



# Data Science And Big Data Analytics Course

## Copyright

Copyright © 1996, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012 2013, 2104 EMC Corporation. All Rights Reserved. EMC believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

THE INFORMATION IN THIS PUBLICATION IS PROVIDED “AS IS.” EMC CORPORATION MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WITH RESPECT TO THE INFORMATION IN THIS PUBLICATION, AND SPECIFICALLY DISCLAIMS IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Use, copying, and distribution of any EMC software described in this publication requires an applicable software license.

EMC2, EMC, Data Domain, RSA, EMC Centera, EMC ControlCenter, EMC LifeLine, EMC OnCourse, EMC Proven, EMC Snap, EMC SourceOne, EMC Storage Administrator, Acartus, Access Logix, AdvantEdge, AlphaStor, ApplicationXtender, ArchiveXtender, Atmos, Authentica, Authentic Problems, Automated Resource Manager, AutoStart, AutoSwap, AVALONidm, Avamar, Captiva, Catalog Solution, C-Clip, Celerra, Celerra Replicator, Centera, CenterStage, CentraStar, ClaimPack, ClaimsEditor, CLARiiON, ClientPak, Codebook Correlation Technology, Common Information Model, Configuration Intelligence, Configuresoft, Connectrix, CopyCross, CopyPoint, Dantz, DatabaseXtender, Direct Matrix Architecture, DiskXtender, DiskXtender 2000, Document Sciences, Documentum, elnput, E-Lab, EmailXaminer, EmailXtender, Enginuity, eRoom, Event Explorer, FarPoint, FirstPass, FLARE, FormWare, Geosynchrony, Global File Virtualization, Graphic Visualization, Greenplum, HighRoad, HomeBase, InfoMover, Infoscape, Infra, InputAccel, InputAccel Express, Invista, Ionix, ISIS, Max Retriever, MediaStor, MirrorView, Navisphere, NetWorker, nLayers, OnAlert, OpenScale, PixTools, Powerlink, PowerPath, PowerSnap, QuickScan, Rainfinity, RepliCare, RepliStor, ResourcePak, Retrospect, RSA, the RSA logo, SafeLine, SAN Advisor, SAN Copy, SAN Manager, Smarts, SnapImage, SnapSure, SnapView, SRDF, StorageScope, SupportMate, SymmAPI, SymmEnabler, Symmetrix, Symmetrix DMX, Symmetrix VMAX, TimeFinder, UltraFlex, UltraPoint, UltraScale, Unisphere, VMAX, Vblock, Viewlets, Virtual Matrix, Virtual Matrix Architecture, Virtual Provisioning, VisualSAN, VisualSRM, Voyence, VPLEX, VSAM-Assist, WebXtender, xPression, xPresso, YottaYotta, the EMC logo, and where information lives, are registered trademarks or trademarks of EMC Corporation in the United States and other countries.

All other trademarks used herein are the property of their respective owners.

© Copyright 2014 EMC Corporation. All rights reserved. Published in the USA.

Revision Date: September 2014

Revision Number: MR-1CP-DSBDA .1.5

# LAB 1

## Introduction to Data Environment

✓ Lab Content:

1. Lab and environment Deployment Guide
2. Database Environment – Retail Data Exercise.
3. Database Environment – Census Data Exercise.

# 1. 1 Lab Deployment Guide

✓ In order to have full advantage of the course, please make sure you have good internet connection and a pc with the following specs:

- Dual Core x86-64 bit processor i5
- 8 GB RAM Memory min
- 80 GB of free Hard Disk space or external Hard Disk
- Windows 10 installed and updated to latest update

✓ Needed materials and softwares:

1. VM sandbox disk files (6 zipped files with total size 18.6 GB)

2. On your host machine download and install the following:

a. VMware Workstation Player:

<https://www.vmware.com/mena/products/workstation-player/workstation-player-evaluation.html>

b. Putty as terminal tool: <https://www.putty.org/>

c. WinSCP for files transformer to/from VM:

