

# Greenplum Architecture, Administration, and Implementation Lab Guide

March 2016

## Copyright

Copyright © 1996, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015 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, Enginity, 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, RepliStar, 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 2015 EMC Corporation. All rights reserved. Published in the USA.

Revision Date: May 2015

Revision Number: MR-1CN-GRNADM.4.3.4

## Copyright

Copyright © 2016 Pivotal Software, Inc. All rights reserved. This manual and its accompanying materials are protected by U.S. and international copyright and intellectual property laws.

Pivotal products are covered by one or more patents listed at <http://www.pivotal.io/patents>.

Pivotal is a registered trademark or trademark of Pivotal Software, Inc. in the United States and/or other jurisdictions. All other marks and names mentioned herein may be trademarks of their respective companies. The training material is provided "as is," and all express or implied conditions, representations, and warranties, including any implied warranty of merchantability, fitness for a particular purpose or noninfringement, are disclaimed, even if Pivotal Software, Inc., has been advised of the possibility of such claims. This training material is designed to support an instructor-led training course and is intended to be used for reference purposes in conjunction with the instructor-led training course. The training material is not a standalone training tool. Use of the training material for self-study without class attendance is not recommended.

These materials and the computer programs to which it relates are the property of, and embody trade secrets and confidential information proprietary to, Pivotal Software, Inc., and may not be reproduced, copied, disclosed, transferred, adapted or modified without the express written approval of Pivotal Software, Inc.

## Document Revision History

Rev #	File Name	Date
1	MR-1CN-GRNADM_LabGuide.docx	03/11
2	MR-1CN-GRNADM_v2_LabGuide.docx	05/11
3	4326_MR-1CN-GRNADM_LabGuide.docx	02/12
4	MR-1CN-GRNADM_v2.1_LabGuide	04/12
5	MR-1CN-GRNADM_v2.2_LabGuide	11/13
6	MR-1CN-GRNADM_4.3.4_LabGuide	05/15
7	MR-1CN-GRNADM_4.3.4_LabGuide modifications to support Pivotal lab environment	03/02/16

## Table of Contents

<b>COPYRIGHT.....</b>	<b>2</b>
<b>DOCUMENT REVISION HISTORY.....</b>	<b>4</b>
<b>TABLE OF CONTENTS .....</b>	<b>5</b>
<b>INTRODUCTION TO LAB EXERCISES .....</b>	<b>8</b>
<b>LAB 1: GREENPLUM PRODUCT OVERVIEW .....</b>	<b>12</b>
LAB 1: TASK – GREENPLUM PRODUCT REVIEW.....	13
<b>LAB 2: SYSTEMS PREPARATION AND VERIFICATION .....</b>	<b>16</b>
LAB 2: TASK 1 – INSTALL THE PIVOTAL GREENPLUM DATABASE BINARIES .....	17
LAB 2: TASK 2 – INSTALL AND CONFIGURE GREENPLUM ON ALL HOSTS .....	26
LAB 2: TASK 3 – CREATING THE DATA STORAGE AREAS.....	30
LAB 2: TASK 4 – SYNCHRONIZING SYSTEM CLOCKS.....	32
LAB 2: TASK 5 – PERFORM SYSTEM VERIFICATION TESTS.....	39
<b>LAB 3: PIVOTAL GREENPLUM DATABASE INITIALIZATION .....</b>	<b>49</b>
LAB 3: TASK 1 – INITIALIZE THE DATABASE .....	50
LAB 3: TASK 2 – DELETE THE DATABASE AND INITIALIZE A NEW DATABASE WITH ONE COMMAND	69
<b>LAB 4: USING THE PSQL CLIENT AND GREENPLUM UTILITIES .....</b>	<b>78</b>
LAB 4: TASK 1 – ACCESS THE DATABASE WITH PSQTL .....	79
LAB 4: TASK 2 – INSTALL AND CONFIGURE PIVOTAL GREENPLUM COMMAND CENTER .....	86
LAB 4: TASK 3 – NAVIGATING PIVOTAL GREENPLUM COMMAND CENTER.....	97
<b>LAB 5: GREENPLUM DATABASE SERVER CONFIGURATION .....</b>	<b>103</b>
LAB 5: TASK 1 – SET CONFIGURATION PARAMETERS .....	104
LAB 5: TASK 2 – CONFIGURE HOST-BASED AUTHENTICATION .....	112
<b>LAB 6: DATABASE INTERNALS .....</b>	<b>115</b>
LAB 6: TASK 1 – EXPLORE DATABASE FILES ON DISK.....	116
LAB 6: TASK 2 – EXAMINE SERVER PROCESSES .....	121
<b>LAB 7: DATA DEFINITION LANGUAGE.....</b>	<b>124</b>
LAB 7: TASK 1 – CREATE DATABASES AND SCHEMAS .....	125
LAB 7: TASK 2 – CREATE TABLES .....	129
LAB 7: TASK 3 – VIEW INDEXES AND SCHEMAS.....	131
<b>LAB 8: DATA MANIPULATION LANGUAGE AND DATA QUERY.....</b>	<b>134</b>
LAB 8: TASK 1 – INSERT, UPDATE, AND DELETE RECORDS.....	135
LAB 8: TASK 2 – ACCESSING DATA TO GENERATE REPORTS.....	137
<b>LAB 9: ROLES, PRIVILEGES, AND RESOURCES.....</b>	<b>142</b>
LAB 9: TASK 1 – MANAGE USER AND GROUP ROLES .....	143
LAB 9: TASK 2 – MANAGE RESOURCE QUEUES.....	147
<b>LAB 10: CONTROLLING ACCESS .....</b>	<b>153</b>
LAB 10: TASK – IMPLEMENT BASIC SECURITY AT THE GROUP LEVEL.....	154
<b>LAB 11: TABLE MANAGEMENT.....</b>	<b>161</b>

LAB 11: TASK 1 – CREATING TEMPORARY TABLES.....	162
LAB 11: TASK 2 – CREATING COMPRESSED TABLES .....	165
LAB 11: TASK 3 – CREATING A COLUMN-WISE STORAGE TABLE .....	168
<b>LAB 12: DATA LOADING .....</b>	<b>175</b>
LAB 12: TASK 1 – CREATE A DIMENSION TABLE AND LOAD DATA USING COPY .....	176
LAB 12: TASK 2 – CREATE A DIMENSION TABLE AND LOAD DATA USING EXTERNAL TABLES ....	179
LAB 12: TASK 3 – CREATE A FACT TABLE AND LOAD DATA USING GPFDIST .....	182
LAB 12: TASK 4 – CREATE A WRITEABLE EXTERNAL TABLE TO LOAD DATA INTO A DIFFERENT DATABASE USING GPFDIST.....	188
LAB 12: TASK 5 – LOADING A COMPRESSED FILE USING GPFDIST .....	193
<b>LAB 13: TABLE PARTITIONING .....</b>	<b>196</b>
LAB 13: TASK – CREATE AND MANAGE TABLE PARTITIONS .....	197
<b>LAB 14: MANAGING THE GREENPLUM DATABASE.....</b>	<b>210</b>
LAB 14: TASK – PERFORM SYSTEM ADMINISTRATION TASKS ON THE GREENPLUM ENVIRONMENT	211
<b>LAB 15: BACKUPS AND RESTORES .....</b>	<b>226</b>
LAB 15: TASK 1 – BACKUPS AND RESTORES .....	227
LAB 15: TASK 2 – INCREMENTAL BACKUP AND RESTORE .....	237
LAB 15: TASK 3 – RECOVER FROM A FAILED MASTER (OPTIONAL) .....	247
<b>LAB 16: DATA MODELING .....</b>	<b>270</b>
LAB 16: TASK – CREATE THE DATAMART DATABASE AND OBJECTS .....	271
<b>LAB 17: PHYSICAL DESIGN DECISIONS .....</b>	<b>273</b>
LAB 17: TASK 1 – CREATE THE STORE DIMENSION .....	275
LAB 17: TASK 2 – CREATE THE COUNTRY DIMENSION.....	277
LAB 17: TASK 3 – CREATE THE CUSTOMER DIMENSION.....	279
LAB 17: TASK 4 – CREATE THE TRANSACTION FACT TABLE.....	281
LAB 17: TASK 5 – LOAD DIMENSION AND FACT DATA .....	284
<b>LAB 18: DATABASE TUNING .....</b>	<b>294</b>
LAB 18: TASK – TUNE THE DATABASE AND QUERIES.....	295
<b>LAB 19: EXPLAIN THE EXPLAIN PLAN – ANALYZING QUERIES.....</b>	<b>307</b>
LAB 19: TASK 1– CONFIGURE DATABASES IN PGADMIN III .....	308
LAB 19: TASK 2 – ANALYZE QUERIES.....	313
<b>LAB 20: IMPROVE PERFORMANCE WITH STATISTICS.....</b>	<b>320</b>
LAB 20: TASK – GATHER STATISTICS AND ANALYZE QUERIES.....	321
<b>LAB 21: INDEXING STRATEGIES .....</b>	<b>328</b>
LAB 21: TASK – CREATE INDEXES TO SUPPORT QUERIES.....	329
<b>LAB 22: ADVANCED REPORTING USING OLAP .....</b>	<b>335</b>
LAB 22: TASK – ADVANCED REPORTING USING OLAP .....	336
<b>LAB 23: POSTGRESQL FUNCTIONS.....</b>	<b>347</b>
LAB 23: TASK 1 – CREATE SQL FUNCTIONS.....	348
LAB 23: TASK 2 – CREATE PL/PGSQL FUNCTIONS.....	351

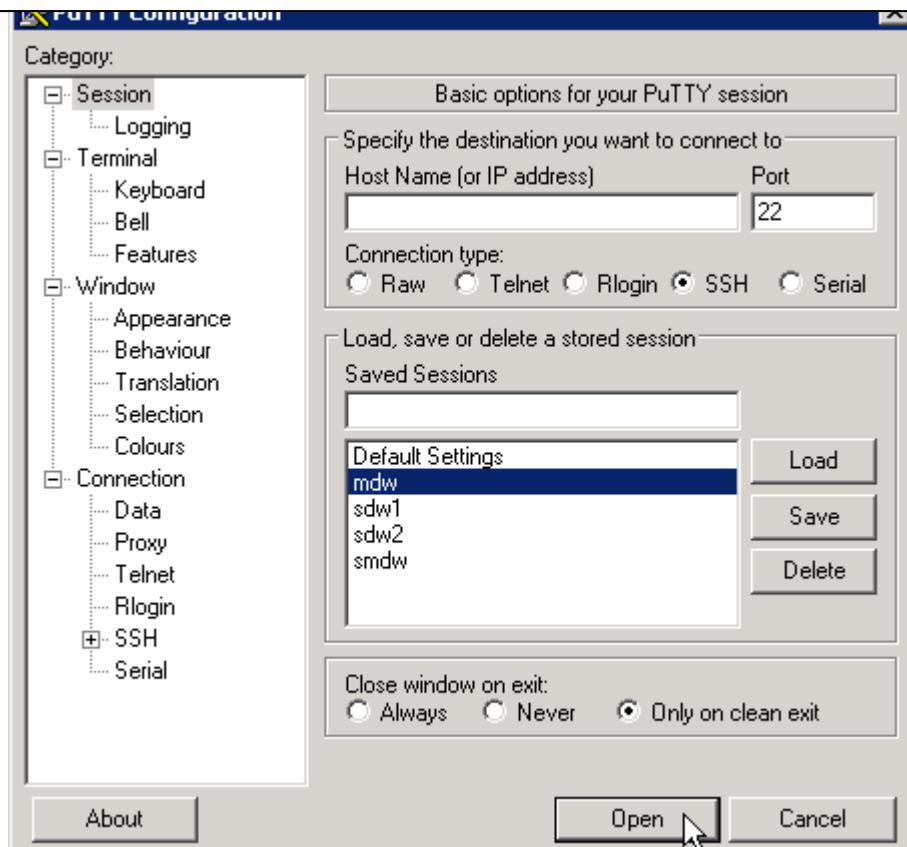


## Introduction to Lab Exercises

<b>Purpose:</b>	The lab exercises contained in this lab guide give you the opportunity to reinforce what you have learned in the Greenplum Architecture, Administration, and Implementation course. The lab exercises are designed to run in the environment provided.
<b>Tasks:</b>	<p>Students will access the lab environment by connecting to and logging in to the Virtual Data Center (VDC) environment in Lab Manager. Students will have access to their own VDC configuration using the credentials provided by the instructor.</p> <p>The virtual lab environment consists of five virtual machines with the following:</p> <ol style="list-style-type: none"><li>1. mdw (172.16.1.11) – Centos 6.5 x64 server with the Greenplum Database 4.3 software installed.</li><li>2. smdw (172.16.1.14) – Centos 6.5 x64 server acting as the standby server for the configuration.</li><li>3. sdw1 (172.16.1.12) – Centos 6.5 x64 server acting as the first segment server in the cluster.</li><li>4. sdw2 (172.16.1.13) – Centos 6.5 x64 server acting as the segment sever in the cluster.</li></ol> <p>Students will perform lab exercises connecting by PuTTY or other terminal application as necessary.</p>

## Lab Introduction

**Note:** Lab access is available in the Instructor Led Training (ILT) program. Your instructor has provided you with a .zip file named **gpdb-arch-admin-impl-4.3.4.RELEASE.zip**. Unzip this file to the root directory on your computer.

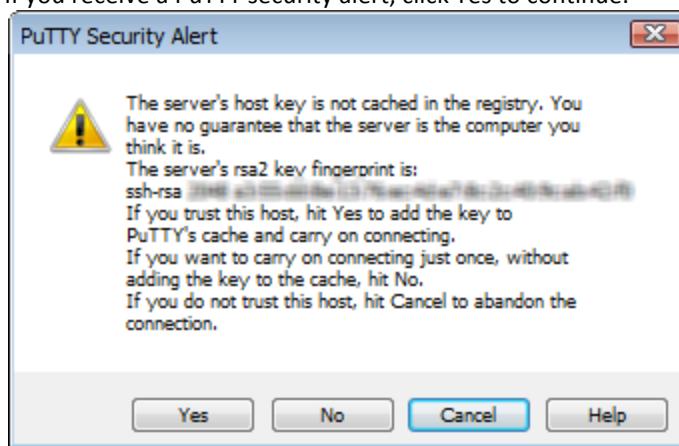


Log in with the following credentials:

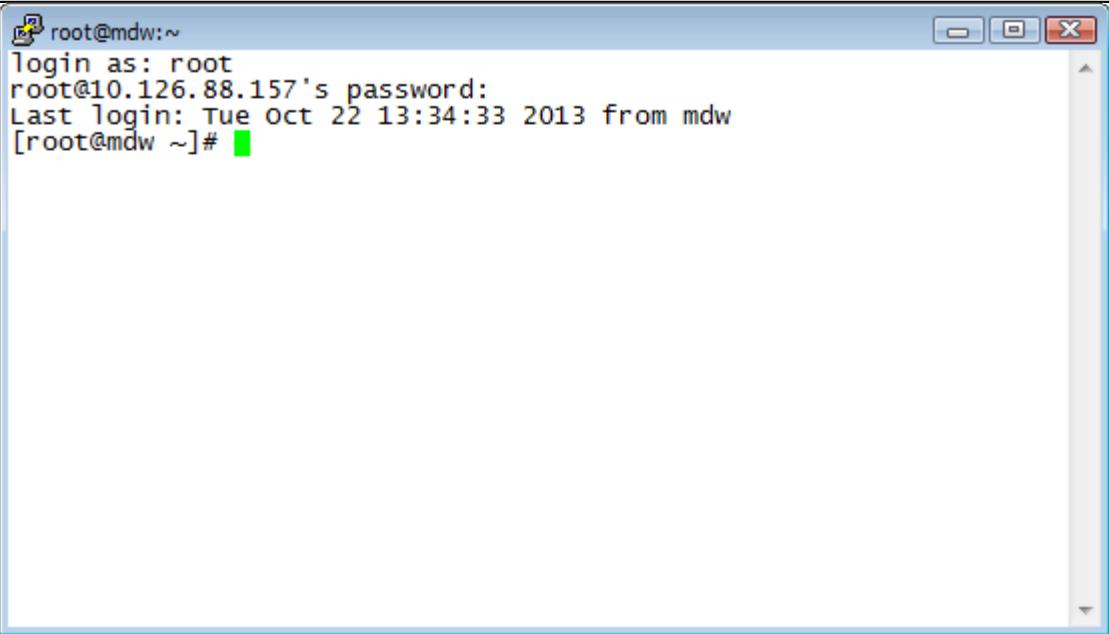
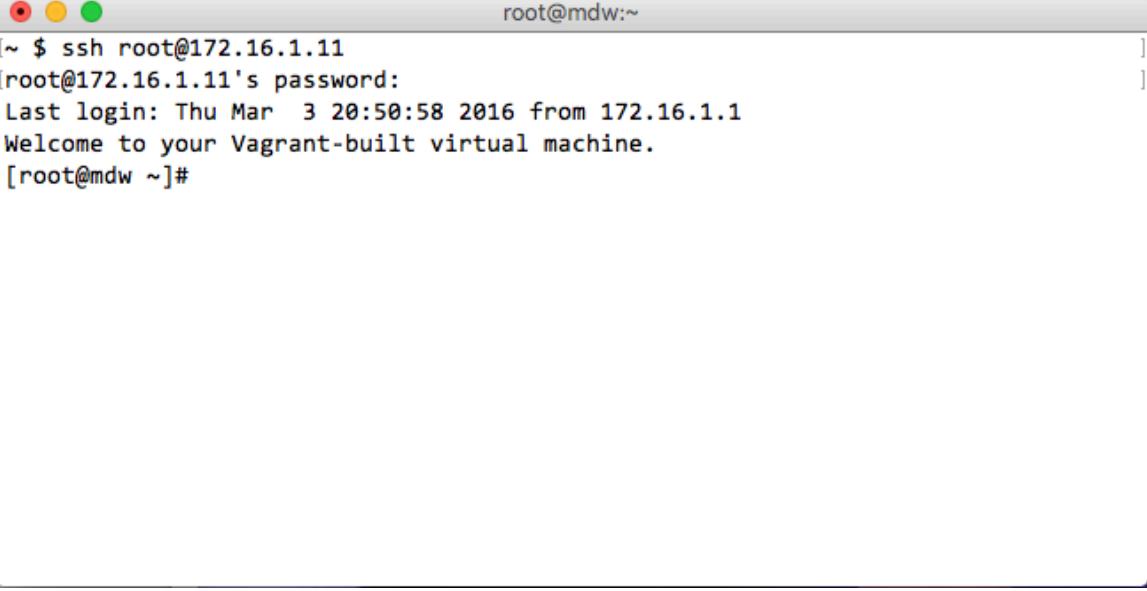
Username: **root**

Password: **Pivotal**

If you receive a PuTTY security alert, click Yes to continue.



You are now logged in to mdw.

Step	Action
	 <pre>root@mdw:~ login as: root root@10.126.88.157's password: Last login: Tue Oct 22 13:34:33 2013 from mdw [root@mdw ~]#</pre>
2.	<p>On a Mac platform open a terminal window and ssh to the master host (mdw):</p> <pre>~ \$ ssh root@172.16.1.11</pre>  <pre>[~ \$ ssh root@172.16.1.11 root@172.16.1.11's password: Last login: Thu Mar  3 20:50:58 2016 from 172.16.1.1 Welcome to your Vagrant-built virtual machine. [root@mdw ~]#</pre> <p>Log in with the following credentials:</p> <p>Username: <b>root</b>    Password: <b>Pivotal</b></p> <p>To make your ssh commands simpler from your Mac, you might consider creating aliases for the master host (mdw), the standby master host (smdw), segment host 1 (sdw1) and segment host 2 (sdw2):</p> <pre>alias mdw='root@172.16.1.11' alias smdw='root@172.16.1.14' alias sdw1='root@172.16.1.12' alias sdw2='root@172.16.1.13'</pre> <p>You will need to restart your shell.</p> <p>You can then ssh to the master (mdw) for example, using the following command:</p> <pre>~ \$ mdw</pre> <p>The ~/.bashrc file on each of the hosts has been modified to include these aliases.</p>

This page intentionally left blank

## Lab 1. Greenplum Product Overview

<b>Purpose:</b>	In this lab, you review your knowledge of Greenplum Database concepts, architecture, and components.
<b>Tasks:</b>	Students perform the following task: Provide answers to the review questions provided.
<b>References:</b>	<p>Module 1 – Greenplum Fundamental Concepts</p> <ul style="list-style-type: none"><li>• Lesson: The Basics of Data Warehousing</li><li>• Lesson: Greenplum Concepts, Features, and Benefits</li><li>• Lesson: Greenplum Architecture – Shared Nothing and MPP</li><li>• Lesson: Greenplum Product Overview</li></ul>

## Lab 1: Task – Greenplum Product Review

Step	Action
1.	What is the master instance and what is its purpose? _____ _____ _____
2.	Name three clients that can connect to a Greenplum database. _____ _____ _____
3.	What is the role of the segment instances in a Greenplum Database system? _____ _____ _____
4.	What is the purpose of mirroring in a Greenplum Database? _____ _____ _____
5.	What is the Interconnect? _____ _____ _____
6.	Where is table data stored in a Greenplum Database system? _____ _____ _____
7.	How are system management tasks performed in a Greenplum Database? _____ _____ _____
8.	How is data distributed in a Greenplum Database? _____ _____ _____

Step	Action
9.	<p>Which redundant components can you deploy in order to have a Greenplum Database system running without a single point of failure?</p> <hr/> <hr/> <hr/>

*End of Lab Exercise*

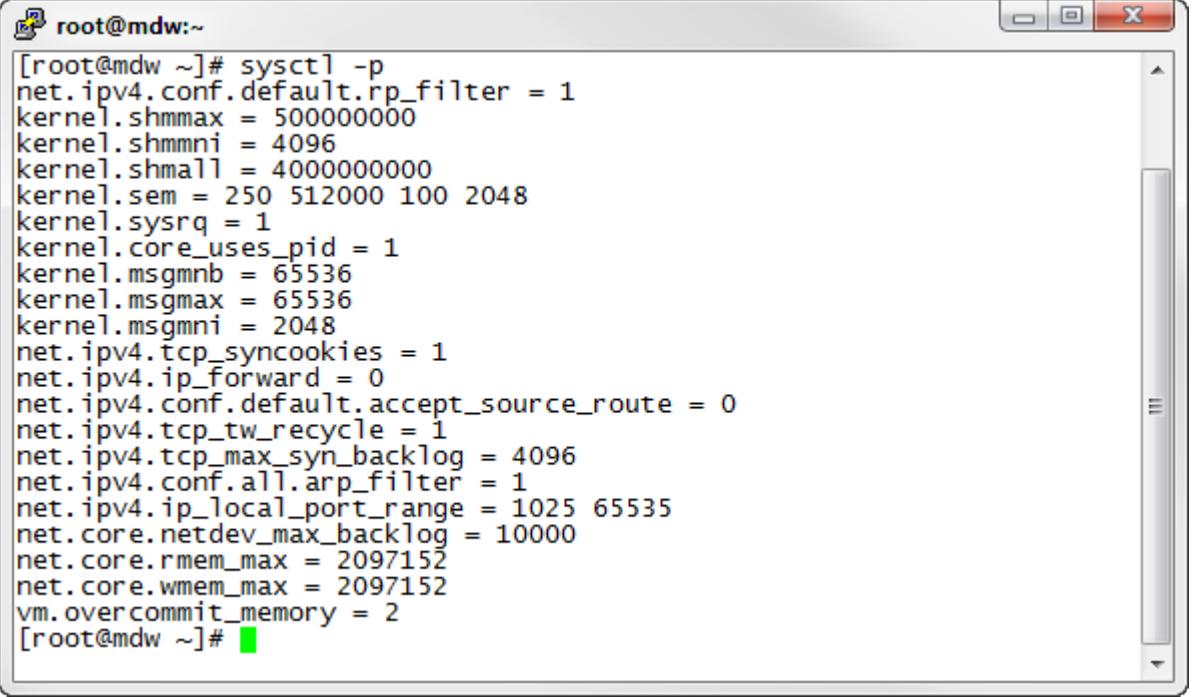
This page intentionally left blank

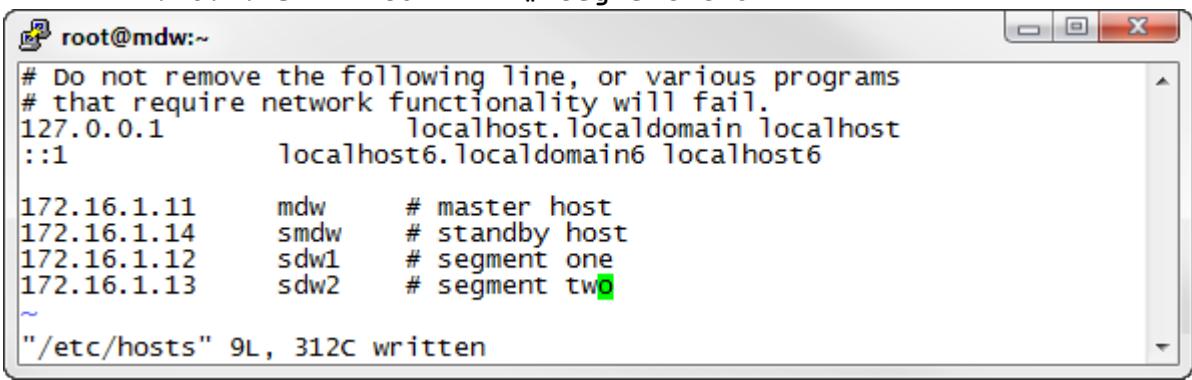
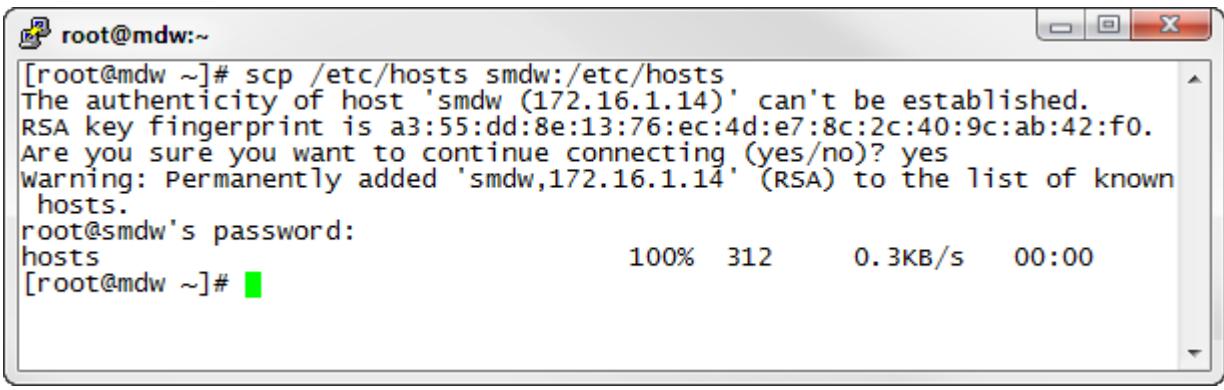
## Lab 2. Systems Preparation and Verification

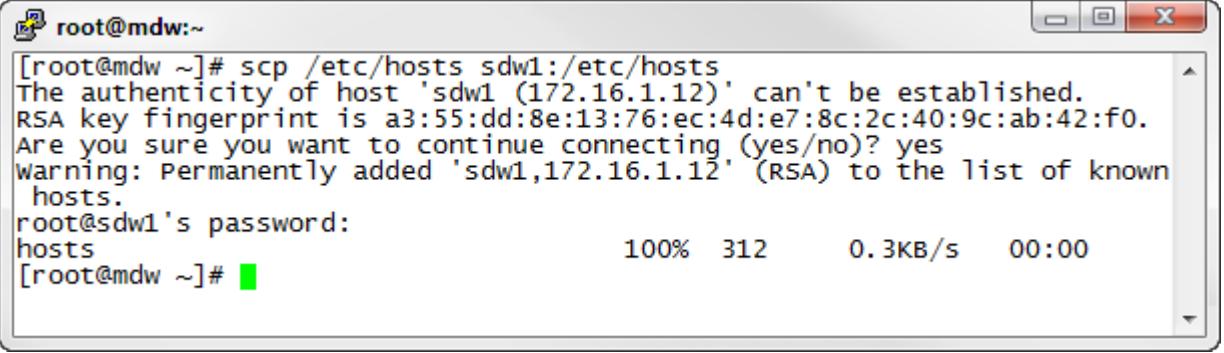
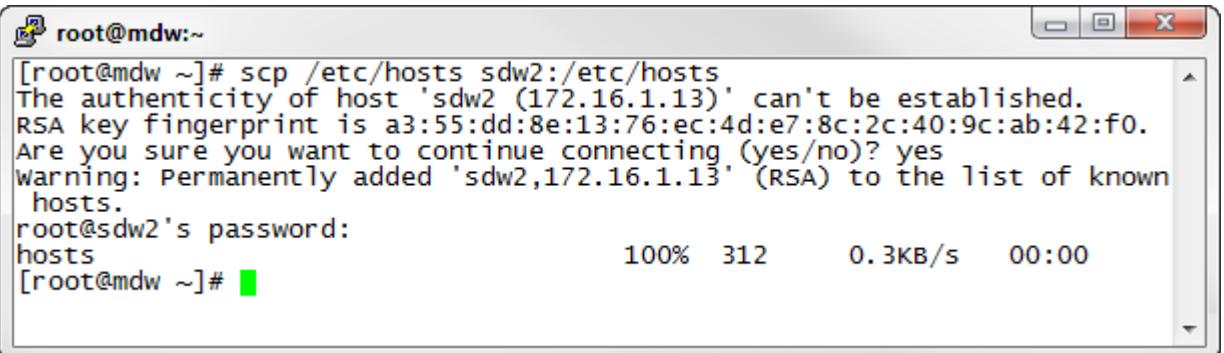
<b>Purpose:</b>	In this lab, you verify and prepare the operating system for installation and configuration of the Greenplum Database.
<b>Tasks:</b>	<p>Students perform the following tasks:</p> <ul style="list-style-type: none"><li>• Install the Greenplum Database binaries.</li><li>• Install and configure Greenplum in all hosts that comprise the Greenplum environment.</li><li>• Create the data storage areas for the Greenplum database.</li><li>• Synchronize system clocks.</li><li>• Perform system verification tasks.</li></ul>
<b>References:</b>	<p>Module 2 – Database Installation and Initialization</p> <ul style="list-style-type: none"><li>• Lesson: Systems Preparation and Verification</li></ul>

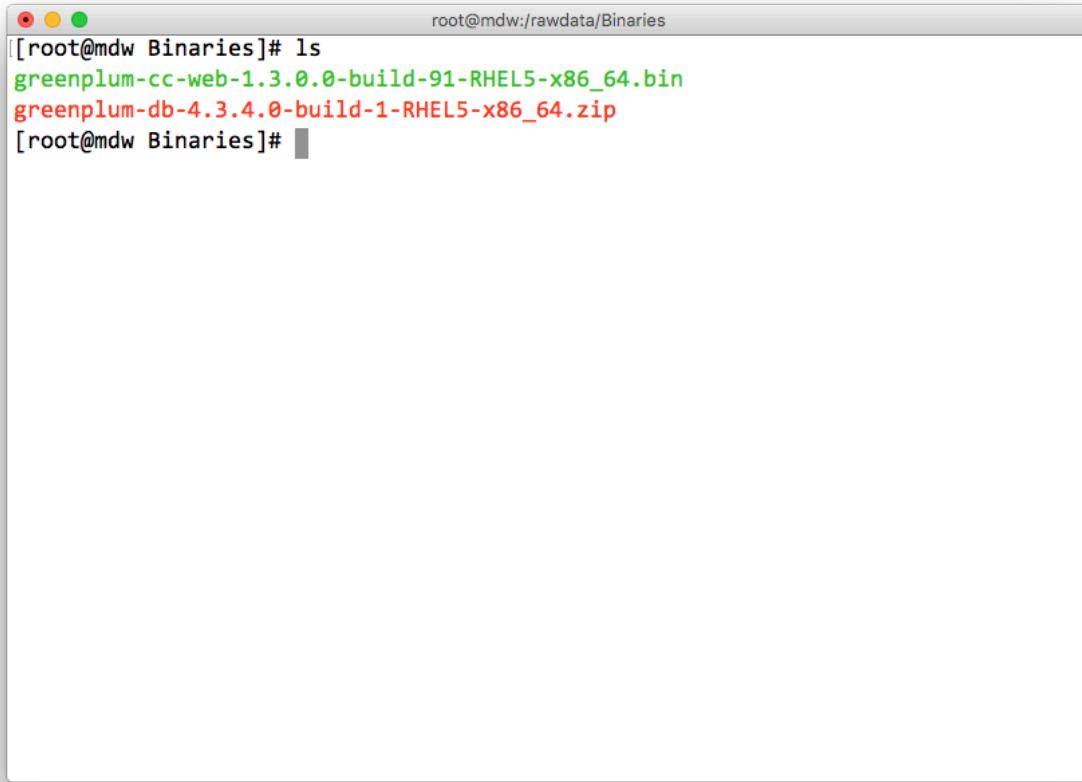
## Lab 2: Task 1 – Install the Pivotal Greenplum Database Binaries

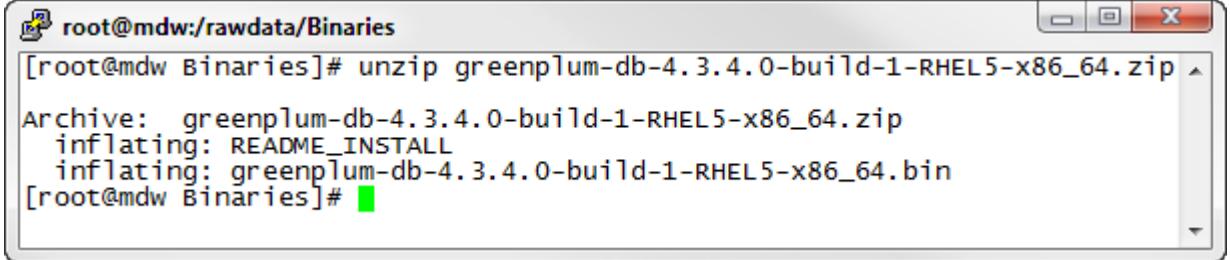
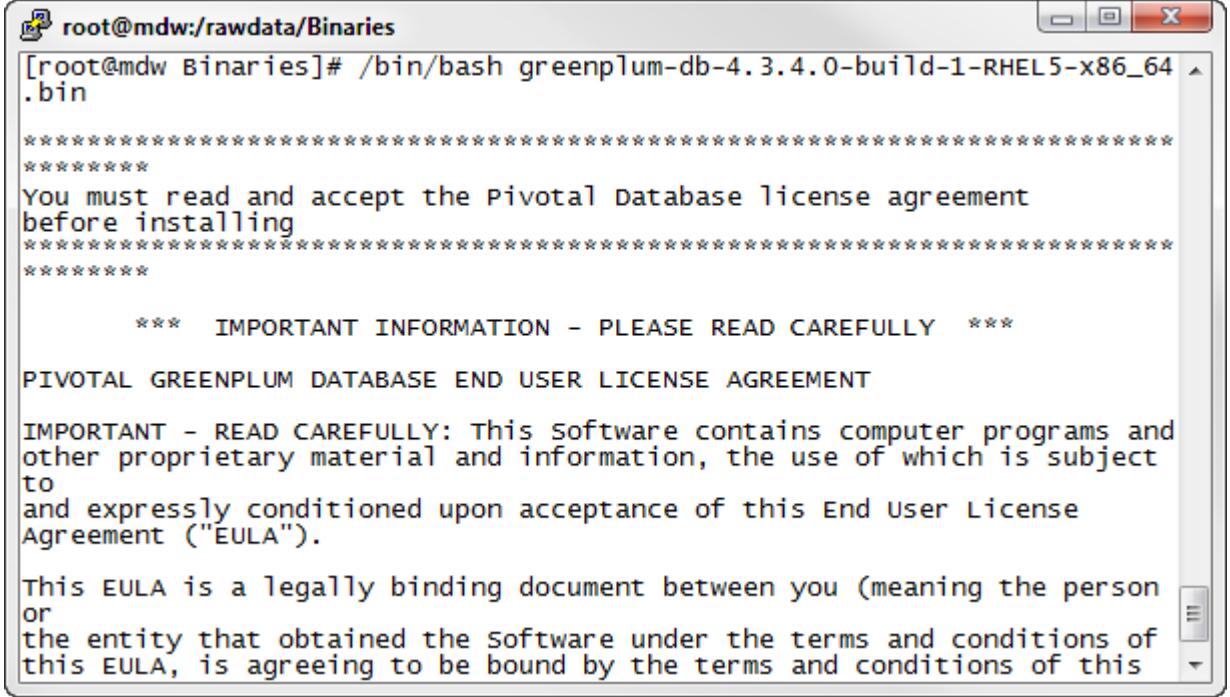
Step	Action
1.	<p>If not already connected, open a terminal session to mdw and log in with the following credentials:</p> <ul style="list-style-type: none"><li>• Login: <b>root</b></li><li>• Password: <b>Piv0tal</b></li></ul>
2.	<p>Using the vi editor, edit the <code>/etc/sysctl.conf</code> file and confirm that the parameters shown in this step correspond to the values shown. If the parameters do not exist, or the values are less than the values shown, add or modify the existing line as shown to the <code>/etc/sysctl.conf</code> file.</p> <p>You are only verifying the values of these parameters as they exist at the bottom of the file.</p> <p>Note: These parameter changes are required in your production environment to ensure better performance for the Greenplum Database.</p> <pre>[root@mdw ~]# vi /etc/sysctl.conf  kernel.shmmmax = 500000000 kernel.shmmni = 4096 kernel.shmall = 4000000000 kernel.sem = 250 512000 100 2048 kernel.sysrq = 1 kernel.core_uses_pid = 1 kernel.msgmnb = 65536 kernel.msgmax = 65536 kernel.msgmni = 2048 net.ipv4.tcp_syncookies = 1 net.ipv4.ip_forward = 0 net.ipv4.conf.default.accept_source_route = 0 net.ipv4.tcp_tw_recycle = 1 net.ipv4.tcp_max_syn_backlog = 4096 net.ipv4.conf.all.arp_filter = 1 net.ipv4.ip_local_port_range = 1025 65535 net.core.netdev_max_backlog = 10000 <b>net.core.rmem_max = 2097152</b> <b>net.core.wmem_max = 2097152</b> vm.overcommit_memory = 2</pre> <p>The settings above for <code>kernel.shmmmax</code> and <code>kernel.shmall</code> are minimum and maximum values.</p> <p><b>Note:</b> Commands that you edit or change are marked in bold. Each UNIX command typed in a terminal session will be preceded by the prompt. Do not type that as part of the command.</p> <p>If you are not familiar with the vi editor, refer to <b>Appendix A</b> in your student guide.</p>

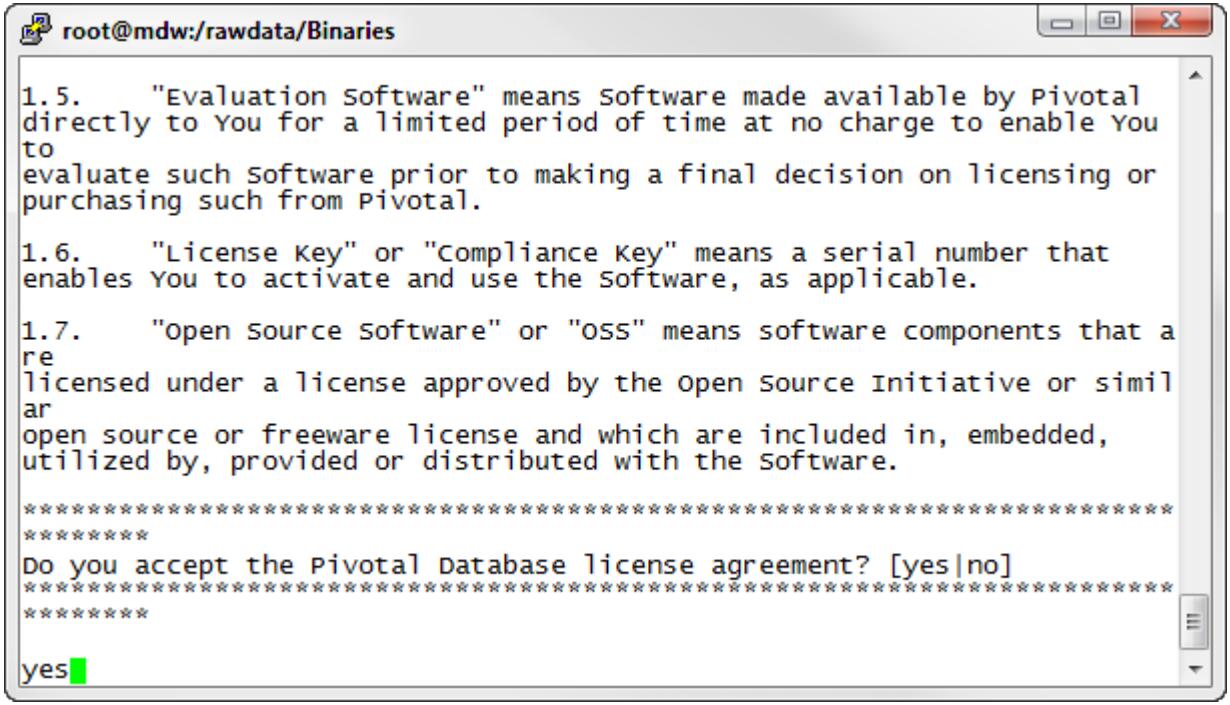
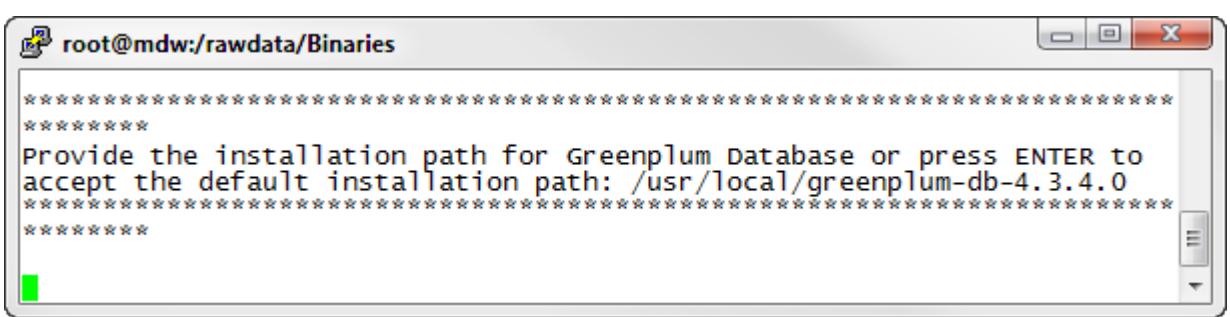
Step	Action
3.	<p>Using the <code>sysctl</code> command, incorporate any changes that you have made to <code>/etc/sysctl.conf</code> take effect immediately. This command would normally accept a file argument, but the <code>/etc/sysctl.conf</code> file is the default file argument:</p> <pre data-bbox="306 312 714 344">[root@mdw ~]# sysctl -p</pre>  <p>This step is necessary if you have made changes to the <code>/etc/sysctl.conf</code> file. Confirm the settings that you added in step 2 are present.</p>
4.	<p>Using the Linux <code>cat</code> command review <code>/etc/security/limits.conf</code> and verify the settings are as shown.</p> <p>These settings have already been made to the file. If they have not been made to the file use <code>vi</code> and add them.</p> <pre data-bbox="306 1453 633 1600"> *soft nofile 65536 *hard nofile 65536 *soft nproc 131072 *hard nproc 131072 </pre>

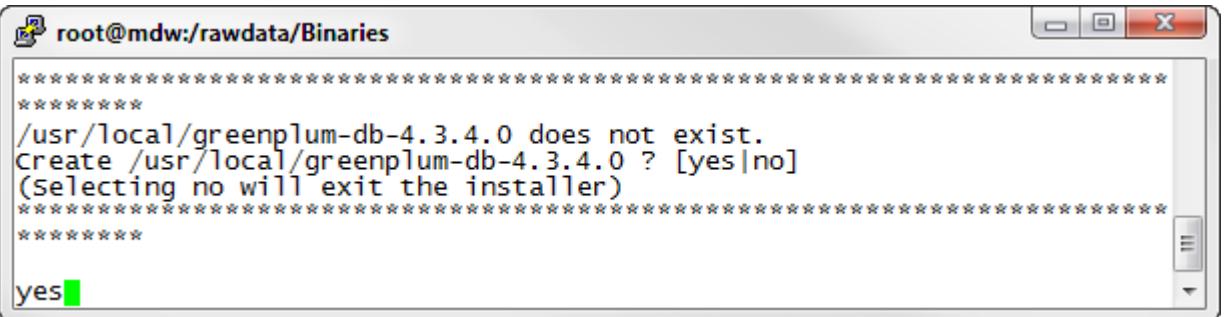
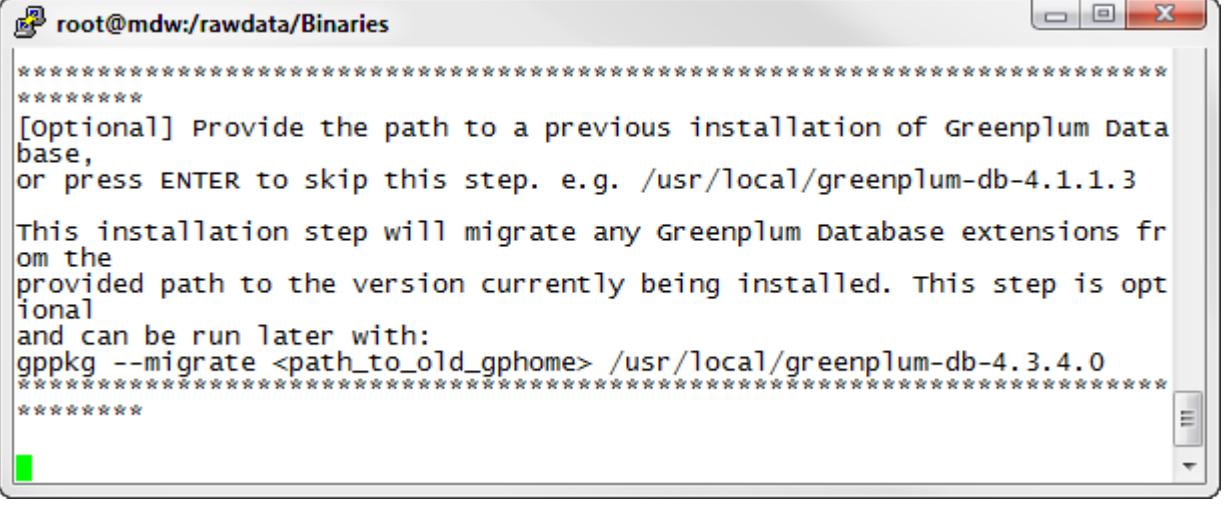
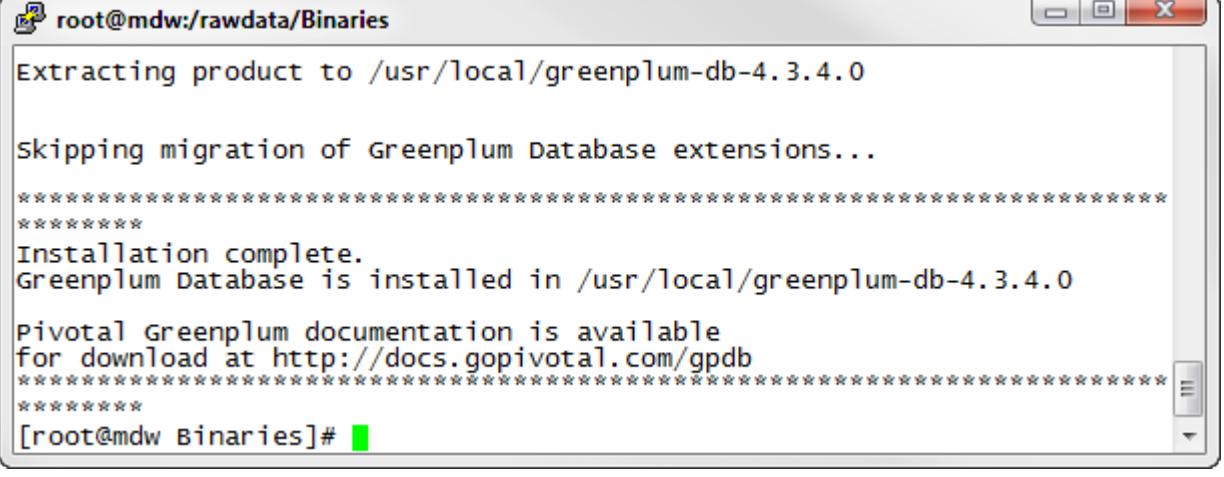
Step	Action
5.	<p>Using the vi editor, edit the /etc/hosts file and add the host names and IP addresses of all machines participating in your Greenplum environment:</p> <pre> 172.16.1.11      mdw      # master host 172.16.1.14      smdw     # standby host 172.16.1.12      sdw1     # segment one 172.16.1.13      sdw2     # segment two </pre>  <p>Save the modified /etc/hosts file.</p>
6.	<p>Using the scp command, copy the file /etc/hosts to all machines participating in your Greenplum environment. Start with the standby host smdw .</p> <pre>[root@mdw ~]# scp /etc/hosts smdw:/etc/hosts</pre>  <p><b>Note:</b> Answer <b>yes</b> to the question <b>Are you sure you want to continue connecting (yes/no)</b> and type the root password to complete the connection.</p> <p><b>Note:</b> The root password is the same for all servers participating on the Greenplum environment.</p>

