,276	14,138	113,103.34	226,206.68	1.02848
MARSHALL ISLANDS	52	26	209.71	629.14	1
MEXICO	117,886	58,943	471,545.62	1,414,636.85	0.98066
MONGOLIA	2,713	1,356	10,850.95	21,701.9	1.12903
MOROCCO	31,642	15,821	126,569.44	253,138.88	1.29629

Pull the data into Libreoffice spreadsheet

Step 1: Install the ODBC Driver According to the platform

Step 2: Create a new database connection in Libreoffice Calc

Resources

High Level Presentation - https://speakerdeck.com/kpeeples/hortonworks-and-red-hat-proof-of-technology

Hortonworks Sandbox - http://hortonworks.com/products/hortonworks-sandbox/#install

Videos:

http://vimeo.com/user16928011/hortonworksusecase1short
http://vimeo.com/user16928011/hortonworksusecase1long

Source :: Github (Data Virtualization)

https://github.com/DataVirtualizationByExample/HortonworksUseCase1