<https://winscp.net/eng/download.php>

d. R and RStudio Desktop for additional labs:

<https://cloud.r-project.org/> (base and Rtools)

<https://rstudio.com/products/rstudio/download/#download>

e. Anaconda Individual Edition for Python adoption of labs:

[https://repo.anaconda.com/archive/Anaconda3-2020.11-Windows-x86\\_64.exe](https://repo.anaconda.com/archive/Anaconda3-2020.11-Windows-x86_64.exe)

[https://repo.anaconda.com/archive/Anaconda3-2020.11-MacOSX-x86\\_64.pkg](https://repo.anaconda.com/archive/Anaconda3-2020.11-MacOSX-x86_64.pkg)

- ✓ The Vmware sandbox contains:



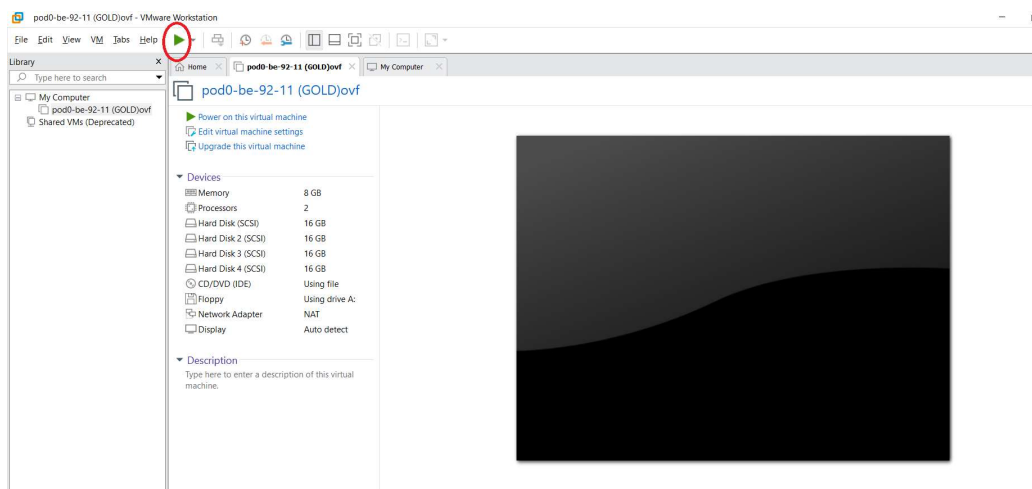
1. CentOS: an operating system from Linux distribution.
2. Greenplum: database engine; an open-source massively parallel data platform for analytics, machine learning and AI.
3. MadLib on greenplum: is an open-source library for scalable in-database analytics, Big Data Machine Learning in SQL. MADlib provides data-parallel implementations of mathematical, statistical and machine-learning methods for structured and unstructured data.
4. Hadoop: is a free java based programming framework that manages Big Data storage in a distributed way and processes it parallelly. It's big data storage and data processing platform. It runs on clusters of low cost commodity hardware.

NP: GreenPlum and Hadoop both manages data at a scale i.e. BigData. However, GreenPlum is still based on relational model using tables to store data and native SQL. Hadoop on the other hand uses files to store data and needs APIs programming to handle data processing. You can add additional components atop Hadoop to use SQL like Hive.