Step	Action
7.	<p>Copy the /etc/hosts file to the first segment server, sdw1, using the <b>scp</b> command.</p> <pre data-bbox="306 228 1078 259">[root@mdw ~]# scp /etc/hosts sdw1:/etc/hosts</pre>  <p><b>Note:</b> Answer <b>yes</b> to the question <b>Are you sure you want to continue connecting (yes/no)</b> and type the root password to complete the connection.</p>
8.	<p>Copy the /etc/hosts file to the second segment server, sdw2, using the <b>scp</b> command.</p> <pre data-bbox="306 868 1078 899">[root@mdw ~]# scp /etc/hosts sdw2:/etc/hosts</pre>  <p><b>Note:</b> Answer <b>yes</b> to the question <b>Are you sure you want to continue connecting (yes/no)</b> and type the root password to complete the connection.</p>

Step	Action
9.	<p>Typically, you would download or copy the Greenplum Database installer file to the system that will be the Greenplum Master host.</p> <p>In this lab environment, the installer was preloaded in the /rawdata/Binaries directory.</p> <p>Change to the /rawdata/Binaries directory and list the contents of the directory.</p> <pre>[root@mdw ~]# cd /rawdata/Binaries [root@mdw ~]# ls</pre> 

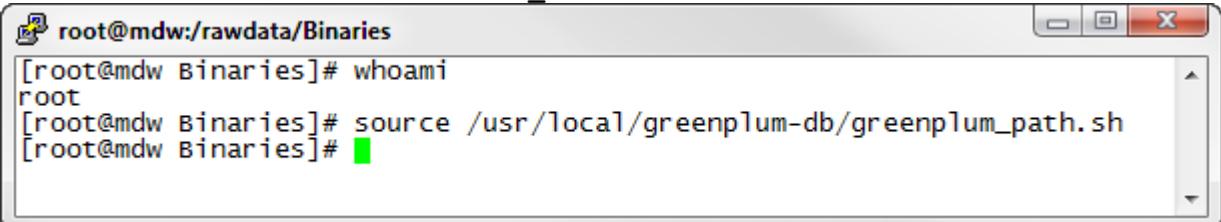
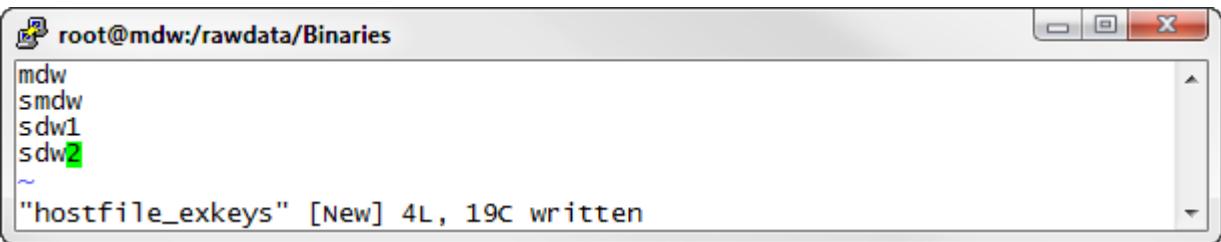
Step	Action
10.	<p>The greenplum-db-4.3.4.0-build-1-RHEL5-x86_64.zip file is the Greenplum binary that you will install.</p> <p>Unzip the greenplum-db-4.3.4.0-build-1-RHEL5-x86_64.zip file.</p> <pre data-bbox="300 333 1416 405">[root@mdw Binaries]# unzip \greenplum-db-4.3.4.0-build-1-RHEL5-x86_64.zip</pre>  <p><b>Note:</b> A non-quoted backslash '\' is the Bash escape character. It preserves the literal value of the next character that follows, with the exception of newline. If a \newline pair appears, and the backslash itself is not quoted, the \newline is treated as a line continuation (that is, it is removed from the input stream and effectively ignored).</p>
11.	<p>Launch the installer using bash. The following is an example of the command:</p> <pre data-bbox="300 1079 1099 1151">[root@mdw Binaries]# /bin/bash \ greenplum-db-4.3.4.0-build-1-RHEL5-x86_64.bin</pre> 

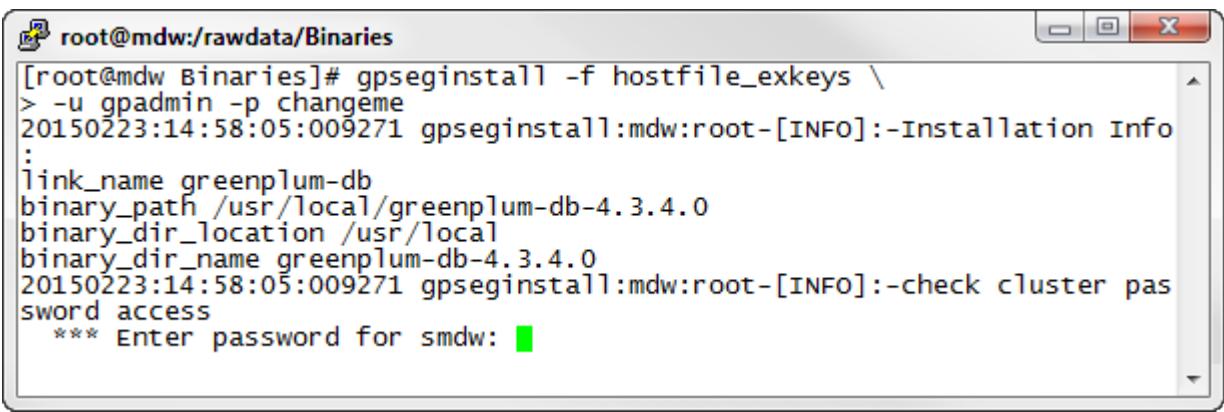
Step	Action
12.	<p>Press the <b>space bar</b> to page through and read the license agreement. You may also press <b>q</b> at any time to stop reading the license agreement. At the end, you will be prompted to accept the license agreement.</p>  <p>Type <b>yes</b> to accept the Greenplum Database License Agreement and press <b>ENTER</b>.</p>
13.	<p>Accept the default installation path and press <b>ENTER</b>.</p> 
14.	<p>Type <b>yes</b> and press <b>ENTER</b> to accept the default install path.</p> 

Step	Action
15.	<p>Type <b>yes</b> and press <b>ENTER</b> to create the Greenplum Database directory.</p>  <pre>root@mdw:/rawdata/Binaries ***** /usr/local/greenplum-db-4.3.4.0 does not exist. Create /usr/local/greenplum-db-4.3.4.0 ? [yes no] (Selecting no will exit the installer) ***** yes</pre>
16.	<p>This optional step applies if a previous installation of Greenplum Database could be found on the system. As this does not currently apply, press <b>ENTER</b> to continue.</p>  <pre>root@mdw:/rawdata/Binaries ***** [Optional] Provide the path to a previous installation of Greenplum Data base, or press ENTER to skip this step. e.g. /usr/local/greenplum-db-4.1.1.3  This installation step will migrate any Greenplum Database extensions fr om the provided path to the version currently being installed. This step is opt ional and can be run later with: gppkg --migrate &lt;path_to_old_gphome&gt; /usr/local/greenplum-db-4.3.4.0 ***** </pre>
17.	<p>The installer has installed the Greenplum Database software and created a symbolic link named <code>greenplum-db</code> at the same level as your version-specific Greenplum Database installation directory. The symbolic link is used to facilitate patch maintenance and upgrades between versions. The installed location is referred to as <code>\$GPHOME</code>.</p>  <pre>root@mdw:/rawdata/Binaries Extracting product to /usr/local/greenplum-db-4.3.4.0  Skipping migration of Greenplum Database extensions...  ***** Installation complete. Greenplum Database is installed in /usr/local/greenplum-db-4.3.4.0  Pivotal Greenplum documentation is available for download at http://docs.gopivotal.com/gpdb ***** [root@mdw Binaries]#</pre>



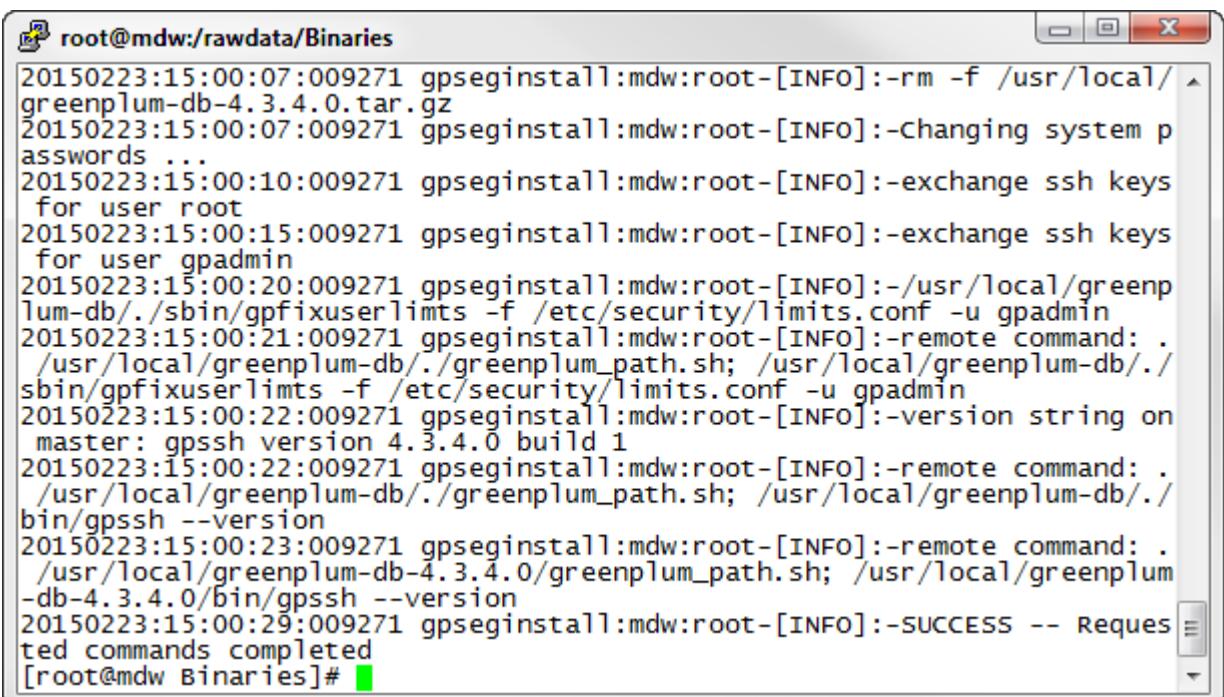
## Lab 2: Task 2 – Install and Configure Greenplum on all Hosts

Step	Action
	<p><b>Task Description</b></p> <p>You have just installed Greenplum Database on the master server. The following tasks will copy the Greenplum database installation from the <b>master</b> host and install the binaries to the hosts you specify. These hosts include the segment servers and the standby host.</p> <p>In this task, you will execute the <code>gpseginstall</code> command which will:</p> <ol style="list-style-type: none"> <li>1. Create the Greenplum system user, <code>gpadmin</code>, on all hosts in the cluster and set the password to <code>changeme</code>.</li> <li>2. Change the ownership of the Greenplum installation directory</li> <li>3. Exchange ssh keys among the master, standby, and the segments</li> <li>4. Create the directories for storage data on hosts in the cluster</li> </ol>
1.	<p>You should login as <code>root</code> to perform the following tasks. You can confirm whether or not you are logged as <code>root</code> with the command <code>whoami</code> as shown below.</p> <p>Source the path from the master host Greenplum database installation as shown.</p> <pre>[root@mdw Binaries]# source \ /usr/local/greenplum-db/greenplum_path.sh</pre>  <p>If you open a separate terminal window, you will need to source the <code>greenplum_path.sh</code> file in that window to easily access the Greenplum commands.</p>
2.	<p>Use the <code>vi</code> editor to create a file called <code>hostfile_exkeys</code>. This file should contain the hostnames of each host in the Greenplum environment.</p> <p>Add the following host names to <code>hostfile_exkeys</code>:</p> <p>mdw smdw sdw1 sdw2</p> <p><b>Note:</b> if you are not familiar with the <code>vi</code> editor, refer to the <i>Linux Basic Commands Appendix A</i>.</p> 

Step	Action
3.	<p>Execute the gpseginstall utility and reference the file you just created, hostfile_exkeys. The -u option will create the Greenplum administrative user in all servers. You may or may not be asked for the password for each server.</p> <pre>[root@mdw Binaries]# gpseginstall -f hostfile_exkeys \ -u gpadmin -p changeme</pre> 

Step	Action
4.	<p>The utility may prompt you to confirm the password for all of the servers on the Greenplum cluster. If prompted you must enter the password for each server in the cluster, except the master. Press <b>Enter</b> after entering the password. If you do not see the prompt for the next password required, enter the password again and press <b>Enter</b>.</p>  <pre data-bbox="311 340 1535 1036"> root@mdw:/rawdata/Binaries 20150223:15:31:54:003933 gpseginstall:mdw:root-[INFO]:-cd /usr/local; tar cf greenplum-db-4.3.4.0.tar greenplum-db-4.3.4.0 20150223:15:31:59:003933 gpseginstall:mdw:root-[INFO]:-gzip /usr/local/greenplum-db-4.3.4.0.tar 20150223:15:32:26:003933 gpseginstall:mdw:root-[INFO]:-remote command: mkdir -p /usr/local 20150223:15:32:26:003933 gpseginstall:mdw:root-[INFO]:-remote command: rm -rf /usr/local/greenplum-db-4.3.4.0 20150223:15:32:27:003933 gpseginstall:mdw:root-[INFO]:-scp software to remote location root@sdw1's password: root@smdw's password: root@sdw2's password:  root@smdw's password:  root@smdw's password: root@sdw2's password:  root@sdw2's password: 20150223:15:32:59:003933 gpseginstall:mdw:root-[INFO]:-remote command: gzip -f -d /usr/local/greenplum-db-4.3.4.0.tar.gz </pre>

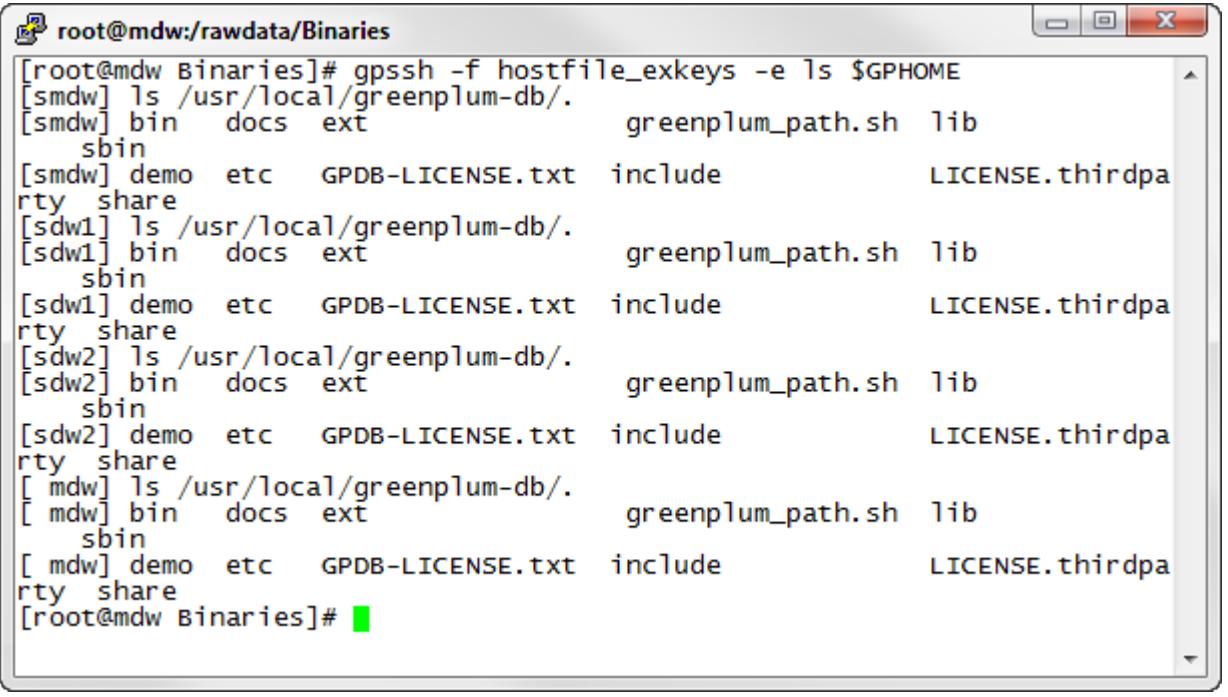
When gpseginstall finishes with no errors, the bottom portion of your screen should be similar to the screen below.



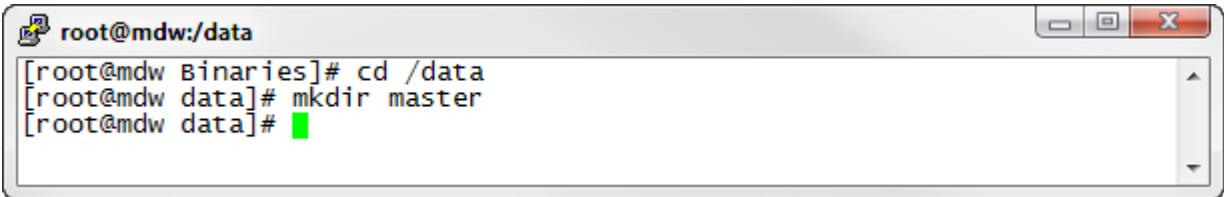
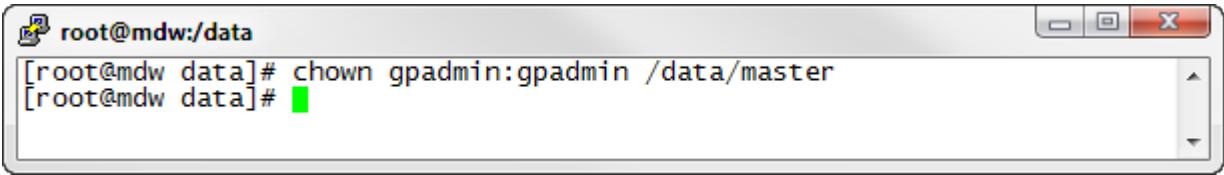
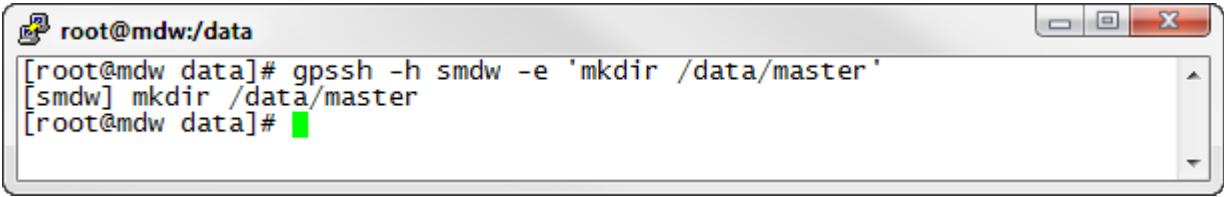
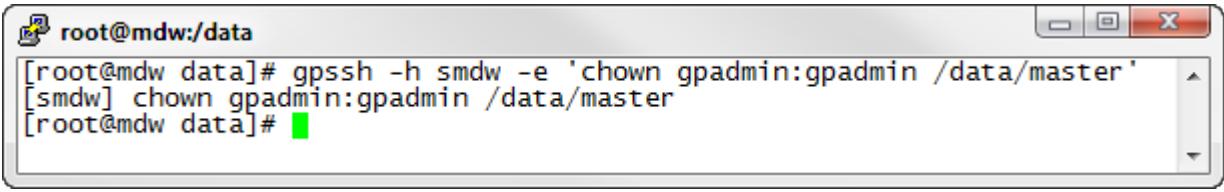
```

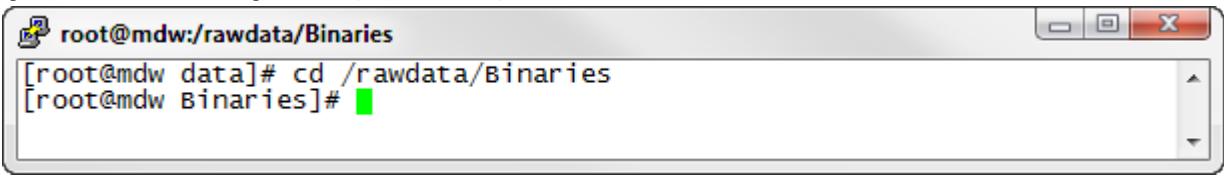
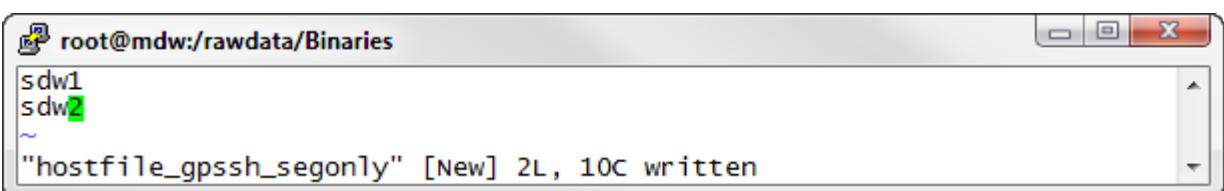
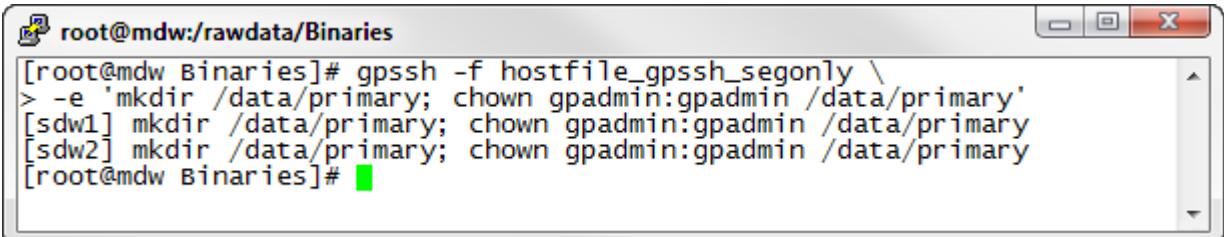
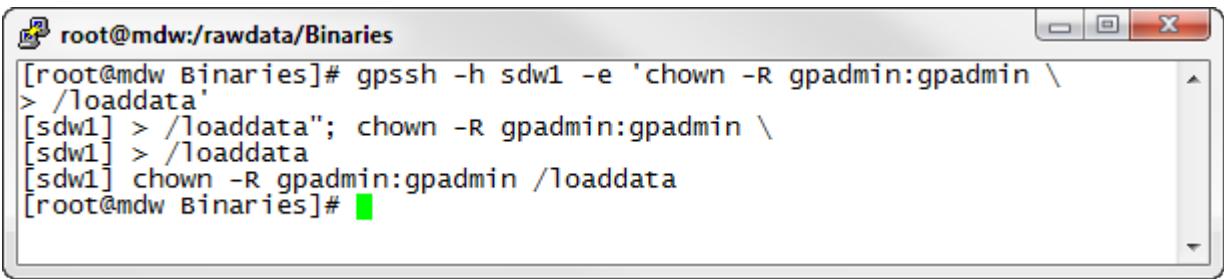
root@mdw:/rawdata/Binaries
20150223:15:00:07:009271 gpseginstall:mdw:root-[INFO]:-rm -f /usr/local/greenplum-db-4.3.4.0.tar.gz
20150223:15:00:07:009271 gpseginstall:mdw:root-[INFO]:-changing system passwords ...
20150223:15:00:10:009271 gpseginstall:mdw:root-[INFO]:-exchange ssh keys for user root
20150223:15:00:15:009271 gpseginstall:mdw:root-[INFO]:-exchange ssh keys for user gpadmin
20150223:15:00:20:009271 gpseginstall:mdw:root-[INFO]:-/usr/local/greenplum-db./sbin/gpfixuserlimits -f /etc/security/limits.conf -u gpadmin
20150223:15:00:21:009271 gpseginstall:mdw:root-[INFO]:-remote command: ./usr/local/greenplum-db./greenplum_path.sh; /usr/local/greenplum-db./sbin/gpfixuserlimits -f /etc/security/limits.conf -u gpadmin
20150223:15:00:22:009271 gpseginstall:mdw:root-[INFO]:-version string on master: gpssh version 4.3.4.0 build 1
20150223:15:00:22:009271 gpseginstall:mdw:root-[INFO]:-remote command: ./usr/local/greenplum-db./greenplum_path.sh; /usr/local/greenplum-db./bin/gpssh --version
20150223:15:00:23:009271 gpseginstall:mdw:root-[INFO]:-remote command: ./usr/local/greenplum-db-4.3.4.0/greenplum_path.sh; /usr/local/greenplum-db-4.3.4.0/bin/gpssh --version
20150223:15:00:29:009271 gpseginstall:mdw:root-[INFO]:-SUCCESS -- Requested commands completed
[root@mdw Binaries]#

```

Step	Action
5.	<p>Verify that all hosts in the cluster are accessible and have their own copy of the Greenplum software installed. Use the gpssh command to accomplish this task.</p> <pre data-bbox="314 264 1411 295">[root@mdw Binaries]# gpssh -f hostfile_exkeys -e ls -F \$GPHOME</pre> 

## Lab 2: Task 3 – Creating the Data Storage Areas

Step	Action
	<p><b>Task Description</b></p> <p>Every Greenplum Database master and segment instance has a designated storage area on disk that is called the data directory location. This is the file system location where the directories that store segment instance data will be created. The master host needs a data storage location for the master data directory. Each segment host needs a data directory storage location for its primary segments, and another for its mirror segments.</p>
1.	<p>On your master server host, change to /data directory and create the directory that will be your master data storage area as shown.</p> <pre>[root@mdw Binaries]# cd /data [root@mdw data]# mkdir master</pre> 
2.	<p>Change the ownership and the group of the new directory.</p> <pre>[root@mdw data]# chown gpadmin:gpadmin /data/master</pre> 
3.	<p>Using gpssh, create the master data directory location on your standby master as well.</p> <pre>[root@mdw data]# gpssh -h smdw -e 'mkdir /data/master'</pre> 
4.	<p>Change the ownership of the data directory to gpadmin.</p> <pre>[root@mdw data]# gpssh -h smdw -e 'chown gpadmin:gpadmin \ /data/master'</pre> 

Step	Action
5.	<p>Change to the /rawdata/Binaries directory.</p> <pre>[root@mdw data]# cd /rawdata/Binaries</pre> 
6.	<p>Use the vi editor to create a file called hostfile_gpssh_segonly. This file should contain the hostnames of each segment host in the Greenplum environment.</p> <p>Add the following segment host names to hostfile_gpssh_segonly:</p> <pre>sdw1 sdw2</pre> 
7.	<p>Using gpssh, create the primary and mirror data directory locations on all segment hosts at once using the hostfile_gpssh_segonly file you just created.</p> <pre>[root@mdw Binaries]# gpssh -f hostfile_gpssh_segonly \ -e 'mkdir /data/primary; chown gpadmin:gpadmin /data/primary'</pre> 
8.	<p>Using gpssh, change the permissions on the directory /loaddata in the sdw1 server. This directory will be used in future labs.</p> <pre>[root@mdw Binaries]# gpssh -h sdw1 -e 'chown -R gpadmin:gpadmin \ /loaddata'</pre> 

## Lab 2: Task 4 – Synchronizing System Clocks

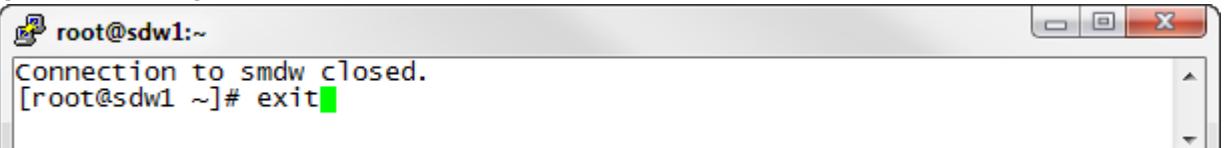
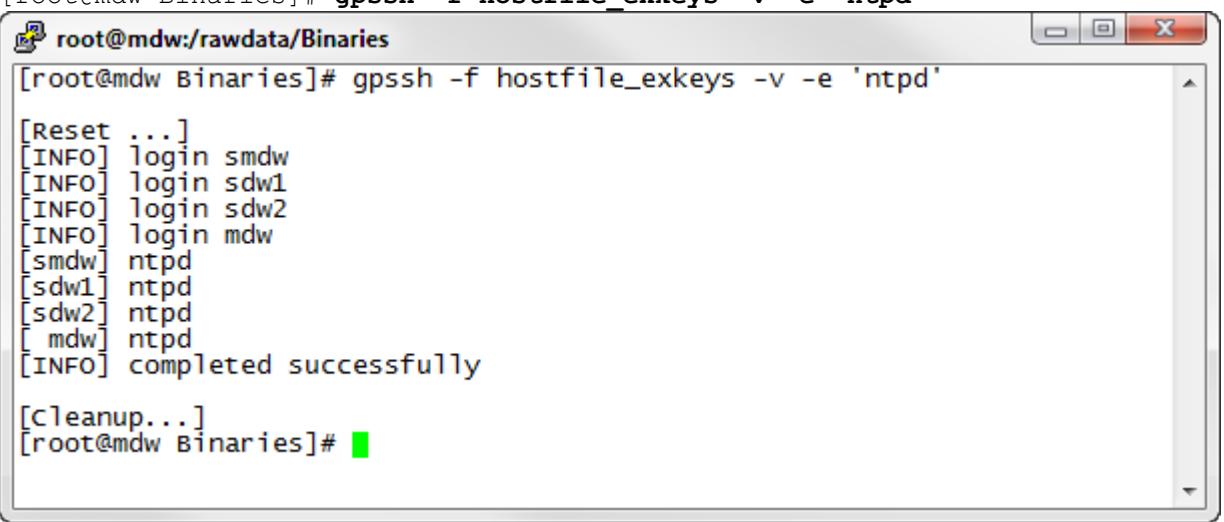
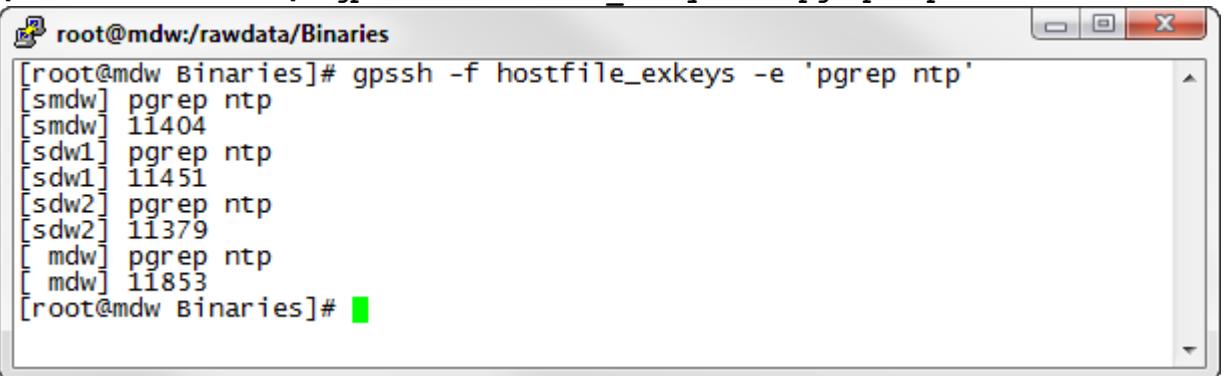
Step	Action
1.	<p>Greenplum recommends using NTP (Network Time Protocol) to synchronize the system clocks on all hosts that comprise your Greenplum Database system.</p> <p>NTP on the segment hosts should be configured to use the master host as the primary time source and the standby master as the secondary time source. On the master and standby master hosts, configure NTP to point to your preferred time server.</p> <p>On the master host, edit the <code>/etc/ntp.conf</code> file and perform the following tasks:</p> <p>Comment the following lines</p> <pre># server 0.centos.pool.ntp.org iburst # server 1.centos.pool.ntp.org iburst # server 2.centos.pool.ntp.org iburst # server 3.centos.pool.ntp.org iburst</pre> <p>Add the line server IP address indicated as shown:</p> <pre>server 172.16.1.11</pre>

Step	Action
2.	<p>On the master host ( mdw ), modify the <code>ntp.conf</code> file as follows.</p> <pre data-bbox="372 255 1465 1712"> root@mdw:/rawdata/Binaries # For more information about this file, see the man pages # ntp.conf(5), ntp_acc(5), ntp_auth(5), ntp_clock(5), ntp_misc(5), ntp_mon(5).  driftfile /var/lib/ntp/drift  # Permit time synchronization with our time source, but do not # permit the source to query or modify the service on this system. restrict default kod nomodify notrap nopeer noquery restrict -6 default kod nomodify notrap nopeer noquery  # Permit all access over the loopback interface. This could # be tightened as well, but to do so would effect some of # the administrative functions. restrict 127.0.0.1 restrict -6 ::1  # Hosts on local network are less restricted. #restrict 192.168.1.0 mask 255.255.255.0 nomodify notrap  # Use public servers from the pool.ntp.org project. # Please consider joining the pool (<a href="http://www.pool.ntp.org/join.html">http://www.pool.ntp.org/join.html</a>). #server 0.centos.pool.ntp.org iburst #server 1.centos.pool.ntp.org iburst #server 2.centos.pool.ntp.org iburst #server 3.centos.pool.ntp.org iburst <b>Comment These Lines Out</b>  #broadcast 192.168.1.255 autokey      # broadcast server #broadcastclient                      # broadcast client #broadcast 224.0.1.1 autokey          # multicast server #multicastclient 224.0.1.1            # multicast client #multicastserver 239.255.254.254     # multicast server #multicastclient 239.255.254.254 autokey # multicast client  <b>Add This Line</b> server 172.16.1.11  # Enable public key cryptography. #crypto  includefile /etc/ntp/crypto/pw  # Key file containing the keys and key identifiers used when operating # with symmetric key cryptography. keys /etc/ntp/keys  # Specify the key identifiers which are trusted. #trustedkey 4 8 42  :wq! </pre> <p>Save your changes and exit.</p>

Step	Action
3.	<p>Connect to segment 1 host ( sdw1) using ssh from the current terminal window. Use vi to edit the /etc/ntp.conf file, and modify it as shown below.</p> <pre data-bbox="372 255 1465 1712"> root@sdw1:~ # For more information about this file, see the man pages # ntp.conf(5), ntp_acc(5), ntp_auth(5), ntp_clock(5), ntp_misc(5), ntp_mon(5).  driftfile /var/lib/ntp/drift  # Permit time synchronization with our time source, but do not # permit the source to query or modify the service on this system. restrict default kod nomodify notrap nopeer noquery restrict -6 default kod nomodify notrap nopeer noquery  # Permit all access over the loopback interface. This could # be tightened as well, but to do so would effect some of # the administrative functions. restrict 127.0.0.1 restrict -6 ::1  # Hosts on local network are less restricted. #restrict 192.168.1.0 mask 255.255.255.0 nomodify notrap  # Use public servers from the pool.ntp.org project. # Please consider joining the pool (<a href="http://www.pool.ntp.org/join.html">http://www.pool.ntp.org/join.html</a>). #server 0.centos.pool.ntp.org iburst #server 1.centos.pool.ntp.org iburst #server 2.centos.pool.ntp.org iburst #server 3.centos.pool.ntp.org iburst Comment These Lines Out  #broadcast 192.168.1.255 autokey      # broadcast server #broadcastclient                      # broadcast client #broadcast 224.0.1.1 autokey          # multicast server #multicastclient 224.0.1.1            # multicast client #multicastserver 239.255.254.254     # multicast server #multicastclient 239.255.254.254 autokey # multicast client  server mdw prefer server smdw Add These Lines  # Enable public key cryptography. #crypto  includefile /etc/ntp/crypto/pw  # Key file containing the keys and key identifiers used when operating # with symmetric key cryptography. keys /etc/ntp/keys  # Specify the key identifiers which are trusted. #trustedkey 4 8 42 :wq </pre> <p>Save your changes and exit vi.</p>

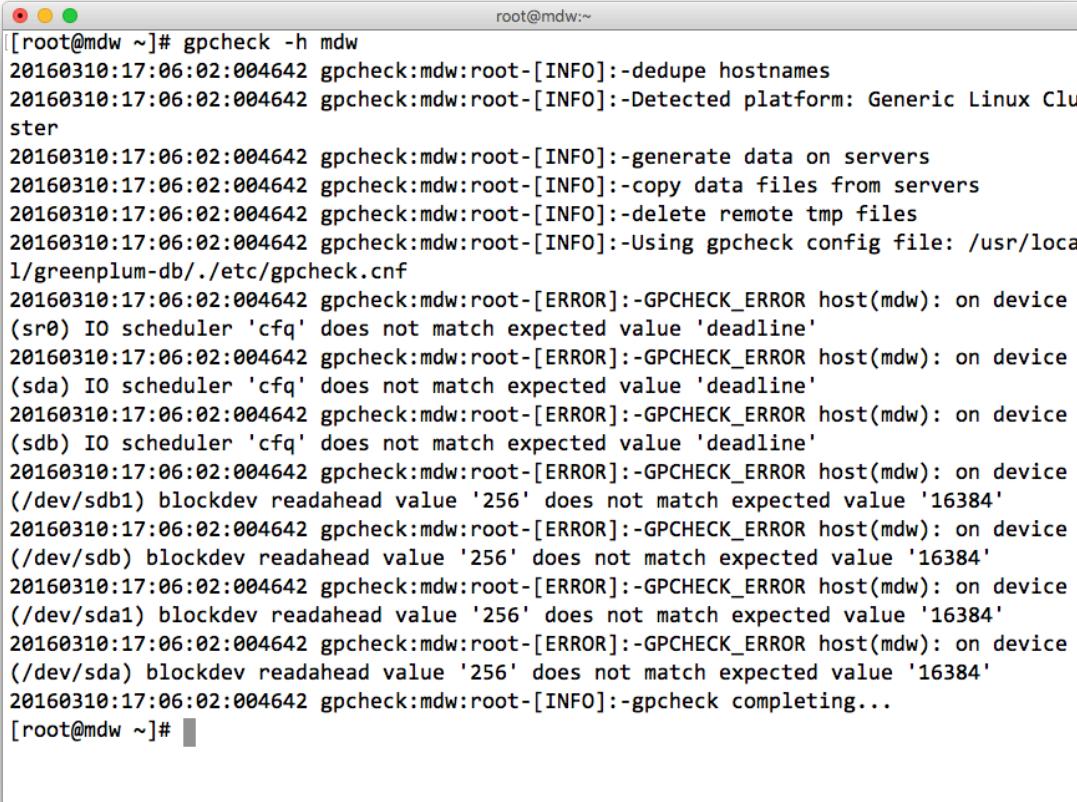
Step	Action
4.	<p>Using gpcp, copy the /etc/ntp.conf file to segment 2, sdw2.</p> <pre>[root@sdw1 ~]# source /usr/local/greenplum-db/greenplum_path.sh [root@sdw1 ~]# gpcp -h sdw2 /etc/ntp.conf =:/etc/ntp.conf</pre>  <p>Note that there is a space between “ntp.conf” and “=:”</p>

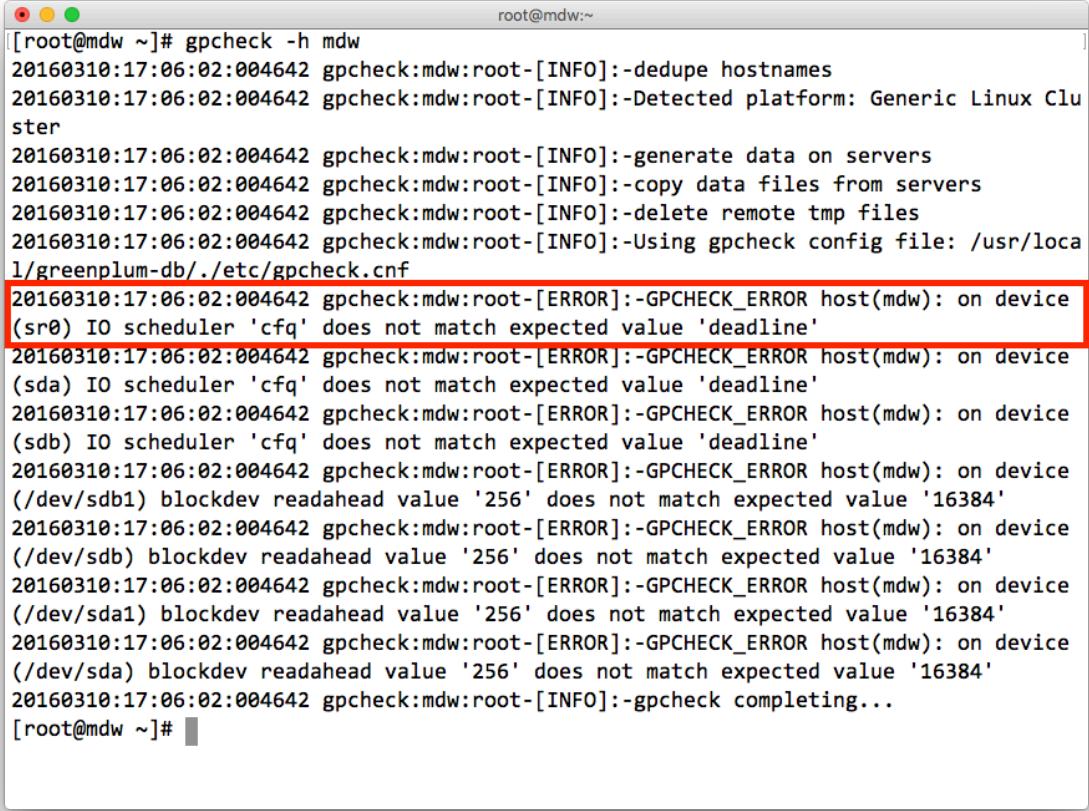
Step	Action
5.	<p>Connect to the standby master server ( smdw ) using ssh. Edit /etc/ntp.conf using vi, and modify the file as shown on the picture below.</p> <pre data-bbox="372 255 1465 1712"> root@smdw:~ # For more information about this file, see the man pages # ntp.conf(5), ntp_acc(5), ntp_auth(5), ntp_clock(5), ntp_misc(5), ntp_mon(5).  driftfile /var/lib/ntp/drift  # Permit time synchronization with our time source, but do not # permit the source to query or modify the service on this system. restrict default kod nomodify notrap nopeer noquery restrict -6 default kod nomodify notrap nopeer noquery  # Permit all access over the loopback interface. This could # be tightened as well, but to do so would effect some of # the administrative functions. restrict 127.0.0.1 restrict -6 ::1  # Hosts on local network are less restricted. #restrict 192.168.1.0 mask 255.255.255.0 nomodify notrap  # Use public servers from the pool.ntp.org project. # Please consider joining the pool (<a href="http://www.pool.ntp.org/join.html">http://www.pool.ntp.org/join.html</a>). #server 0.centos.pool.ntp.org iburst #server 1.centos.pool.ntp.org iburst #server 2.centos.pool.ntp.org iburst #server 3.centos.pool.ntp.org iburst Comment These Lines Out  #broadcast 192.168.1.255 autokey      # broadcast server #broadcastclient                      # broadcast client #broadcast 224.0.1.1 autokey          # multicast server #multicastclient 224.0.1.1            # multicast client #multicastserver 239.255.254.254     # multicast server #multicastclient 239.255.254.254 autokey # multicast client Add These Lines  server mdw prefer server 172.16.1.14  # Enable public key cryptography. #crypto  includefile /etc/ntp/crypto/pw  # Key file containing the keys and key identifiers used when operating # with symmetric key cryptography. keys /etc/ntp/keys  # Specify the key identifiers which are trusted. #trustedkey 4 8 42 :wq! </pre> <p>Save your changes and exit vi.</p>

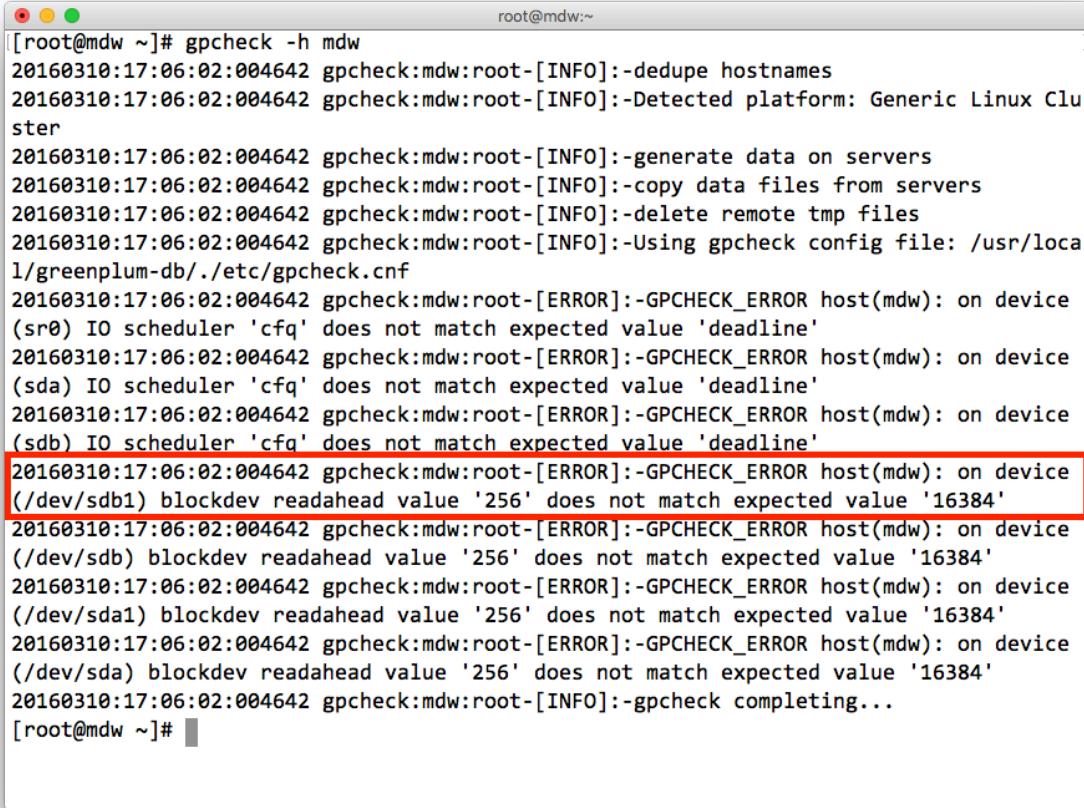
Step	Action
6.	<p>Exit from the standby master server.</p> <pre>[root@smdw ~]# exit</pre> 
7.	<p>Exit from the sdw1 segment server.</p> <pre>[root@sdw1 ~]# exit</pre> 
8.	<p>Synchronize the system clocks on all Greenplum hosts. This effectively starts or resets the ntpd service on all hosts within the cluster.</p> <pre>[root@mdw Binaries]# gpssh -f hostfile_exkeys -v -e 'ntpd'</pre> 
9.	<p>Verify the ntpd service is executing on all hosts in the cluster.</p> <pre>[root@mdw Binaries]# gpssh -f hostfile_exkeys -e 'pgrep ntp'</pre> 

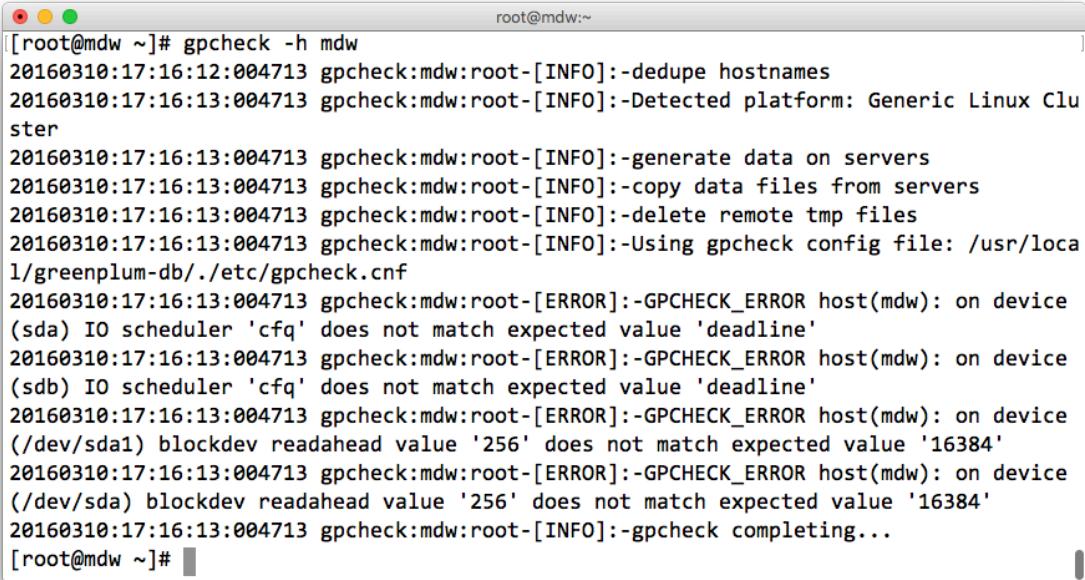
Step	Action
10.	<p>Go back to the root directory on the mdw server.</p> <pre>[root@mdw ~]# cd root@mdw:~ [root@mdw Binaries]# cd [root@mdw ~]#</pre>