5. R & RStudio.

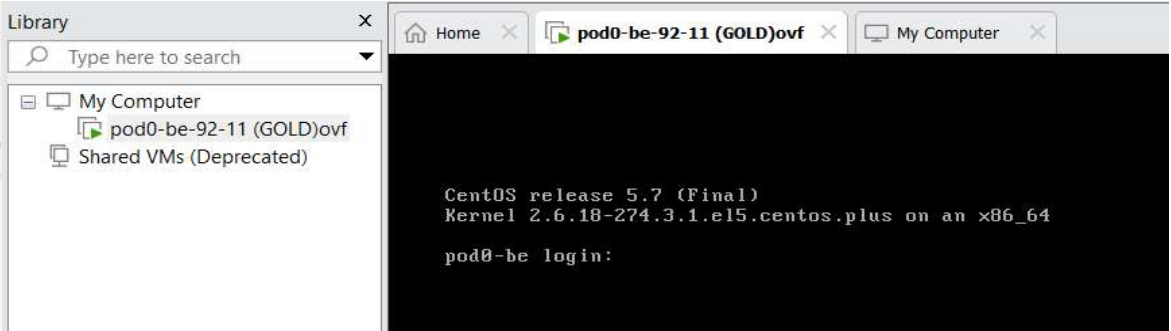
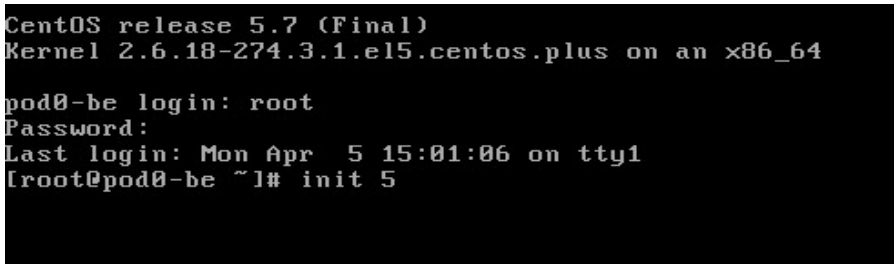
✓ Deploying the Pre-Installed VM sandbox steps:

1. Unzip the 6 files all in the same folder (folder1).
2. Make a new empty folder (folder2) in any disk drive that contains at least 80 GB free space.
3. Open the VMware Player, from the menu bar choose file → open → browse for the folder contains your unzipped files (folder1) and select the file “pod0-be-92-11 (GOLD)ovf” and click open.
4. “Import Virtual Machine” wizard will pop up, in the “Storage path for the new virtual machine”; browse for (folder2) that you created in step 2. Then click import. (It will take few time, be patient)
5. After finishing, it will look like this



6. To start the VM press the play button. Wait for loading to finish (takes few minutes) and finally will ask you for login credentials.

## 1. 2 Database Environment – Retail Data

Step	Action
1	 <p>Login with VM Administrator Account:</p> <p>Username: root                      password: gpsneroot</p>
	<p>After login with the username and password (invisible), your virtual machine is now powered on and on the command line mode, to convert to the GUI mode type this command: init 5</p> 
	<p>GUI Login screen will appear, to login as admin type:</p> <p>Username: gpadmin                      password: changeme</p>
	<p>Currently you are logged in as gpadmin and you have administrative access to the Greenplum Database Environment, in which you will be working. You must first verify if the database up and running. Right click on your desktop and choose “Open Terminal”</p>





Command window will open, type following commands:

1. Type: `gpstate`
2. Review the output; you should be able to see that the database is active with the following output. Please note that because of the large output size we only show selected lines and that your configuration details may slightly differ from the one below.

```
[INFO]:-Starting gpstate with args:
[INFO]:-local Greenplum Version: 'postgres (Greenplum Database)
4.1.1.1 build 1'
[INFO]:-Obtaining Segment details from master...
[INFO]:-Gathering data from segments...
[INFO]:-Greenplum instance status summary
[INFO]:-----
[INFO]:-   Master instance                = Active
[INFO]:-   Master standby                = No master standby
configured
...
[INFO]:-   Total primary segments                = 2
[INFO]:-   Total primary segment valid (at master) = 2
[INFO]:-   Total primary segment failures (at master) = 0
...
[INFO]:-   Mirrors not configured on this array
[INFO]:-----
```

Step	Action
2	<p>Now you're ready to open a PSQL session and check all available databases.</p> <p>Refer to the <i>PSQL Commands – Quick Reference</i>, located in your <b>Student Resource Guide Appendix</b>, for the PSQL meta commands.</p> <p><b>Note:</b> PSQL meta commands start with a backslash (\). To review all available meta commands type backslash and question mark (\?).</p> <p>To review all available databases in your environment:</p> <ol style="list-style-type: none"> <li>1. Type: <code>psql</code> This will open a new PSQL session to the default database.</li> <li>2. Next type: <code>\l</code> (<b>lowercase "L" for list</b>) Notice a list of databases and record databases named "training*".</li> </ol> <p><b>Note:</b> Another way of listing all available databases (without opening a PSQL session) is to call PSQL executable with parameter (-l): <code>psql -l</code></p>
3	<p><b><u>Connect to the training1 database:</u></b></p> <ol style="list-style-type: none"> <li>1. At the PSQL prompt type : <code>\c training1</code> at the OS level prompt type: <code>psql training1</code></li> </ol> <p>To see the schemas you have in this database:</p> <ol style="list-style-type: none"> <li>2. Type: <code>\dn</code> <ul style="list-style-type: none"> <li>• You should see "ddemo" schema, listed.</li> <li>• You should also ensure that this schema is included in the search path.</li> </ul> </li> <li>3. Execute your first PSQL command, type:    <code>SET search_path TO ddemo, public;</code> </li> </ol> <p><b>Note:</b> PSQL commands are terminated with a semi-colon- ";"</p>

Step	Action						
4	<p>You can now view the tables in this database.</p> <ol style="list-style-type: none"><li>1. Type: <code>\dt</code></li><li>2. Record the number of tables in the database: _____</li><li>3. Locate the table, "customers_dim".</li><li>4. Review the column descriptions for this table:</li><li>5. Type: <code>\d+ customers_dim</code></li><li>6. Record the column descriptions, their types and column name(s) by which the table is distributed (aka: the distribution key):</li></ol> <table><thead><tr><th>Column Descriptions</th><th>Type</th><th>Distribution Key Column(s)</th></tr></thead><tbody><tr><td></td><td></td><td></td></tr></tbody></table>	Column Descriptions	Type	Distribution Key Column(s)			
Column Descriptions	Type	Distribution Key Column(s)					
5	<p><b><u>Analyze the gender distribution of the customer base:</u></b></p> <ol style="list-style-type: none"><li>1. To locate the number of males and females type:  <code>SELECT gender,count(*) FROM customers_dim GROUP BY gender;</code></li><li>2. Record the number of female customers: _____</li><li>3. Record the number of male customers: _____</li><li>4. Record the total number of customers: _____</li></ol>						

Step	Action
6	<p>1. Using PSQL, generate a report on the average spending by gender, Type:</p> <pre> SELECT     c.gender   , AVG(o.item_price) AS avg_price FROM     ddemo.order_lineitems AS o JOIN     ddemo.customers_dim AS c   ON o.customer_id = c.customer_id GROUP BY c.gender ; </pre> <p><b>Note:</b> You can find this code in the LAB01 directory. This script can be executed using the following command from the OS prompt:</p> <p>2. To exit the PSQL environment, use the following meta command, type:</p> <pre>\q</pre> <p>You are now at the OS prompt.</p> <p>3. To execute the SQL script type at the OS prompt:</p> <pre>cd LAB01 psql -d training1 -f lab1p1step6.sql</pre> <p><b>Note 1:</b> In the <i>psql</i> command above option “-d” specifies the database name to connect to (“training1”). This is equivalent to specifying <i>dbname</i> as the first <b>non-option argument</b> on the command line. As a convention we have used the option “-d” throughout this document. However <i>dbname</i> can be specified without option “-d” as long as it is the first argument of the <i>psql</i> command.</p> <p><b>Note 2:</b> This query may take some time to execute as it is processing a million rows of data.</p> <p>4. Record the average expenditures by gender:</p> <p>Male : _____ Female: _____</p>