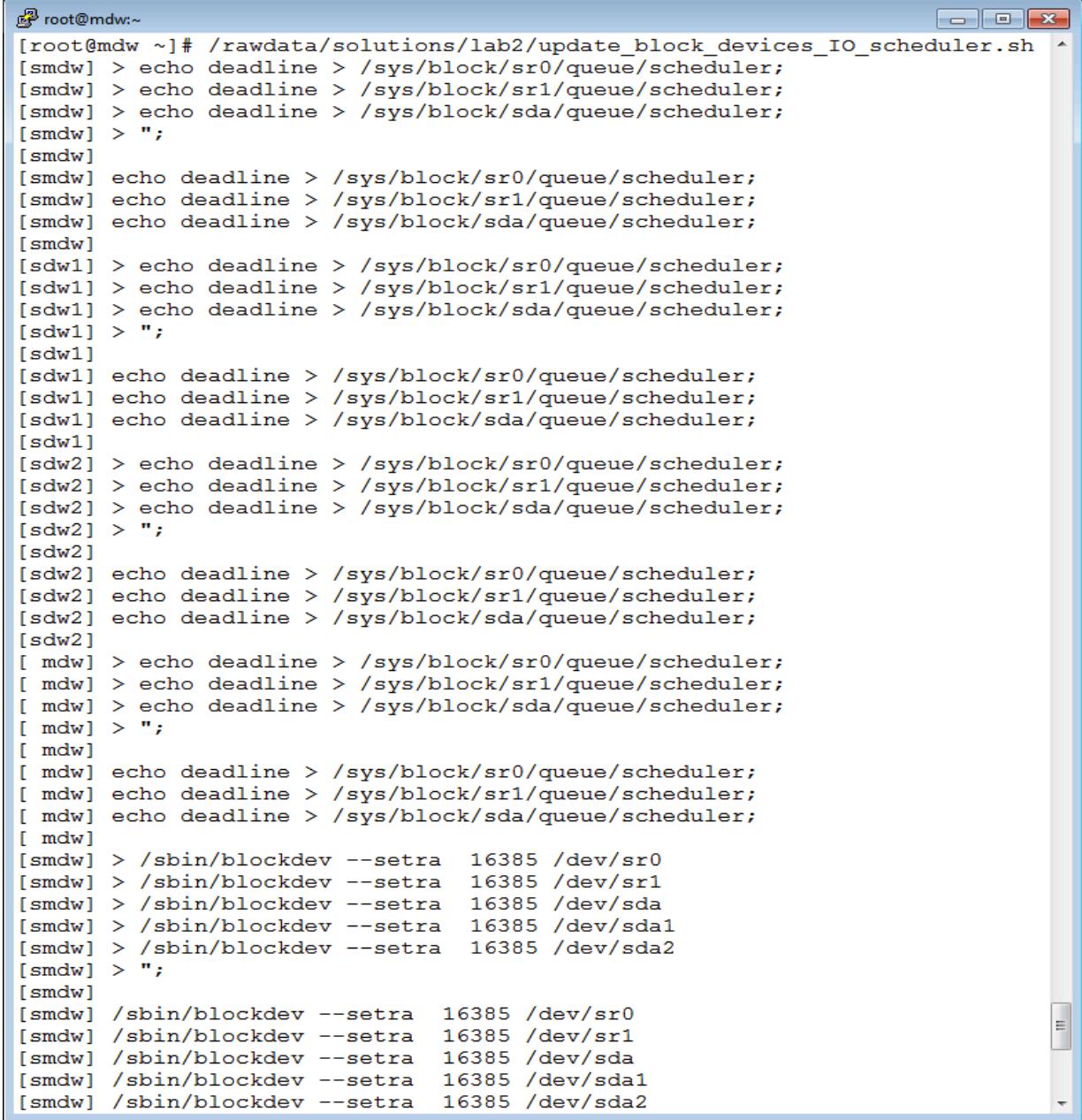
## Lab 2: Task 5 – Perform System Verification Tests

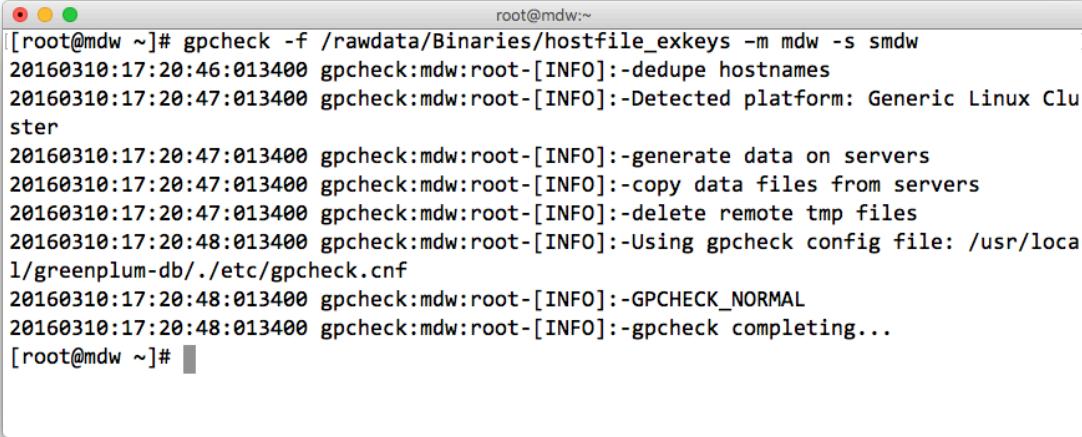
Step	Action
1.	<p>You will execute several Greenplum Database utilities to verify the operating system settings and hardware performance of the servers within the Greenplum Cluster. These tests should be executed prior to initializing your Greenplum Database system.</p> <p>If not already connected to the master server, open a new terminal session to <code>mdw</code> and log in as <code>root</code> with the password, <code>Piv0tal</code>.</p>
2.	<p>Execute the <code>gpcheck</code> command to verify the operating system settings. This command should normally be executed against all hosts in the cluster, but due to the amount of information it provides, we will examine it first for the master server.</p> <pre>[root@mdw ~]# gpcheck -h mdw</pre>  <p>The terminal window shows the output of the <code>gpcheck -h mdw</code> command. It includes informational messages about deduplicating hostnames, detecting the platform as a Generic Linux Cluster, generating data on servers, copying data files, deleting remote tmp files, and using the gpcheck config file. It also lists multiple errors related to IO scheduler 'cfq' not matching expected values for devices <code>(sr0)</code>, <code>(sda)</code>, and <code>(sdb)</code>, as well as blockdev readahead values for devices <code>(/dev/sdb1)</code>, <code>(/dev/sdb)</code>, <code>(/dev/sda1)</code>, and <code>(/dev/sda)</code>. The final message indicates the process is completing.</p> <p>What is the result of the command? Are there any configuration issues that need to be fixed?</p>

Step	Action
3.	<p>There are likely several errors visible for the master server, mdw. The following highlights some of the I/O scheduler and the read ahead errors seen in this environment.</p>  <pre> root@mdw ~]# gpcheck -h mdw 20160310:17:06:02:004642 gpcheck:mdw:root-[INFO]:-dedupe hostnames 20160310:17:06:02:004642 gpcheck:mdw:root-[INFO]:-Detected platform: Generic Linux Cluster 20160310:17:06:02:004642 gpcheck:mdw:root-[INFO]:-generate data on servers 20160310:17:06:02:004642 gpcheck:mdw:root-[INFO]:-copy data files from servers 20160310:17:06:02:004642 gpcheck:mdw:root-[INFO]:-delete remote tmp files 20160310:17:06:02:004642 gpcheck:mdw:root-[INFO]:-Using gpcheck config file: /usr/local/greenplum-db/.etc/gpcheck.cnf 20160310:17:06:02:004642 gpcheck:mdw:root-[ERROR]:-GPCHECK_ERROR host(mdw): on device (sr0) IO scheduler 'cfq' does not match expected value 'deadline' 20160310:17:06:02:004642 gpcheck:mdw:root-[ERROR]:-GPCHECK_ERROR host(mdw): on device (sda) IO scheduler 'cfq' does not match expected value 'deadline' 20160310:17:06:02:004642 gpcheck:mdw:root-[ERROR]:-GPCHECK_ERROR host(mdw): on device (sdb) IO scheduler 'cfq' does not match expected value 'deadline' 20160310:17:06:02:004642 gpcheck:mdw:root-[ERROR]:-GPCHECK_ERROR host(mdw): on device (/dev/sdb1) blockdev readahead value '256' does not match expected value '16384' 20160310:17:06:02:004642 gpcheck:mdw:root-[ERROR]:-GPCHECK_ERROR host(mdw): on device (/dev/sdb) blockdev readahead value '256' does not match expected value '16384' 20160310:17:06:02:004642 gpcheck:mdw:root-[ERROR]:-GPCHECK_ERROR host(mdw): on device (/dev/sda1) blockdev readahead value '256' does not match expected value '16384' 20160310:17:06:02:004642 gpcheck:mdw:root-[ERROR]:-GPCHECK_ERROR host(mdw): on device (/dev/sda) blockdev readahead value '256' does not match expected value '16384' 20160310:17:06:02:004642 gpcheck:mdw:root-[INFO]:-gpcheck completing... [root@mdw ~]# </pre>
4.	<p>Update I/O scheduler value for sr0 device using the command below.</p> <pre>[root@mdw ~]# echo deadline &gt; /sys/block/sr0/queue/scheduler</pre> 

Step	Action
5.	<p>Another error identified by running <code>gpcheck -h mdw</code> is highlighted in this window.</p>  <pre>[root@mdw ~]# gpcheck -h mdw 20160310:17:06:02:004642 gpcheck:mdw:root-[INFO]:-dedupe hostnames 20160310:17:06:02:004642 gpcheck:mdw:root-[INFO]:-Detected platform: Generic Linux Cluster 20160310:17:06:02:004642 gpcheck:mdw:root-[INFO]:-generate data on servers 20160310:17:06:02:004642 gpcheck:mdw:root-[INFO]:-copy data files from servers 20160310:17:06:02:004642 gpcheck:mdw:root-[INFO]:-delete remote tmp files 20160310:17:06:02:004642 gpcheck:mdw:root-[INFO]:-Using gpcheck config file: /usr/local/greenplum-db/.etc/gpcheck.cnf 20160310:17:06:02:004642 gpcheck:mdw:root-[ERROR]:-GPCHECK_ERROR host(mdw): on device (sr0) IO scheduler 'cfq' does not match expected value 'deadline' 20160310:17:06:02:004642 gpcheck:mdw:root-[ERROR]:-GPCHECK_ERROR host(mdw): on device (sda) IO scheduler 'cfq' does not match expected value 'deadline' 20160310:17:06:02:004642 gpcheck:mdw:root-[ERROR]:-GPCHECK_ERROR host(mdw): on device (sdb) IO scheduler 'cfq' does not match expected value 'deadline' 20160310:17:06:02:004642 gpcheck:mdw:root-[ERROR]:-GPCHECK_ERROR host(mdw): on device (/dev/sdb1) blockdev readahead value '256' does not match expected value '16384' 20160310:17:06:02:004642 gpcheck:mdw:root-[ERROR]:-GPCHECK_ERROR host(mdw): on device (/dev/sdb) blockdev readahead value '256' does not match expected value '16384' 20160310:17:06:02:004642 gpcheck:mdw:root-[ERROR]:-GPCHECK_ERROR host(mdw): on device (/dev/sda1) blockdev readahead value '256' does not match expected value '16384' 20160310:17:06:02:004642 gpcheck:mdw:root-[ERROR]:-GPCHECK_ERROR host(mdw): on device (/dev/sda) blockdev readahead value '256' does not match expected value '16384' 20160310:17:06:02:004642 gpcheck:mdw:root-[INFO]:-gpcheck completing... [root@mdw ~]#</pre>
6.	<p>Update readahead value for <code>sda1</code> device using the command below.</p> <pre>[root@mdw ~]# /sbin/blockdev --setra 16384 /dev/sdb1</pre>  <pre>[root@mdw ~]# /sbin/blockdev --setra 16384 /dev/sdb1 [root@mdw ~]#</pre>

Step	Action
7.	<p>Re-execute the <code>gpcheck</code> command to confirm that the errors for the devices <code>srx0</code> and <code>sdb1</code> on <code>mdw</code> are no longer present. However some errors still exist on <code>mdw</code>. In addition all of the errors that were present on <code>mdw</code> are also present on <code>smdw</code>, <code>sdw1</code> and <code>sdw2</code> and need to be cleared.</p> <pre data-bbox="323 264 816 295">[root@mdw ~]# gpcheck -h mdw</pre>  <pre data-bbox="383 369 1468 929"> root@mdw:~ [root@mdw ~]# gpcheck -h mdw 20160310:17:16:12:004713 gpcheck:mdw:root-[INFO]:-dedupe hostnames 20160310:17:16:13:004713 gpcheck:mdw:root-[INFO]:-Detected platform: Generic Linux Cluster 20160310:17:16:13:004713 gpcheck:mdw:root-[INFO]:-generate data on servers 20160310:17:16:13:004713 gpcheck:mdw:root-[INFO]:-copy data files from servers 20160310:17:16:13:004713 gpcheck:mdw:root-[INFO]:-delete remote tmp files 20160310:17:16:13:004713 gpcheck:mdw:root-[INFO]:-Using gpcheck config file: /usr/local/greenplum-db/.etc/gpcheck.cnf 20160310:17:16:13:004713 gpcheck:mdw:root-[ERROR]:-GPCHECK_ERROR host(mdw): on device (sda) IO scheduler 'cfq' does not match expected value 'deadline' 20160310:17:16:13:004713 gpcheck:mdw:root-[ERROR]:-GPCHECK_ERROR host(mdw): on device (sdb) IO scheduler 'cfq' does not match expected value 'deadline' 20160310:17:16:13:004713 gpcheck:mdw:root-[ERROR]:-GPCHECK_ERROR host(mdw): on device (/dev/sda1) blockdev readahead value '256' does not match expected value '16384' 20160310:17:16:13:004713 gpcheck:mdw:root-[ERROR]:-GPCHECK_ERROR host(mdw): on device (/dev/sda) blockdev readahead value '256' does not match expected value '16384' 20160310:17:16:13:004713 gpcheck:mdw:root-[INFO]:-gpcheck completing... [root@mdw ~]# </pre>

Step	Action
8.	<p>Execute the script, update_block_devices_IO_scheduler.sh, to update the devices on all hosts.</p> <pre>[root@mdw ~]# /radata/solutions/lab2/update_block_devices_IO_scheduler.sh</pre>  <p><b>Note:</b> This script is provided within the training environment to quickly update the master, standby, and segment servers. This is not provided by default with the Greenplum Database installation. The script uses gpssh to update the block devices on each system. You must be root to successfully execute the commands within the script.</p>

Step	Action
9.	<p>Execute the gpcheck command to verify the configuration in all hosts. Use the /rawdata/Binaries/hostfile_exkeys file to provide the list of all hosts in the cluster.</p> <pre>[root@mdw ~]# gpcheck -f /rawdata/Binaries/hostfile_exkeys -m mdw \ -s smdw</pre>  <pre>root@mdw:~# [root@mdw ~]# gpcheck -f /rawdata/Binaries/hostfile_exkeys -m mdw -s smdw 20160310:17:20:46:013400 gpcheck:mdw:root-[INFO]:-dedupe hostnames 20160310:17:20:47:013400 gpcheck:mdw:root-[INFO]:-Detected platform: Generic Linux Cluster 20160310:17:20:47:013400 gpcheck:mdw:root-[INFO]:-generate data on servers 20160310:17:20:47:013400 gpcheck:mdw:root-[INFO]:-copy data files from servers 20160310:17:20:47:013400 gpcheck:mdw:root-[INFO]:-delete remote tmp files 20160310:17:20:48:013400 gpcheck:mdw:root-[INFO]:-Using gpcheck config file: /usr/local/greenplum-db/.etc/gpcheck.cnf 20160310:17:20:48:013400 gpcheck:mdw:root-[INFO]:-GPCHECK_NORMAL 20160310:17:20:48:013400 gpcheck:mdw:root-[INFO]:-gpcheck completing... [root@mdw ~]#</pre>

Step	Action
10.	<p>Verify the baseline hardware performance of the segment host systems. The test you are performing will take a few minutes as it performs a stress test of memory, I/O, and network performance.</p> <p>Using gpssh, execute the three commands below for all Greenplum hosts. These commands will disable the firewall among the Greenplum servers. The commands are:</p> <ol style="list-style-type: none"> <li>1. <b>service iptables save</b></li> <li>2. <b>service iptables stop</b></li> <li>3. <b>chkconfig iptables off</b></li> </ol> <pre>[root@mdw ~]# cd /rawdata/Binaries [root@mdw Binaries]# gpssh -v -f hostfile_exkeys -e 'service iptables save' [root@mdw Binaries]# gpssh -v -f hostfile_exkeys -e 'service iptables stop' [root@mdw Binaries]# gpssh -v -f hostfile_exkeys -e 'chkconfig iptables off'</pre> <p>The screenshot shows a terminal window titled 'root@mdw:/rawdata/Binaries'. It displays the output of three gpssh commands. The first command saves the iptables configuration ('service iptables save'). The second command stops the iptables service ('service iptables stop'). The third command disables the iptables service at boot ('chkconfig iptables off'). The output shows successful completion on all four hosts (smdw, mdw, sdw2, sdw1).</p>

Step	Action
	<p><b>Note:</b> It is necessary to either disable the firewall or to allow specific ports through the Linux firewall for specific Greenplum services. When testing the servers, the network performance test uses port 23000 by default. If this port is not open on all systems, the test will fail. Greenplum utilities may also occasionally use remote copy (rcp) services to copy files from one system to another.</p>
11.	<p>Execute the gpcheckperf utility to verify the configuration. This is executed against the segment servers within the cluster. As there are only two segment hosts, we will simply specify the hostnames on the command line. If you do create a file, use the <code>-f</code> option followed by the filename to specify the segment servers to include in the validation tests.</p> <p>Either of the following commands will work:</p> <pre>[root@mdw Binaries]# gpcheckperf -h sdw1 -h sdw2 -d /data -D</pre> <p>or</p> <pre>[root@mdw Binaries]# gpcheckperf -f hostfile_gpssh_only -d /data -D</pre>

Step	Action
	<pre>root@mdw:/rawdata/Binaries [root@mdw Binaries]# gpcheckperf -h sdw1 -h sdw2 -d /data -D /usr/local/greenplum-db./bin/gpcheckperf -h sdw1 -h sdw2 -d /data -D  ----- -- DISK WRITE TEST -----  ----- -- DISK READ TEST -----  ----- -- STREAM TEST -----  ----- -- NETPERF TEST -----  ===== == RESULT =====  disk write avg time (sec): 64.16 disk write tot bytes: 16566648832 disk write tot bandwidth (MB/s): 246.23 disk write min bandwidth (MB/s): 122.89 [sdw1] disk write max bandwidth (MB/s): 123.33 [sdw2] -- per host bandwidth --   disk write bandwidth (MB/s): 122.89 [sdw1]   disk write bandwidth (MB/s): 123.33 [sdw2]  disk read avg time (sec): 38.27 disk read tot bytes: 16566648832 disk read tot bandwidth (MB/s): 412.89 disk read min bandwidth (MB/s): 204.12 [sdw1] disk read max bandwidth (MB/s): 208.76 [sdw2] -- per host bandwidth --   disk read bandwidth (MB/s): 204.12 [sdw1]   disk read bandwidth (MB/s): 208.76 [sdw2]  stream tot bandwidth (MB/s): 21864.97 stream min bandwidth (MB/s): 10913.78 [sdw2] stream max bandwidth (MB/s): 10951.19 [sdw1] -- per host bandwidth --   stream bandwidth (MB/s): 10951.19 [sdw1]   stream bandwidth (MB/s): 10913.78 [sdw2]  Netperf bisection bandwidth test sdw1 -&gt; sdw2 = 467.120000 sdw2 -&gt; sdw1 = 472.520000  Summary: sum = 939.64 MB/sec min = 467.12 MB/sec max = 472.52 MB/sec avg = 469.82 MB/sec median = 472.52 MB/sec  [root@mdw Binaries]#</pre>

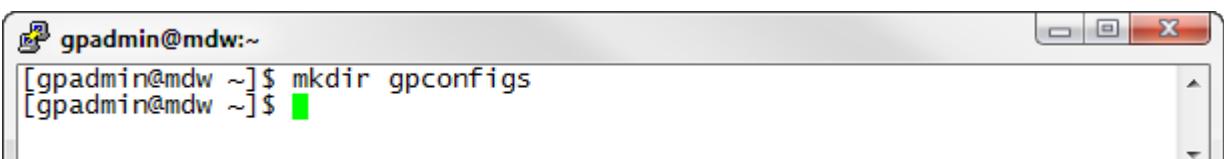
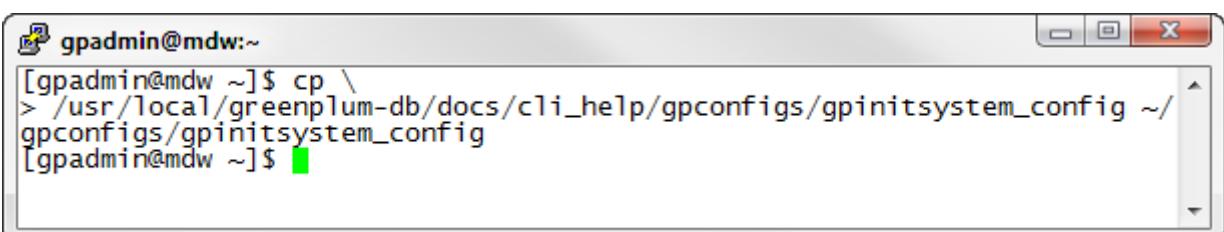
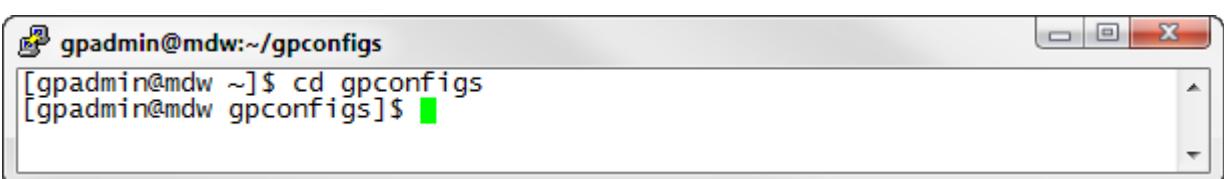
Step	Action
12.	<p>If you wish to perform a network only test, execute the <code>gpcheckperf</code> command with the <code>-r n</code> option. You will also need to specify a segment directory using the <code>-d</code> option.</p> <p>What is the total bandwidth for:</p> <ul style="list-style-type: none"> <li>• Disk Writes? _____</li> <li>• Disk Reads? _____</li> <li>• Sustainable Memory? _____</li> <li>• Network? _____</li> </ul> <p><b>Note:</b> Network performance in the virtual environment may be slightly degraded and so you may see messages that the connection between systems does not meet the guidelines established by Greenplum as adequate for a production environment.</p>
	<p><b>Tips and Best Practices</b></p> <p>In a single-host environment, or virtual environment, performance will be lower than on a distributed cluster. On production systems, you should run <code>gpcheckperf</code> when the system is idle to get accurate performance metrics.</p> <p><b>Summary</b></p> <p>As with any database system, the performance of the Greenplum Database is dependent upon the hardware and IT infrastructure on which it is running. The Greenplum Database is comprised of several servers (or hosts) acting together as one cohesive system. The Greenplum Database's runtime performance will be as fast as the slowest segment host in the array. It is important to know your systems' expected level of performance before setting database performance expectations.</p> <p>The Greenplum Database requires that the operating systems of the hosts on which it runs be properly tuned. These tuning parameters are especially important on large systems with complex query workloads, as queries can fail when they do not get the resources they need from the operating system. The <code>gpcheck</code> utility checks the OS environment of each host to ensure that they have the Greenplum recommended settings.</p> <p>The expected results of the <code>gpcheckperf</code> tests depend on the total capacity of the server hardware you are using. If the expected disk I/O of a system is 2 GBytes per second, multiply the expected rate by the number of segment servers for the total bandwidth. If there are two segment servers, as in this environment, the expected total bandwidth is 4 GB/s.</p> <p>When looking at the output from <code>gpcheckperf</code>, you want to make sure that your disk I/O rate is what you would expect from your hardware platform and that the memory and network bandwidth are not bottlenecks to optimal performance. (They should be comparable to or greater than disk I/O.)</p>

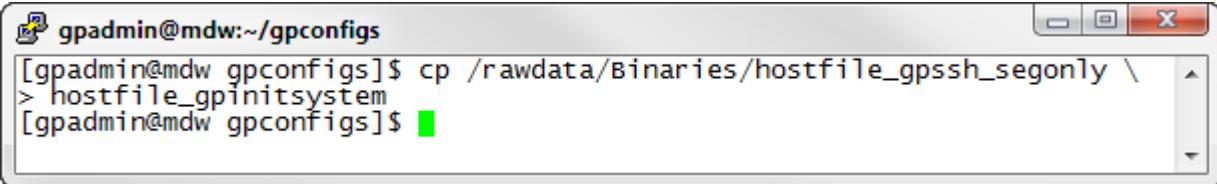
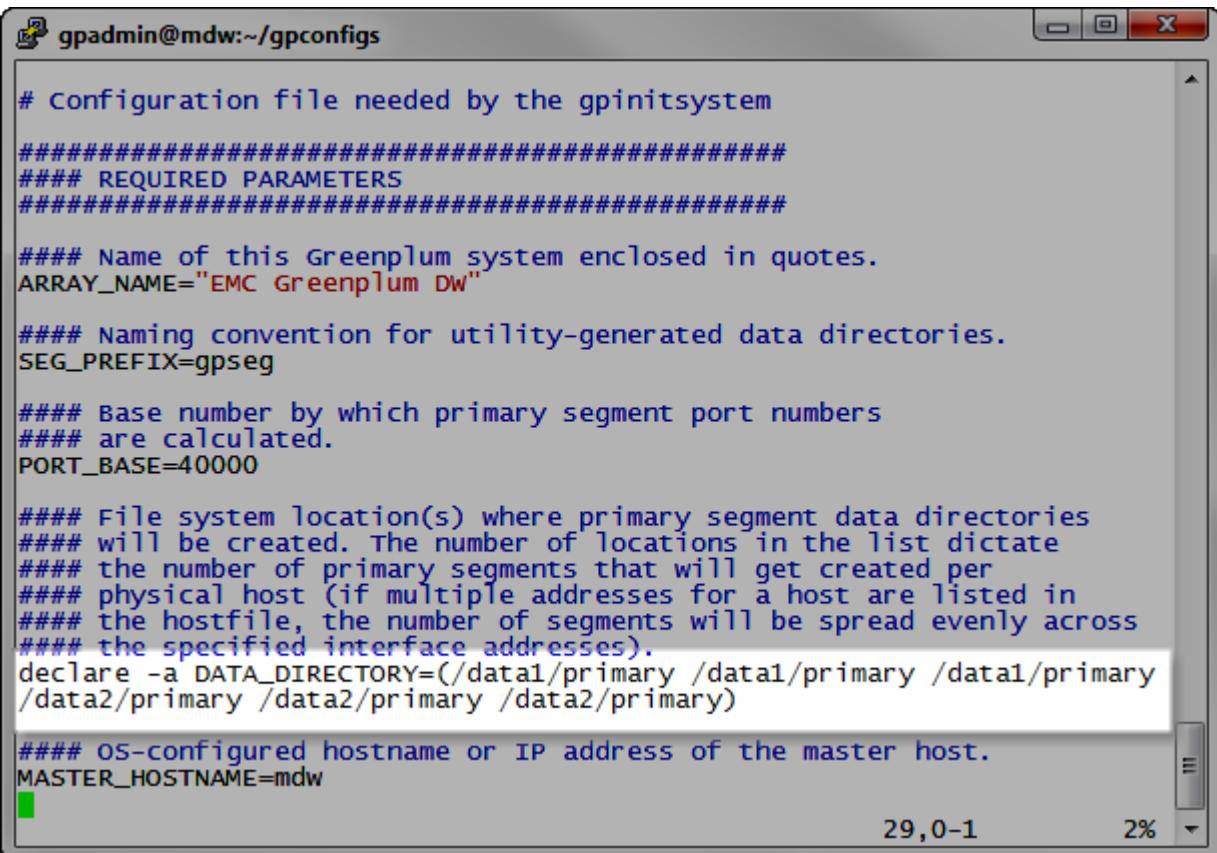
End of Lab Exercise

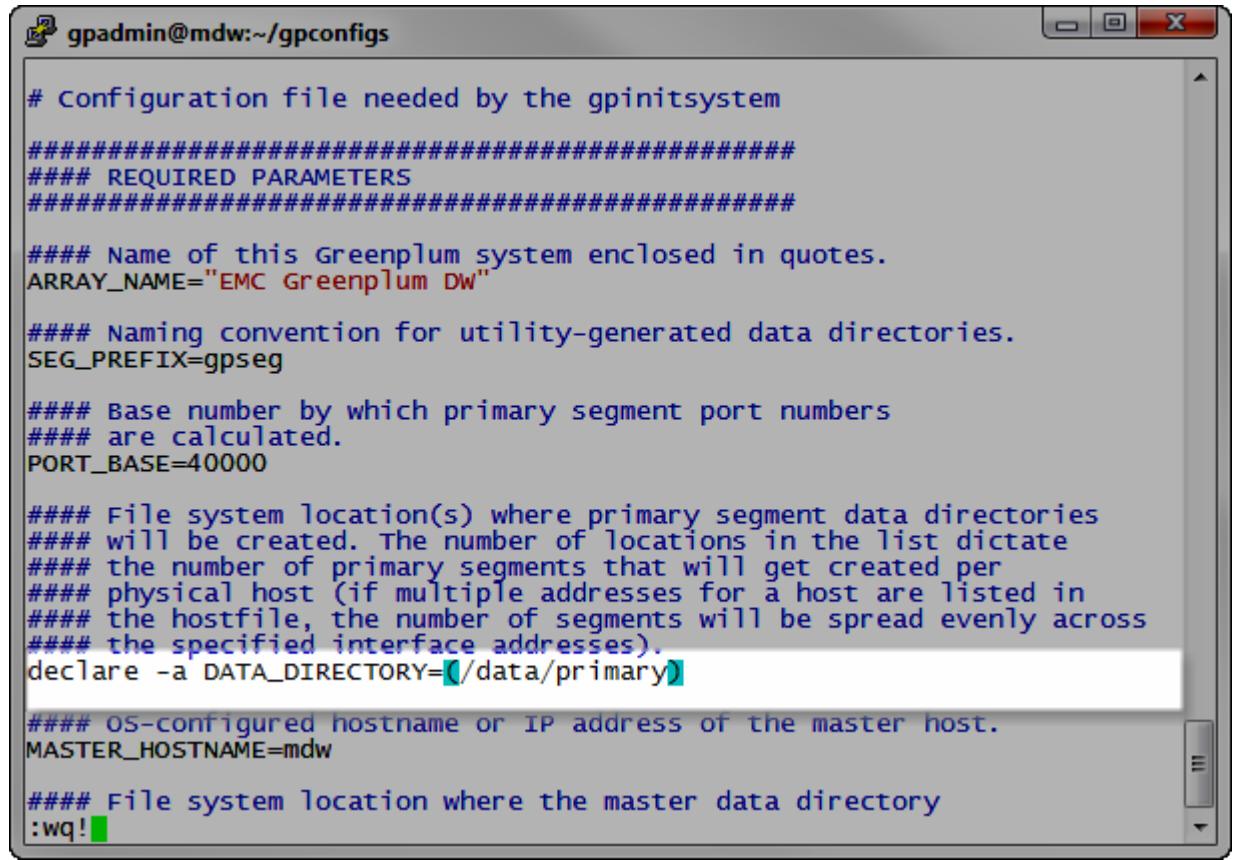
## Lab 3. Pivotal Greenplum Database Initialization

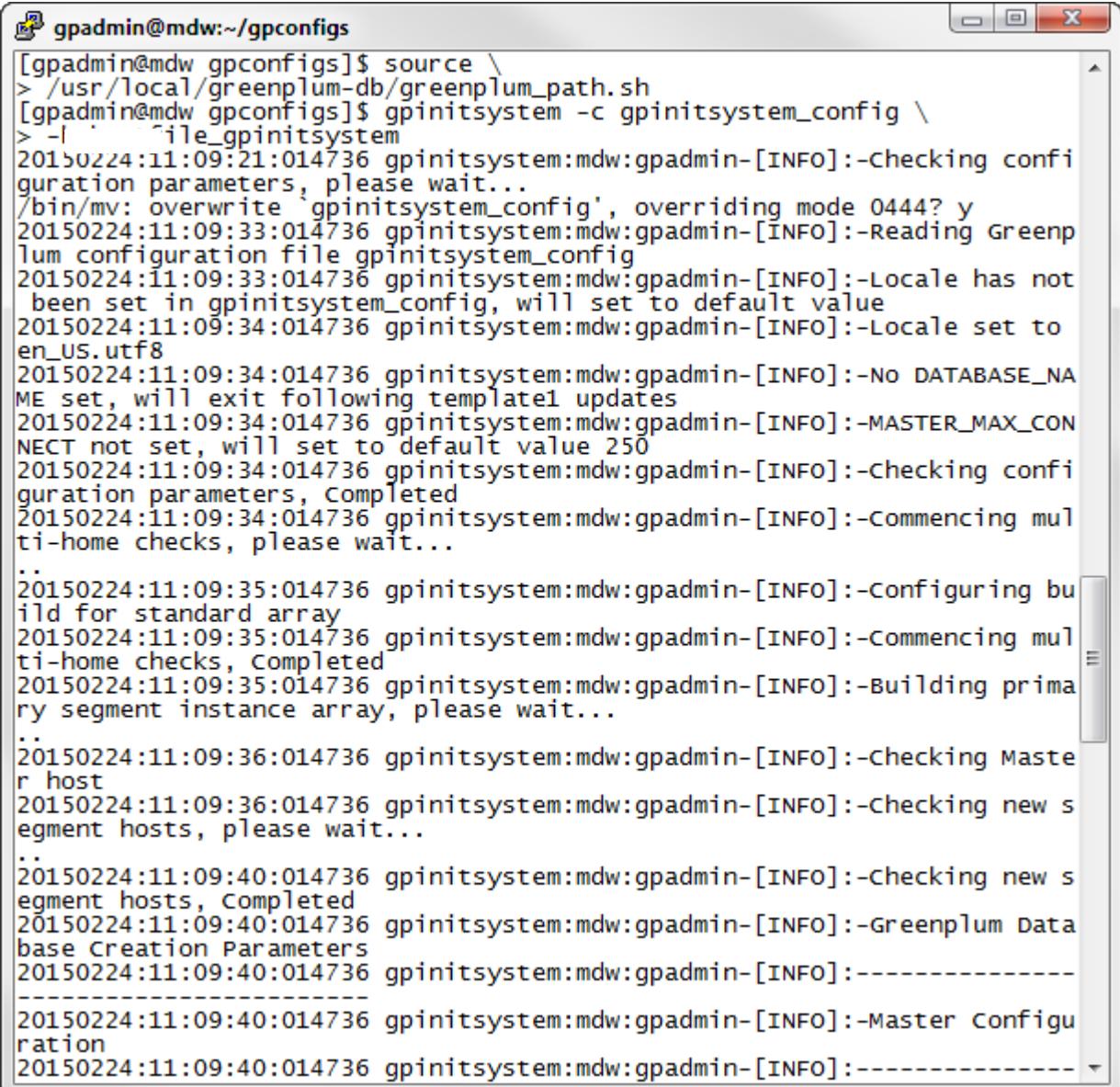
<b>Purpose:</b>	<p>You will perform the following installation and setup tasks necessary for the Greenplum Database software to run:</p> <ul style="list-style-type: none"><li>• Verify that the Greenplum Database software binaries are installed on all hosts.</li><li>• Ensure the required file systems are created on all hosts.</li><li>• Ensure the Greenplum administrator user account is created on all hosts.</li><li>• Verify that SSH keys are exchanged between all hosts.</li><li>• Set environment variables on the master as needed.</li></ul> <p>Many of these tasks are now automated, but you should continue to verify that the tasks have been completed. These installation tasks should be performed from your assigned master host.</p> <p>You will initialize a Greenplum Database array by examining the initialization configuration file and executing the <code>gpinitsystem</code> utility. You will also troubleshoot errors that may occur during initialization of a Greenplum Database array.</p>
<b>Tasks:</b>	<p>Students perform the following tasks:</p> <ol style="list-style-type: none"><li>1. Initialize the database with no mirrors and standby master using the <code>gpinitsystem</code> utility</li><li>2. Add mirrors with the <code>gpaddmirrors</code> utility</li><li>3. Add a standby master with the <code>gpinitstandby</code> utility</li><li>4. Delete the Greenplum database using the <code>gpdelete system</code> utility</li><li>5. Initialize a new Greenplum Database with mirrors and a standby master using the <code>gpinitsystem</code> utility.</li></ol>
<b>References:</b>	<p>Module 2 – Database Installation and Initialization</p> <ul style="list-style-type: none"><li>• Lesson: Greenplum Database Initialization</li></ul>

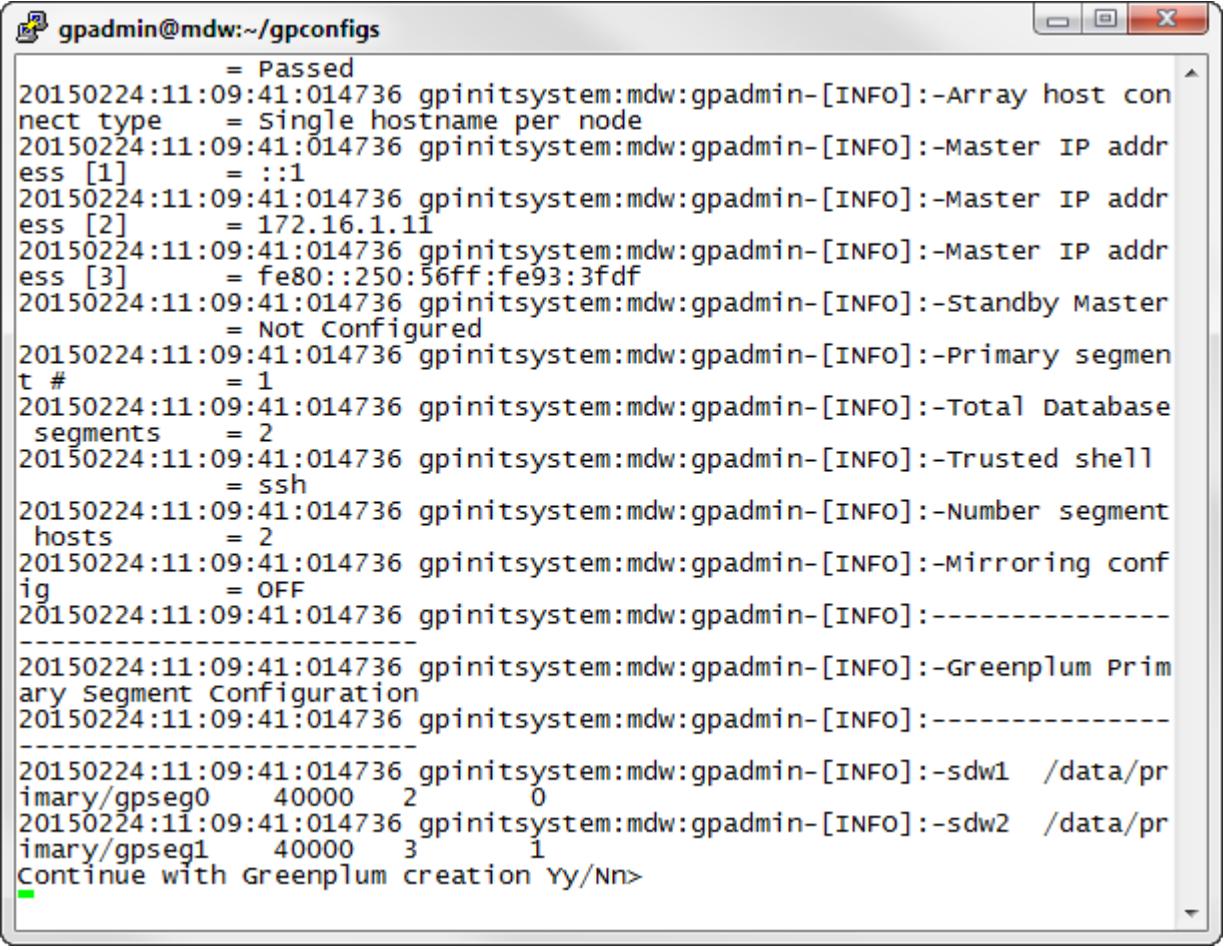
## Lab 3: Task 1 – Initialize the Database

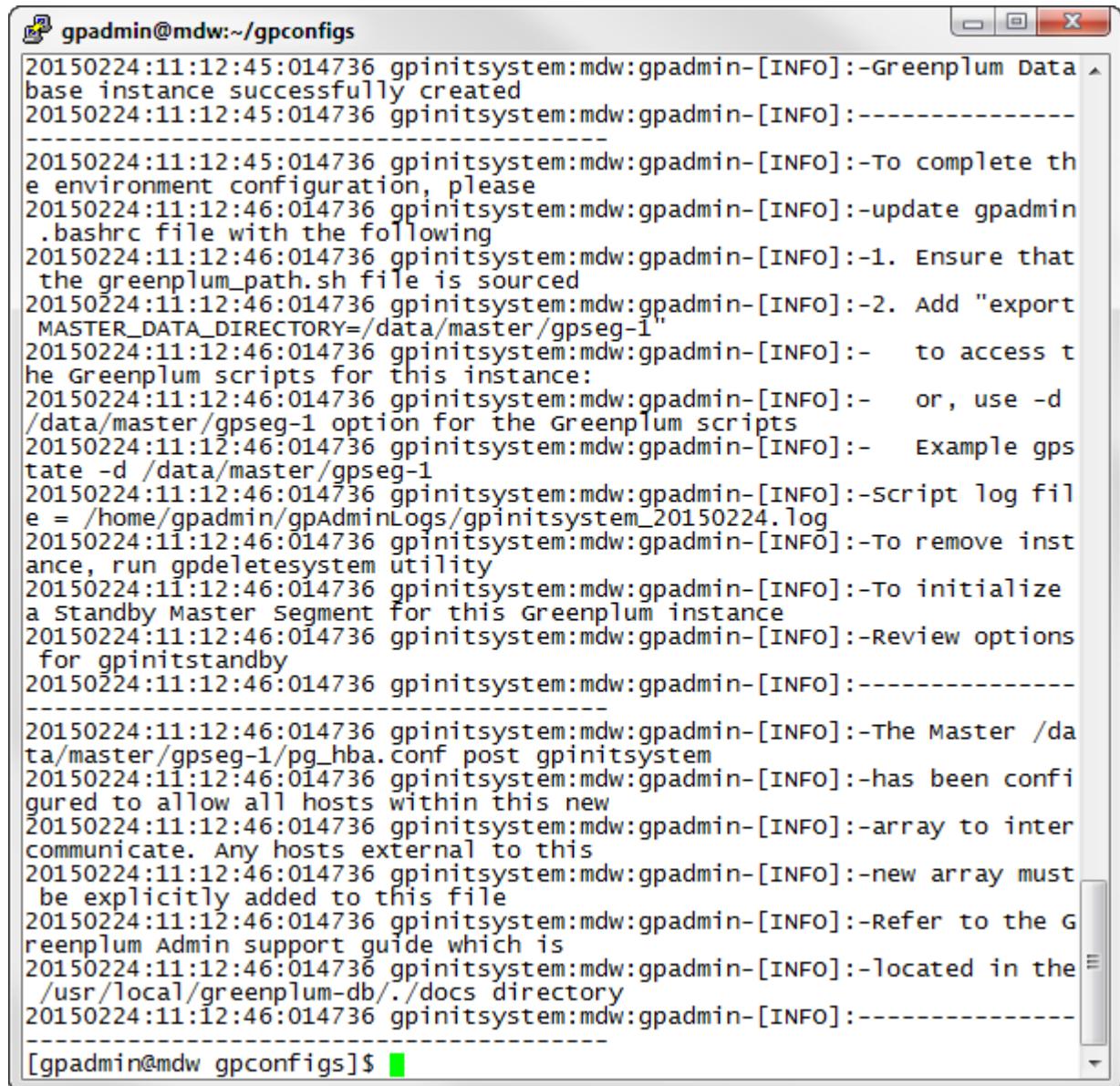
Step	Action
1.	If not already connected to mdw, open a terminal session to mdw and log in as root with the password, Pivotal. Switch to gpadmin user. [root@mdw ~]# <b>su - gpadmin</b>   A screenshot of a terminal window titled "gpadmin@mdw:~". The window shows the command [root@mdw ~]# su - gpadmin being entered, followed by a prompt [gpadmin@mdw ~]\$.
2.	Create a directory named gpconfigs in the home directory of the gpadmin user as shown. [gpadmin@mdw ~]\$ <b>mkdir gpconfigs</b>   A screenshot of a terminal window titled "gpadmin@mdw:~". The window shows the command [gpadmin@mdw ~]\$ mkdir gpconfigs being entered, followed by a prompt [gpadmin@mdw ~]\$.
3.	Copy the file /usr/local/greenplum-db/docs/cli_help/gpconfigs/gpinit_system_config into the directory /home/gpadmin/gpconfigs.  [gpadmin@mdw ~]\$ <b>cp \ /usr/local/greenplum-db/docs/cli_help/gpconfigs/gpinit_system_config ~/gpconfigs</b>   A screenshot of a terminal window titled "gpadmin@mdw:~". The window shows the command [gpadmin@mdw ~]\$ cp \ /usr/local/greenplum-db/docs/cli_help/gpconfigs/gpinit_system_config ~/gpconfigs being entered, followed by a prompt [gpadmin@mdw ~]\$.
4.	Change to the gpconfigs directory.  [gpadmin@mdw ~]\$ <b>cd gpconfigs</b>   A screenshot of a terminal window titled "gpadmin@mdw:~/gpconfigs". The window shows the command [gpadmin@mdw ~]\$ cd gpconfigs being entered, followed by a prompt [gpadmin@mdw gpconfigs]\$.

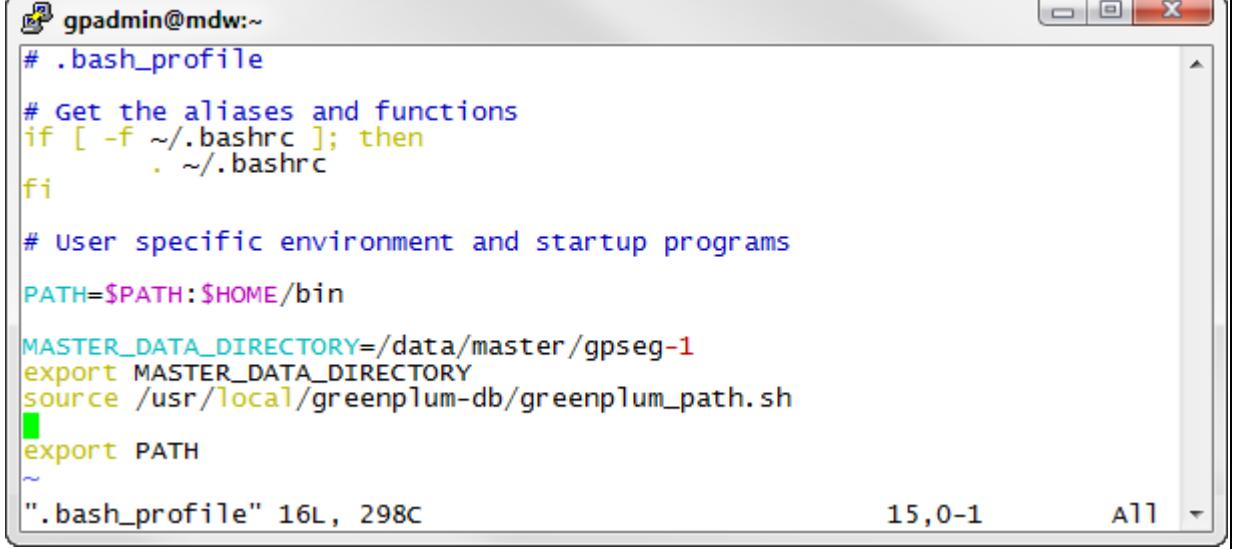
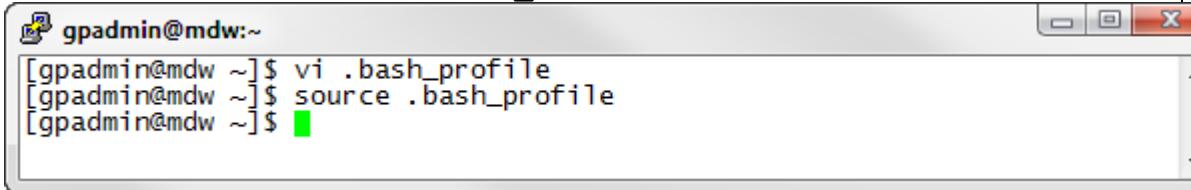
Step	Action
5.	<p>Create a copy of the segment only file you created earlier. This is simply for ease of use. This file should contain the hostname of all segment servers participating in the Greenplum environment. This file cannot contain any extra lines or spaces.</p> <pre>[gpadmin@mdw ~]\$ cp /rawdata/Binaries/hostfile_gpssh_seonly \ hostfile_gpinitSystem</pre> 
6.	<p>Using vi, edit the gpinitSystem_config file. Locate the 'Required Parameters' section and replace the highlighted section with the following:</p> <pre>declare -a DATA_DIRECTORY=(/data/primary)</pre> 

Step	Action
7.	<p>The final file appears as follows:</p>  <pre> gpadmin@mdw:~/gpconfigs  # Configuration file needed by the gpinit system ##### ##### REQUIRED PARAMETERS #####  ##### Name of this Greenplum system enclosed in quotes. ARRAY_NAME="EMC Greenplum DW"  ##### Naming convention for utility-generated data directories. SEG_PREFIX=gpseg  ##### Base number by which primary segment port numbers ##### are calculated. PORT_BASE=40000  ##### File system location(s) where primary segment data directories ##### will be created. The number of locations in the list dictate ##### the number of primary segments that will get created per ##### physical host (if multiple addresses for a host are listed in ##### the hostfile, the number of segments will be spread evenly across ##### the specified interface addresses). declare -a DATA_DIRECTORY=(/data/primary)  ##### OS-configured hostname or IP address of the master host. MASTER_HOSTNAME=mdw  ##### File system location where the master data directory :wq! </pre> <p>This is a read-only file. Use :wq! to save the file.</p> <p>Verify there are no comment symbols (#) in front of the line you are adding or replacing. If so, remove the comment symbol. Failure to do so may result in the initialization failing.</p>

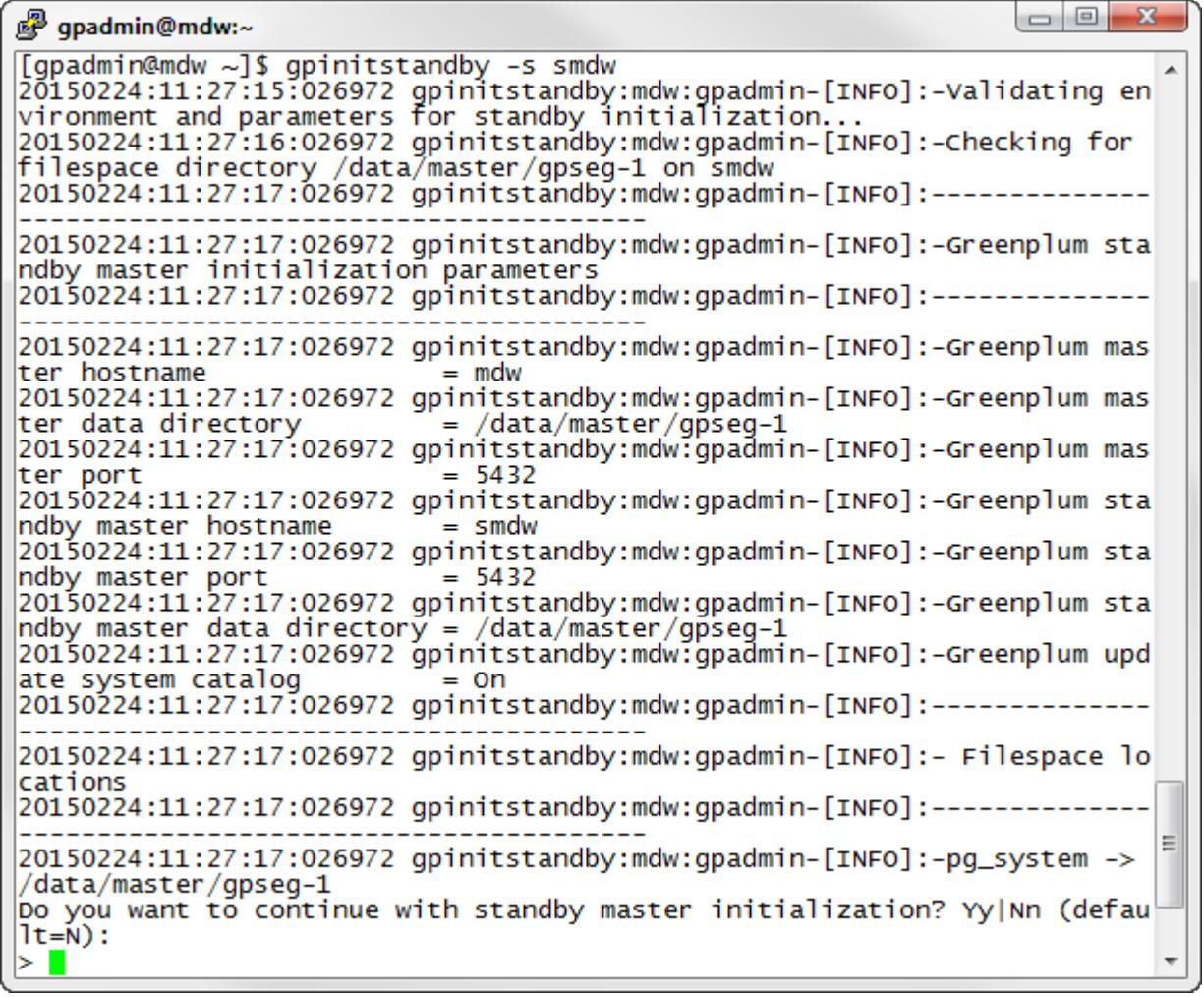
Step	Action
8.	<p>Run the gpinit system utility to create a Greenplum Database system using the values defined in gpinit system_config.j</p> <pre>[gpadmin@mdw gpconfigs]\$ source \ /usr/local/greenplum-db/greenplum_path.sh [gpadmin@mdw gpconfigs]\$ gpinit system -c gpinit system_config \ -h hostfile_gpinit system</pre>  <p>The utility will ask your permission to overwrite the configuration file. Confirm by pressing <b>y</b> and press <b>ENTER</b>.</p>

Step	Action
9.	<p>The utility displays the master, and primary segments configuration. It also asks you whether or not you want to continue with the Greenplum creation.</p>  <pre> gpadmin@mdw:~/gpconfigs       = Passed 20150224:11:09:41:014736 gpinitconfig:mdw:gpadmin-[INFO]:-Array host connect type      = Single hostname per node 20150224:11:09:41:014736 gpinitconfig:mdw:gpadmin-[INFO]:-Master IP address [1]      = ::1 20150224:11:09:41:014736 gpinitconfig:mdw:gpadmin-[INFO]:-Master IP address [2]      = 172.16.1.11 20150224:11:09:41:014736 gpinitconfig:mdw:gpadmin-[INFO]:-Master IP address [3]      = fe80::250:56ff:fe93:3fdf 20150224:11:09:41:014736 gpinitconfig:mdw:gpadmin-[INFO]:-Standby Master      = Not Configured 20150224:11:09:41:014736 gpinitconfig:mdw:gpadmin-[INFO]:-Primary segment #      = 1 20150224:11:09:41:014736 gpinitconfig:mdw:gpadmin-[INFO]:-Total database segments      = 2 20150224:11:09:41:014736 gpinitconfig:mdw:gpadmin-[INFO]:-Trusted shell      = ssh 20150224:11:09:41:014736 gpinitconfig:mdw:gpadmin-[INFO]:-Number segment hosts      = 2 20150224:11:09:41:014736 gpinitconfig:mdw:gpadmin-[INFO]:-Mirroring config      = OFF 20150224:11:09:41:014736 gpinitconfig:mdw:gpadmin-[INFO]:----- -----[REDACTED] 20150224:11:09:41:014736 gpinitconfig:mdw:gpadmin-[INFO]:-Greenplum Primary Segment Configuration 20150224:11:09:41:014736 gpinitconfig:mdw:gpadmin-[INFO]:----- -----[REDACTED] 20150224:11:09:41:014736 gpinitconfig:mdw:gpadmin-[INFO]:-sdw1 /data/primary/gpseg0 40000 2 0 20150224:11:09:41:014736 gpinitconfig:mdw:gpadmin-[INFO]:-sdw2 /data/primary/gpseg1 40000 3 1 Continue with Greenplum creation Yy/Nn&gt; </pre> <p>Press <b>y</b> followed by <b>Enter</b> to continue.</p>

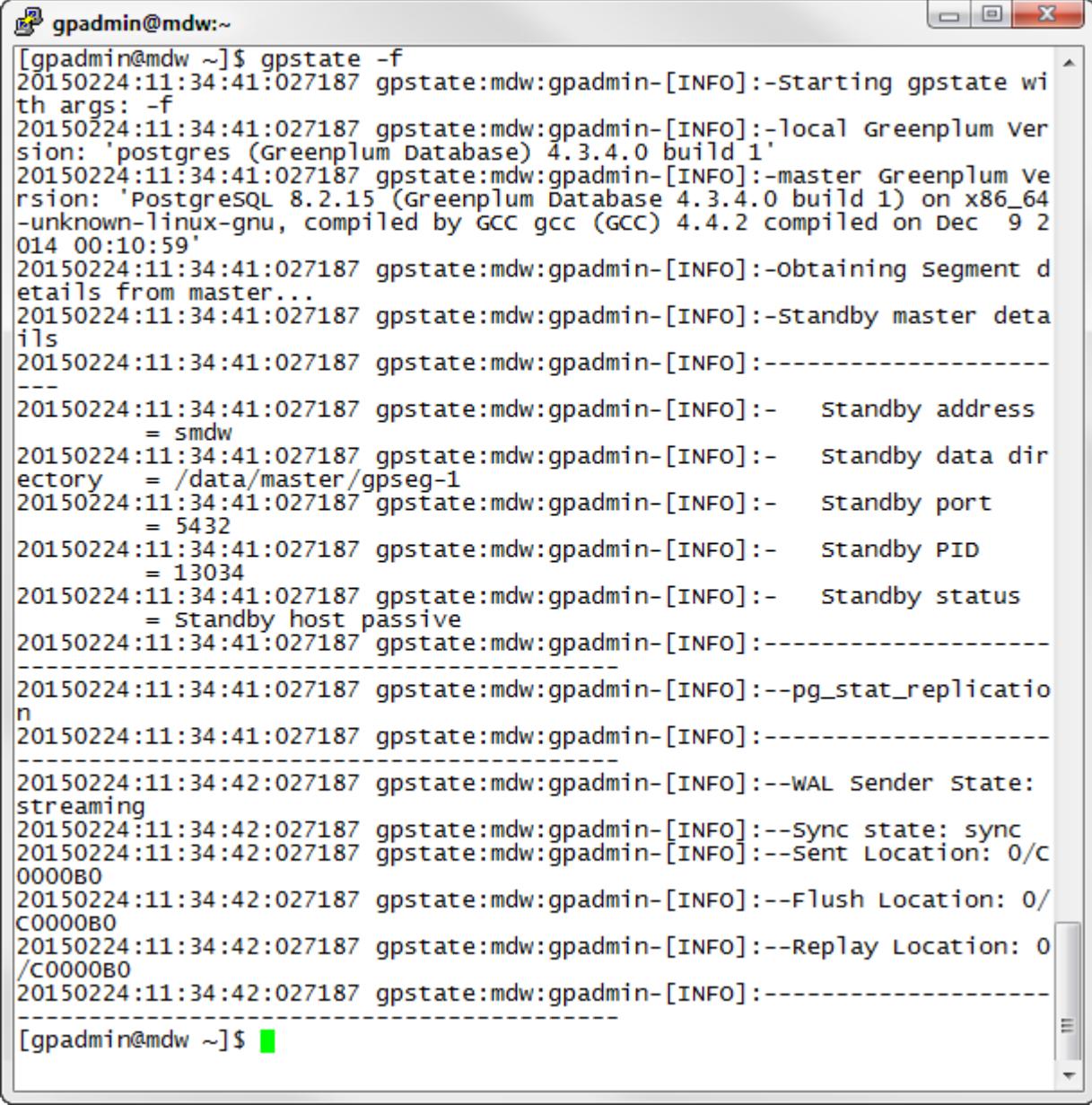
Step	Action
10.	<p>This screen shows that the gpinit system ended with no errors. This will take a few minutes.</p>  <pre data-bbox="319 219 1547 1410"> gpadmin@gpadmin@mdw:~/gpconfigs 20150224:11:12:45:014736 gpinitsystem:mdw:gpadmin-[INFO]:-Greenplum Data base instance successfully created 20150224:11:12:45:014736 gpinitsystem:mdw:gpadmin-[INFO]:----- 20150224:11:12:45:014736 gpinitsystem:mdw:gpadmin-[INFO]:-To complete th e environment configuration, please 20150224:11:12:46:014736 gpinitsystem:mdw:gpadmin-[INFO]:-update gpadmin .bashrc file with the following 20150224:11:12:46:014736 gpinitsystem:mdw:gpadmin-[INFO]:-1. Ensure that the greenplum_path.sh file is sourced 20150224:11:12:46:014736 gpinitsystem:mdw:gpadmin-[INFO]:-2. Add "export MASTER_DATA_DIRECTORY=/data/master/gpseg-1" 20150224:11:12:46:014736 gpinitsystem:mdw:gpadmin-[INFO]:- to access t he Greenplum scripts for this instance: 20150224:11:12:46:014736 gpinitsystem:mdw:gpadmin-[INFO]:- or, use -d /data/master/gpseg-1 option for the Greenplum scripts 20150224:11:12:46:014736 gpinitsystem:mdw:gpadmin-[INFO]:- Example gps tate -d /data/master/gpseg-1 20150224:11:12:46:014736 gpinitsystem:mdw:gpadmin-[INFO]:-script log fil e = /home/gpadmin/gpAdminLogs/gpinitsystem_20150224.log 20150224:11:12:46:014736 gpinitsystem:mdw:gpadmin-[INFO]:-To remove inst ance, run gpdeleteutility 20150224:11:12:46:014736 gpinitsystem:mdw:gpadmin-[INFO]:-To initialize a Standby Master Segment for this Greenplum instance 20150224:11:12:46:014736 gpinitsystem:mdw:gpadmin-[INFO]:-Review options for gpinitstandby 20150224:11:12:46:014736 gpinitsystem:mdw:gpadmin-[INFO]:----- 20150224:11:12:46:014736 gpinitsystem:mdw:gpadmin-[INFO]:-The Master /da ta/master/gpseg-1/pg_hba.conf post gpinitsystem 20150224:11:12:46:014736 gpinitsystem:mdw:gpadmin-[INFO]:-has been config ured to allow all hosts within this new 20150224:11:12:46:014736 gpinitsystem:mdw:gpadmin-[INFO]:-array to inter communicate. Any hosts external to this 20150224:11:12:46:014736 gpinitsystem:mdw:gpadmin-[INFO]:-new array must be explicitly added to this file 20150224:11:12:46:014736 gpinitsystem:mdw:gpadmin-[INFO]:-Refer to the G reenplum Admin support guide which is 20150224:11:12:46:014736 gpinitsystem:mdw:gpadmin-[INFO]:-located in the /usr/local/greenplum-db/.docs directory 20150224:11:12:46:014736 gpinitsystem:mdw:gpadmin-[INFO]:----- [gpadmin@gpadmin gpconfigs]\$ </pre>

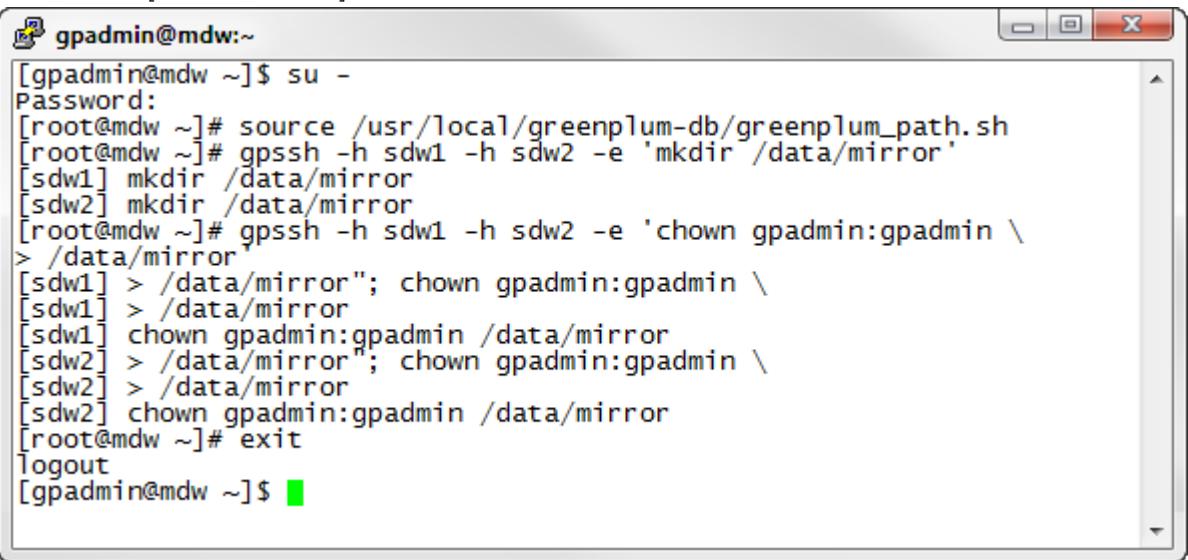
Step	Action
11.	<p>Perform the following steps on the master server (mdw):</p> <ol style="list-style-type: none"> <li>Change to the gpadmin home directory, /home/gpadmin, and using the vi editor, edit the file /home/gpadmin/.bash_profile.</li> </ol> <pre>[gpadmin@mdw gpconfigs] \$ cd gpadmin@mdw:~</pre>  <ol style="list-style-type: none"> <li>Add the following entries to the bottom of the file using vi:</li> </ol> <pre>MASTER_DATA_DIRECTORY=/data/master/gpseg-1 export MASTER_DATA_DIRECTORY source /usr/local/greenplum-db/greenplum_path.sh</pre> <pre># .bash_profile # Get the aliases and functions if [ -f ~/.bashrc ]; then     . ~/.bashrc fi  # User specific environment and startup programs PATH=\$PATH:\$HOME/bin  MASTER_DATA_DIRECTORY=/data/master/gpseg-1 export MASTER_DATA_DIRECTORY source /usr/local/greenplum-db/greenplum_path.sh export PATH ~  ".bash_profile" 16L, 298C</pre>  <ol style="list-style-type: none"> <li>Run the command below to make the changes active immediately.</li> </ol> <pre>[gpadmin@mdw ~] \$ source .bash_profile</pre> 
12.	Execute the gpstate utility to verify the Greenplum instance status summary.

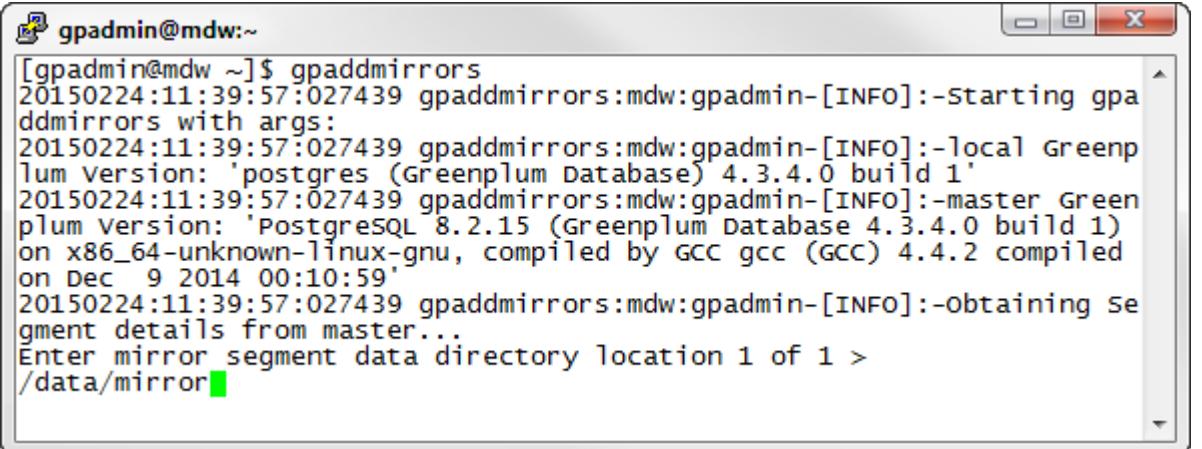
Step	Action
	<pre>[gpadmin@mdw ~]\$ gpstate 20150224:11:21:21:026793 gpstate:mdw:gpadmin-[INFO]:-Starting gpstate with args: 20150224:11:21:22:026793 gpstate:mdw:gpadmin-[INFO]:-local Greenplum version: 'postgres (Greenplum Database) 4.3.4.0 build 1' 20150224:11:21:22:026793 gpstate:mdw:gpadmin-[INFO]:-master Greenplum version: 'PostgreSQL 8.2.15 (Greenplum Database 4.3.4.0 build 1) on x86_64-unknown-linux-gnu, compiled by GCC gcc (GCC) 4.4.2 compiled on Dec 9 2014 00:10:59' 20150224:11:21:22:026793 gpstate:mdw:gpadmin-[INFO]:-obtaining Segment details from master... 20150224:11:21:22:026793 gpstate:mdw:gpadmin-[INFO]:-Gathering data from segments... . 20150224:11:21:23:026793 gpstate:mdw:gpadmin-[INFO]:-Greenplum instance status summary 20150224:11:21:23:026793 gpstate:mdw:gpadmin-[INFO]:----- 20150224:11:21:23:026793 gpstate:mdw:gpadmin-[INFO]:- Master instance = Active 20150224:11:21:23:026793 gpstate:mdw:gpadmin-[INFO]:- Master standby = No master standby configured 20150224:11:21:23:026793 gpstate:mdw:gpadmin-[INFO]:- Total segment instance count from metadata = 2 20150224:11:21:23:026793 gpstate:mdw:gpadmin-[INFO]:----- 20150224:11:21:23:026793 gpstate:mdw:gpadmin-[INFO]:- Primary Segment Status 20150224:11:21:23:026793 gpstate:mdw:gpadmin-[INFO]:----- 20150224:11:21:23:026793 gpstate:mdw:gpadmin-[INFO]:- Total primary segments = 2 20150224:11:21:23:026793 gpstate:mdw:gpadmin-[INFO]:- Total primary segment valid (at master) = 2 20150224:11:21:23:026793 gpstate:mdw:gpadmin-[INFO]:- Total primary segment failures (at master) = 0 20150224:11:21:23:026793 gpstate:mdw:gpadmin-[INFO]:- Total number of postmaster.pid files missing = 0 20150224:11:21:23:026793 gpstate:mdw:gpadmin-[INFO]:- Total number of postmaster.pid files found = 2 20150224:11:21:23:026793 gpstate:mdw:gpadmin-[INFO]:- Total number of postmaster.pid missing = 0 20150224:11:21:23:026793 gpstate:mdw:gpadmin-[INFO]:- Total number of postmaster.pid found = 2 20150224:11:21:23:026793 gpstate:mdw:gpadmin-[INFO]:- Total number of /tmp lock files missing = 0 20150224:11:21:23:026793 gpstate:mdw:gpadmin-[INFO]:- Total number of /tmp lock files found = 2 20150224:11:21:23:026793 gpstate:mdw:gpadmin-[INFO]:- Total number pos tmaster processes missing = 0 20150224:11:21:23:026793 gpstate:mdw:gpadmin-[INFO]:- Total number pos tmaster processes found = 2 20150224:11:21:23:026793 gpstate:mdw:gpadmin-[INFO]:----- 20150224:11:21:23:026793 gpstate:mdw:gpadmin-[INFO]:- Mirror Segment status 20150224:11:21:23:026793 gpstate:mdw:gpadmin-[INFO]:----- 20150224:11:21:23:026793 gpstate:mdw:gpadmin-[INFO]:- Mirrors not configured on this array 20150224:11:21:23:026793 gpstate:mdw:gpadmin-[INFO]:----- [gpadmin@mdw ~]\$</pre>

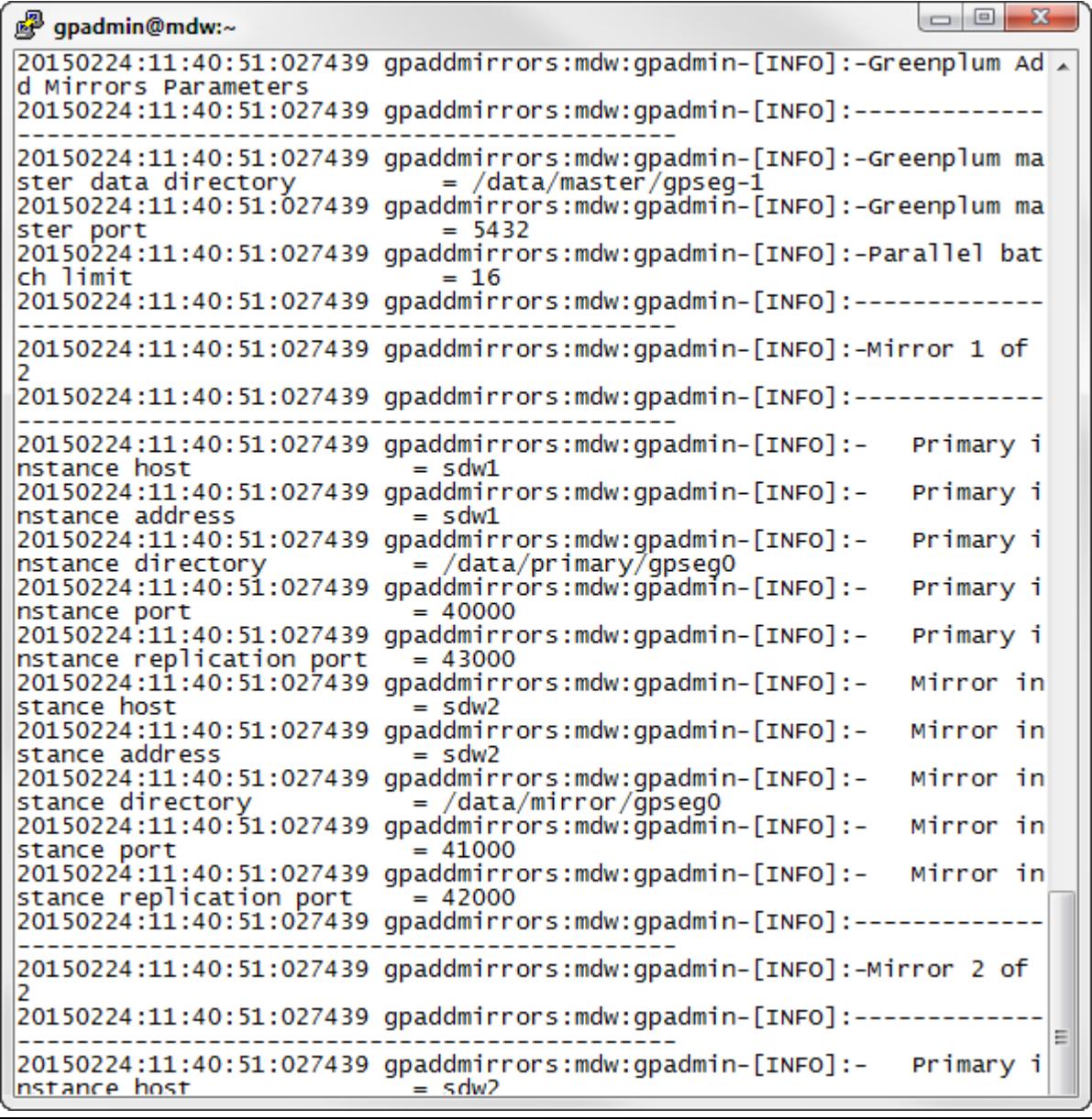
Step	Action
13.	<p>The next step is to manually add a standby master to the Greenplum Database that you just created. Run the <b>gpinitstandby</b> utility to do this.</p> <pre data-bbox="328 264 1003 295">[gpadmin@mdw ]\$ gpinitstandby -s smdw</pre>  <p>Press <b>y</b> to continue.</p>

Step	Action
14.	<p>This screen shows that the gpinitstandby ended with no errors.</p>  <pre> gpadmin@mdw:~\$ gpinitstandby 20150224:11:28:09:026972 gpinitstandby:mdw:gpadmin-[INFO]:-Syncing Green plum Database extensions to standby 20150224:11:28:12:026972 gpinitstandby:mdw:gpadmin-[INFO]:-The packages on smdw are consistent. 20150224:11:28:12:026972 gpinitstandby:mdw:gpadmin-[INFO]:-Updating pg_h ba.conf file... 20150224:11:28:13:026972 gpinitstandby:mdw:gpadmin-[INFO]:-Updating pg_h ba.conf file on segments... 20150224:11:28:18:026972 gpinitstandby:mdw:gpadmin-[INFO]:-Adding standb y master to catalog... 20150224:11:28:18:026972 gpinitstandby:mdw:gpadmin-[INFO]:-Database cata log updated successfully. 20150224:11:28:20:026972 gpinitstandby:mdw:gpadmin-[INFO]:-Updating file space flat files 20150224:11:28:20:026972 gpinitstandby:mdw:gpadmin-[INFO]:-Updating file space flat files 20150224:11:28:20:026972 gpinitstandby:mdw:gpadmin-[INFO]:-Removing pg_h ba.conf backup... 20150224:11:28:21:026972 gpinitstandby:mdw:gpadmin-[INFO]:-Starting stan dby master 20150224:11:28:21:026972 gpinitstandby:mdw:gpadmin-[INFO]:-Checking if s tandby master is running on host: smdw in directory: /data/master/gpseg -1 20150224:11:28:25:026972 gpinitstandby:mdw:gpadmin-[INFO]:-Successfully created standby master on smdw [gpadmin@mdw ~]\$ </pre>

Step	Action
15.	<p>Run the <code>gpstate -f</code> ( the <code>-f</code> option displays the details of the standby master ( <code>smdw</code> ) utility to verify that the standby master was created and is synchronized with the master.</p> <pre>[gpadmin@mdw ~]\$ gpstate -f</pre>  <pre>gpadmin@mdw ~]\$ gpstate -f 20150224:11:34:41:027187 gpstate:mdw:gpadmin-[INFO]:-Starting gpstate with args: -f 20150224:11:34:41:027187 gpstate:mdw:gpadmin-[INFO]:-local Greenplum version: 'postgres (Greenplum Database) 4.3.4.0 build 1' 20150224:11:34:41:027187 gpstate:mdw:gpadmin-[INFO]:-master Greenplum version: 'PostgreSQL 8.2.15 (Greenplum Database 4.3.4.0 build 1) on x86_64-unknown-linux-gnu, compiled by GCC gcc (GCC) 4.4.2 compiled on Dec 9 2014 00:10:59' 20150224:11:34:41:027187 gpstate:mdw:gpadmin-[INFO]:-obtaining Segment details from master... 20150224:11:34:41:027187 gpstate:mdw:gpadmin-[INFO]:-standby master details 20150224:11:34:41:027187 gpstate:mdw:gpadmin-[INFO]:----- 20150224:11:34:41:027187 gpstate:mdw:gpadmin-[INFO]:- Standby address = smdw 20150224:11:34:41:027187 gpstate:mdw:gpadmin-[INFO]:- Standby data directory = /data/master/gpseg-1 20150224:11:34:41:027187 gpstate:mdw:gpadmin-[INFO]:- Standby port = 5432 20150224:11:34:41:027187 gpstate:mdw:gpadmin-[INFO]:- Standby PID = 13034 20150224:11:34:41:027187 gpstate:mdw:gpadmin-[INFO]:- Standby status = Standby host passive 20150224:11:34:41:027187 gpstate:mdw:gpadmin-[INFO]:----- 20150224:11:34:41:027187 gpstate:mdw:gpadmin-[INFO]:--pg_stat_replication 20150224:11:34:41:027187 gpstate:mdw:gpadmin-[INFO]:----- 20150224:11:34:42:027187 gpstate:mdw:gpadmin-[INFO]:--WAL Sender state: streaming 20150224:11:34:42:027187 gpstate:mdw:gpadmin-[INFO]:--Sync state: sync 20150224:11:34:42:027187 gpstate:mdw:gpadmin-[INFO]:--Sent Location: 0/c0000BO 20150224:11:34:42:027187 gpstate:mdw:gpadmin-[INFO]:--Flush Location: 0/c0000BO 20150224:11:34:42:027187 gpstate:mdw:gpadmin-[INFO]:--Replay Location: 0/c0000BO 20150224:11:34:42:027187 gpstate:mdw:gpadmin-[INFO]:----- [gpadmin@mdw ~]\$</pre>

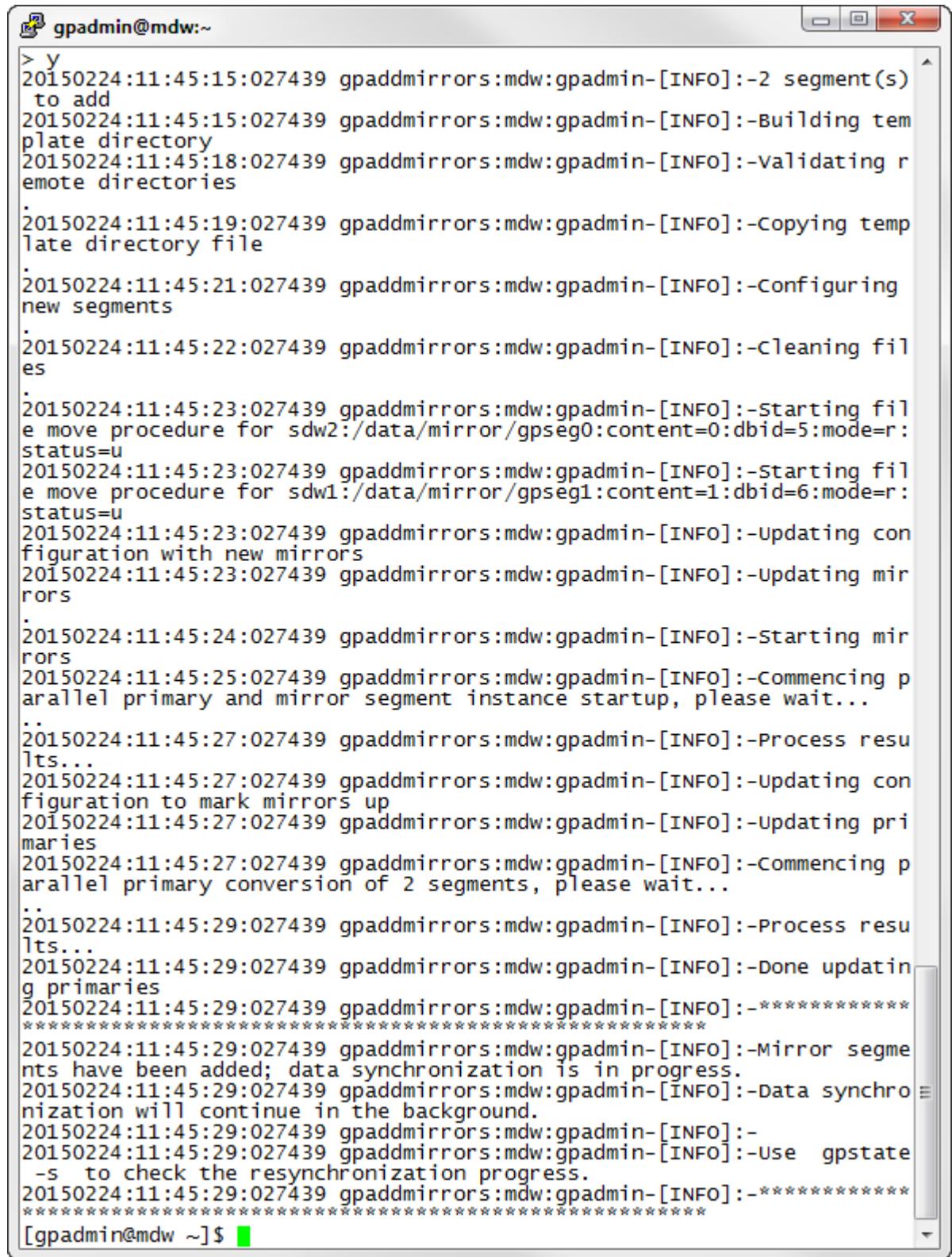
Step	Action
16.	<p>Copy the file /home/gpadmin/.bash_profile to smdw using the scp or gpscp command:</p> <pre data-bbox="331 228 1274 264">[gpadmin@mdw ~]\$ scp /home/gpadmin/.bash_profile smdw:</pre> 
17.	<p>Create the /data/mirror directory on all segment hosts within the cluster. After creating the directory, you must change the ownership of the directory so that it is owned by the gpadmin user account.</p> <ol style="list-style-type: none"> <li>1. Switch to the root user with the following command:  <code>[gpadmin@mdw ~]\$ su -</code> </li> <li>2. Enter the password, Pivotal.</li> <li>3. Source the greenplum_path.sh file to access the Greenplum binaries:  <code>[root@mdw ~]# source /usr/local/greenplum-db/greenplum_path.sh</code> </li> <li>4. Use gpssh to connect to sdw1 and sdw2 and create the mirror directory, /data/mirror:  <code>[root@mdw ~]# gpssh -h sdw1 -h sdw2 -e 'mkdir /data/mirror'</code> </li> <li>5. Change the ownership of these directories to gpadmin using the gpssh command:  <code>[root@mdw ~]# gpssh -h sdw1 -h sdw2 -e 'chown gpadmin:gpadmin /data/mirror'</code> <p>Note: Steps 4 and 5 can be combined into a single command by using a semicolon to separate the mkdir and chown statements.</p> </li> <li>6. Exit from the root account session:  <code>[root@mdw ~]# exit</code> </li> </ol> 

Step	Action
18.	<p>The next step is to manually add mirror segments to the Greenplum Database that you just created. Run the <b>gpaddmirrors</b> utility to do this.</p> <pre>[gpadmin@mdw ~]\$ <b>gpaddmirrors</b></pre> <p>When prompted, type in the mirror segment data directory and press <b>Enter</b>:</p> <pre>/data/mirror</pre> 

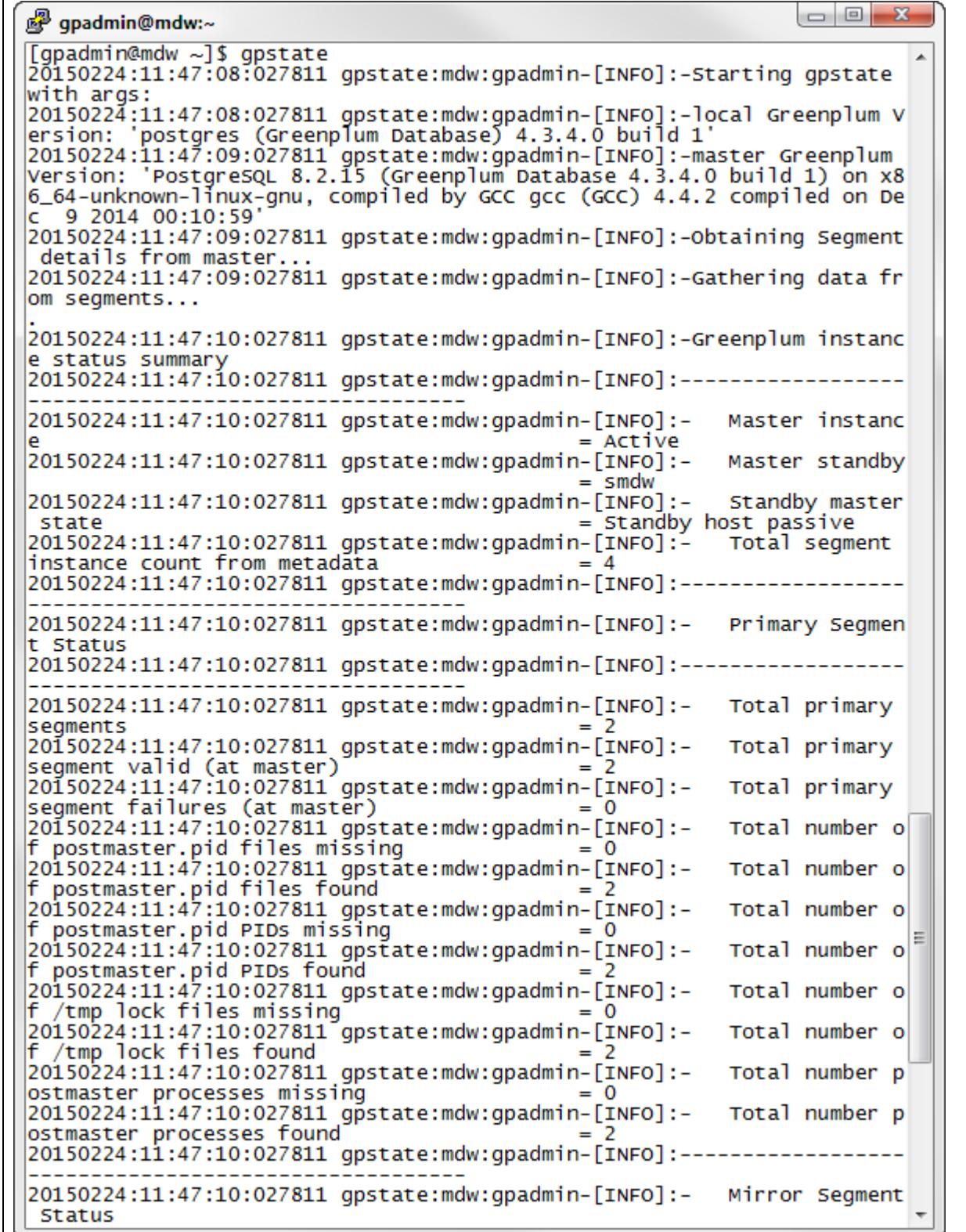
Step	Action
19.	<p>The following screen shows the configuration output from the gpaddmirrors utility:</p>  <pre data-bbox="328 228 1519 1450"> gpadmin@mdw:~\$ gpaddmirrors --list 20150224:11:40:51:027439 gpaddmirrors:mdw:gpadmin-[INFO]:-Greenplum Ad d Mirrors Parameters 20150224:11:40:51:027439 gpaddmirrors:mdw:gpadmin-[INFO]:----- 20150224:11:40:51:027439 gpaddmirrors:mdw:gpadmin-[INFO]:-Greenplum ma ster data directory          = /data/master/gpseg-1 20150224:11:40:51:027439 gpaddmirrors:mdw:gpadmin-[INFO]:-Greenplum ma ster port                   = 5432 20150224:11:40:51:027439 gpaddmirrors:mdw:gpadmin-[INFO]:-Parallel bat ch limit                   = 16 20150224:11:40:51:027439 gpaddmirrors:mdw:gpadmin-[INFO]:----- 20150224:11:40:51:027439 gpaddmirrors:mdw:gpadmin-[INFO]:-Mirror 1 of 2 20150224:11:40:51:027439 gpaddmirrors:mdw:gpadmin-[INFO]:----- 20150224:11:40:51:027439 gpaddmirrors:mdw:gpadmin-[INFO]:- Primary i nstance host                = sdw1 20150224:11:40:51:027439 gpaddmirrors:mdw:gpadmin-[INFO]:- Primary i nstance address              = sdw1 20150224:11:40:51:027439 gpaddmirrors:mdw:gpadmin-[INFO]:- Primary i nstance directory            = /data/primary/gpseg0 20150224:11:40:51:027439 gpaddmirrors:mdw:gpadmin-[INFO]:- Primary i nstance port                 = 40000 20150224:11:40:51:027439 gpaddmirrors:mdw:gpadmin-[INFO]:- Primary i nstance replication port     = 43000 20150224:11:40:51:027439 gpaddmirrors:mdw:gpadmin-[INFO]:- Mirror in stance host                  = sdw2 20150224:11:40:51:027439 gpaddmirrors:mdw:gpadmin-[INFO]:- Mirror in stance address                = sdw2 20150224:11:40:51:027439 gpaddmirrors:mdw:gpadmin-[INFO]:- Mirror in stance directory              = /data/mirror/gpseg0 20150224:11:40:51:027439 gpaddmirrors:mdw:gpadmin-[INFO]:- Mirror in stance port                   = 41000 20150224:11:40:51:027439 gpaddmirrors:mdw:gpadmin-[INFO]:- Mirror in stance replication port       = 42000 20150224:11:40:51:027439 gpaddmirrors:mdw:gpadmin-[INFO]:----- 20150224:11:40:51:027439 gpaddmirrors:mdw:gpadmin-[INFO]:-Mirror 2 of 2 20150224:11:40:51:027439 gpaddmirrors:mdw:gpadmin-[INFO]:----- 20150224:11:40:51:027439 gpaddmirrors:mdw:gpadmin-[INFO]:- Primary i nstance host                = sdw2 </pre>

Step	Action
	 gpadmin@mdw:~ <pre> instance host      = sdw2 20150224:11:40:51:027439 gpaddmirrors:mdw:gpadmin-[INFO]:: Primary i instance address   = sdw2 20150224:11:40:51:027439 gpaddmirrors:mdw:gpadmin-[INFO]:: Primary i instance directory = /data/primary/gpseg1 20150224:11:40:51:027439 gpaddmirrors:mdw:gpadmin-[INFO]:: Primary i instance port       = 40000 20150224:11:40:51:027439 gpaddmirrors:mdw:gpadmin-[INFO]:: Primary i instance replication port = 43000 20150224:11:40:51:027439 gpaddmirrors:mdw:gpadmin-[INFO]:: Mirror in instance host      = sdw1 20150224:11:40:51:027439 gpaddmirrors:mdw:gpadmin-[INFO]:: Mirror in instance address   = sdw1 20150224:11:40:51:027439 gpaddmirrors:mdw:gpadmin-[INFO]:: Mirror in instance directory = /data/mirror/gpseg1 20150224:11:40:51:027439 gpaddmirrors:mdw:gpadmin-[INFO]:: Mirror in instance port       = 41000 20150224:11:40:51:027439 gpaddmirrors:mdw:gpadmin-[INFO]:: Mirror in instance replication port = 42000 20150224:11:40:51:027439 gpaddmirrors:mdw:gpadmin-[INFO]::----- -----</pre> <p>Continue with add mirrors procedure Yy Nn (default=N):  &gt; █</p>

Press **y** to continue.