Step	Action																		
7	<p>Use the script, “lab1p1step7”, with the appropriate modifications to list the top five product categories ordered by men and women.</p> <table><tr><td></td><td><i>Men</i></td><td><i>Women</i></td></tr><tr><td>1</td><td></td><td></td></tr><tr><td>2</td><td></td><td></td></tr><tr><td>3</td><td></td><td></td></tr><tr><td>4</td><td></td><td></td></tr><tr><td>5</td><td></td><td></td></tr></table>		<i>Men</i>	<i>Women</i>	1			2			3			4			5		
	<i>Men</i>	<i>Women</i>																	
1																			
2																			
3																			
4																			
5																			

### 1.3 Database Environment-Census Data

Step	Action
1	Follow the steps detailed in, Lab 1 - Data Set 1, to connect to and inspect another database “training2”.
2	Record the tables in database (Schema – Public)“training2”
3	Describe the type of data in the database.
4	Record the number of rows in each table.
5	<p><b><u>Data Preparation &amp; Cleanup – 1:</u></b></p> <p>(Scenario) You realize that the Intern who loaded the “housing” data has copied records into the table twice. Each different row is represented by a unique combination of “serialno” and “state” columns.</p> <p>1. Execute the following code:</p> <pre> SELECT     SUM(c) AS total_records   , SUM(CASE WHEN c&gt;1 THEN c-1 ELSE 0 END) AS total_dupes   , COUNT(*) AS total_uniques FROM (     SELECT         COUNT(*) AS c     FROM         housing     GROUP BY         serialno         , state     ) AS dupes ; </pre> <p><b>Note:</b> This code is also available at,</p> <p><b><u>/home/gpadmin/LAB01/countdupes.sql,</u></b></p> <p>2. Record the total number of records in the table: _____</p> <p>3. Record the total number of duplicate records: _____</p> <p>4. Record the total number of unique records: _____</p>

Step	Action
6	<p><b><u>Data Preparation &amp; Cleanup – 2:</u></b></p> <p>To prepare and clean the data you need to create a “housing_nodupes” table. Make sure that you are in the PSQL environment if you have previously exited to the OS command line.</p> <ol style="list-style-type: none"> <li>1. Check to see if a table already exists with the name (“housing_nodupes”). Type <code>\dt</code>  Note: the command <code>\dt</code> will list all tables in the database. <code>\dt public.*</code> will list all tables in the public schema.</li> <li>2. If this table already exists execute the following SQL statement:  <pre>DROP TABLE IF EXISTS housing_nodupes;</pre></li> <li>3. Execute the following SQL statement:  <pre>CREATE TABLE housing_nodupes AS SELECT DISTINCT ON (serialno, state) * FROM housing DISTRIBUTED BY (serialno, state) ;</pre> <p><b>Note:</b> This code is also available at, <code>/home/gpadmin/LAB01/lab1p2step6.sql</code></p></li> <li>4. Repeat the queries in Step 5 (previous step) to ensure that there are no duplicate records in the housing_nodupes table.</li> </ol>

Step	Action
7	<p><b><u>Basic Analytics Using the “Housing” Data:</u></b></p> <ol style="list-style-type: none"> <li>Execute the following SQL statement to calculate correlation between household income and number of rooms: <pre> SELECT     corr(hinc, rooms) FROM     housing_nodupes WHERE     state = 25 ; </pre> </li> <li>Record your result:</li> <li>Using the fips table, determine which U.S. state corresponds to 25. <pre> SELECT * FROM fips WHERE code = 25; </pre> </li> <li>Execute the following SQL statement to calculate the R-squared of the regression line of household income and number of rooms: <pre> SELECT     regr_r2(hinc, rooms) FROM     housing_nodupes WHERE     state = 25 ; </pre> </li> <li>Record your result:</li> </ol>