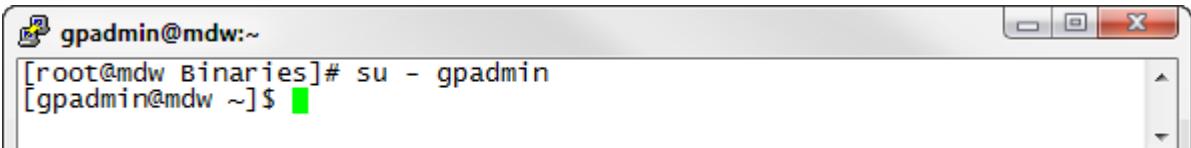
Step	Action
20.	<p>This screen shows that the gpaddmirrors ended with no errors.</p>  <pre data-bbox="319 219 1519 1803">&gt; y 20150224:11:45:15:027439 gpaddmirrors:mdw:gpadmin-[INFO]:-2 segment(s) to add 20150224:11:45:15:027439 gpaddmirrors:mdw:gpadmin-[INFO]:-Building tem plate directory 20150224:11:45:18:027439 gpaddmirrors:mdw:gpadmin-[INFO]:-validating r emote directories . 20150224:11:45:19:027439 gpaddmirrors:mdw:gpadmin-[INFO]:-Copying temp late directory file . 20150224:11:45:21:027439 gpaddmirrors:mdw:gpadmin-[INFO]:-Configuring new segments . 20150224:11:45:22:027439 gpaddmirrors:mdw:gpadmin-[INFO]:-cleaning fil es . 20150224:11:45:23:027439 gpaddmirrors:mdw:gpadmin-[INFO]:-starting fil e move procedure for sdw2:/data/mirror/gpseg0:content=0:dbid=5:mode=r: status=u 20150224:11:45:23:027439 gpaddmirrors:mdw:gpadmin-[INFO]:-starting fil e move procedure for sdw1:/data/mirror/gpseg1:content=1:dbid=6:mode=r: status=u 20150224:11:45:23:027439 gpaddmirrors:mdw:gpadmin-[INFO]:-Updating con figuration with new mirrors 20150224:11:45:23:027439 gpaddmirrors:mdw:gpadmin-[INFO]:-Updating mir rors . 20150224:11:45:24:027439 gpaddmirrors:mdw:gpadmin-[INFO]:-Starting mir rors 20150224:11:45:25:027439 gpaddmirrors:mdw:gpadmin-[INFO]:-Commencing p arallel primary and mirror segment instance startup, please wait... . 20150224:11:45:27:027439 gpaddmirrors:mdw:gpadmin-[INFO]:-Process resu lts... 20150224:11:45:27:027439 gpaddmirrors:mdw:gpadmin-[INFO]:-Updating con figuration to mark mirrors up 20150224:11:45:27:027439 gpaddmirrors:mdw:gpadmin-[INFO]:-Updating pri maries 20150224:11:45:27:027439 gpaddmirrors:mdw:gpadmin-[INFO]:-commencing p arallel primary conversion of 2 segments, please wait... . 20150224:11:45:29:027439 gpaddmirrors:mdw:gpadmin-[INFO]:-Process resu lts... 20150224:11:45:29:027439 gpaddmirrors:mdw:gpadmin-[INFO]:-Done updatin g primaries 20150224:11:45:29:027439 gpaddmirrors:mdw:gpadmin-[INFO]:-***** *****20150224:11:45:29:027439 gpaddmirrors:mdw:gpadmin-[INFO]:-Mirror segme nts have been added; data synchronization is in progress. 20150224:11:45:29:027439 gpaddmirrors:mdw:gpadmin-[INFO]:-Data synchro nization will continue in the background. 20150224:11:45:29:027439 gpaddmirrors:mdw:gpadmin-[INFO]:- 20150224:11:45:29:027439 gpaddmirrors:mdw:gpadmin-[INFO]:-Use gpstate -s to check the resynchronization progress. 20150224:11:45:29:027439 gpaddmirrors:mdw:gpadmin-[INFO]:-***** *****[gpadmin@mdw ~]\$</pre>

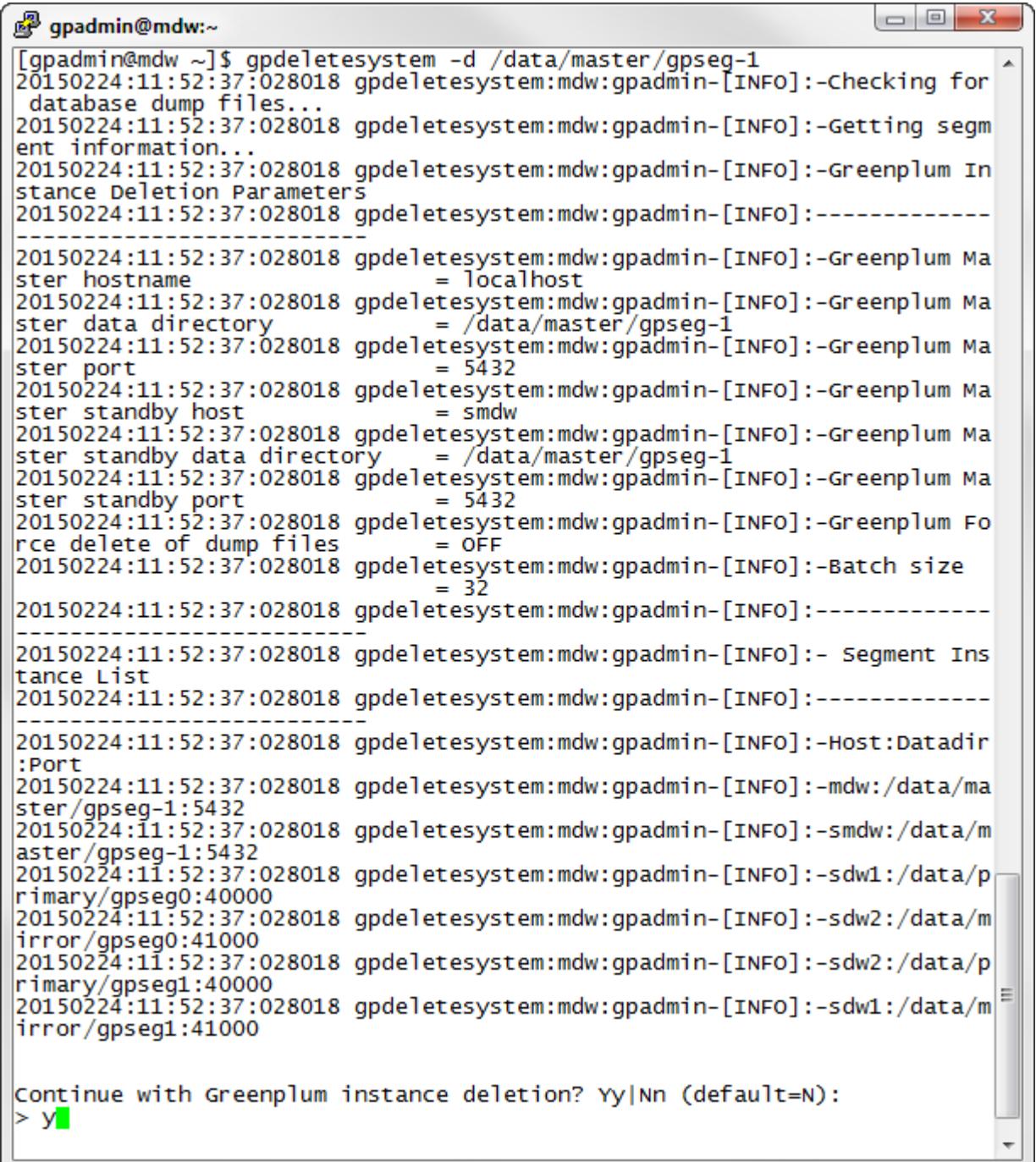
Step	Action
	<p>You can use the <code>gpstate -s</code> utility to check that the mirror segments are being synchronized with their primary segments. This utility will also indicate when the mirrors are fully synchronized.</p> <pre>[gpadmin@mdw ]\$ <b>gpstate -s</b></pre>

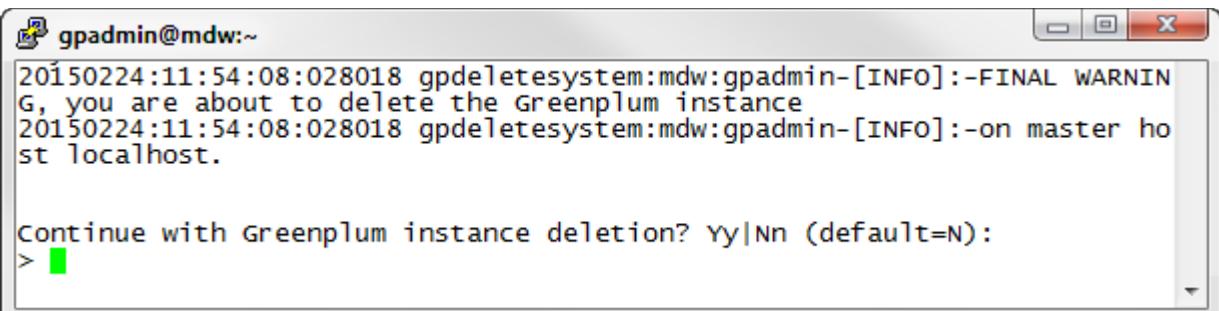
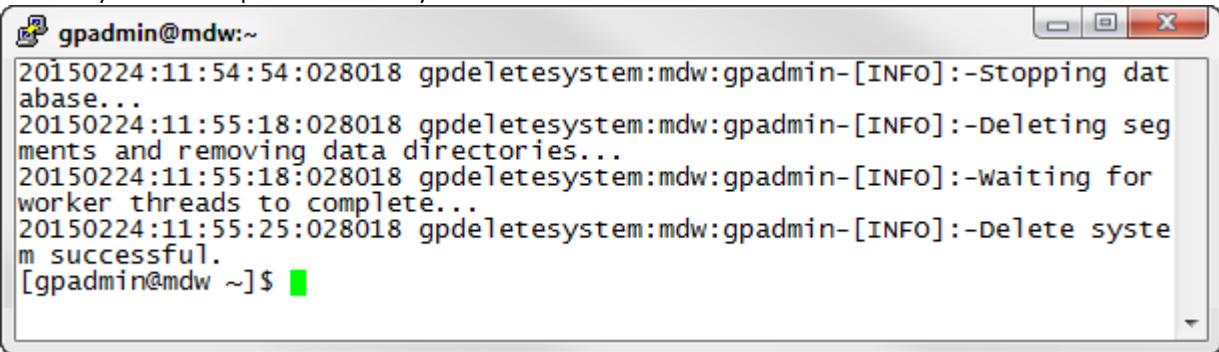
Step	Action
21.	<p>Run the gpstate utility to verify that the Greenplum instance status summary includes the new mirrors that were just added.</p>  <pre data-bbox="311 264 1545 1837"> gpadmin@mdw:~\$ gpstate 20150224:11:47:08:027811 gpstate:mdw:gpadmin-[INFO]:-Starting gpstate with args: 20150224:11:47:08:027811 gpstate:mdw:gpadmin-[INFO]:-local Greenplum v ersion: 'postgres (Greenplum Database) 4.3.4.0 build 1' 20150224:11:47:09:027811 gpstate:mdw:gpadmin-[INFO]:-master Greenplum Version: 'PostgreSQL 8.2.15 (Greenplum Database 4.3.4.0 build 1) on x8 6_64-unknown-linux-gnu, compiled by GCC gcc (GCC) 4.4.2 compiled on De c 9 2014 00:10:59' 20150224:11:47:09:027811 gpstate:mdw:gpadmin-[INFO]:-obtaining Segment details from master... 20150224:11:47:09:027811 gpstate:mdw:gpadmin-[INFO]:-Gathering data fr om segments... . 20150224:11:47:10:027811 gpstate:mdw:gpadmin-[INFO]:-Greenplum instanc e status summary 20150224:11:47:10:027811 gpstate:mdw:gpadmin-[INFO]:----- ----- 20150224:11:47:10:027811 gpstate:mdw:gpadmin-[INFO]:- Master instanc e = Active 20150224:11:47:10:027811 gpstate:mdw:gpadmin-[INFO]:- Master standby = smdw 20150224:11:47:10:027811 gpstate:mdw:gpadmin-[INFO]:- Standby master state = Standby host passive 20150224:11:47:10:027811 gpstate:mdw:gpadmin-[INFO]:- Total segment instance count from metadata = 4 20150224:11:47:10:027811 gpstate:mdw:gpadmin-[INFO]:----- ----- 20150224:11:47:10:027811 gpstate:mdw:gpadmin-[INFO]:- Primary Segmen t Status 20150224:11:47:10:027811 gpstate:mdw:gpadmin-[INFO]:----- ----- 20150224:11:47:10:027811 gpstate:mdw:gpadmin-[INFO]:- Total primary segments = 2 20150224:11:47:10:027811 gpstate:mdw:gpadmin-[INFO]:- Total primary segment valid (at master) = 2 20150224:11:47:10:027811 gpstate:mdw:gpadmin-[INFO]:- Total primary segment failures (at master) = 0 20150224:11:47:10:027811 gpstate:mdw:gpadmin-[INFO]:- Total number o f postmaster.pid files missing = 0 20150224:11:47:10:027811 gpstate:mdw:gpadmin-[INFO]:- Total number o f postmaster.pid files found = 2 20150224:11:47:10:027811 gpstate:mdw:gpadmin-[INFO]:- Total number o f postmaster.pid IDs missing = 0 20150224:11:47:10:027811 gpstate:mdw:gpadmin-[INFO]:- Total number o f postmaster.pid IDs found = 2 20150224:11:47:10:027811 gpstate:mdw:gpadmin-[INFO]:- Total number o f /tmp lock files missing = 0 20150224:11:47:10:027811 gpstate:mdw:gpadmin-[INFO]:- Total number o f /tmp lock files found = 2 20150224:11:47:10:027811 gpstate:mdw:gpadmin-[INFO]:- Total number p ostmaster processes missing = 0 20150224:11:47:10:027811 gpstate:mdw:gpadmin-[INFO]:- Total number p ostmaster processes found = 2 20150224:11:47:10:027811 gpstate:mdw:gpadmin-[INFO]:----- ----- 20150224:11:47:10:027811 gpstate:mdw:gpadmin-[INFO]:- Mirror Segment Status </pre>

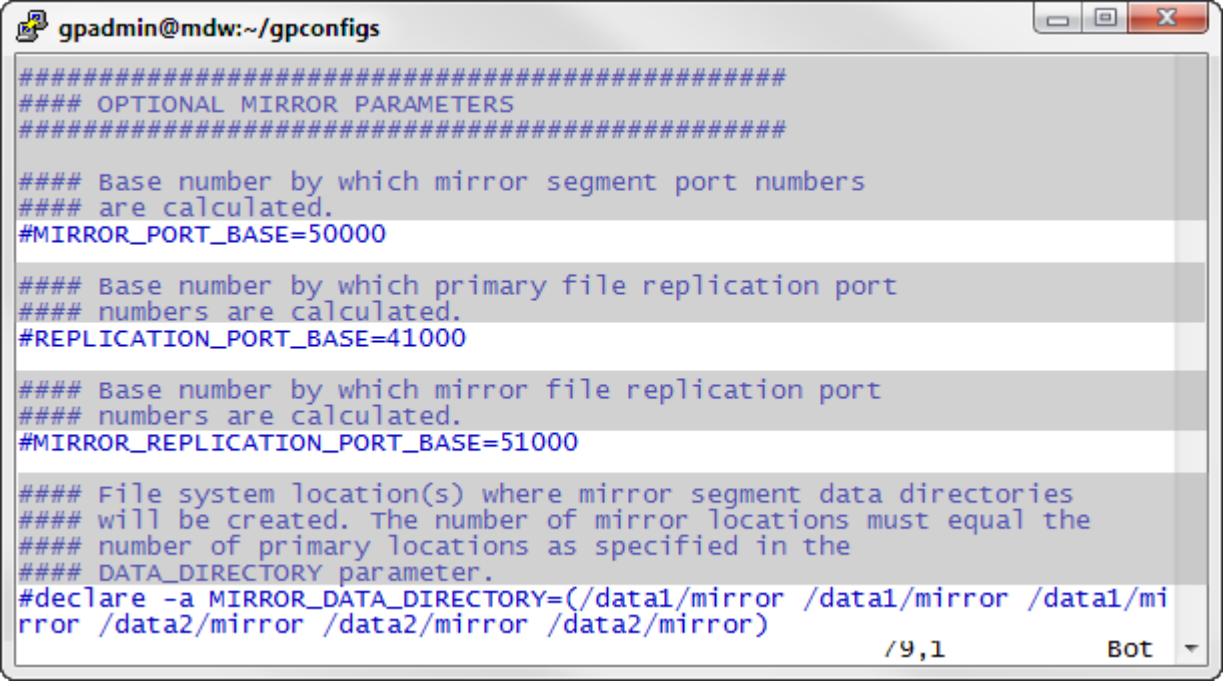
Step	Action
	<pre>gpadmin@mdw:~\$ gpstate 20150224:11:47:10:027811 gpstate:mdw:gpadmin-[INFO]:- Mirror Segment Status 20150224:11:47:10:027811 gpstate:mdw:gpadmin-[INFO]:----- 20150224:11:47:10:027811 gpstate:mdw:gpadmin-[INFO]:- Total mirror s egments = 2 20150224:11:47:10:027811 gpstate:mdw:gpadmin-[INFO]:- Total mirror s egment valid (at master) = 2 20150224:11:47:10:027811 gpstate:mdw:gpadmin-[INFO]:- Total mirror s egment failures (at master) = 0 20150224:11:47:10:027811 gpstate:mdw:gpadmin-[INFO]:- Total number o f postmaster.pid files missing = 0 20150224:11:47:10:027811 gpstate:mdw:gpadmin-[INFO]:- Total number o f postmaster.pid files found = 2 20150224:11:47:10:027811 gpstate:mdw:gpadmin-[INFO]:- Total number o f postmaster.pid PID missing = 0 20150224:11:47:10:027811 gpstate:mdw:gpadmin-[INFO]:- Total number o f postmaster.pid PIDs found = 2 20150224:11:47:10:027811 gpstate:mdw:gpadmin-[INFO]:- Total number o f /tmp lock files missing = 0 20150224:11:47:10:027811 gpstate:mdw:gpadmin-[INFO]:- Total number o f /tmp lock files found = 2 20150224:11:47:10:027811 gpstate:mdw:gpadmin-[INFO]:- Total number p ostmaster processes missing = 0 20150224:11:47:10:027811 gpstate:mdw:gpadmin-[INFO]:- Total number p ostmaster processes found = 2 20150224:11:47:10:027811 gpstate:mdw:gpadmin-[INFO]:- Total number m irror segments acting as primary segments = 0 20150224:11:47:10:027811 gpstate:mdw:gpadmin-[INFO]:- Total number m irror segments acting as mirror segments = 2 20150224:11:47:10:027811 gpstate:mdw:gpadmin-[INFO]:----- [gpadmin@mdw ~]\$</pre>

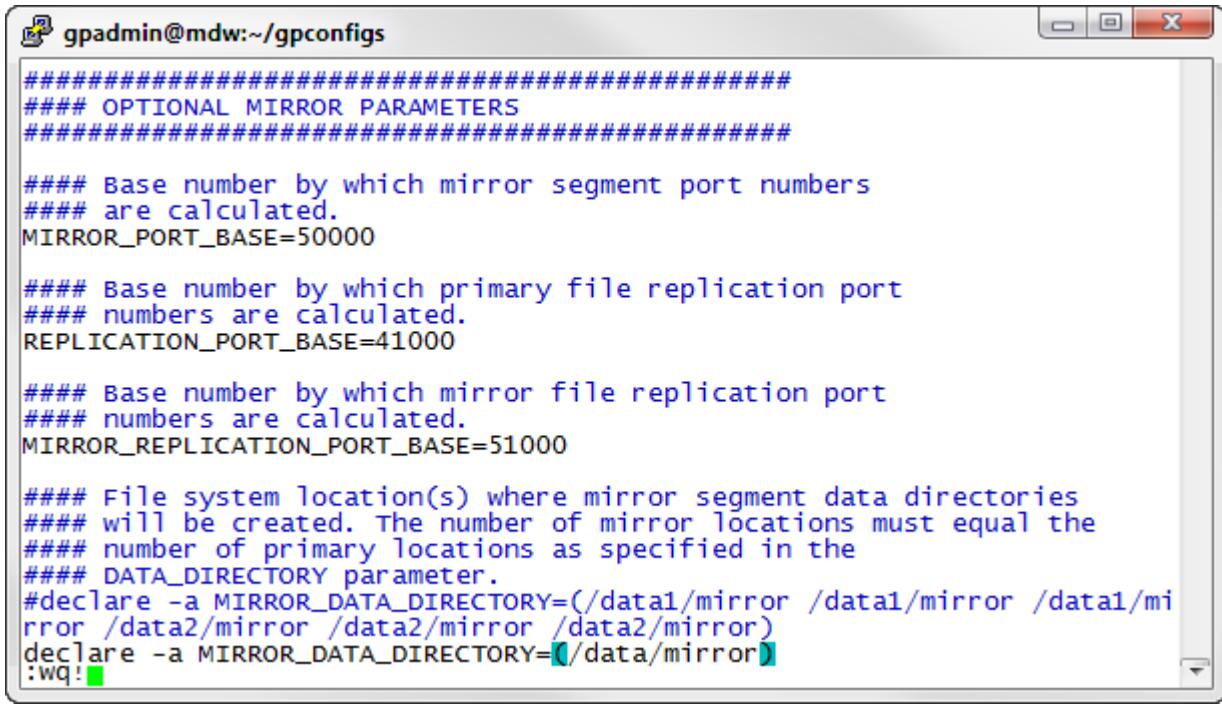
## Lab 3: Task 2 – Delete the Database and Initialize a New Database with One Command

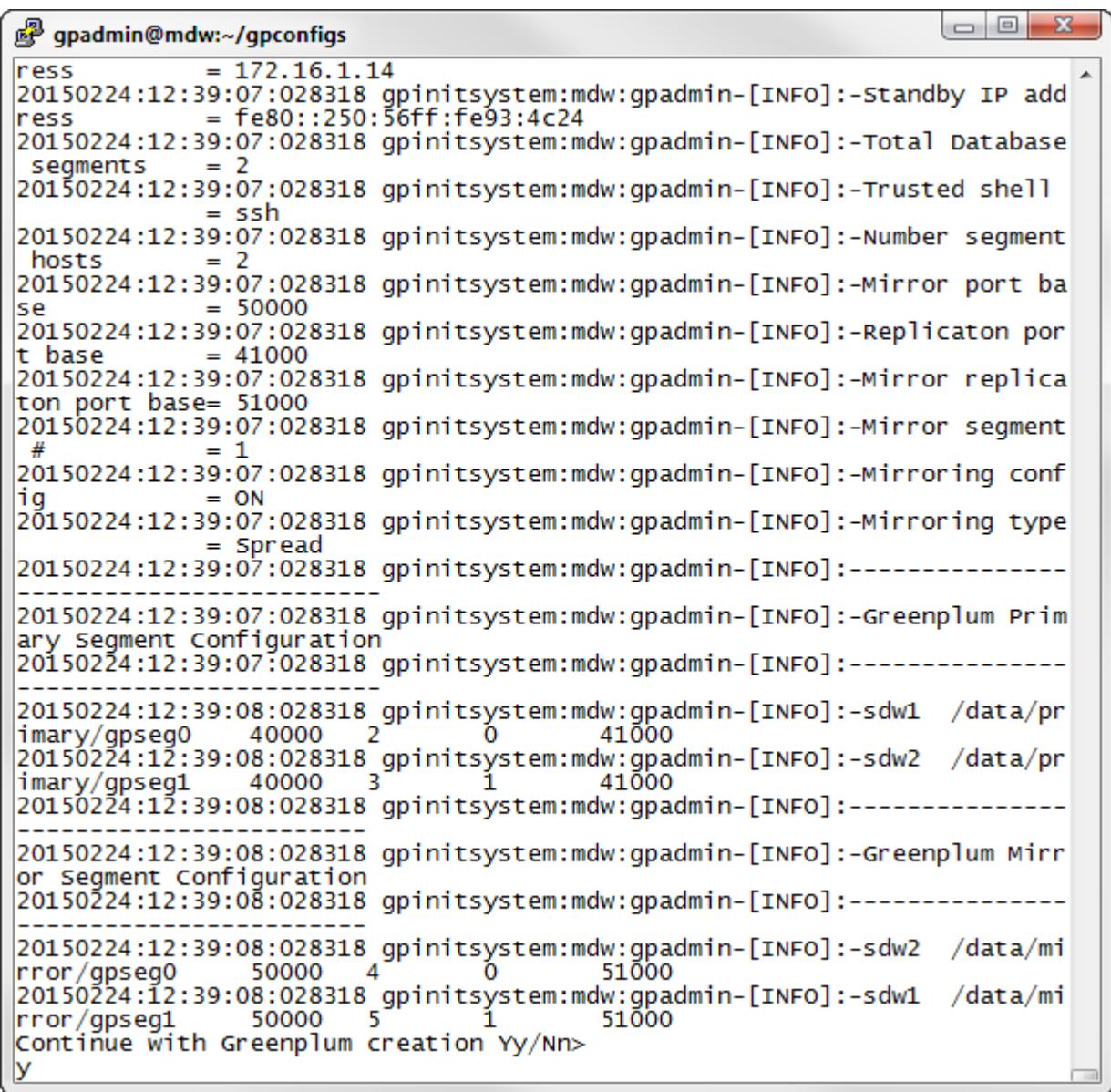
Step	Action
1.	<p>If not already connected to mdw, open a terminal session to mdw and log in as root with the password, Piv0tal.</p> <p>Switch to gpadmin user. If you were already in the gpadmin session from the previous lab task, you can skip this step.</p> <pre>[root@mdw ~]# su - gpadmin</pre>  A screenshot of a terminal window titled "gpadmin@mdw:~". The window contains the command "[root@mdw Binaries]# su - gpadmin" and its output "[gpadmin@mdw ~]\$". The terminal has a standard window frame with minimize, maximize, and close buttons.

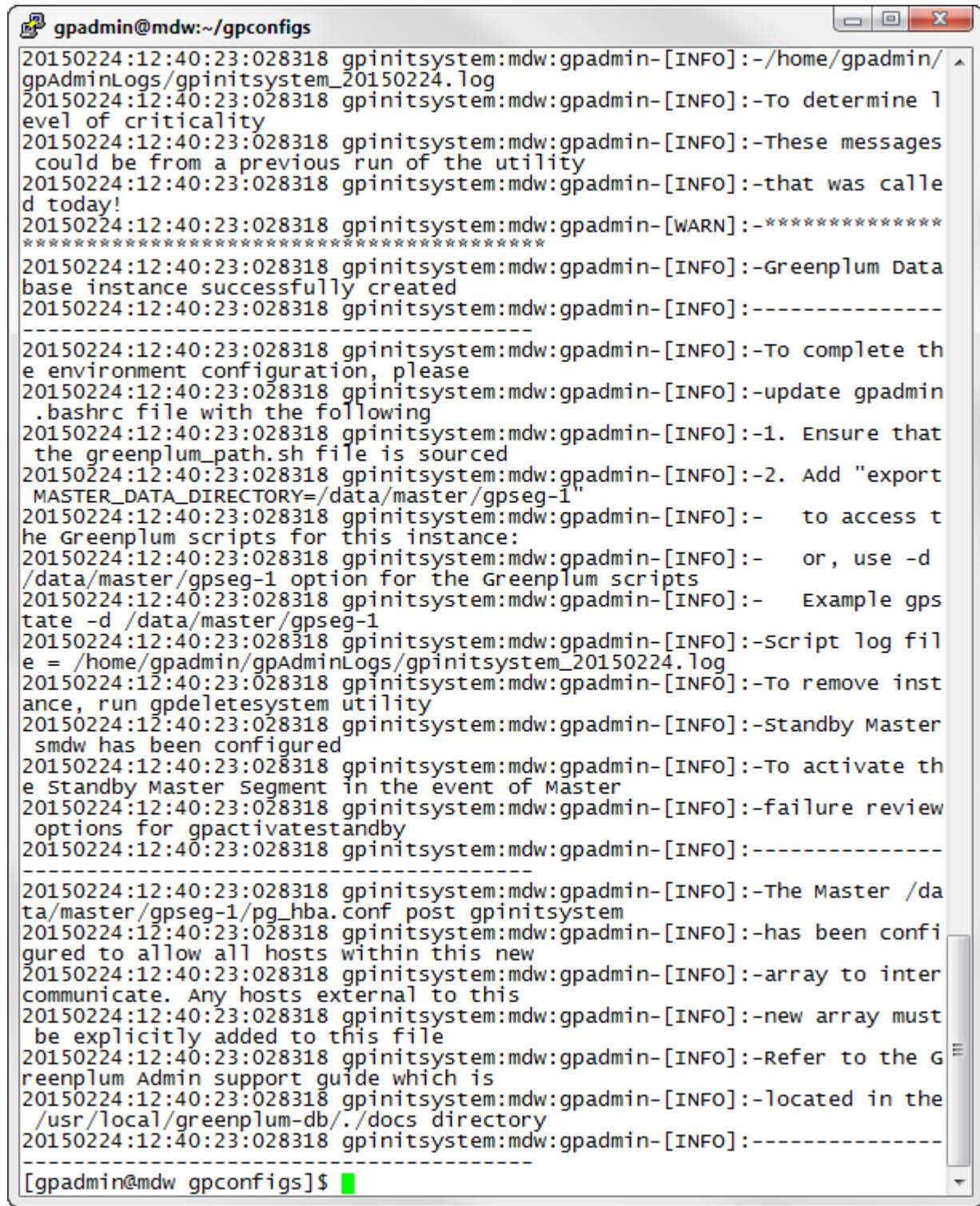
Step	Action
2.	<p>You will now delete the Greenplum Database that you created using the <code>gpdelete system</code> utility. The <code>-d</code> option is pointing at the directory where the master database instance was created.</p> <pre>[gpadmin@mdw ~]\$ gpdelete system -d /data/master/gpseg-1</pre>  <p>Press <b>y</b> to continue.</p>

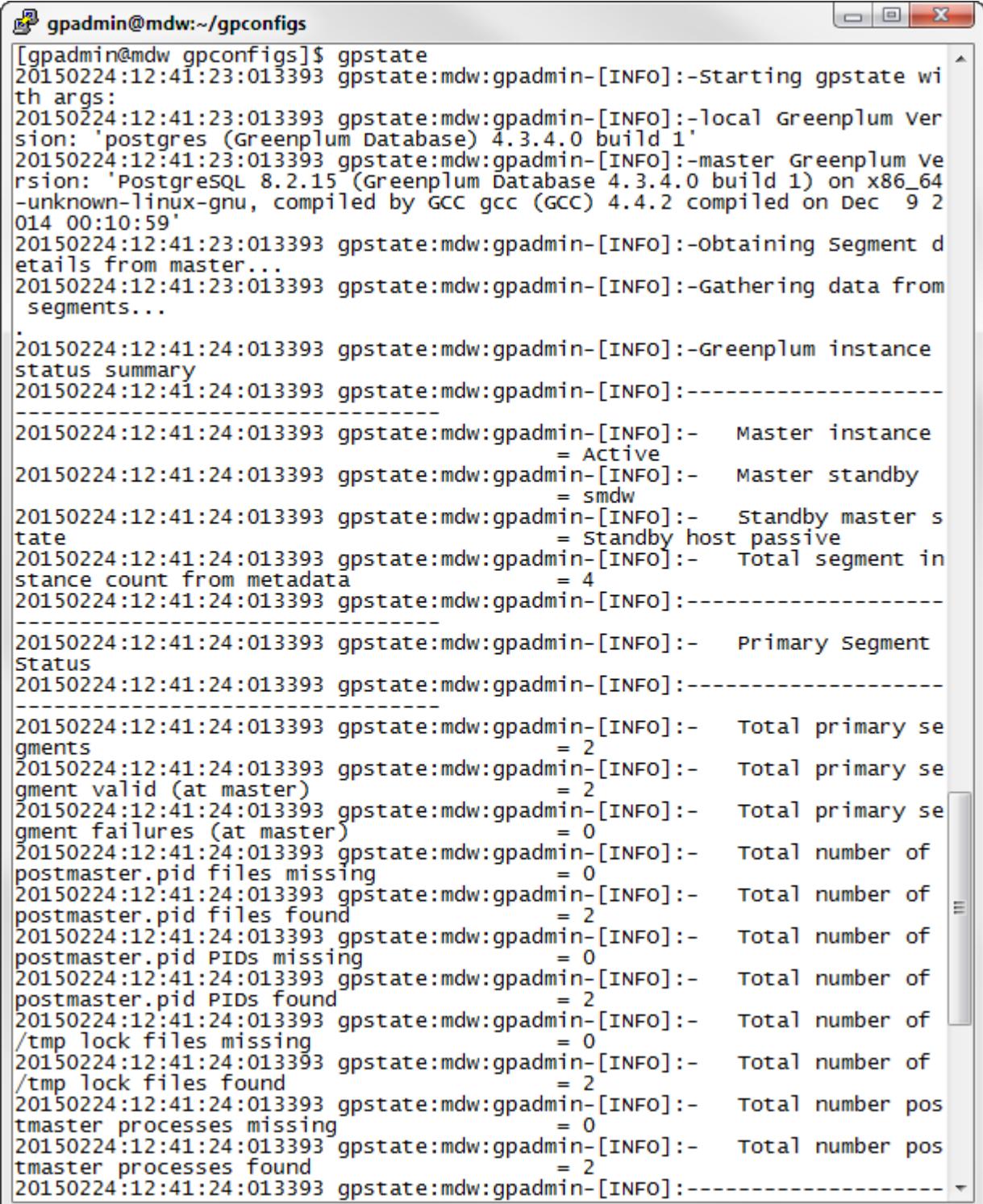
Step	Action
3.	 <pre>gpadmin@mdw:~\$ 20150224:11:54:08:028018 gpdeletesystem:mdw:gpadmin-[INFO]:-FINAL WARNING, you are about to delete the Greenplum instance 20150224:11:54:08:028018 gpdeletesystem:mdw:gpadmin-[INFO]:-on master host localhost.  Continue with Greenplum instance deletion? Yy Nn (default=N): &gt; [green square]</pre>
	<p>The utility will warn you one more time about deleting the database, and prompt you for confirmation. Press <b>y</b> to continue.</p> <p>The utility should complete successfully as shown here.</p>  <pre>gpadmin@mdw:~\$ 20150224:11:54:54:028018 gpdeletesystem:mdw:gpadmin-[INFO]:-stopping database... 20150224:11:55:18:028018 gpdeletesystem:mdw:gpadmin-[INFO]:-Deleting segments and removing data directories... 20150224:11:55:18:028018 gpdeletesystem:mdw:gpadmin-[INFO]:-Waiting for worker threads to complete... 20150224:11:55:25:028018 gpdeletesystem:mdw:gpadmin-[INFO]:-Delete system successful. [gpadmin@mdw ~]\$ [green square]</pre>

Step	Action
5.	<p>Using vi, edit the <code>gpinitSystem_config</code> file and replace the highlighted sections with the following:</p> <p>Scroll down to the <b>OPTIONAL MIRROR PARAMETERS</b> section of the file and perform the following tasks:</p> <ol style="list-style-type: none"> <li>1. Uncomment the lines shown in the snapshot.</li> <li>2. Replace the <code>declare</code> line with the following:</li> </ol> <pre><code>declare -a MIRROR_DATA_DIRECTORY=(/data/mirror)</code></pre>  <pre><code>gpadmin@mdw:~/gpconfigs ##### #### OPTIONAL MIRROR PARAMETERS #####  #### Base number by which mirror segment port numbers #### are calculated. #MIRROR_PORT_BASE=50000  #### Base number by which primary file replication port #### numbers are calculated. #REPLICATION_PORT_BASE=41000  #### Base number by which mirror file replication port #### numbers are calculated. #MIRROR_REPLICATION_PORT_BASE=51000  #### File system location(s) where mirror segment data directories #### will be created. The number of mirror locations must equal the #### number of primary locations as specified in the #### DATA_DIRECTORY parameter. #declare -a MIRROR_DATA_DIRECTORY=(/data1/mirror /data1/mirror /data1/mirror /data2/mirror /data2/mirror /data2/mirror)</code></pre>

Step	Action
6.	<p>The final file appears as follows:</p>  <pre data-bbox="339 291 1514 903"> ##### ##### OPTIONAL MIRROR PARAMETERS #####  ##### ## Base number by which mirror segment port numbers ## are calculated. MIRROR_PORT_BASE=50000  ##### ## Base number by which primary file replication port ## numbers are calculated. REPLICATION_PORT_BASE=41000  ##### ## Base number by which mirror file replication port ## numbers are calculated. MIRROR_REPLICATION_PORT_BASE=51000  ##### ## File system location(s) where mirror segment data directories ## will be created. The number of mirror locations must equal the ## number of primary locations as specified in the ## DATA_DIRECTORY parameter. #declare -a MIRROR_DATA_DIRECTORY=(/data1/mirror /data1/mirror /data1/mirror #/data2/mirror /data2/mirror /data2/mirror) declare -a MIRROR_DATA_DIRECTORY=(/data/mirror)  :wq! </pre> <p>This is a read-only file. Use :wq! to save the file.</p>
7.	<p>Run the <code>gpinitSystem</code> utility to create a Greenplum Database system using the values defined in <code>gpinitSystem_config</code>. The command is shown below:</p> <pre data-bbox="323 1157 1421 1227"> gpinitSystem -c gpinitSystem_config -h hostfile_gpinitSystem \ -s standby_master_hostname -S </pre> <ul data-bbox="372 1262 1530 1537" style="list-style-type: none"> <li data-bbox="372 1262 1530 1332"><code>standby_master_hostname</code> is a variable and should be replaced with the standby hostname of your pod.</li> <li data-bbox="372 1360 1530 1429"><code>-s</code> is an option used for a redundant system with a standby server. For more information on this utility, refer to Greenplum Database Installation Guide.</li> <li data-bbox="372 1457 1530 1526"><code>-S</code> is the option used to create a spread mirror configuration. For more information on this utility, refer to Greenplum Database Installation Guide.</li> </ul> <pre data-bbox="323 1564 1421 1634"> [gpadmin@mdw gpconfigs]\$ gpinitSystem -c gpinitSystem_config \ -h hostfile_gpinitSystem -s smdw -S </pre>

Step	Action
8.	<p>The utility displays the master, standby master, primary segments, and mirror segments configuration. It also asks you whether or not you want to continue with the Greenplum creation.</p>  <pre data-bbox="328 255 1547 1453"> ress          = 172.16.1.14 20150224:12:39:07:028318 gpinit[system:mdw:gpadmin-[INFO]]:-Standby IP add ress          = fe80::250:56ff:fe93:4c24 20150224:12:39:07:028318 gpinit[system:mdw:gpadmin-[INFO]]:-Total Database segments      = 2 20150224:12:39:07:028318 gpinit[system:mdw:gpadmin-[INFO]]:-Trusted shell                   = ssh 20150224:12:39:07:028318 gpinit[system:mdw:gpadmin-[INFO]]:-Number segment hosts          = 2 20150224:12:39:07:028318 gpinit[system:mdw:gpadmin-[INFO]]:-Mirror port ba se            = 50000 20150224:12:39:07:028318 gpinit[system:mdw:gpadmin-[INFO]]:-Replicaton por t base        = 41000 20150224:12:39:07:028318 gpinit[system:mdw:gpadmin-[INFO]]:-Mirror replica tion port base= 51000 20150224:12:39:07:028318 gpinit[system:mdw:gpadmin-[INFO]]:-Mirror segment #              = 1 20150224:12:39:07:028318 gpinit[system:mdw:gpadmin-[INFO]]:-Mirroring conf ig            = ON 20150224:12:39:07:028318 gpinit[system:mdw:gpadmin-[INFO]]:-Mirroring type                   = Spread 20150224:12:39:07:028318 gpinit[system:mdw:gpadmin-[INFO]]:----- 20150224:12:39:07:028318 gpinit[system:mdw:gpadmin-[INFO]]:-Greenplum Prim ary Segment Configuration 20150224:12:39:07:028318 gpinit[system:mdw:gpadmin-[INFO]]:----- 20150224:12:39:08:028318 gpinit[system:mdw:gpadmin-[INFO]]:-sdw1  /data/pr imary/gpseg0  40000  2    0    41000 20150224:12:39:08:028318 gpinit[system:mdw:gpadmin-[INFO]]:-sdw2  /data/pr imary/gpseg1  40000  3    1    41000 20150224:12:39:08:028318 gpinit[system:mdw:gpadmin-[INFO]]:----- 20150224:12:39:08:028318 gpinit[system:mdw:gpadmin-[INFO]]:-Greenplum Mirr or Segment Configuration 20150224:12:39:08:028318 gpinit[system:mdw:gpadmin-[INFO]]:----- 20150224:12:39:08:028318 gpinit[system:mdw:gpadmin-[INFO]]:-sdw2  /data/mi rror/gpseg0  50000  4    0    51000 20150224:12:39:08:028318 gpinit[system:mdw:gpadmin-[INFO]]:-sdw1  /data/mi rror/gpseg1  50000  5    1    51000 Continue with Greenplum creation Yy/Nn&gt; y </pre> <p>Press <b>y</b> to continue.</p>

Step	Action
	<p>This screen shows that the gpinit system completed with no errors.</p>  <pre data-bbox="319 219 1545 1727"> gpadmin@mdw:~/gpconfigs 20150224:12:40:23:028318 gpinit system:mdw:gpadmin-[INFO]:-/home/gpadmin/ gpAdminLogs/gpinit system_20150224.log 20150224:12:40:23:028318 gpinit system:mdw:gpadmin-[INFO]:-To determine 1 level of criticality 20150224:12:40:23:028318 gpinit system:mdw:gpadmin-[INFO]:-These messages could be from a previous run of the utility 20150224:12:40:23:028318 gpinit system:mdw:gpadmin-[INFO]:-that was calle d today! 20150224:12:40:23:028318 gpinit system:mdw:gpadmin-[WARN]:-***** ***** 20150224:12:40:23:028318 gpinit system:mdw:gpadmin-[INFO]:-Greenplum Data base instance successfully created 20150224:12:40:23:028318 gpinit system:mdw:gpadmin-[INFO]:----- 20150224:12:40:23:028318 gpinit system:mdw:gpadmin-[INFO]:-To complete th e environment configuration, please 20150224:12:40:23:028318 gpinit system:mdw:gpadmin-[INFO]:-update gpadmin .bashrc file with the following 20150224:12:40:23:028318 gpinit system:mdw:gpadmin-[INFO]:-1. Ensure that the greenplum_path.sh file is sourced 20150224:12:40:23:028318 gpinit system:mdw:gpadmin-[INFO]:-2. Add "export MASTER_DATA_DIRECTORY=/data/master/gpseg-1" 20150224:12:40:23:028318 gpinit system:mdw:gpadmin-[INFO]:- to access t he Greenplum scripts for this instance: 20150224:12:40:23:028318 gpinit system:mdw:gpadmin-[INFO]:- or, use -d /data/master/gpseg-1 option for the Greenplum scripts 20150224:12:40:23:028318 gpinit system:mdw:gpadmin-[INFO]:- Example gps tate -d /data/master/gpseg-1 20150224:12:40:23:028318 gpinit system:mdw:gpadmin-[INFO]:-Script log fil e = /home/gpadmin/gpAdminLogs/gpinit system_20150224.log 20150224:12:40:23:028318 gpinit system:mdw:gpadmin-[INFO]:-To remove inst ance, run gpdelete system utility 20150224:12:40:23:028318 gpinit system:mdw:gpadmin-[INFO]:-Standby Master smdw has been configured 20150224:12:40:23:028318 gpinit system:mdw:gpadmin-[INFO]:-To activate th e Standby Master Segment in the event of Master 20150224:12:40:23:028318 gpinit system:mdw:gpadmin-[INFO]:-failure review options for gpactivatestandby 20150224:12:40:23:028318 gpinit system:mdw:gpadmin-[INFO]:----- 20150224:12:40:23:028318 gpinit system:mdw:gpadmin-[INFO]:-The Master /da ta/master/gpseg-1/pg_hba.conf post gpinit system 20150224:12:40:23:028318 gpinit system:mdw:gpadmin-[INFO]:-has been config ured to allow all hosts within this new 20150224:12:40:23:028318 gpinit system:mdw:gpadmin-[INFO]:-array to inter communicate. Any hosts external to this 20150224:12:40:23:028318 gpinit system:mdw:gpadmin-[INFO]:-new array must be explicitly added to this file 20150224:12:40:23:028318 gpinit system:mdw:gpadmin-[INFO]:-Refer to the G reenplum Admin support guide which is 20150224:12:40:23:028318 gpinit system:mdw:gpadmin-[INFO]:-located in the /usr/local/greenplum-db/.docs directory 20150224:12:40:23:028318 gpinit system:mdw:gpadmin-[INFO]:----- [gpadmin@mdw gpconfigs]\$ </pre>

Step	Action
9.	<p>Run the gpstate utility to verify the Greenplum instance status summary.</p>  <pre>[gpadmin@mdw gpconfigs]\$ gpstate 20150224:12:41:23:013393 gpstate:mdw:gpadmin-[INFO]:-starting gpstate with args: 20150224:12:41:23:013393 gpstate:mdw:gpadmin-[INFO]:-local Greenplum version: 'postgres (Greenplum Database) 4.3.4.0 build 1' 20150224:12:41:23:013393 gpstate:mdw:gpadmin-[INFO]:-master Greenplum version: 'PostgreSQL 8.2.15 (Greenplum Database 4.3.4.0 build 1) on x86_64 -unknown-linux-gnu, compiled by GCC gcc (GCC) 4.4.2 compiled on Dec 9 2014 00:10:59' 20150224:12:41:23:013393 gpstate:mdw:gpadmin-[INFO]:-obtaining Segment details from master... 20150224:12:41:23:013393 gpstate:mdw:gpadmin-[INFO]:-Gathering data from segments... . 20150224:12:41:24:013393 gpstate:mdw:gpadmin-[INFO]:-Greenplum instance status summary 20150224:12:41:24:013393 gpstate:mdw:gpadmin-[INFO]:----- 20150224:12:41:24:013393 gpstate:mdw:gpadmin-[INFO]:- Master instance = Active 20150224:12:41:24:013393 gpstate:mdw:gpadmin-[INFO]:- Master standby = smdw 20150224:12:41:24:013393 gpstate:mdw:gpadmin-[INFO]:- Standby master state = Standby host passive 20150224:12:41:24:013393 gpstate:mdw:gpadmin-[INFO]:- Total segment instance count from metadata = 4 20150224:12:41:24:013393 gpstate:mdw:gpadmin-[INFO]:----- 20150224:12:41:24:013393 gpstate:mdw:gpadmin-[INFO]:- Primary Segment Status 20150224:12:41:24:013393 gpstate:mdw:gpadmin-[INFO]:----- 20150224:12:41:24:013393 gpstate:mdw:gpadmin-[INFO]:- Total primary segments = 2 20150224:12:41:24:013393 gpstate:mdw:gpadmin-[INFO]:- Total primary segment valid (at master) = 2 20150224:12:41:24:013393 gpstate:mdw:gpadmin-[INFO]:- Total primary segment failures (at master) = 0 20150224:12:41:24:013393 gpstate:mdw:gpadmin-[INFO]:- Total number of postmaster.pid files missing = 0 20150224:12:41:24:013393 gpstate:mdw:gpadmin-[INFO]:- Total number of postmaster.pid files found = 2 20150224:12:41:24:013393 gpstate:mdw:gpadmin-[INFO]:- Total number of postmaster.pid PIDs missing = 0 20150224:12:41:24:013393 gpstate:mdw:gpadmin-[INFO]:- Total number of postmaster.pid PIDs found = 2 20150224:12:41:24:013393 gpstate:mdw:gpadmin-[INFO]:- Total number of /tmp lock files missing = 0 20150224:12:41:24:013393 gpstate:mdw:gpadmin-[INFO]:- Total number of /tmp lock files found = 2 20150224:12:41:24:013393 gpstate:mdw:gpadmin-[INFO]:- Total number pos tmaster processes missing = 0 20150224:12:41:24:013393 gpstate:mdw:gpadmin-[INFO]:- Total number pos tmaster processes found = 2 20150224:12:41:24:013393 gpstate:mdw:gpadmin-[INFO]:-----</pre>

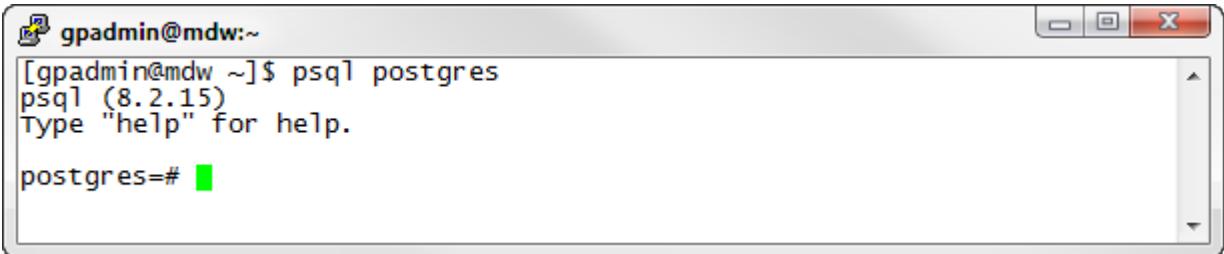
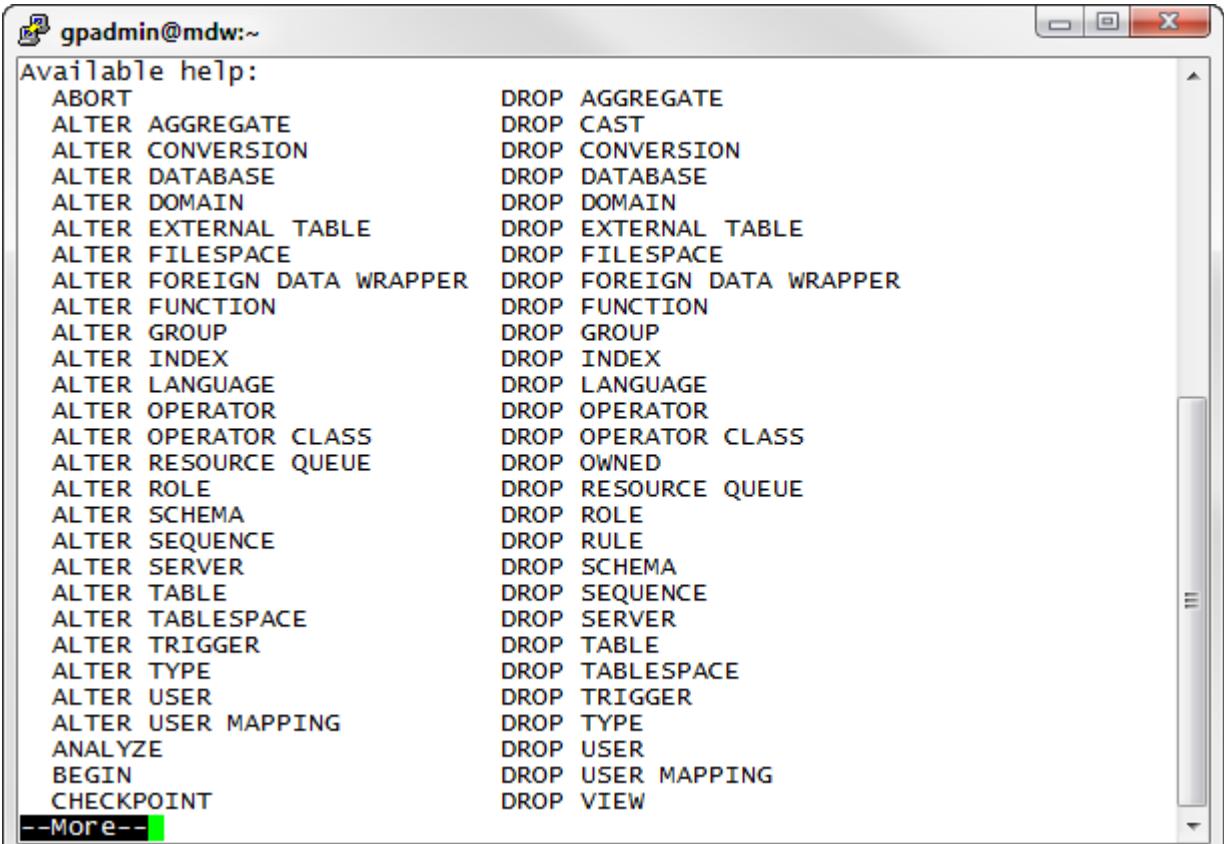
Step	Action
10.	<p>Change to the home directory.</p> <pre>[gpadmin@mdw ~]\$ cd gpadmin@mdw:~ [gpadmin@mdw gpconfigs]\$ cd [gpadmin@mdw ~]\$ █</pre>

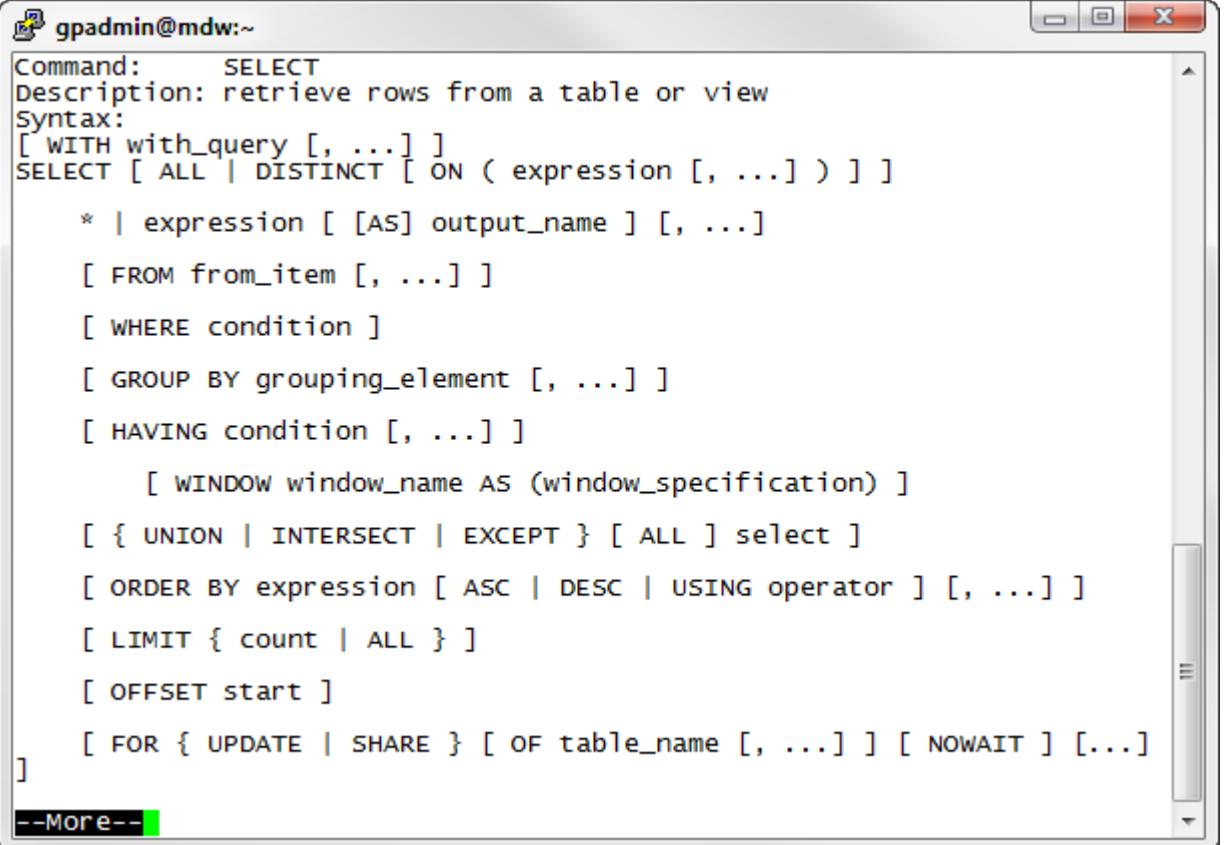
*End of Lab Exercise*

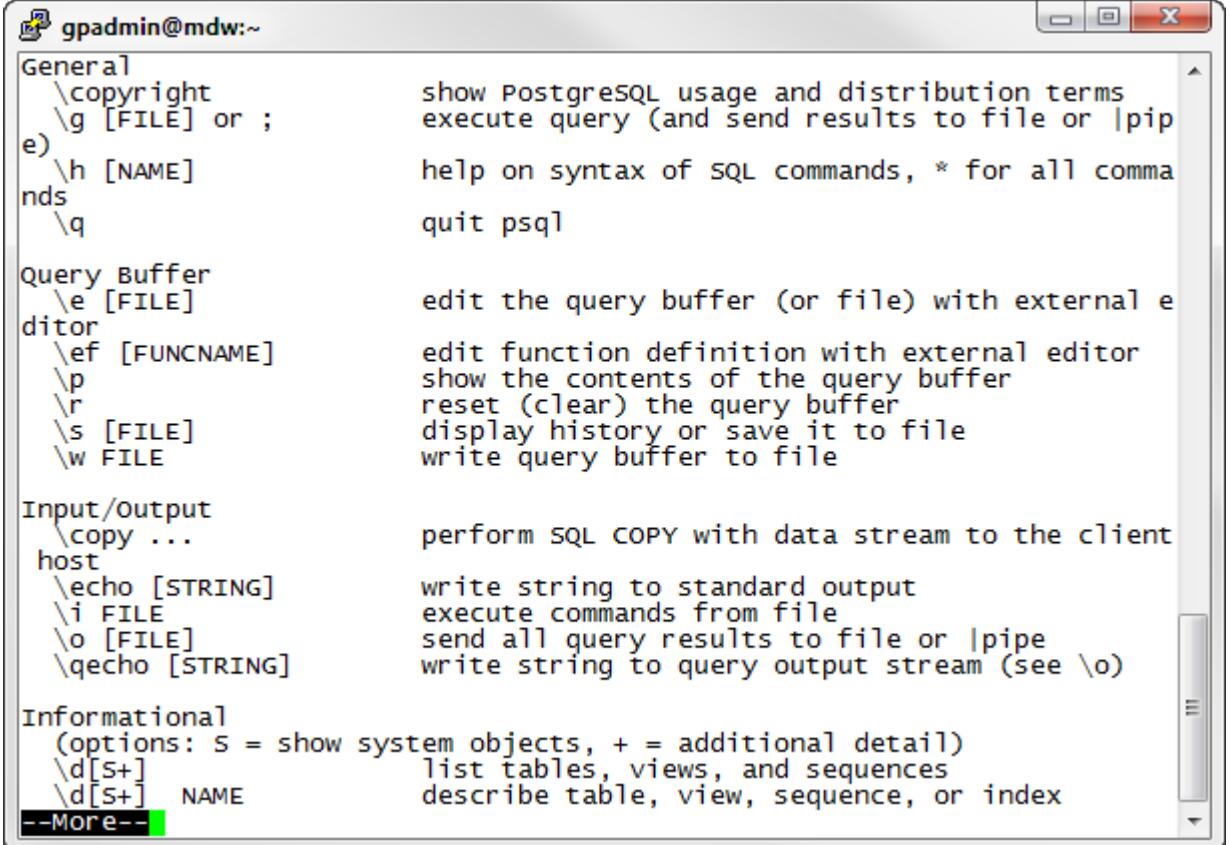
## Lab 4. Using the PSQL Client and Greenplum Utilities

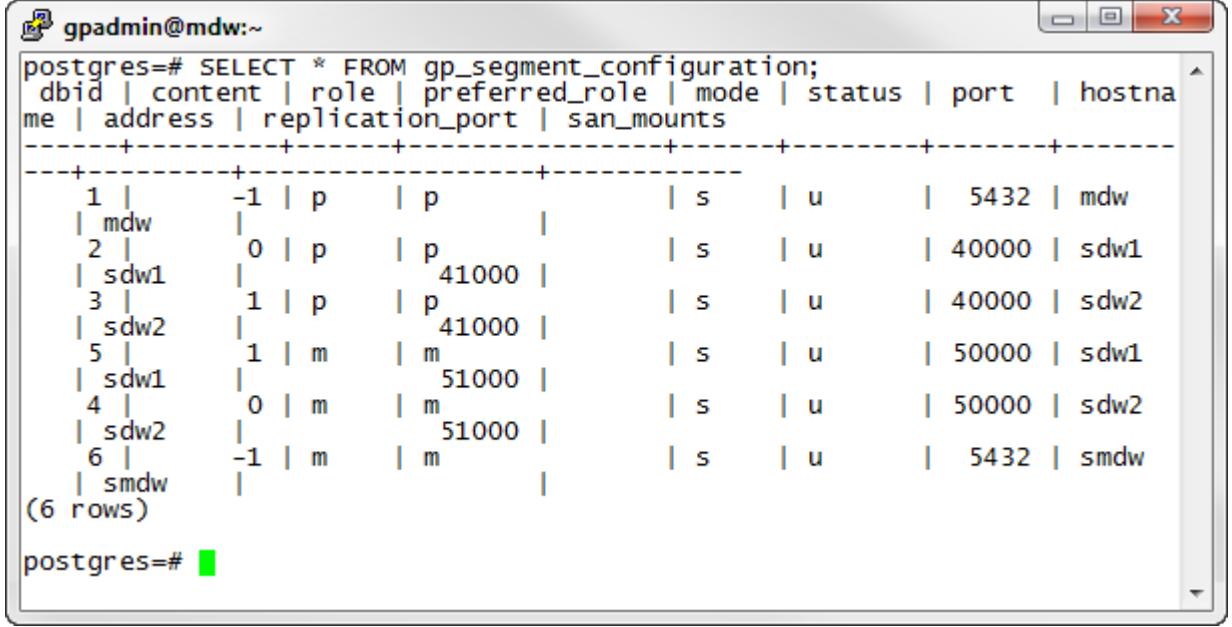
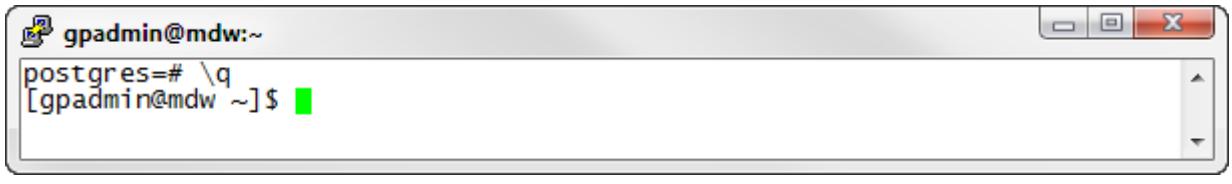
<b>Purpose:</b>	In this lab you will learn how to connect to the database using <code>psql</code> , the command line client interface to the Greenplum Database. You will learn how to use <code>psql</code> to run SQL commands both in interactive mode and non-interactive mode. You will also learn how to view the help in <code>psql</code> and about <code>psql</code> meta-commands.
<b>Tasks:</b>	Students perform the following tasks: <ul style="list-style-type: none"><li>• Access the database with <code>psql</code></li><li>• Install and configure Greenplum Command Center</li></ul>
<b>References:</b>	Module 3 – Greenplum Database Tools, Utilities, and Internals <ul style="list-style-type: none"><li>• Lesson: Using the PSQL Client and Greenplum Utilities</li><li>• Lesson: Pivotal Greenplum Command Center</li></ul>

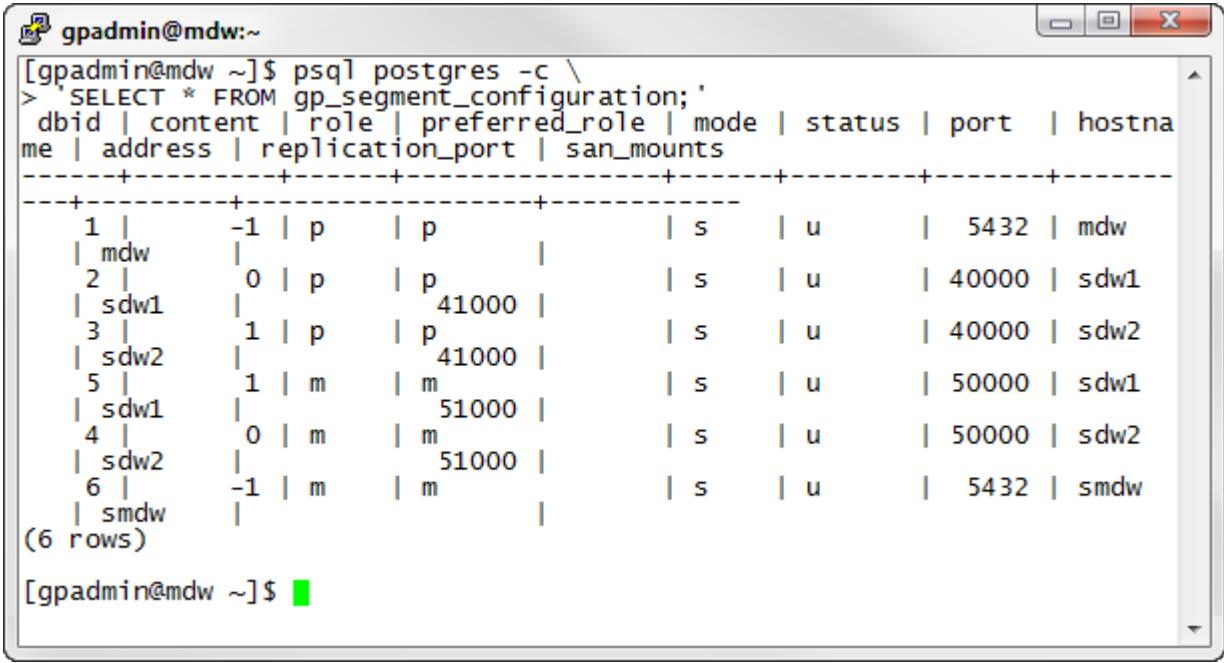
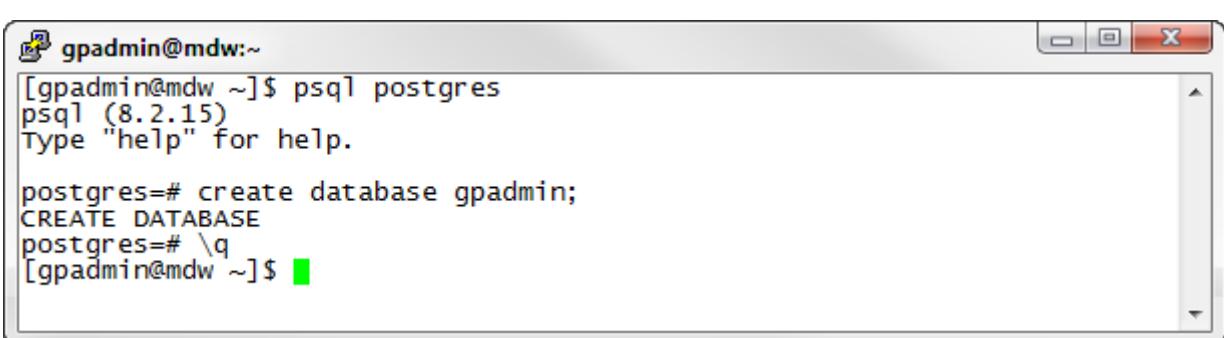
## Lab 4: Task 1 – Access the Database with psql

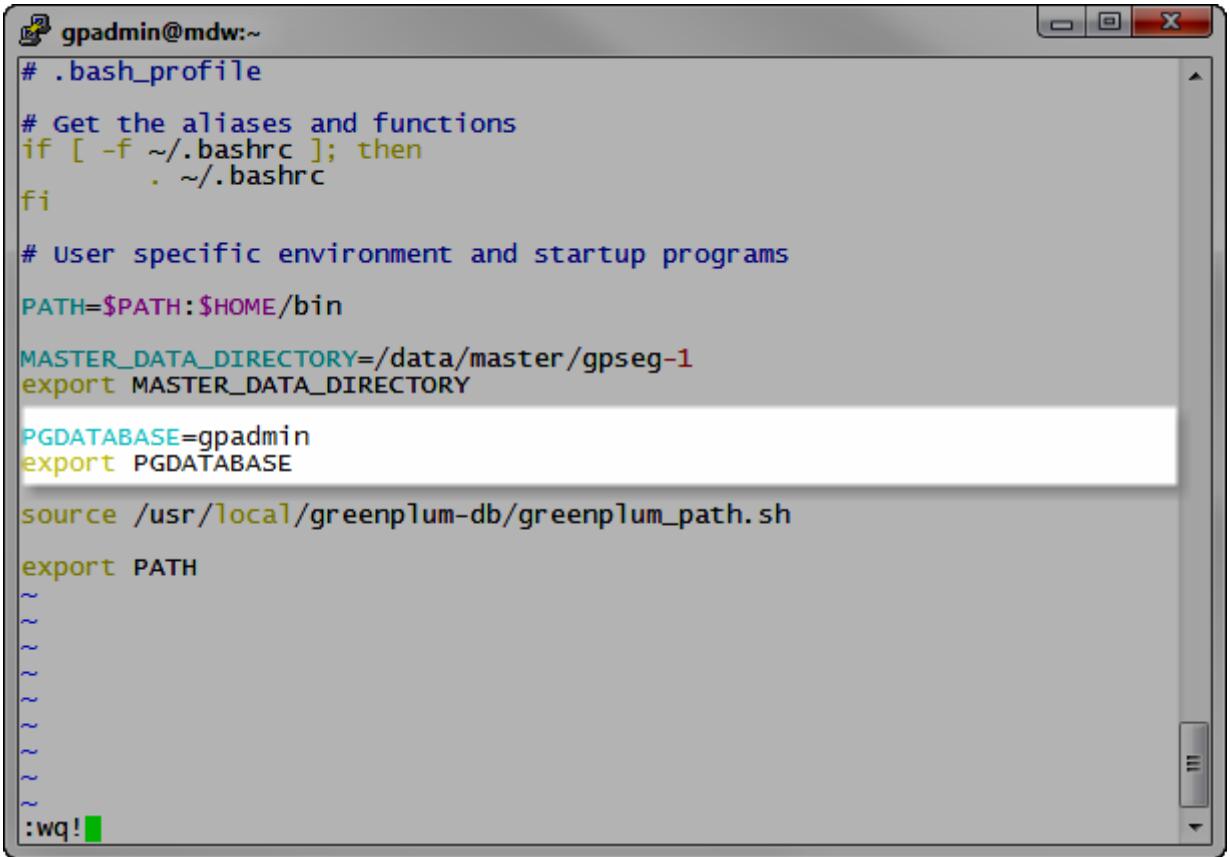
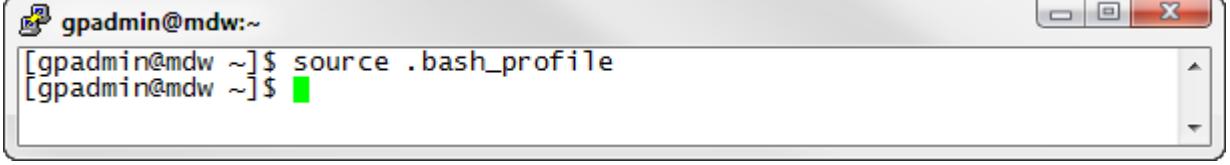
Step	Action
1.	<p>Start a psql session by connecting to the database, postgres: [gpadmin@mdw ~]\$ <b>psql postgres</b></p> 
2.	<p>The psql prompt displays the database you are connected to followed by =#, if you are logged in as the superuser account. At the psql prompt, type the \h meta-command to see a list of available SQL commands:</p> <pre>postgres=# \h</pre> 

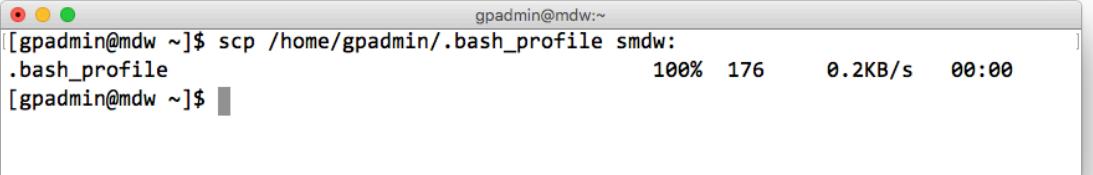
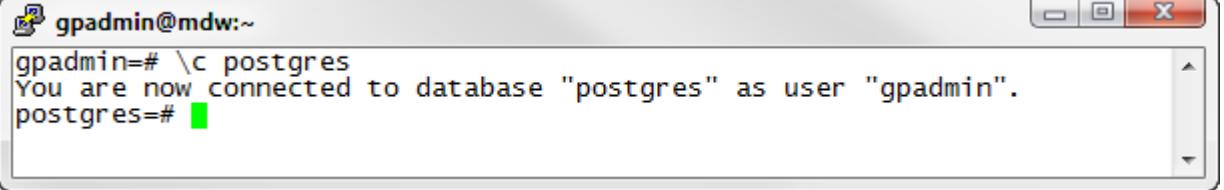
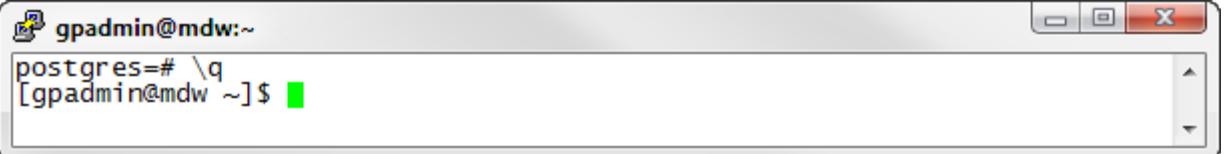
Step	Action
3.	<p>Use the \h meta-command to view the help for the SELECT SQL command only:</p> <pre>postgres=# \h SELECT</pre>  <pre> gpadmin@mdw:~ Command:      SELECT Description:  retrieve rows from a table or view Syntax: [ WITH with_query [, ...] ] SELECT [ ALL   DISTINCT [ ON ( expression [, ...] ) ] ]        *   expression [ [AS] output_name ] [, ...]        [ FROM from_item [, ...] ]        [ WHERE condition ]        [ GROUP BY grouping_element [, ...] ]        [ HAVING condition [, ...] ]            [ WINDOW window_name AS (window_specification) ]        [ { UNION   INTERSECT   EXCEPT } [ ALL ] select ]        [ ORDER BY expression [ ASC   DESC   USING operator ] [, ...] ]        [ LIMIT { count   ALL } ]        [ OFFSET start ]        [ FOR { UPDATE   SHARE } [ OF table_name [, ...] ] [ NOWAIT ] [...] ]  --More-- </pre>

Step	Action
4.	<p>At the <code>psql</code> prompt, type the <code>\?</code> meta-command to see a list of available <code>psql</code> meta-commands:</p> <pre>postgres=# \?</pre>  <pre> General \copyright          show PostgreSQL usage and distribution terms \g [FILE] or ;     execute query (and send results to file or  pipe) e)                 help on syntax of SQL commands, * for all commands \h [NAME]           help on syntax of SQL commands, * for all commands \nds                quit psql  Query Buffer \edit [FILE]        edit the query buffer (or file) with external editor ditor               edit function definition with external editor \ef [FUNCNAME]      show the contents of the query buffer \p                  reset (clear) the query buffer \q                  display history or save it to file \w FILE             write query buffer to file  Input/Output \copy ...           perform SQL COPY with data stream to the client host \echo [STRING]      write string to standard output \i FILE             execute commands from file \o [FILE]           send all query results to file or  pipe \qecho [STRING]     write string to query output stream (see \o)  Informational (options: s = show system objects, + = additional detail) \d[s+]              list tables, views, and sequences \d[s+] NAME         describe table, view, sequence, or index --More--</pre>

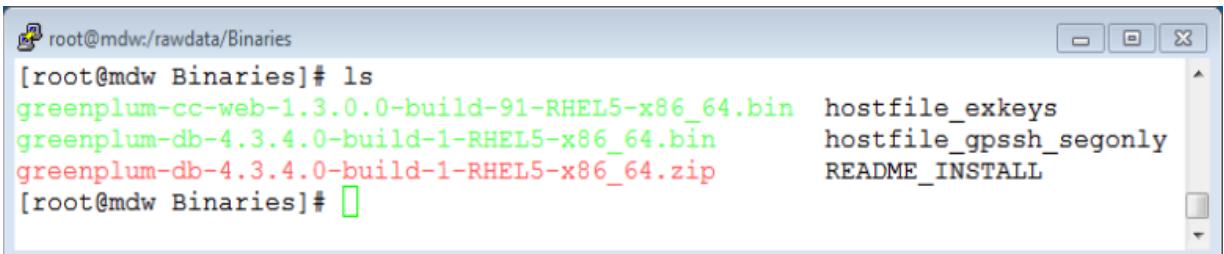
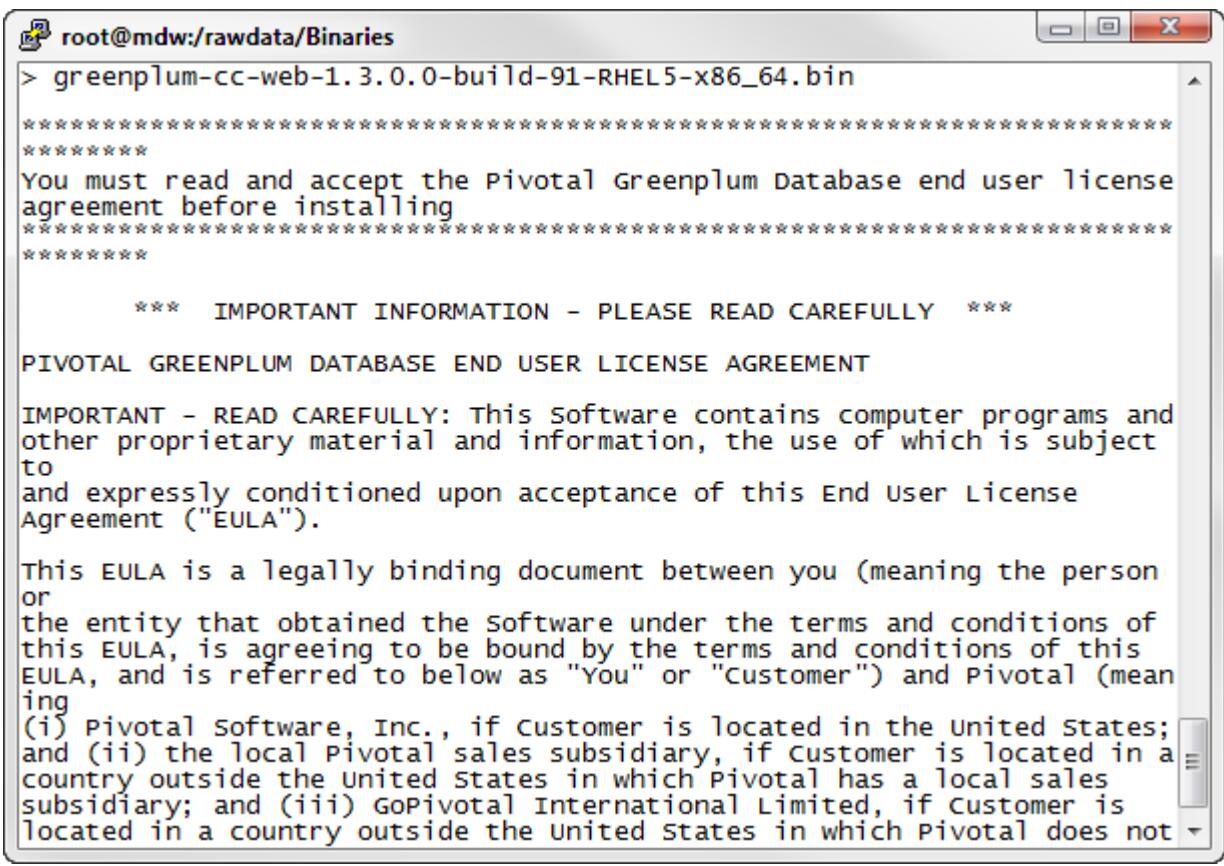
Step	Action
5.	<p>Run a SELECT statement on the gp_segment_configuration table. This system catalog table shows the entire master and segment instances in your array.</p> <pre data-bbox="347 270 1228 302">postgres=# SELECT * FROM gp_segment_configuration;</pre> <p>Note the use of the semi-colon (;) at the end of the statement. This completes the SQL command, allowing pSQL to execute the command. If you do not include the semi-colon, but press return, your command will continue on the next line.</p> <p>Your results should be similar to the following:</p>  <pre data-bbox="363 608 1538 1136"> postgres=# SELECT * FROM gp_segment_configuration; dbid   content   role   preferred_role   mode   status   port   hostna me   address   replication_port   san_mounts -----+-----+-----+-----+-----+-----+-----+-----+ 1   mdw   p   p   s   u   5432   mdw 2   sdw1   0   p   p   s   u   40000   sdw1 3   sdw1   1   p   p   s   u   40000   sdw2 4   sdw2   1   m   m   s   u   50000   sdw1 5   sdw1   0   m   m   s   u   50000   sdw2 6   smdw   -1   m   m   s   u   5432   smdw (6 rows)  postgres=# </pre>
6.	<p>Exit the psql session:</p> <pre data-bbox="347 1294 584 1326">postgres=# \q</pre>  <pre data-bbox="363 1406 649 1469"> postgres=# \q [gpadmin@mdw ~]\$ </pre>

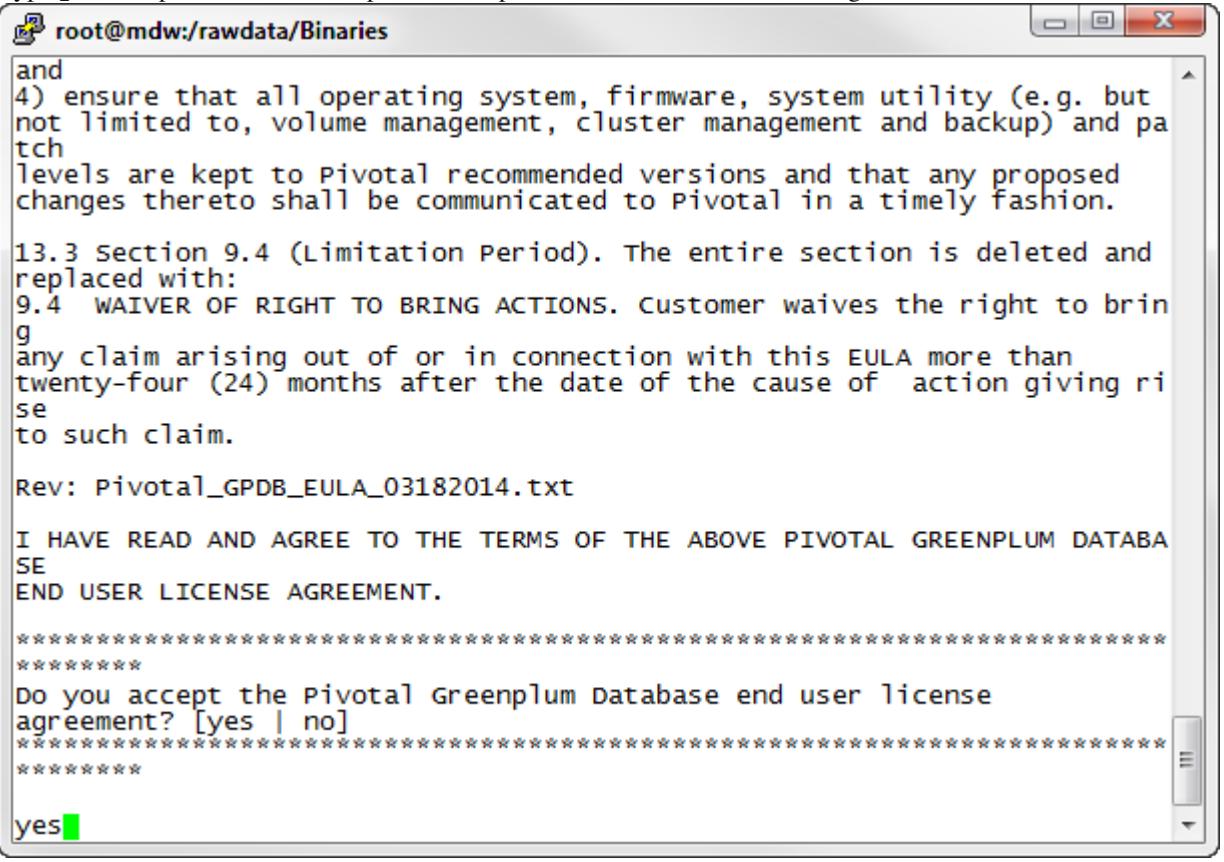
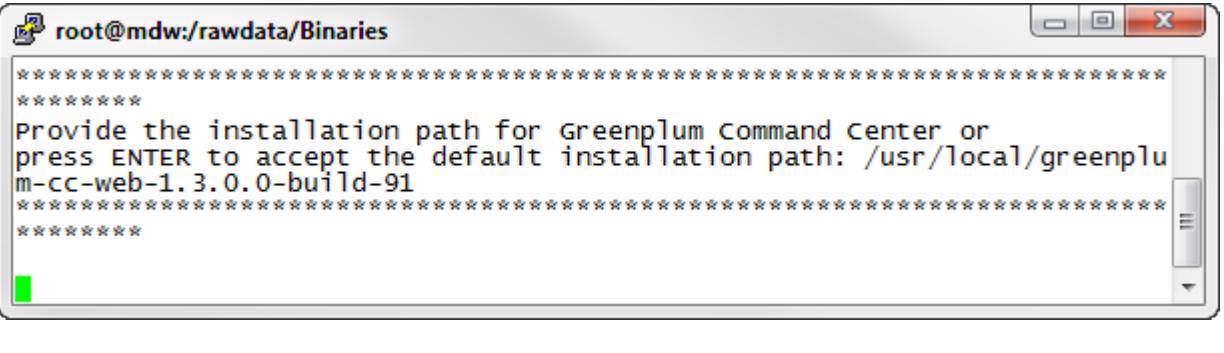
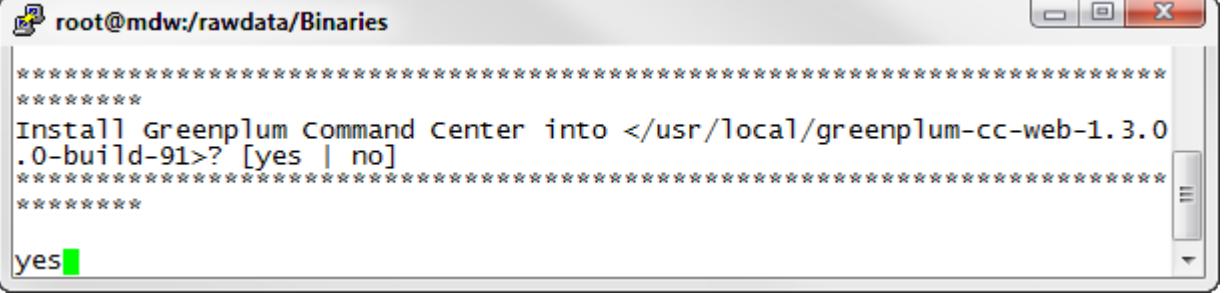
Step	Action																																																								
7.	<p>Execute the same SELECT statement you ran earlier, but this time, in non-interactive mode:</p> <pre>[gpadmin@mdw ~]\$ psql postgres -c 'SELECT * FROM gp_segment_configuration;'</pre>  <table border="1"> <thead> <tr> <th>dbid</th> <th>content</th> <th>role</th> <th>preferred_role</th> <th>mode</th> <th>status</th> <th>port</th> <th>hostname</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>-1</td> <td>p</td> <td>p</td> <td></td> <td>s</td> <td>5432</td> <td>mdw</td> </tr> <tr> <td>2</td> <td>0</td> <td>p</td> <td>p</td> <td>41000</td> <td>s</td> <td>40000</td> <td>sdw1</td> </tr> <tr> <td>3</td> <td>1</td> <td>p</td> <td>p</td> <td>41000</td> <td>s</td> <td>40000</td> <td>sdw2</td> </tr> <tr> <td>5</td> <td>1</td> <td>m</td> <td>m</td> <td>51000</td> <td>s</td> <td>50000</td> <td>sdw1</td> </tr> <tr> <td>4</td> <td>0</td> <td>m</td> <td>m</td> <td>51000</td> <td>s</td> <td>50000</td> <td>sdw2</td> </tr> <tr> <td>6</td> <td>-1</td> <td>m</td> <td>m</td> <td></td> <td>s</td> <td>5432</td> <td>smdw</td> </tr> </tbody> </table> <p>(6 rows)</p> <pre>[gpadmin@mdw ~]\$</pre>	dbid	content	role	preferred_role	mode	status	port	hostname	1	-1	p	p		s	5432	mdw	2	0	p	p	41000	s	40000	sdw1	3	1	p	p	41000	s	40000	sdw2	5	1	m	m	51000	s	50000	sdw1	4	0	m	m	51000	s	50000	sdw2	6	-1	m	m		s	5432	smdw
dbid	content	role	preferred_role	mode	status	port	hostname																																																		
1	-1	p	p		s	5432	mdw																																																		
2	0	p	p	41000	s	40000	sdw1																																																		
3	1	p	p	41000	s	40000	sdw2																																																		
5	1	m	m	51000	s	50000	sdw1																																																		
4	0	m	m	51000	s	50000	sdw2																																																		
6	-1	m	m		s	5432	smdw																																																		
8.	<p>Create a database with the same name as the user <code>gpadmin</code>. By creating a database with the same name as the current user, you will automatically be connected to this database if you do not specify the database name as part of the <code>psql</code> command line.</p> <ol style="list-style-type: none"> <li>1. Connect to the template1 database:</li> </ol> <pre>[gpadmin@mdw ~]\$ psql postgres</pre> <ol style="list-style-type: none"> <li>2. Create a new database and name it <code>gpadmin</code>:</li> </ol> <pre>postgres=# create database gpadmin;</pre> <ol style="list-style-type: none"> <li>3. Exit the database session:</li> </ol> <pre>postgres=# \q</pre>  <pre>[gpadmin@mdw ~]\$ psql postgres psql (8.2.15) Type "help" for help.  postgres=# create database gpadmin; CREATE DATABASE postgres=# \q [gpadmin@mdw ~]\$</pre>																																																								

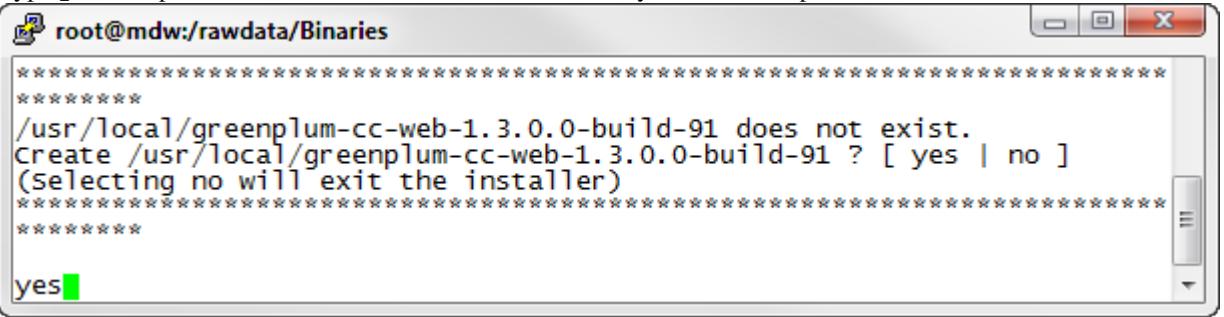
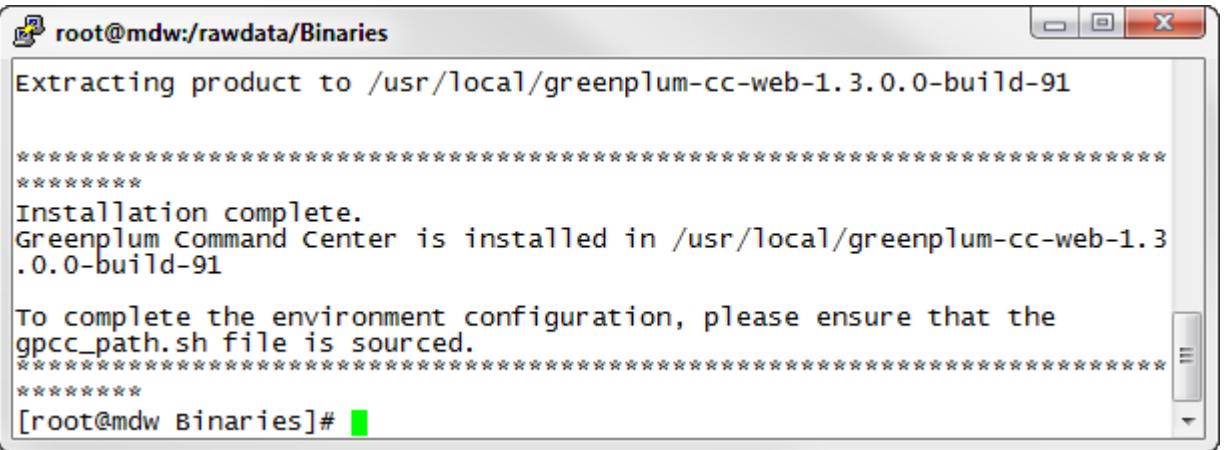
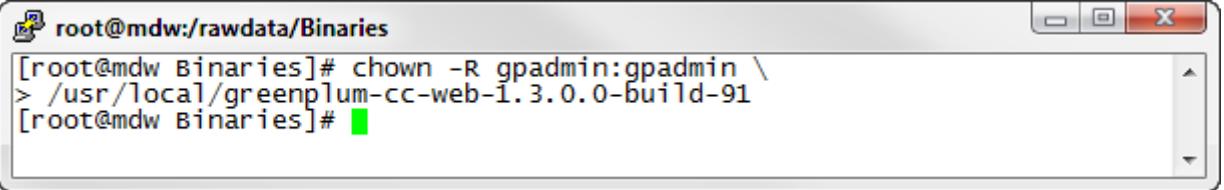
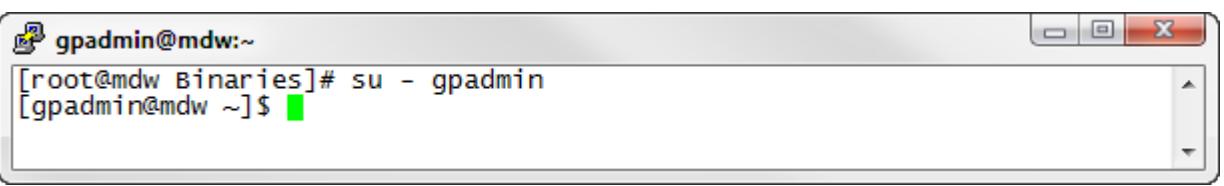
Step	Action
9.	<p>Using vi, edit the <code>.bash_profile</code> and add the PGDATABASE variable as shown below.</p> <p>This variable contains your <b>default login database name</b>. <code>psql</code> chooses a default database based on your username. The PGDATABASE environment variable takes precedence over this action, so if it is set, you will automatically be logged in to the database specified in the variable if you do not specify one on the command line.</p>  <pre data-bbox="349 418 1579 1275"> gpadmin@mdw:~ # .bash_profile # Get the aliases and functions if [ -f ~/.bashrc ]; then     . ~/.bashrc fi  # User specific environment and startup programs PATH=\$PATH:\$HOME/bin  MASTER_DATA_DIRECTORY=/data/master/gpseg-1 export MASTER_DATA_DIRECTORY  PGDATABASE=gpadmin export PGDATABASE  source /usr/local/greenplum-db/greenplum_path.sh export PATH ~ ~ ~ ~ ~ ~ ~ ~ :wq! </pre> <p>Run the command below to make the changes active immediately.</p> <pre data-bbox="349 1381 1008 1415">[gpadmin@mdw ~] \$ source .bash_profile</pre>  <pre data-bbox="349 1448 1579 1613"> gpadmin@mdw:~ [gpadmin@mdw ~] \$ source .bash_profile [gpadmin@mdw ~] \$ </pre>

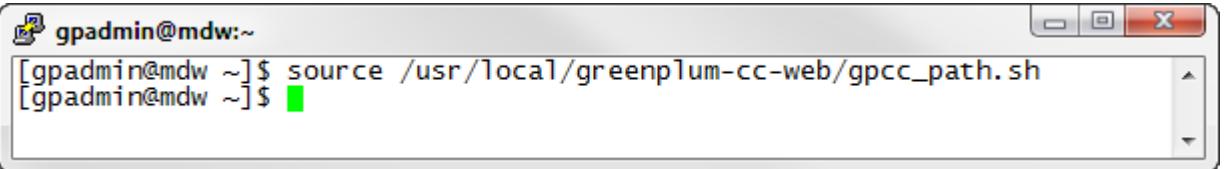
Step	Action
10.	<p>Copy the updated .bash_profile file to the standby server smdw.</p> <pre data-bbox="355 228 1302 264">[gpadmin@mdw ~]\$ scp /home/gpadmin/.bash_profile smdw:</pre> 
11.	<p>Log in to Greenplum with the default database of gpadmin.</p> <pre data-bbox="355 692 724 728">[gpadmin@mdw ~]\$ psql</pre> 
12.	<p>From within the database session, connect to the postgres database.</p> <pre data-bbox="355 1123 724 1159">gpadmin=# \c postgres</pre> 
13.	<p>Close your psql session.</p> <pre data-bbox="355 1469 584 1505">postgres=# \q</pre> 

## Lab 4: Task 2 – Install and Configure Pivotal Greenplum Command Center

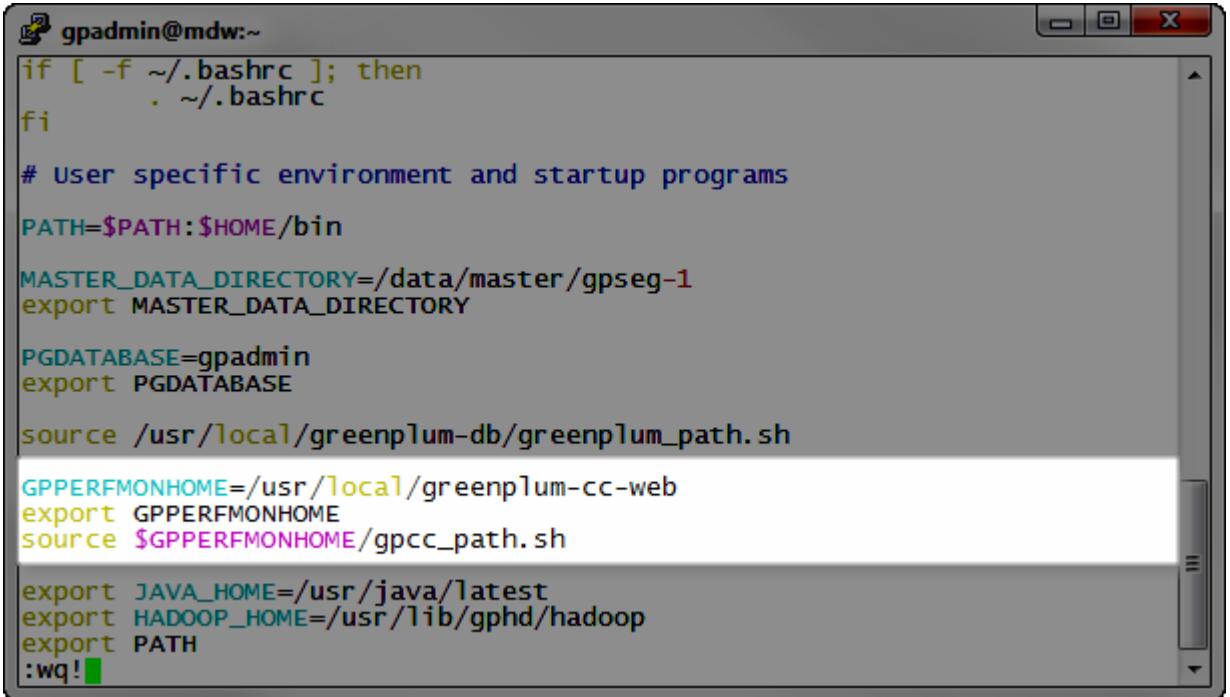
Step	Action
1.	<p>As root, connect to the master server, mdw, and change to the directory /rawdata/Binaries and list the contents of the directory:</p> <pre>[root@mdw ~]# cd /rawdata/Binaries [root@mdw Binaries]# ls</pre> 
2.	<p>Install the Greenplum Command Center binary to the master server, mdw, using the bash command.</p> <pre>[root@mdw Binaries]# bash \ greenplum-cc-web-1.3.0.0-build-91-RHEL5-x86_64.bin</pre>  <p>Hit <b>q</b> at any time to bypass the license agreement or space to read the next page of the license agreement.</p>

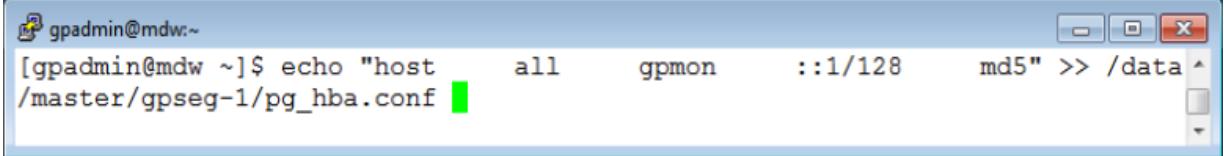
Step	Action
3.	<p>Type <b>yes</b> and press <b>Enter</b> to accept the Greenplum Command Center license agreement.</p>  <pre> root@mdw:/rawdata/Binaries and 4) ensure that all operating system, firmware, system utility (e.g. but not limited to, volume management, cluster management and backup) and pa tch levels are kept to Pivotal recommended versions and that any proposed changes thereto shall be communicated to Pivotal in a timely fashion.  13.3 Section 9.4 (Limitation Period). The entire section is deleted and replaced with: 9.4 WAIVER OF RIGHT TO BRING ACTIONS. Customer waives the right to brin g any claim arising out of or in connection with this EULA more than twenty-four (24) months after the date of the cause of action giving ri se to such claim.  Rev: Pivotal_GPDB_EULA_03182014.txt  I HAVE READ AND AGREE TO THE TERMS OF THE ABOVE PIVOTAL GREENPLUM DATA BASE END USER LICENSE AGREEMENT.  ***** Do you accept the Pivotal Greenplum Database end user license agreement? [yes   no] ***** yes </pre>
4.	<p>Hit <b>Enter</b> to accept the default installation directory location:  <code>/usr/local/greenplum-cc-web-1.3.0.0-build-91.</code></p>  <pre> root@mdw:/rawdata/Binaries ***** Provide the installation path for Greenplum Command Center or press ENTER to accept the default installation path: /usr/local/greenplu m-cc-web-1.3.0.0-build-91 ***** </pre>
5.	<p>Type <b>yes</b> and press <b>Enter</b> to install the Greenplum Command Center.</p>  <pre> root@mdw:/rawdata/Binaries ***** Install Greenplum Command Center into &lt;/usr/local/greenplum-cc-web-1.3.0 .0-build-91&gt;? [yes   no] ***** yes </pre>