Step	Action
8	<p><b><u>Prepare “Housing” Data for Subsequent Analytic Exercises:</u></b></p> <p>You need to prepare data from the, “housing_nodupes” and “persons” tables, for subsequent analysis with “R” in the next module.</p> <ol style="list-style-type: none"> <li>Run the following commands and SQL query to move (pipe) the results into a text file  <b>Note:</b> Use the meta commands from the PSQL Meta Command-Quick Reference to render your output to a file and remove the white spaces (formatting) <pre> \a \o lab1_01.txt SELECT     serialno , hinc , rooms FROM     housing_nodupes WHERE     hinc &gt; 0     AND state = 25 ; </pre> <b>Note:</b> The SQL query is also available at the following location:  /home/gpadmin/LAB01/lab1p2step8.sql </li> <li>Alternatively you can execute the following command from the OS prompt: <pre>psql -d training2 -f lab1p2step8.sql</pre> Now, your data is ready for the lab exercise in the next module. </li> <li>Remove the summary line at the end of the output file lab1_01.txt using WinSCP or a Linux editor of your choice such as VI.</li> </ol>

Step	Action																		
9	<p><b><u>Prepare “Persons” Data for Subsequent Analytic Exercises:</u></b></p> <p>Prepare a summary table with the number of people by race and by education level.</p> <p><b>Note:</b> Use the following Races: White, Black, American Indian/Alaska Native, Asian, Hawaiian /Pacific Islander, and Others.</p> <div><div>(white) White,</div><div>(black) Black,</div><div>(aian) American_Indian_Alaska_native,</div><div>(asian) Asian,</div><div>(nhpi) Hawaii_pacific_islander,</div><div>(other) Others</div></div> <p><b><u>Use the following Education Levels:</u></b></p> <table><tr><td>01. No schooling completed</td><td>06. 10th grade</td><td>11. One or more years of college, no degree</td></tr><tr><td>02. Nursery school to 4th grade</td><td>07. 11th grade</td><td>12. Associate degree</td></tr><tr><td>03. 5th grade or 6th grade</td><td>08. 12th grade, no diploma</td><td>13. Bachelor’s degree</td></tr><tr><td>04. 7th grade or 8th grade</td><td>09. High school graduate</td><td>14. Master’s degree</td></tr><tr><td>05. 9th grade</td><td>10. Some college, but less than 1 year</td><td>15. Professional degree</td></tr><tr><td></td><td></td><td>16. Doctorate degree</td></tr></table> <p>Create a table with columns for Races and rows for Educational Level. (The cells denote the number of “persons” for each category.) Prepare a text file with headers to use in the next module. SQL code necessary for this task is presented below:</p> <pre>\a \o lab1_02.txt SELECT     educ AS Education_Level     , SUM(white) AS White     , SUM(black) AS Black     , SUM(aian) AS American_Indian_Alaska_Native     , SUM(asian) AS Asian     , SUM(nhpi) AS Hawaii_Pacific_Islander     , SUM(other) AS Others FROM     persons WHERE     age &gt; 17     AND educ &gt; 0 GROUP BY educ ORDER BY educ ;</pre>	01. No schooling completed	06. 10th grade	11. One or more years of college, no degree	02. Nursery school to 4th grade	07. 11th grade	12. Associate degree	03. 5th grade or 6th grade	08. 12th grade, no diploma	13. Bachelor’s degree	04. 7th grade or 8th grade	09. High school graduate	14. Master’s degree	05. 9th grade	10. Some college, but less than 1 year	15. Professional degree			16. Doctorate degree
01. No schooling completed	06. 10th grade	11. One or more years of college, no degree																	
02. Nursery school to 4th grade	07. 11th grade	12. Associate degree																	
03. 5th grade or 6th grade	08. 12th grade, no diploma	13. Bachelor’s degree																	
04. 7th grade or 8th grade	09. High school graduate	14. Master’s degree																	
05. 9th grade	10. Some college, but less than 1 year	15. Professional degree																	
		16. Doctorate degree																	

Step	Action
10	<p>The code in step 9 is also available at the following location: /home/gpadmin/LAB01/lab1p2step9.sql</p> <p>Execute the following command from the OS prompt:</p> <pre>psql -d training2 -f lab1p2step9.sql</pre> <p>Remove the last “summary” line as you did in Step 8 and prepare the file “lab1_02.txt” for the lab exercise in the next module.</p>

*End of Lab Exercise*