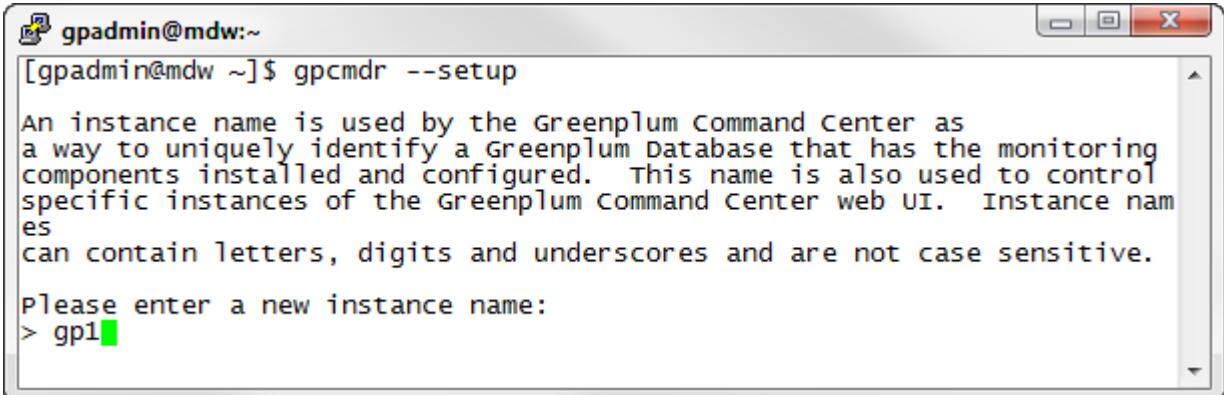
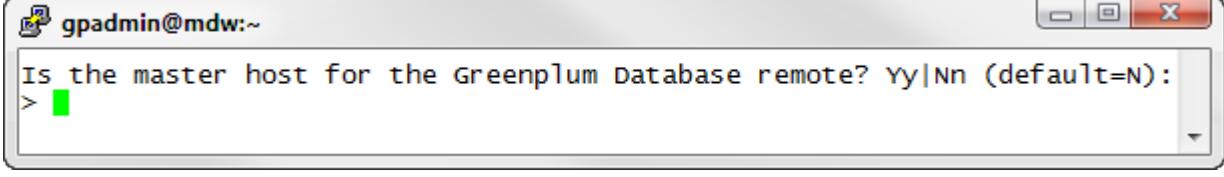
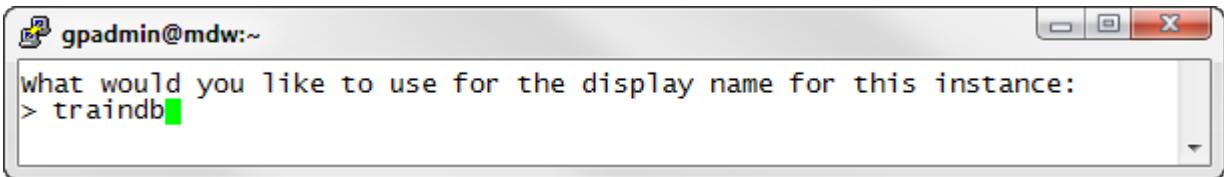
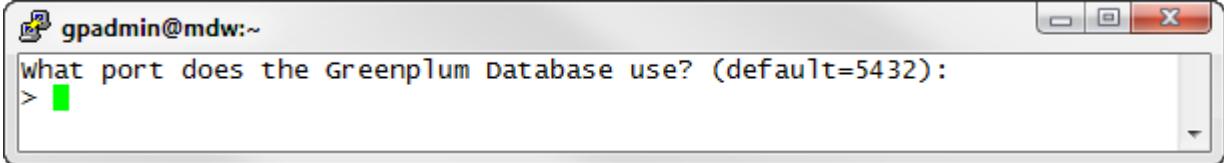
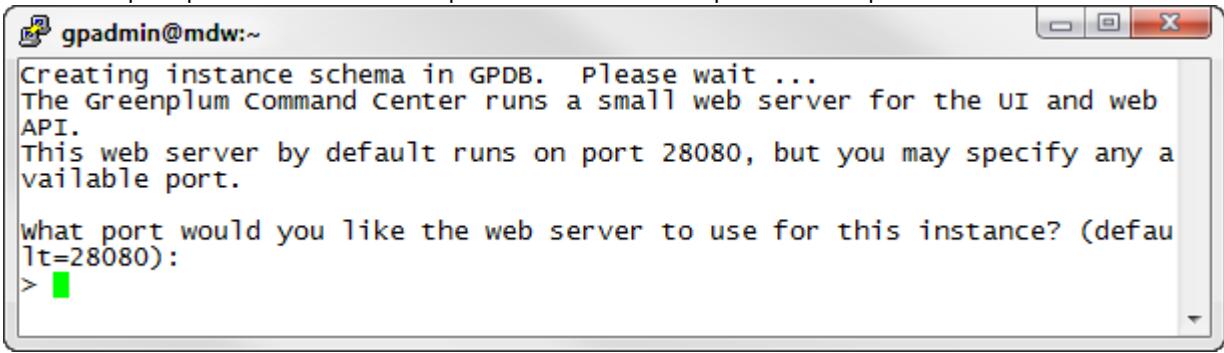
Step	Action
6.	<p>Type <b>yes</b> and press <b>Enter</b> to create the installation directory for the Greenplum Command Center.</p>  <pre>root@mdw:/rawdata/Binaries ***** /usr/local/greenplum-cc-web-1.3.0.0-build-91 does not exist. Create /usr/local/greenplum-cc-web-1.3.0.0-build-91 ? [ yes   no ] (selecting no will exit the installer) ***** yes</pre>
7.	<p>Greenplum Command Center binaries will be installed into the default directory:  <code>/usr/local/greenplum-cc-web-1.3.0.0-build-91</code>.</p> <p>You will see the following output confirming that the binaries have been installed:</p>  <pre>root@mdw:/rawdata/Binaries Extracting product to /usr/local/greenplum-cc-web-1.3.0.0-build-91 ***** Installation complete. Greenplum Command Center is installed in /usr/local/greenplum-cc-web-1.3 .0.0-build-91 To complete the environment configuration, please ensure that the gpcc_path.sh file is sourced. ***** [root@mdw Binaries]#</pre>
8.	<p>Change the user and group permissions of Pivotal Greenplum Command Center directory to gpadmin.</p> <pre>[root@mdw Binaries]# chown -R gpadmin:gpadmin \ /usr/local/greenplum-cc-web-1.3.0.0-build-91</pre>  <pre>[root@mdw Binaries]# chown -R gpadmin:gpadmin \ &gt; /usr/local/greenplum-cc-web-1.3.0.0-build-91 [root@mdw Binaries]#</pre>
9.	<p>Change user to gpadmin.</p> <pre>[root@mdw Binaries]# su - gpadmin</pre>  <pre>gpadmin@mdw:~ [root@mdw Binaries]# su - gpadmin [gpadmin@mdw ~]\$</pre>

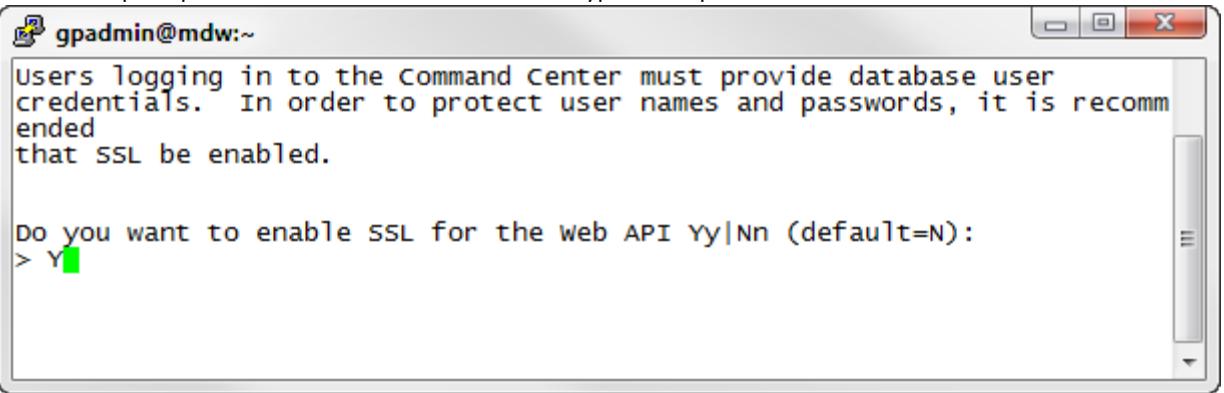
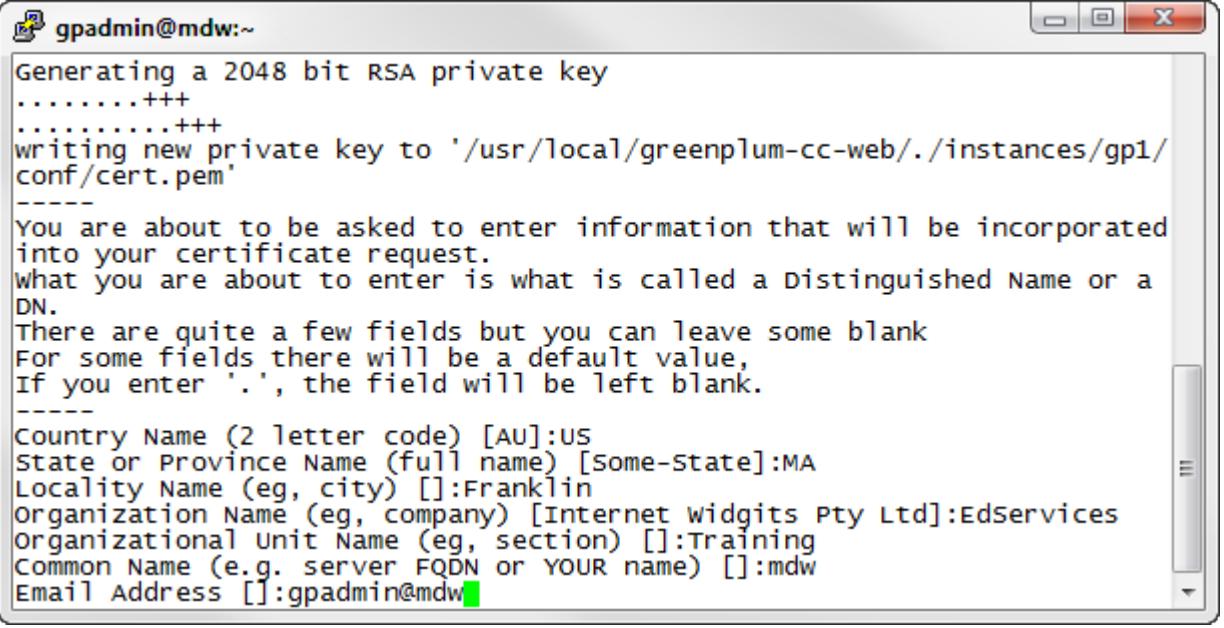
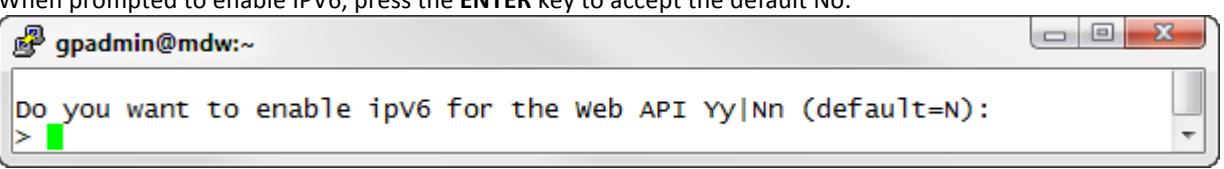
Step	Action
10.	<p>The installation directory contains a <code>gpcc_path.sh</code> file with path and environment settings for the Console. Source the file as follows:</p> <pre data-bbox="355 255 1481 291">[gpadmin@mdw ~]\$ source /usr/local/greenplum-cc-web/gpcc_path.sh</pre> 
11.	<p>Run the <code>gpperfmon_install</code> utility to perform the following:</p> <ol style="list-style-type: none"> <li>1. Create the Greenplum Command Center database (<code>gpperfmon</code>).</li> <li>2. Creates the Greenplum Command Center superuser role (<code>gpmon</code>).</li> <li>3. Configures Greenplum Database server to accept connections from the Greenplum Command Center superuser role (edits the <code>pg_hba.conf</code> and <code>.pgpass</code> files).</li> <li>4. Sets the Greenplum Command Center server configuration parameters in the Greenplum Database server <code>postgresql.conf</code> files.</li> </ol> <pre data-bbox="447 794 1155 857">[gpadmin@mdw ~]\$ gpperfmon_install \ --enable --password changeme --port 5432</pre> <p>You will see the following output to the console:</p>

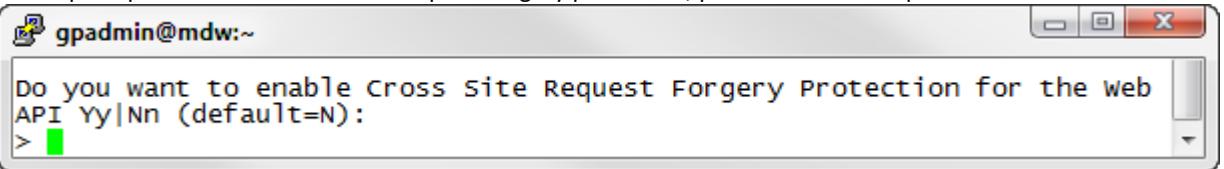
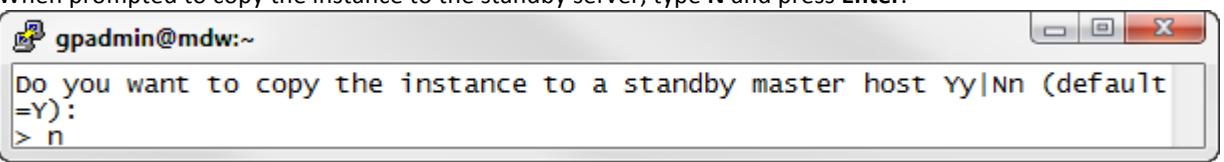
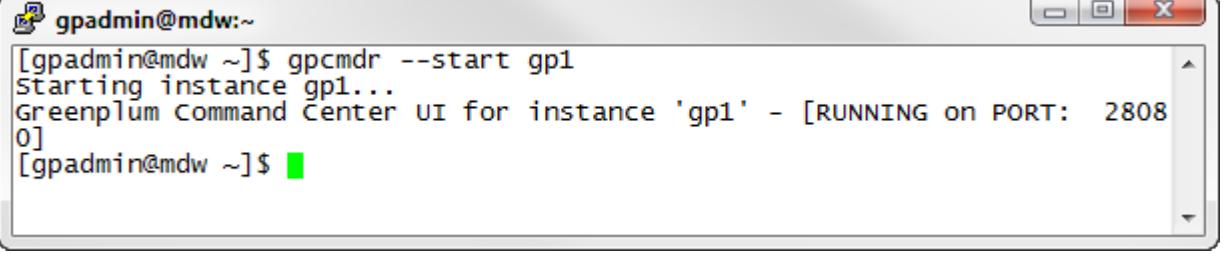
Step	Action
	 gpadmin@mdw:~ [gpadmin@mdw ~]\$ gpperfmon_install \ > --enable --password changeme --port 5432 20150224:16:21:29:017708 gpperfmon_install:mdw:gpadmin-[INFO]:-PGPORT=54 32 psql -f /usr/local/greenplum-db./lib/gpperfmon/gpperfmon3.sql template1 >& /dev/null 20150224:16:21:31:017708 gpperfmon_install:mdw:gpadmin-[INFO]:-PGPORT=54 32 psql -f /usr/local/greenplum-db./lib/gpperfmon/gpperfmon4.sql gpperfmon >& /dev/null 20150224:16:21:31:017708 gpperfmon_install:mdw:gpadmin-[INFO]:-PGPORT=54 32 psql -f /usr/local/greenplum-db./lib/gpperfmon/gpperfmon41.sql gpperfmon >& /dev/null 20150224:16:21:31:017708 gpperfmon_install:mdw:gpadmin-[INFO]:-PGPORT=54 32 psql -f /usr/local/greenplum-db./lib/gpperfmon/gpperfmon42.sql gpperfmon >& /dev/null 20150224:16:21:32:017708 gpperfmon_install:mdw:gpadmin-[INFO]:-PGPORT=54 32 psql -f /usr/local/greenplum-db./lib/gpperfmon/gpperfmonC.sql template1 >& /dev/null 20150224:16:21:32:017708 gpperfmon_install:mdw:gpadmin-[INFO]:-PGPORT=54 32 psql template1 -c "DROP ROLE IF EXISTS gpmon" >& /dev/null 20150224:16:21:32:017708 gpperfmon_install:mdw:gpadmin-[INFO]:-PGPORT=54 32 psql template1 -c "CREATE ROLE gpmon WITH SUPERUSER CREATEDB LOGIN ENCRYPTIONED PASSWORD 'changeme'" >& /dev/null 20150224:16:21:32:017708 gpperfmon_install:mdw:gpadmin-[INFO]:-echo "local gpperfmon gpmon md5" >> /data/master/gpseg-1/pg_hba.conf 20150224:16:21:32:017708 gpperfmon_install:mdw:gpadmin-[INFO]:-echo "host all gpmon 127.0.0.1/28 md5" >> /data/master/gps eg-1/pg_hba.conf 20150224:16:21:32:017708 gpperfmon_install:mdw:gpadmin-[INFO]:-touch /home/gpadmin/.pgpass >& /dev/null 20150224:16:21:32:017708 gpperfmon_install:mdw:gpadmin-[INFO]:-mv -f /home/gpadmin/.pgpass /home/gpadmin/.pgpass.1424812889 >& /dev/null 20150224:16:21:32:017708 gpperfmon_install:mdw:gpadmin-[INFO]:-echo "*:5432:gpperfmon:gpmon:changeme" >> /home/gpadmin/.pgpass 20150224:16:21:32:017708 gpperfmon_install:mdw:gpadmin-[INFO]:-cat /home/gpadmin/.pgpass.1424812889 >> /home/gpadmin/.pgpass 20150224:16:21:32:017708 gpperfmon_install:mdw:gpadmin-[INFO]:-chmod 0600 /home/gpadmin/.pgpass >& /dev/null 20150224:16:21:32:017708 gpperfmon_install:mdw:gpadmin-[INFO]:-PGPORT=54 32 gpconfig -c gp_enable_gpperfmon -v on >& /dev/null 20150224:16:21:41:017708 gpperfmon_install:mdw:gpadmin-[INFO]:-PGPORT=54 32 gpconfig -c gpperfmon_port -v 8888 >& /dev/null 20150224:16:21:46:017708 gpperfmon_install:mdw:gpadmin-[INFO]:-PGPORT=54 32 gpconfig -c gp_external_enable_exec -v on --masteronly >& /dev/null 20150224:16:21:52:017708 gpperfmon_install:mdw:gpadmin-[INFO]:-PGPORT=54 32 gpconfig -c gpperfmon_log_alert_level -v warning >& /dev/null 20150224:16:21:58:017708 gpperfmon_install:mdw:gpadmin-[INFO]:-gpperfmon will be enabled after a full restart of GPDB [gpadmin@mdw ~]\$

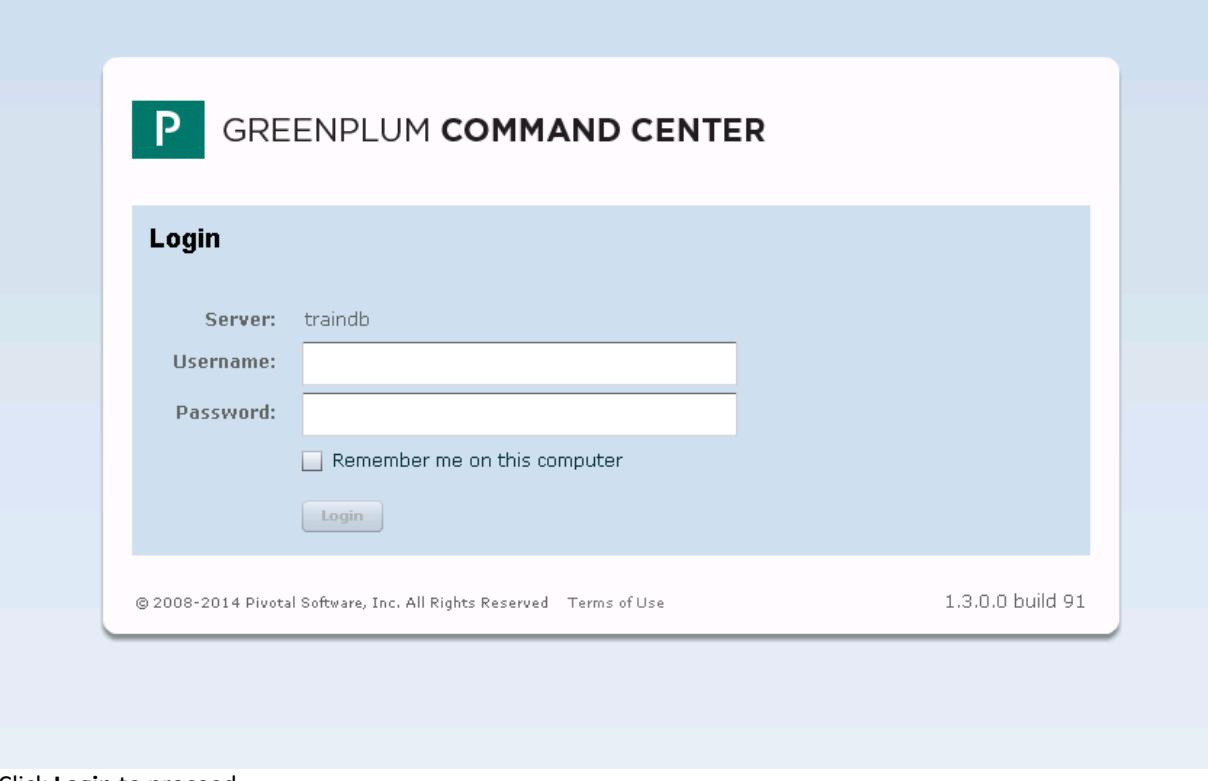
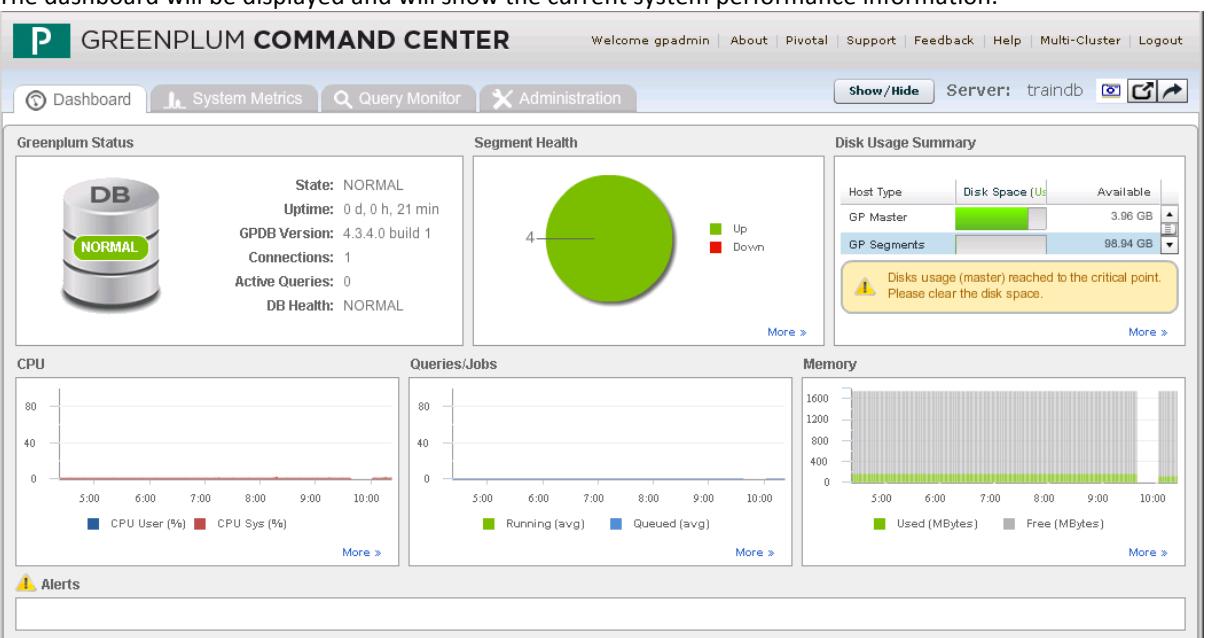
Step	Action
12.	<p>Modify the <code>.bash_profile</code> file and add an entry for the <code>GPPerfmonhome</code> variable. This variable, which specifies the home directory for Greenplum Command Center, should be set to <code>/usr/local/greenplum-cc-web</code>.</p> <p>Add the following to the file:</p> <pre data-bbox="442 291 1171 382"> <b>GPPerfmonhome=/usr/local/greenplum-cc-web</b> <b>export GPPerfmonhome</b> <b>source \$GPPerfmonhome/gpcc_path.sh</b></pre> 
13.	<p>Source <code>/home/gpadmin.bash_profile</code></p> <pre data-bbox="349 1199 1253 1235"> [gpadmin@mdw ~]\$ <b>source /home/gpadmin/.bash_profile</b></pre> 

Step	Action
14.	<p>Add a record to pg_hba.conf to allow Command Center to connect to localhost through the IPv6 address ::1.</p> <pre>[gpadmin@mdw ~]\$ echo "host      all      gpmon    ::1/128      md5" &gt;&gt; \ /data/master/gpseg-1/pg_hba.conf</pre>  <p><b>NOTE:</b> Tab characters are not supported in the pg_hba.conf file. You must use spaces between the record entries.</p>
15.	<p>Restart the Greenplum Database cluster to enable gpperfmon.</p> <pre>[gpadmin@mdw ~]\$ gpstop -a -r</pre> <p>The gpstop command may take some time to stop the database.</p> 

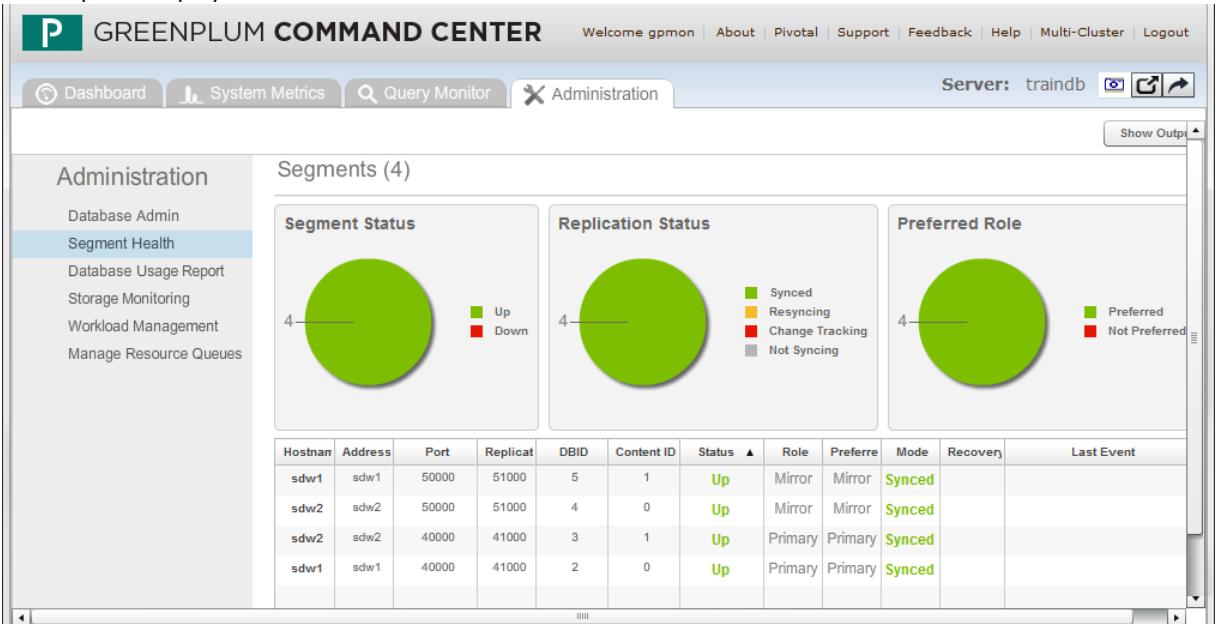
Step	Action
16.	<p>After the Greenplum Database Cluster has restarted, execute the following command:  <code>[gpadmin@mdw ~] \$ gpcmdr --setup</code></p> <p>You will be prompted for the instance name:  Type in <b>gp1</b> and press <b>Enter</b>.</p>  <pre>gpadmin@mdw:~ [gpadmin@mdw ~] \$ gpcmdr --setup An instance name is used by the Greenplum Command Center as a way to uniquely identify a Greenplum Database that has the monitoring components installed and configured. This name is also used to control specific instances of the Greenplum Command Center web UI. Instance nam es can contain letters, digits and underscores and are not case sensitive. Please enter a new instance name: &gt; gp1</pre>
17.	<p>You will be prompted with the question, Is the master host remote?  Press <b>Enter</b> to accept the default answer of No.</p>  <pre>gpadmin@mdw:~ Is the master host for the Greenplum Database remote? Yy Nn (default=N): &gt;</pre>
18.	<p>You will be prompted for the display name of the instance:  Type in <b>traindb</b> and press <b>Enter</b></p>  <pre>gpadmin@mdw:~ what would you like to use for the display name for this instance: &gt; traindb</pre>
19.	<p>You will be prompted for the port number of the database. Press <b>Enter</b> to accept the default port number of 5432. A message will appear indicating that the instance schema is being created.</p>  <pre>gpadmin@mdw:~ what port does the Greenplum Database use? (default=5432): &gt;</pre>
20.	<p>You will be prompted for the web server port. Press <b>Enter</b> to accept the default port of 28080.</p>  <pre>gpadmin@mdw:~ Creating instance schema in GPDB. Please wait ... The Greenplum Command Center runs a small web server for the UI and web API. This web server by default runs on port 28080, but you may specify any a vailable port.  what port would you like the web server to use for this instance? (defau lt=28080): &gt;</pre>

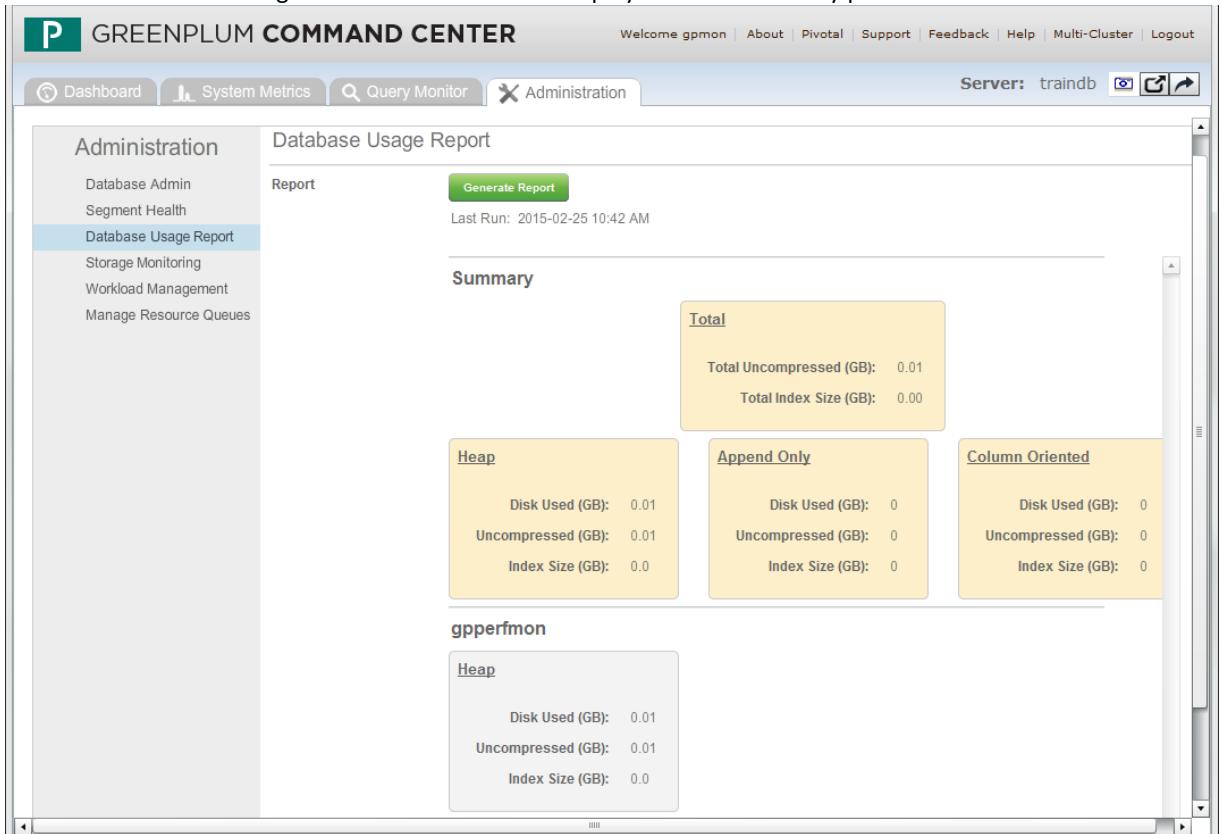
Step	Action
21.	<p>You will be prompted to enable SSL for the Web API. Type <b>Y</b> and press <b>Enter</b> to confirm SSL.</p>  <pre>gpadmin@mdw:~\$  Users logging in to the Command Center must provide database user credentials. In order to protect user names and passwords, it is recommended that SSL be enabled.  Do you want to enable SSL for the Web API Yy Nn (default=N): &gt; Y</pre>
22.	<p>You will be prompted with a series of questions for your certificate. Enter the following information for the certificate:</p> <ul style="list-style-type: none"> <li>• Country name: <b>US</b></li> <li>• State or province name: <b>CA</b></li> <li>• Locality name: <b>Palo Alto</b></li> <li>• Organization name: <b>EdServices</b></li> <li>• Organizational Unit: <b>PivotalEdu</b></li> <li>• Common name: <b>mdw</b></li> <li>• Email address: <b>gpadmin@mdw</b></li> </ul>  <pre>gpadmin@mdw:~\$  Generating a 2048 bit RSA private key .....+++ .....+++ writing new private key to '/usr/local/greenplum-cc-web./instances/gp1/conf/cert.pem'  ----- You are about to be asked to enter information that will be incorporated into your certificate request. What you are about to enter is what is called a Distinguished Name or a DN. There are quite a few fields but you can leave some blank For some fields there will be a default value, If you enter '.', the field will be left blank. ----- Country Name (2 letter code) [AU]:us State or Province Name (full name) [Some-state]:MA Locality Name (eg, city) []:Franklin Organization Name (eg, company) [Internet Widgits Pty Ltd]:Edservices Organizational Unit Name (eg, section) []:Training Common Name (e.g. server FQDN or YOUR name) []:mdw Email Address []:gpadmin@mdw</pre>
23.	<p>When prompted to enable IPV6, press the <b>ENTER</b> key to accept the default No.</p>  <pre>gpadmin@mdw:~\$  Do you want to enable ipv6 for the Web API Yy Nn (default=N): &gt;</pre>

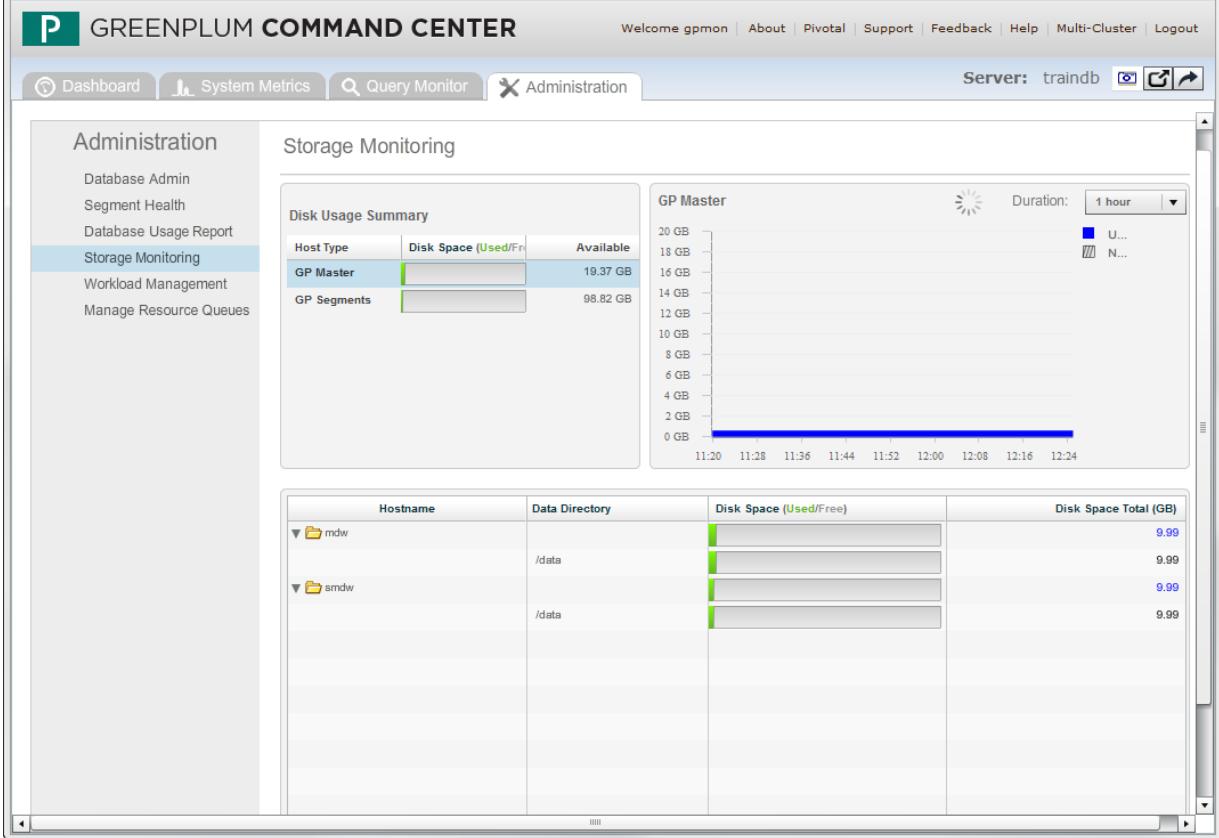
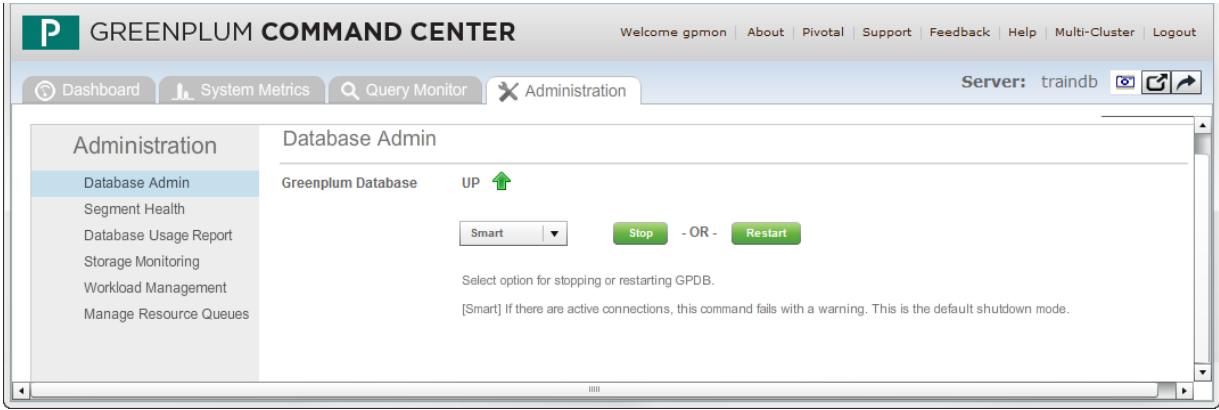
Step	Action
24.	<p>When prompted to enable cross site request forgery protection, press <b>Enter</b> to accept the default of no.</p>  <pre>gpadmin@mdw:~\$ gpcmdr --setup Do you want to enable Cross Site Request Forgery Protection for the Web API Yy Nn (default=N): &gt; </pre>
25.	<p>When prompted to copy the instance to the standby server, type <b>N</b> and press <b>Enter</b>.</p>  <pre>gpadmin@mdw:~\$ gpcmdr --setup Do you want to copy the instance to a standby master host Yy Nn (default=Y): &gt; n</pre>
	<p>The configuration process will complete, displaying the following:</p>  <pre>gpadmin@mdw:~\$ gpcmdr --setup Done writing lighttpd configuration to /usr/local/greenplum-cc-web/.instances/gp1/conf/lighttpd.conf Done writing web UI configuration to /usr/local/greenplum-cc-web/.instances/gp1/conf/gpperfmonui.conf Done writing web UI clusters configuration to /usr/local/greenplum-cc-web/.instances/gp1/conf/clusters.conf  Greenplum Command Center UI configuration is now complete. If at a later date you want to change certain parameters, you can either re-run 'gpcmdr --setup' or edit the configuration file located at /usr/local/greenplum-cc-web/.instances/gp1/conf/gpperfmonui.conf.  The web UI for this instance is available at https://mdw:28080/ You can now start the web UI for this instance by running: gpcmdr --start gp1 No instances [gpadmin@mdw ~]\$ </pre>
26.	<p>Start the Greenplum Command Center web service for the instance you created using the <b>gpcmdr --start gp1</b> command.</p> <pre>[gpadmin@mdw ~]\$ gpcmdr --start gp1</pre>  <pre>gpadmin@mdw:~\$ gpcmdr --start gp1 Starting instance gp1... Greenplum Command Center UI for instance 'gp1' - [RUNNING on PORT: 28080] [gpadmin@mdw ~]\$ </pre>
27.	<p>Using a browser of choice, connect to the URL, <a href="https://172.16.1.11:28080">https://172.16.1.11:28080</a>.</p> <p>Depending on which browser you are using you may receive a "Warning" or "Exception" that the connection is "Insecure". Since Greenplum is running on the same machine simply accept the "Exception" and continue.</p>

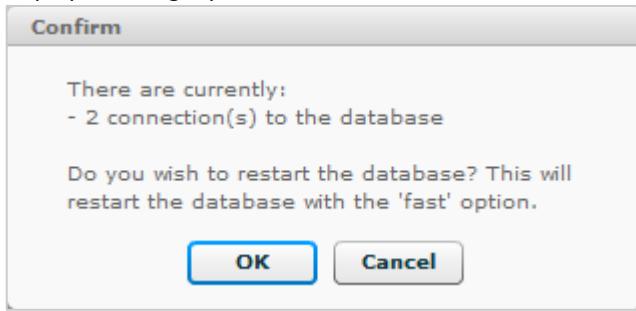
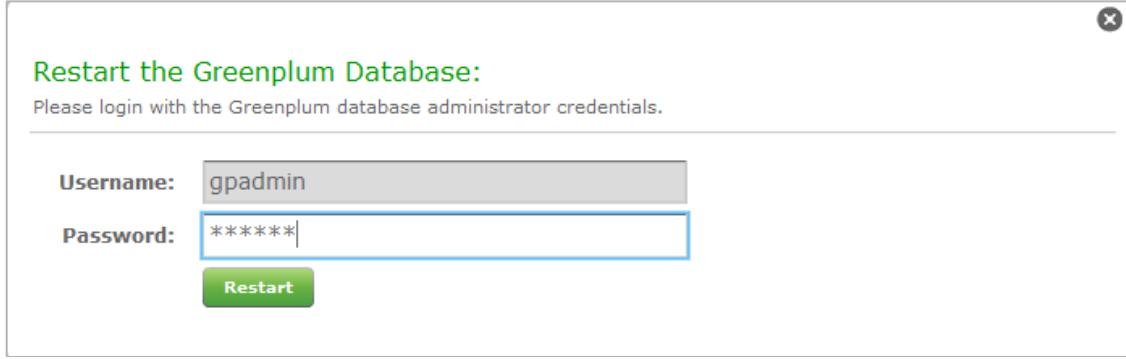
Step	Action
28.	<p>The login screen will now be displayed. Log in with the following credentials:</p> <ul style="list-style-type: none"> <li>• Username: <b>gpadmin</b></li> <li>• Password: <b>changeme</b></li> </ul>  <p>Click <b>Login</b> to proceed.</p> <p><b>Note:</b> The <b>gpadmin</b> user is a trusted account within the Greenplum Database. You can further secure the environment by requiring that the <b>gpadmin</b> user account enter a MD5 or SHA-256 encrypted password.</p>
	<p>The dashboard will be displayed and will show the current system performance information.</p>  <p>The dashboard displays various system metrics and health status:</p> <ul style="list-style-type: none"> <li><b>Greenplum Status:</b> State: NORMAL, Uptime: 0 d, 0 h, 21 min, GPDB Version: 4.3.4.0 build 1, Connections: 1, Active Queries: 0, DB Health: NORMAL.</li> <li><b>Segment Health:</b> A green circle indicating all segments are up (4).</li> <li><b>Disk Usage Summary:</b> Host Type: GP Master, Disk Space (Used): 3.96 GB, Available: 98.94 GB. A warning message: "Disks usage (master) reached to the critical point. Please clear the disk space."</li> <li><b>CPU:</b> CPU User (%): ~0.5, CPU Sys (%): ~0.5 over the last 10 hours.</li> <li><b>Queries/Jobs:</b> Running (avg): ~0, Queued (avg): ~0 over the last 10 hours.</li> <li><b>Memory:</b> Used (MBytes): ~100, Free (MBytes): ~1500 over the last 10 hours.</li> <li><b>Alerts:</b> No alerts present.</li> </ul>

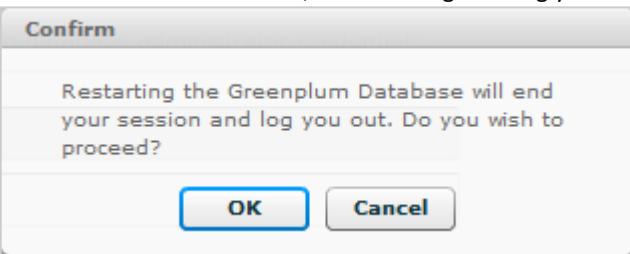
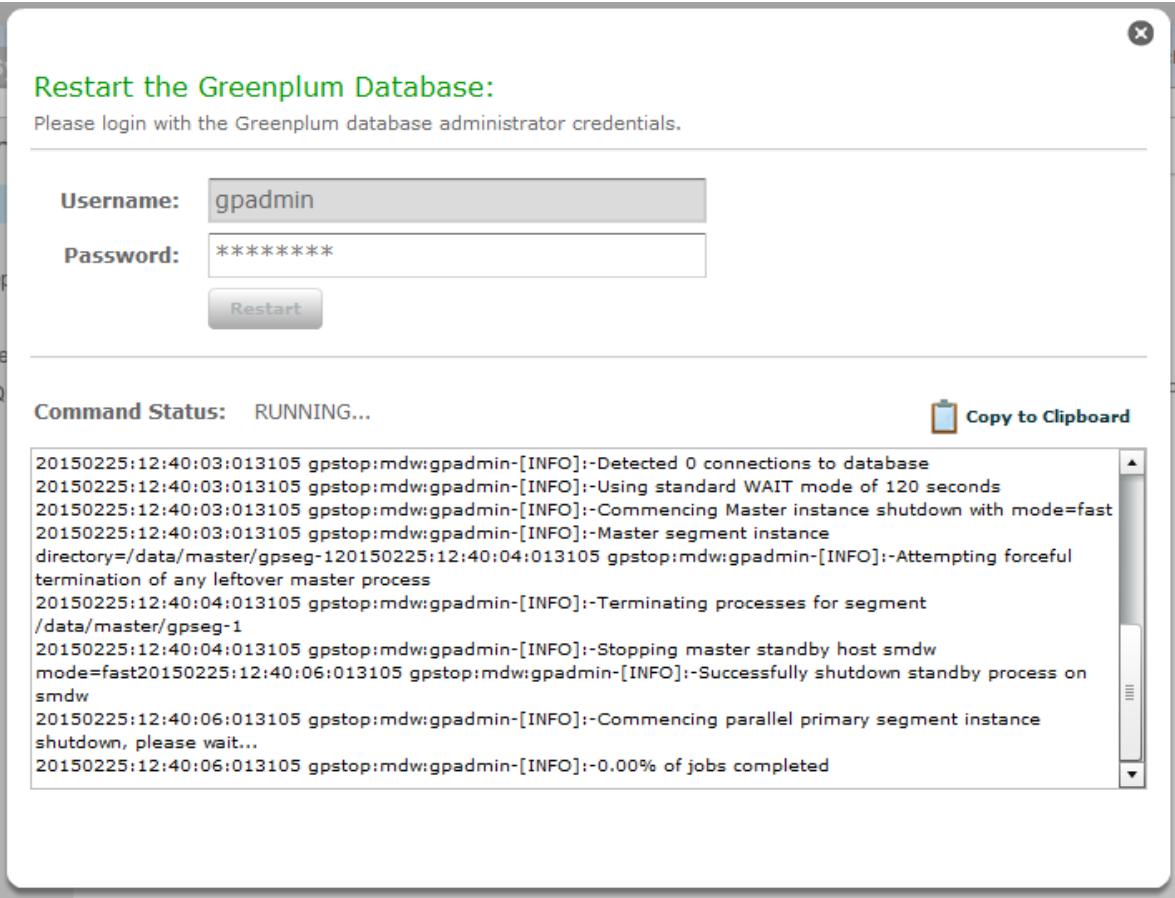
## Lab 4: Task 3 – Navigating Pivotal Greenplum Command Center

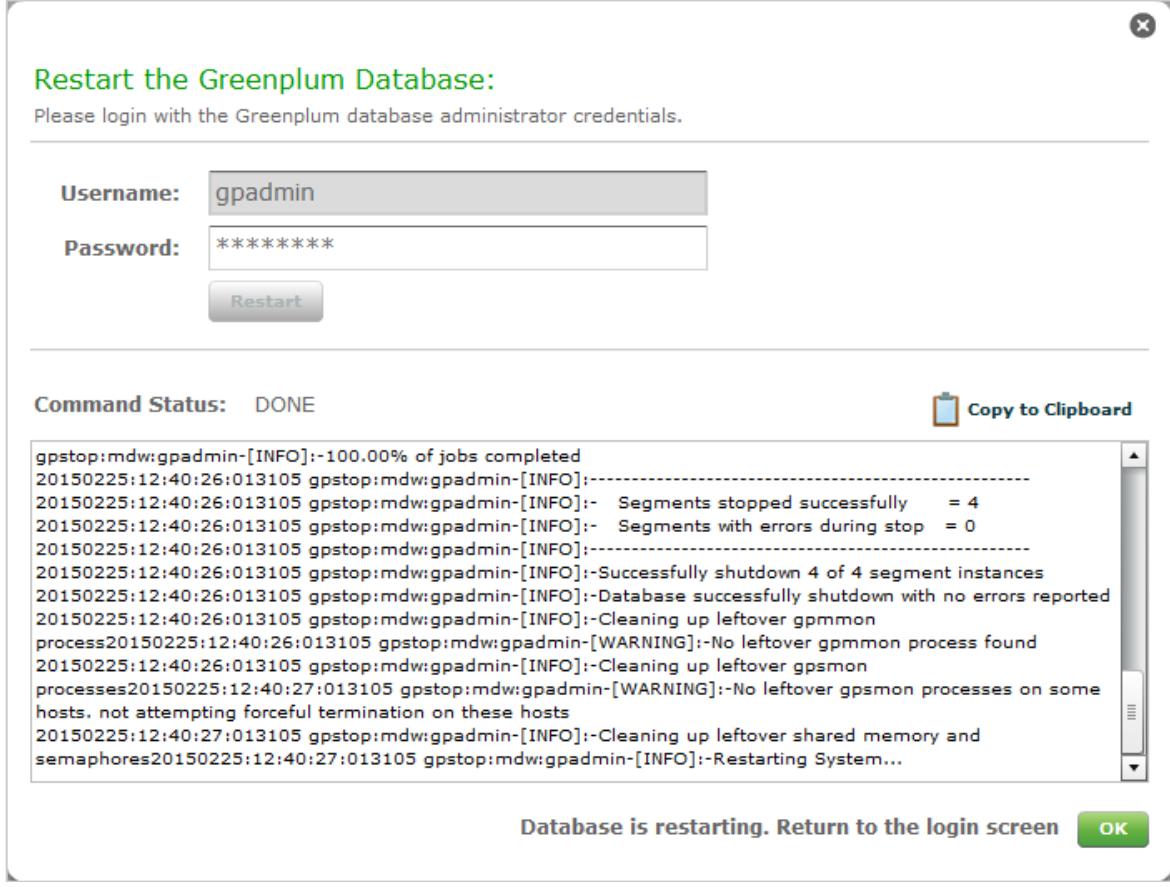
Step	Action																																																												
1.	<p>In this task, you will explore the Pivotal Greenplum Command Center environment, managing and monitoring activities within the environment.</p> <p>The Dashboard provides an overview of the environment. Collect the following information:</p> <ul style="list-style-type: none"> <li>• What is the health of the database?</li> </ul> <hr/> <ul style="list-style-type: none"> <li>• How many segments are visible within the environment and what is their status?</li> </ul> <hr/>																																																												
2.	<p>Click <b>More »</b> in the Segment Health panel. The Administration tab should now be displayed with the Segment Health panel displayed.</p>  <p>The screenshot shows the Administration tab selected in the left sidebar under Segment Health. The main area displays three circular status indicators for Segment Status, Replication Status, and Preferred Role, all showing green 'Up' status. Below these are three tables: Segment Status, Replication Status, and Preferred Role, each showing four entries. The Segment Status table has columns: Hostname, Address, Port, Replicat, DBID, Content ID, Status, Role, Preferre, Mode, Recovery, and Last Event. The data is as follows:</p> <table border="1"> <thead> <tr> <th>Hostname</th> <th>Address</th> <th>Port</th> <th>Replicat</th> <th>DBID</th> <th>Content ID</th> <th>Status</th> <th>Role</th> <th>Preferre</th> <th>Mode</th> <th>Recovery</th> <th>Last Event</th> </tr> </thead> <tbody> <tr> <td>sdw1</td> <td>sdw1</td> <td>50000</td> <td>51000</td> <td>5</td> <td>1</td> <td>Up</td> <td>Mirror</td> <td>Mirror</td> <td>Synced</td> <td></td> <td></td> </tr> <tr> <td>sdw2</td> <td>sdw2</td> <td>50000</td> <td>51000</td> <td>4</td> <td>0</td> <td>Up</td> <td>Mirror</td> <td>Mirror</td> <td>Synced</td> <td></td> <td></td> </tr> <tr> <td>sdw2</td> <td>sdw2</td> <td>40000</td> <td>41000</td> <td>3</td> <td>1</td> <td>Up</td> <td>Primary</td> <td>Primary</td> <td>Synced</td> <td></td> <td></td> </tr> <tr> <td>sdw1</td> <td>sdw1</td> <td>40000</td> <td>41000</td> <td>2</td> <td>0</td> <td>Up</td> <td>Primary</td> <td>Primary</td> <td>Synced</td> <td></td> <td></td> </tr> </tbody> </table> <p>Are all of the segments and mirrors in their preferred roles? _____</p>	Hostname	Address	Port	Replicat	DBID	Content ID	Status	Role	Preferre	Mode	Recovery	Last Event	sdw1	sdw1	50000	51000	5	1	Up	Mirror	Mirror	Synced			sdw2	sdw2	50000	51000	4	0	Up	Mirror	Mirror	Synced			sdw2	sdw2	40000	41000	3	1	Up	Primary	Primary	Synced			sdw1	sdw1	40000	41000	2	0	Up	Primary	Primary	Synced		
Hostname	Address	Port	Replicat	DBID	Content ID	Status	Role	Preferre	Mode	Recovery	Last Event																																																		
sdw1	sdw1	50000	51000	5	1	Up	Mirror	Mirror	Synced																																																				
sdw2	sdw2	50000	51000	4	0	Up	Mirror	Mirror	Synced																																																				
sdw2	sdw2	40000	41000	3	1	Up	Primary	Primary	Synced																																																				
sdw1	sdw1	40000	41000	2	0	Up	Primary	Primary	Synced																																																				

Step	Action
3.	<p>From the left panel, click Database Usage Report. Click Generate Report to run a report highlighting how the database is being used. The results will be displayed in the Summary panel.</p>  <p>Once the database is populated, this provides a graphical method of seeing the structure of the database, including the type of tables and the different databases available in the environment. Based on the report, are there any databases listed in your environment? _____</p>

Step	Action																				
4.	<p>From the left panel, click Storage Monitoring. The page displays a usage summary for your environment.</p>  <table border="1" data-bbox="626 692 1535 967"> <thead> <tr> <th>Hostname</th> <th>Data Directory</th> <th>Disk Space (Used/Free)</th> <th>Disk Space Total (GB)</th> </tr> </thead> <tbody> <tr> <td>mdw</td> <td>/data</td> <td>[Green 1GB bar]</td> <td>9.99</td> </tr> <tr> <td>mdw</td> <td>/data</td> <td>[Green 1GB bar]</td> <td>9.99</td> </tr> <tr> <td>smdw</td> <td>/data</td> <td>[Green 1GB bar]</td> <td>9.99</td> </tr> <tr> <td>smdw</td> <td>/data</td> <td>[Green 1GB bar]</td> <td>9.99</td> </tr> </tbody> </table> <p>Information on the GP Master is displayed by default. This includes all masters, the primary and standby masters. Use the panel below the Disk Usage Summary to answer the following:</p> <ul style="list-style-type: none"> <li>• What is the data directory for the master and standby servers? _____</li> <li>• What is the percentage of data used on sdw1? _____</li> </ul>	Hostname	Data Directory	Disk Space (Used/Free)	Disk Space Total (GB)	mdw	/data	[Green 1GB bar]	9.99	mdw	/data	[Green 1GB bar]	9.99	smdw	/data	[Green 1GB bar]	9.99	smdw	/data	[Green 1GB bar]	9.99
Hostname	Data Directory	Disk Space (Used/Free)	Disk Space Total (GB)																		
mdw	/data	[Green 1GB bar]	9.99																		
mdw	/data	[Green 1GB bar]	9.99																		
smdw	/data	[Green 1GB bar]	9.99																		
smdw	/data	[Green 1GB bar]	9.99																		
5.	<p>From the left panel, click Database Admin. This displays a panel that allows you to change the state of the database.</p>  <p>Select option for stopping or restarting GPDB. [Smart] If there are active connections, this command fails with a warning. This is the default shutdown mode.</p>																				

Step	Action
6.	<p>Restart the database in Smart mode. If there are any open connections to the database, a dialog box will be displayed asking if you wish to continue the restart with the <b>fast</b> option.</p>  <p>Click <b>OK</b> to proceed. You should have at least one connection to the database. Where is that connection coming from?</p>
7.	<p>Enter the password for the gpadmin account and click <b>Restart</b>.</p> 

Step	Action
8.	<p>You will be asked to confirm, with a dialog warning you that you will end your session.</p>  <p>Click <b>OK</b> to proceed. You should see the following as the database is restarted.</p>  <pre data-bbox="388 952 1470 1284"> 20150225:12:40:03:013105 gpstop:mdw:gpadmin-[INFO]:-Detected 0 connections to database 20150225:12:40:03:013105 gpstop:mdw:gpadmin-[INFO]:-Using standard WAIT mode of 120 seconds 20150225:12:40:03:013105 gpstop:mdw:gpadmin-[INFO]:-Commencing Master instance shutdown with mode=fast 20150225:12:40:03:013105 gpstop:mdw:gpadmin-[INFO]:-Master segment instance directory=/data/master/gpseg-120150225:12:40:04:013105 gpstop:mdw:gpadmin-[INFO]:-Attempting forceful termination of any leftover master process 20150225:12:40:04:013105 gpstop:mdw:gpadmin-[INFO]:-Terminating processes for segment /data/master/gpseg-1 20150225:12:40:04:013105 gpstop:mdw:gpadmin-[INFO]:-Stopping master standby host smdw mode=fast20150225:12:40:06:013105 gpstop:mdw:gpadmin-[INFO]:-Successfully shutdown standby process on smdw 20150225:12:40:06:013105 gpstop:mdw:gpadmin-[INFO]:-Commencing parallel primary segment instance shutdown, please wait... 20150225:12:40:06:013105 gpstop:mdw:gpadmin-[INFO]:-0.00% of jobs completed </pre>

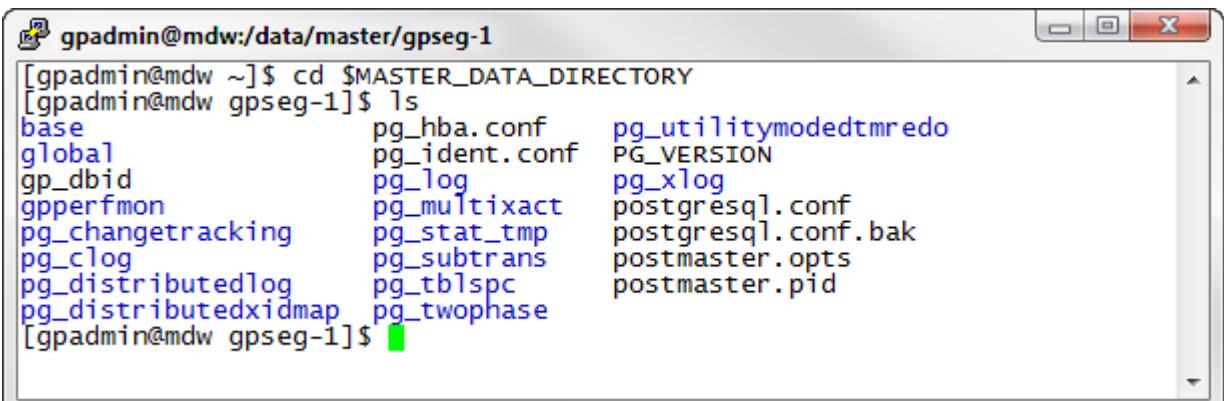
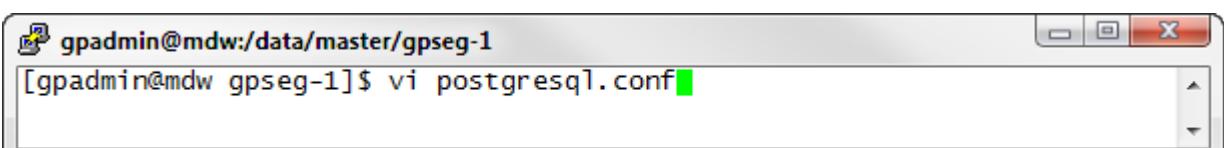
Step	Action
9.	<p>Once the database has restarted, click <b>OK</b> to return to the login screen to log back in.</p>  <p>The screenshot shows a modal dialog box titled "Restart the Greenplum Database:". It contains a "Username:" field with "gpadmin" and a "Password:" field with masked text. A "Restart" button is below the fields. Below the dialog is a "Command Status: DONE" message and a scrollable terminal window displaying gpstop logs. At the bottom is a green "Database is restarting. Return to the login screen" button with an "OK" button next to it.</p> <pre data-bbox="376 629 1470 967"> gpstop:mdw:gpadmin-[INFO]:-100.00% of jobs completed 20150225:12:40:26:013105 gpstop:mdw:gpadmin-[INFO]:----- 20150225:12:40:26:013105 gpstop:mdw:gpadmin-[INFO]:- Segments stopped successfully = 4 20150225:12:40:26:013105 gpstop:mdw:gpadmin-[INFO]:- Segments with errors during stop = 0 20150225:12:40:26:013105 gpstop:mdw:gpadmin-[INFO]:----- 20150225:12:40:26:013105 gpstop:mdw:gpadmin-[INFO]:-Successfully shutdown 4 of 4 segment instances 20150225:12:40:26:013105 gpstop:mdw:gpadmin-[INFO]:-Database successfully shutdown with no errors reported 20150225:12:40:26:013105 gpstop:mdw:gpadmin-[INFO]:-Cleaning up leftover gpmon processes20150225:12:40:26:013105 gpstop:mdw:gpadmin-[WARNING]:-No leftover gpmon process found 20150225:12:40:26:013105 gpstop:mdw:gpadmin-[INFO]:-Cleaning up leftover gpsmon processes20150225:12:40:27:013105 gpstop:mdw:gpadmin-[WARNING]:-No leftover gpsmon processes on some hosts, not attempting forceful termination on these hosts 20150225:12:40:27:013105 gpstop:mdw:gpadmin-[INFO]:-Cleaning up leftover shared memory and semaphores20150225:12:40:27:013105 gpstop:mdw:gpadmin-[INFO]:-Restarting System... </pre>
10.	Log into Pivotal Command Center with the <code>gpadmin</code> user account.

End of Lab Exercise

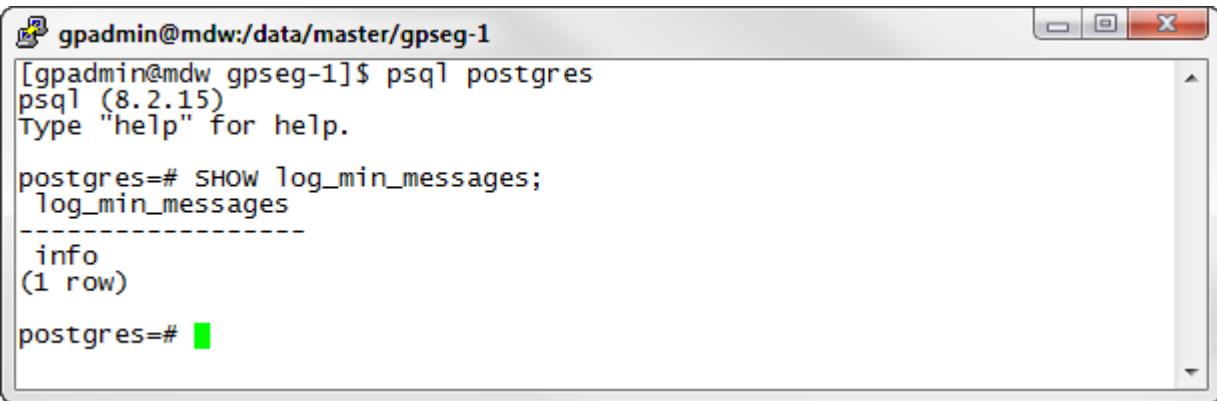
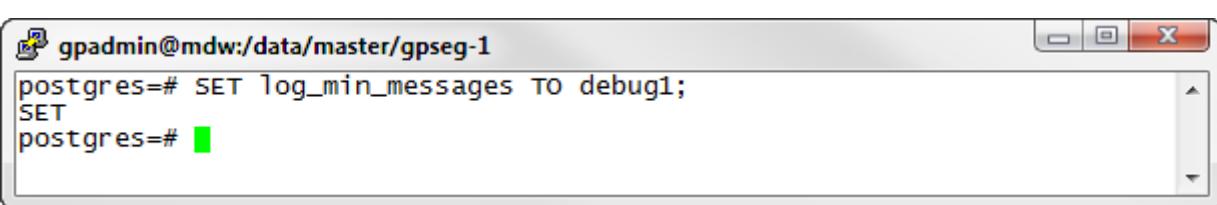
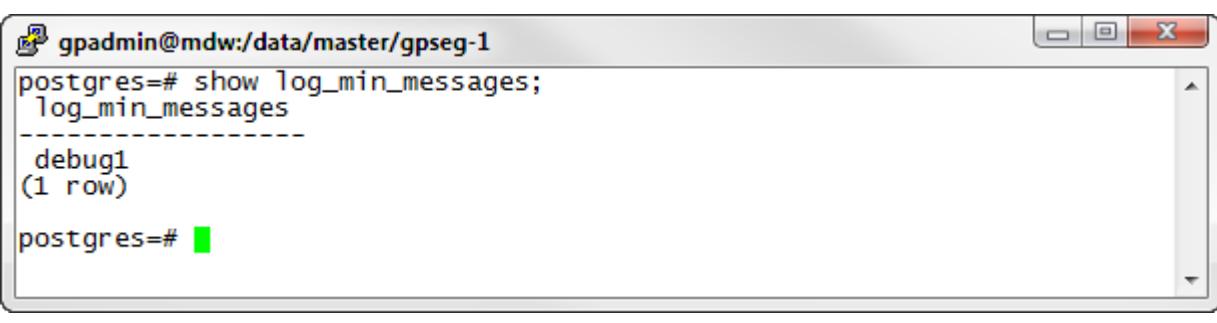
## Lab 5. Greenplum Database Server Configuration

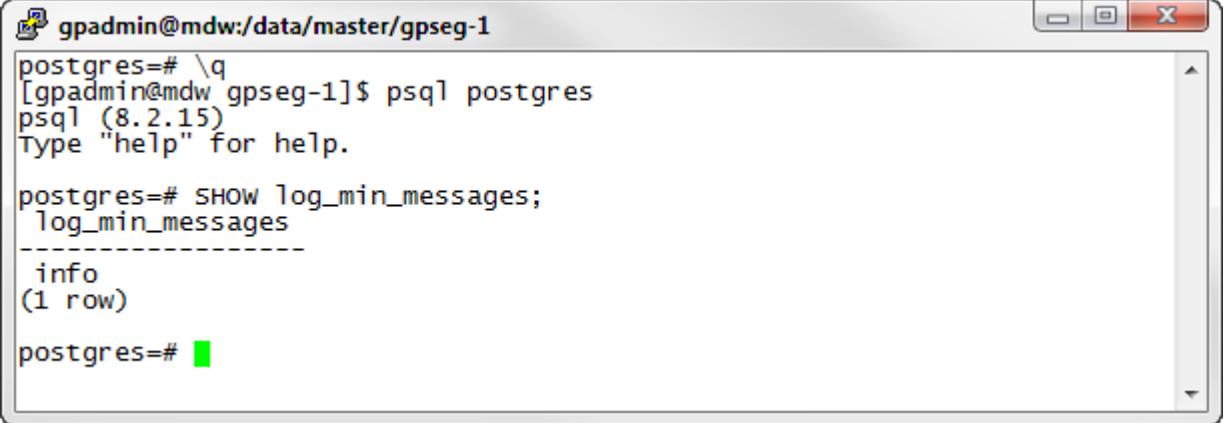
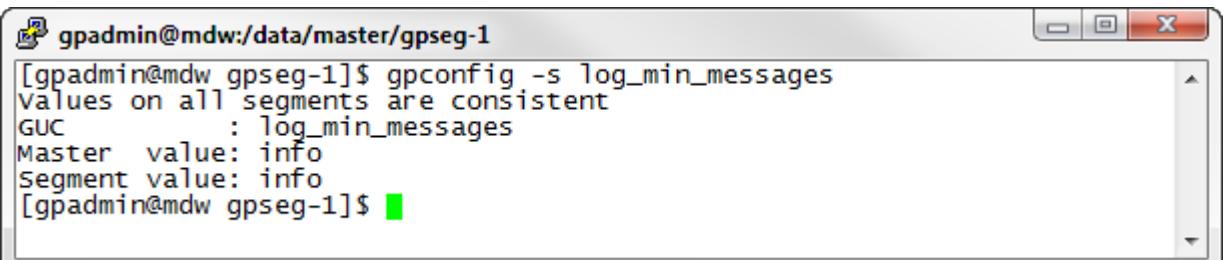
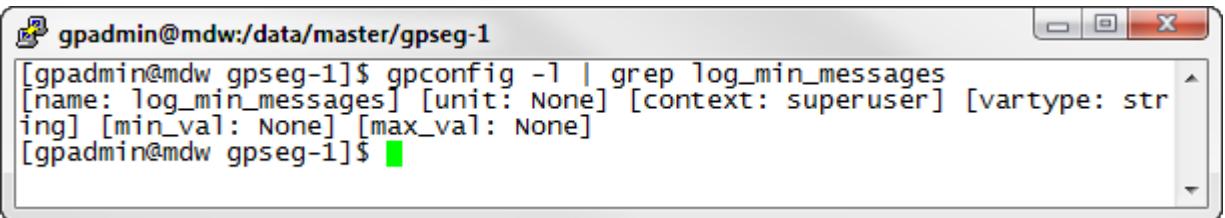
<b>Purpose:</b>	In this lab you will learn how to set configuration parameters in <code>postgresql.conf</code> and use the <code>SHOW</code> command in <code>psql</code> to examine the current values of configuration parameters. You will also learn how to set Greenplum global parameters on a per-session basis using the <code>SET</code> command in <code>psql</code> .
<b>Tasks:</b>	Students perform the following tasks: <ul style="list-style-type: none"><li>• Set Configuration Parameters</li><li>• Configure host-based authentication</li></ul>
<b>References:</b>	Module 3 – Greenplum Database Tools, Utilities, and Internals <ul style="list-style-type: none"><li>• Lesson: Greenplum Database Server Configuration</li></ul>

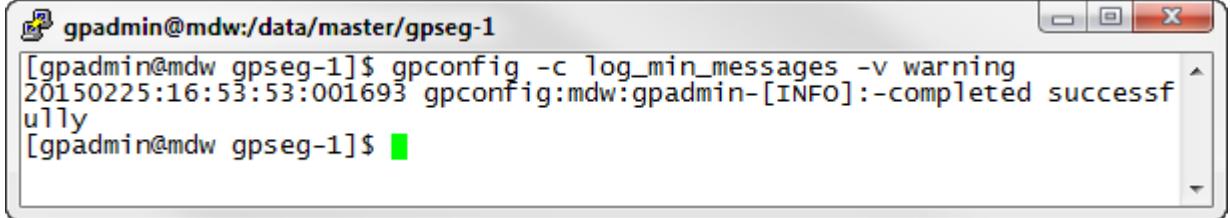
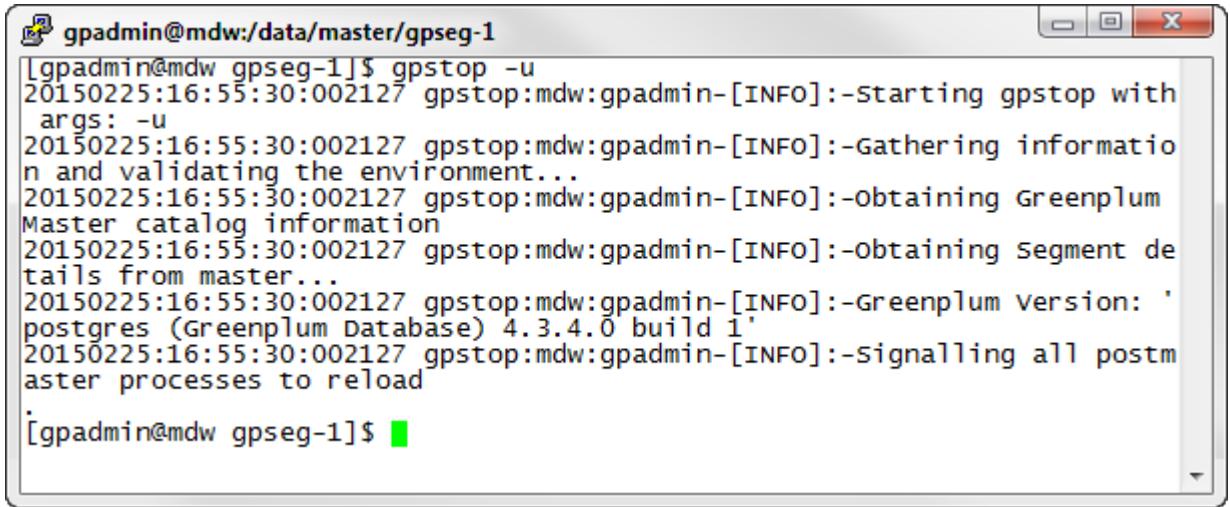
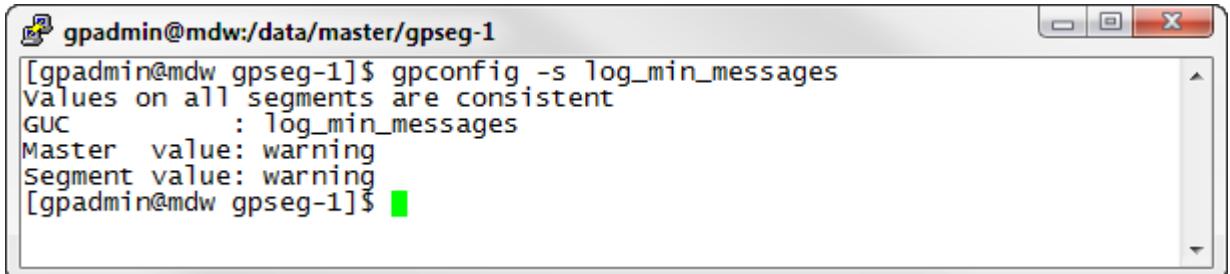
## Lab 5: Task 1 – Set Configuration Parameters

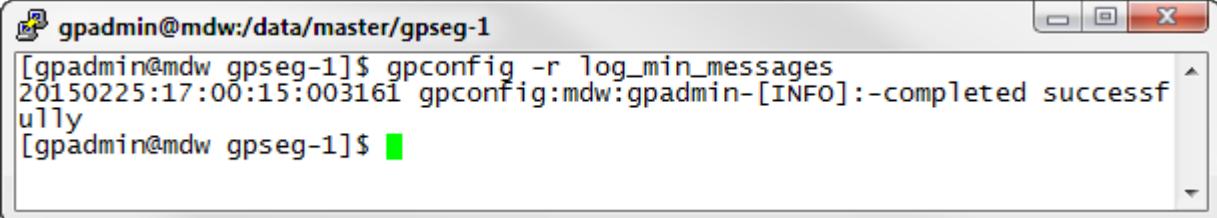
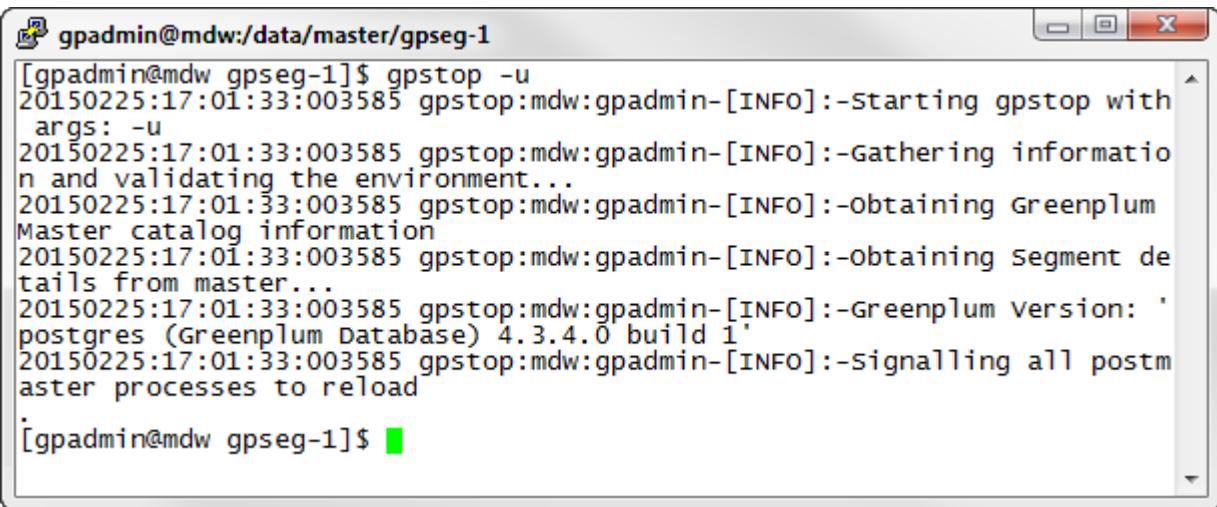
Step	Action
1.	<p>Change to the data directory of your Greenplum Database master instance:</p> <pre>[gpadmin@mdw ~]\$ cd \$MASTER_DATA_DIRECTORY [gpadmin@mdw gpseg-1]\$ ls</pre> 
2.	<p>In a text editor, edit the postgresql.conf file.</p> <pre>[gpadmin@mdw gpseg-1]\$ vi postgresql.conf</pre> 

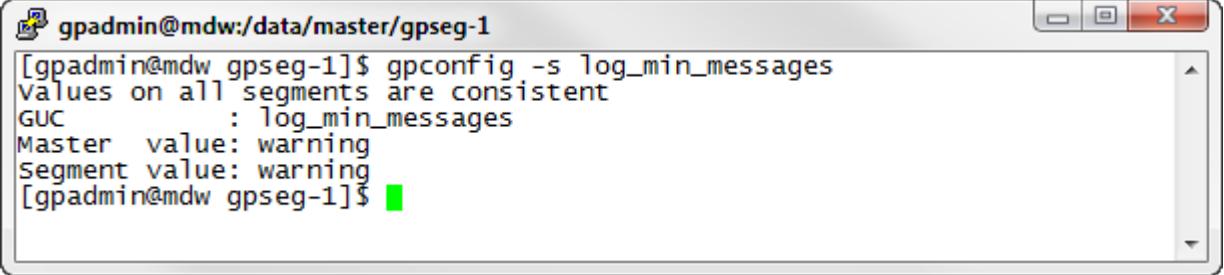
Step	Action
3.	<p>Find the configuration parameter named <code>log_min_messages</code>. This parameter controls the level of detail written to the log files.</p> <p>Uncomment this line by removing the <code>#</code> preceding it. Change the value from <code>warning</code> to <code>info</code>.</p> <pre data-bbox="332 297 1545 1072"><code>gpadmin@mdw:/data/master/gpseg-1</code> # debug5 # debug4 # debug3 # debug2 # debug1 # log # notice # warning # error  <b>log_min_messages = info</b> # values in order of decreasing detail: # debug5 # debug4 # debug3 # debug2 # debug1 # info # notice # warning # error # log # fatal # panic</pre> <p>Save your changes and exit vi.</p>
4.	<p>The <code>postgresql.conf</code> file for the master instance must be reloaded so that the changes are picked up by the Greenplum Database. Use the <code>gpstop</code> utility to complete this task:</p> <pre data-bbox="332 1290 1545 1875">[gpadmin@mdw gpseg-1]\$ <b>gpstop -u</b>  <code>gpadmin@mdw:/data/master/gpseg-1</code> [gpadmin@mdw gpseg-1]\$ gpstop -u 20150225:13:48:39:020201 gpstop:mdw:gpadmin-[INFO]:-starting gpstop with args: -u 20150225:13:48:39:020201 gpstop:mdw:gpadmin-[INFO]:-Gathering information and validating the environment... 20150225:13:48:39:020201 gpstop:mdw:gpadmin-[INFO]:-obtaining Greenplum Master catalog information 20150225:13:48:39:020201 gpstop:mdw:gpadmin-[INFO]:-obtaining segment de tails from master... 20150225:13:48:39:020201 gpstop:mdw:gpadmin-[INFO]:-Greenplum version: 'postgres (Greenplum Database) 4.3.4.0 build 1' 20150225:13:48:39:020201 gpstop:mdw:gpadmin-[INFO]:-signalling all postm aster processes to reload . [gpadmin@mdw gpseg-1]\$</pre>

Step	Action
5.	<p>Once the change has been loaded, start a psql session and check the value using the SHOW command to verify the log level was changed to <b>info</b>:</p> <pre>[gpadmin@mdw gpseg-1]\$ psql postgres postgres=# SHOW log_min_messages;</pre> 
6.	<p>While still in psql, change the value of <code>log_min_messages</code> to <code>debug1</code> using the <code>SET</code> command. This sets the parameter for the current session only.</p> <pre>postgres=# SET log_min_messages TO debug1;</pre> 
7.	<p>Check the value using the <code>SHOW</code> command to make sure it was changed to <code>debug1</code>:</p> <pre>postgres=# SHOW log_min_messages;</pre> 

Step	Action
8.	<p>Exit the current psql session and start a new session.</p> <p>Run the SHOW command again. The value should be info again, the value you set in postgresql.conf</p> <pre>postgres=# \q [gpadmin@mdw gpseg-1]\$ psql postgres postgres=# SHOW log_min_messages;</pre> 
9.	<p>Exit psql:</p> <pre>postgres=# \q</pre>
10.	<p>Using the gpconfig command, list the values of the log_min_messages parameter.</p> <pre>[gpadmin@mdw gpseg-1]\$ gpconfig -s log_min_messages</pre> 
11.	<p>Use the gpconfig command to determine if the same command can be used to set the log_min_messages parameter.</p> <pre>[gpadmin@mdw gpseg-1]\$ gpconfig -l   grep log_min_messages</pre> 

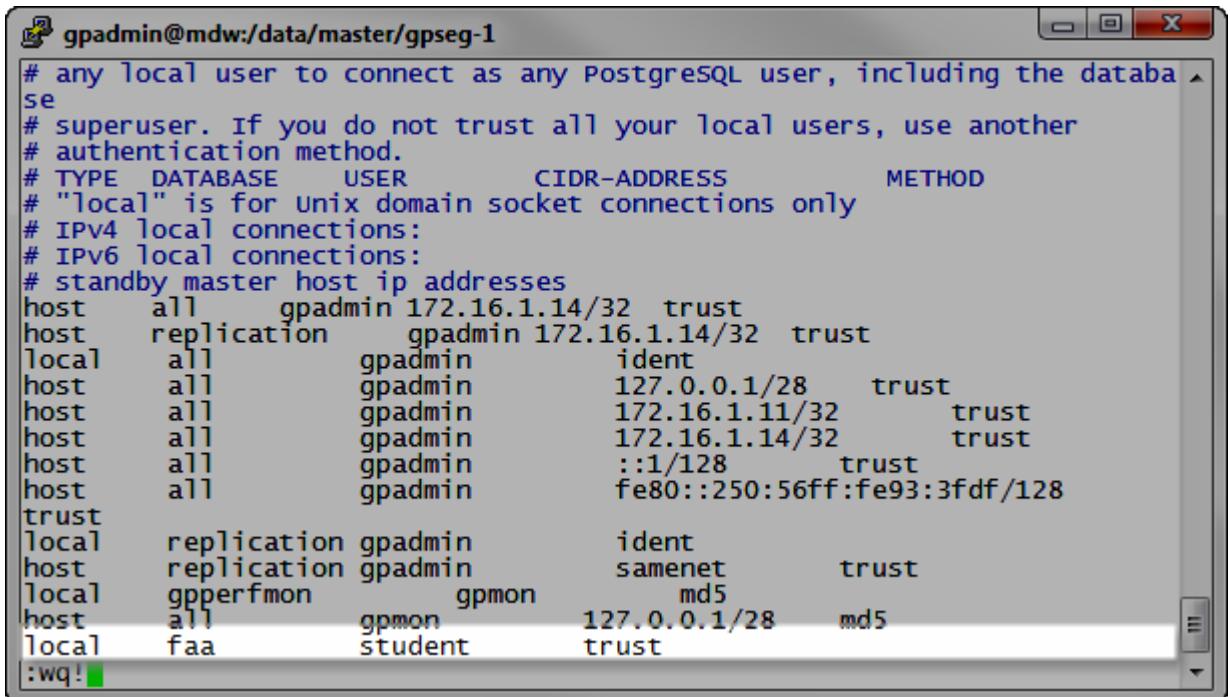
Step	Action
12.	<p>Using the <code>gpconfig</code> command, modify the <code>log_min_messages</code> parameter once more to change it to warning.</p> <pre>[gpadmin@mdw gpseg-1]\$ <b>gpconfig -c log_min_messages -v warning</b></pre> 
13.	<p>Using <code>gpstop</code>, reread the <code>postgresql.conf</code> file.</p> <pre>[gpadmin@mdw gpseg-1]\$ <b>gpstop -u</b></pre> 
14.	<p>Use <code>gpconfig</code> to verify the <code>log_min_messages</code> parameter has been updated on all servers.</p> <pre>[gpadmin@mdw gpseg-1]\$ <b>gpconfig -s log_min_messages</b></pre> 

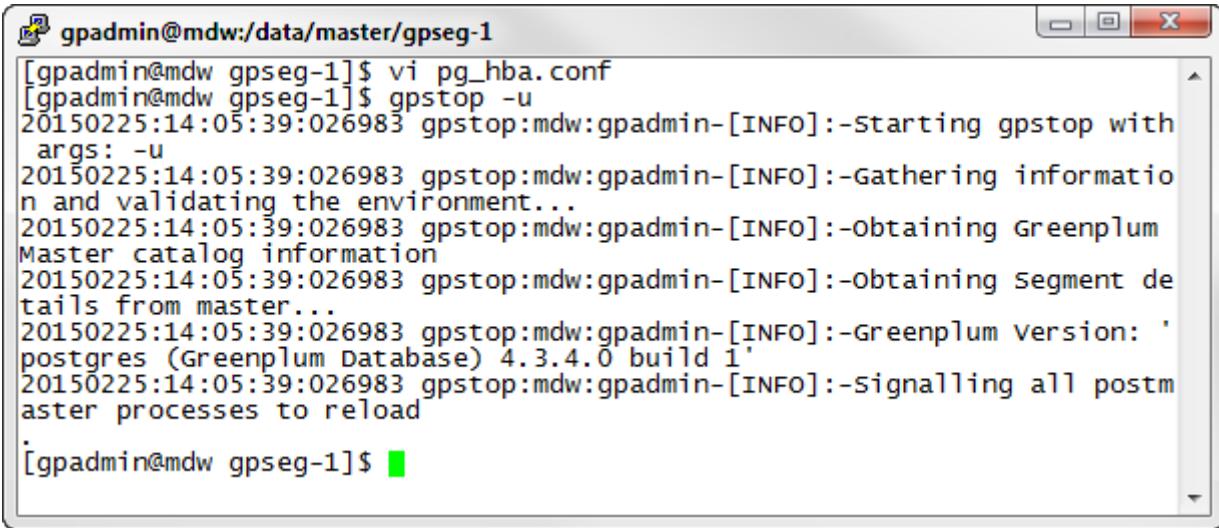
Step	Action
15.	<p>Using the gpconfig command, remove the changes that you made to the log_min_messages parameter.</p> <pre>[gpadmin@mdw gpseg-1]\$ <b>gpconfig -r log_min_messages</b></pre> 
16.	<p>Reload the postgresql.conf file.</p> <pre>[gpadmin@mdw gpseg-1]\$ <b>gpstop -u</b></pre> 

Step	Action
17.	<p>Verify the changes have been reset.</p> <pre>[gpadmin@mdw gpseg-1]\$ gpconfig -s log_min_messages</pre>  <p>Is this the expected value? Why or why not?</p> <hr/> <hr/> <hr/>

Step	Action
	<p><b>Summary</b></p> <p>Setting server configuration parameters can be done at the system level by editing the <code>postgresql.conf</code> configuration file on the Greenplum master. The preferred method for editing some parameters is with the <code>gpconfig</code> command. The command, <code>gpconfig -l</code> lists all the parameters you can change with <code>gpconfig</code>. You must either reload the master <code>postgresql.conf</code> file with the <code>gpstop -u</code> command or restart the Greenplum Database with the <code>gpstop -r</code> before changes are visible.</p> <p>Note that some parameters are global and some are not. Global parameters need only be set on the master and the value is passed on to the segments. The non-global parameters must be set in the <code>postgresql.conf</code> on the master and on all segments as well and require a system restart.</p> <p>You can also set global parameters on a per-session basis using the <code>SET</code> command. Note that it is also possible to set parameters at the database level (using <code>ALTER DATABASE SET</code>) or at the role level (<code>ALTER ROLE SET</code>).</p> <p>To examine the current setting of configuration parameters, use the <code>SHOW</code> command. You can also use the <code>gpconfig -s</code> command to list the values on all servers in the cluster. <code>SHOW ALL</code> will show the values for all parameters. You will have to page through the commands or type <code>q</code> to stop.</p> <p>Optional steps:</p> <ol style="list-style-type: none"> <li>1. Connect to the <code>postgres</code> database:  <code>\$ psql postgres</code></li> <li>2. Issue the <code>SHOW ALL</code> command:  <code>postgres=# SHOW ALL;</code></li> <li>3. Exit <code>psql</code>:  <code>postgres=# \q</code></li> </ol>

## Lab 5: Task 2 – Configure Host-Based Authentication

Step	Action
1.	<p>If not already there, change to the directory represented by the variable, \$MASTER_DATA_DIRECTORY, to perform this exercise. Using vi, edit the pg_hba.conf file.</p> <pre>[gpadmin@mdw gpseg-1]\$ cd \$MASTER_DATA_DIRECTORY [gpadmin@mdw gpseg-1]\$ vi pg_hba.conf</pre>
2.	<p>In follow on labs, you will create a database called faa and you will add a database role for yourself. Add the following line at the bottom of pg_hba.conf, replacing &lt;your_name&gt; with your first name:</p> <pre>local    faa    &lt;your_name&gt;    trust</pre>  <p>Save your changes and exit vi.</p> <p>Before proceeding to the next step:</p> <ol style="list-style-type: none"> <li>1. Verify that all characters entered are lowercase.</li> <li>2. Do not use tabs to separate each field in the pg_hba.conf file. Instead, use spaces.</li> </ol>

Step	Action
3.	<p>Reload the configuration for the changes to take effect:</p> <pre>[gpadmin@mdw gpseg-1]\$ gpstop -u</pre>  <pre>[gpadmin@mdw gpseg-1]\$</pre>
	<p><b>Summary</b></p> <p>The pg_hba.conf file controls the hosts and users that can connect to the Greenplum Database, the databases they can access, and how users are authenticated.</p> <p>After a Greenplum Database system is first initialized, only the Greenplum superuser, the user who ran gpinit孤tem, is allowed to connect from the local host. If you want to allow other users or client machines to connect, or change the default means of authentication, you must edit the pg_hba.conf file of the master instance.</p> <p>It is often necessary to give access with the addition of new databases or when requesting remote access through a query editor such as pgAdmin III.</p> <p><b>Note:</b> You must reload the configuration (or restart the Greenplum Database) for changes to pg_hba.conf to take effect.</p>

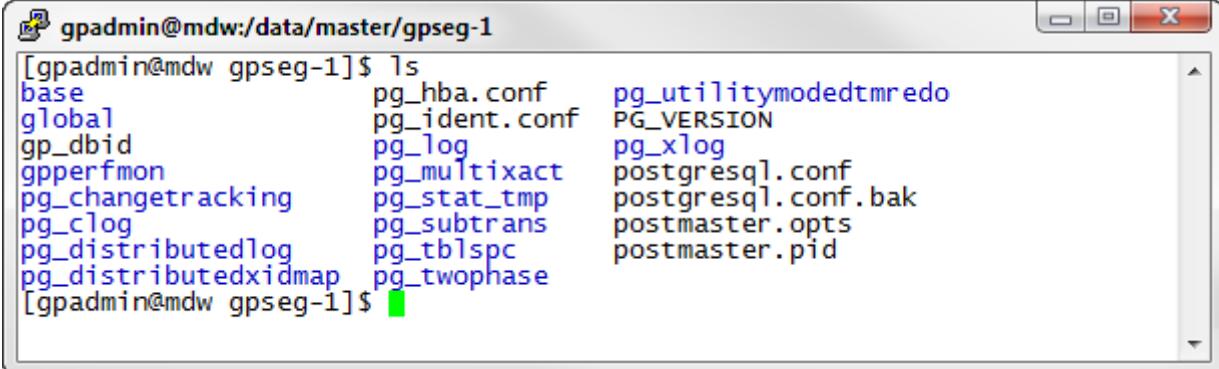
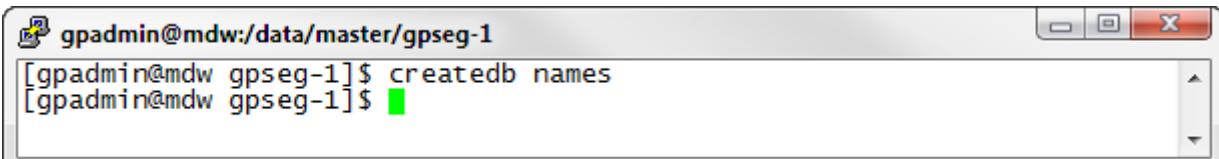
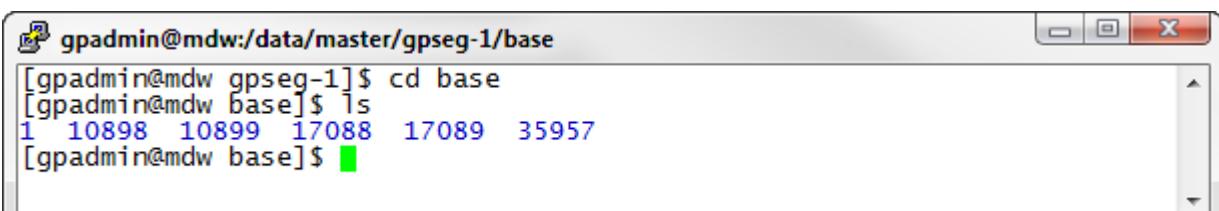
End of Lab Exercise

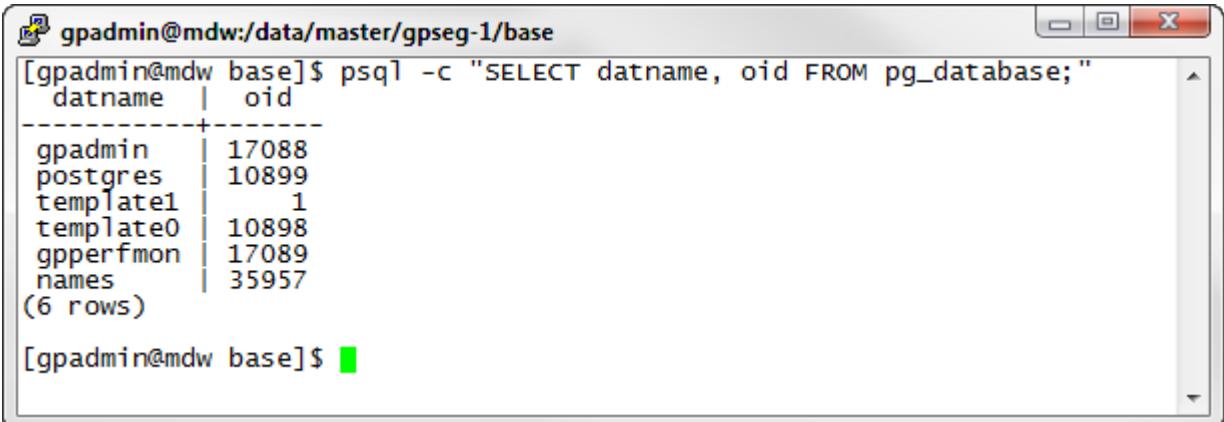
This page intentionally left blank

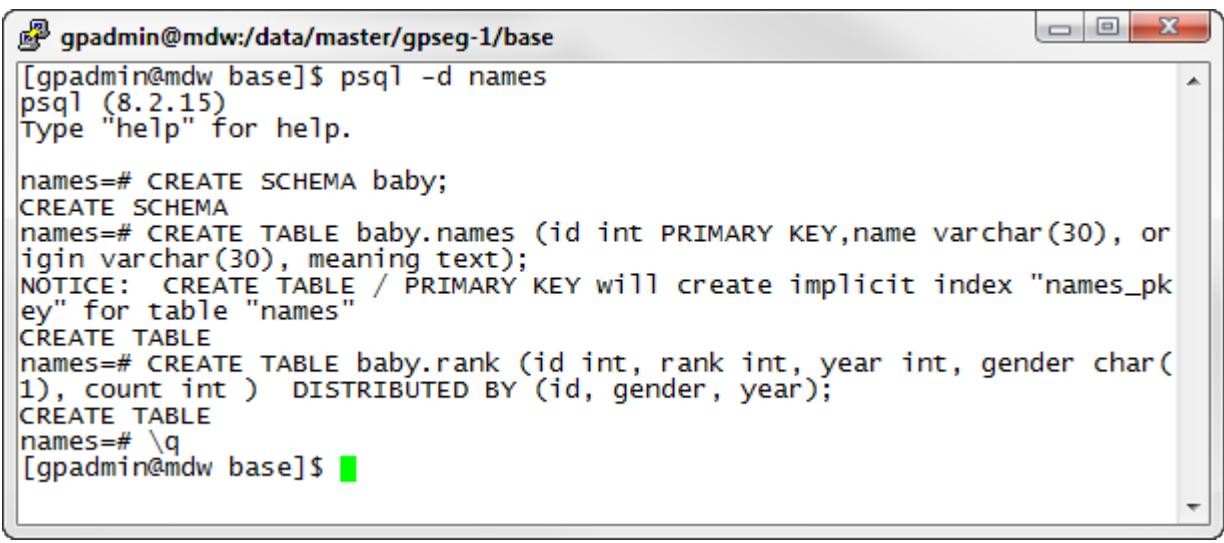
## Lab 6. Database Internals

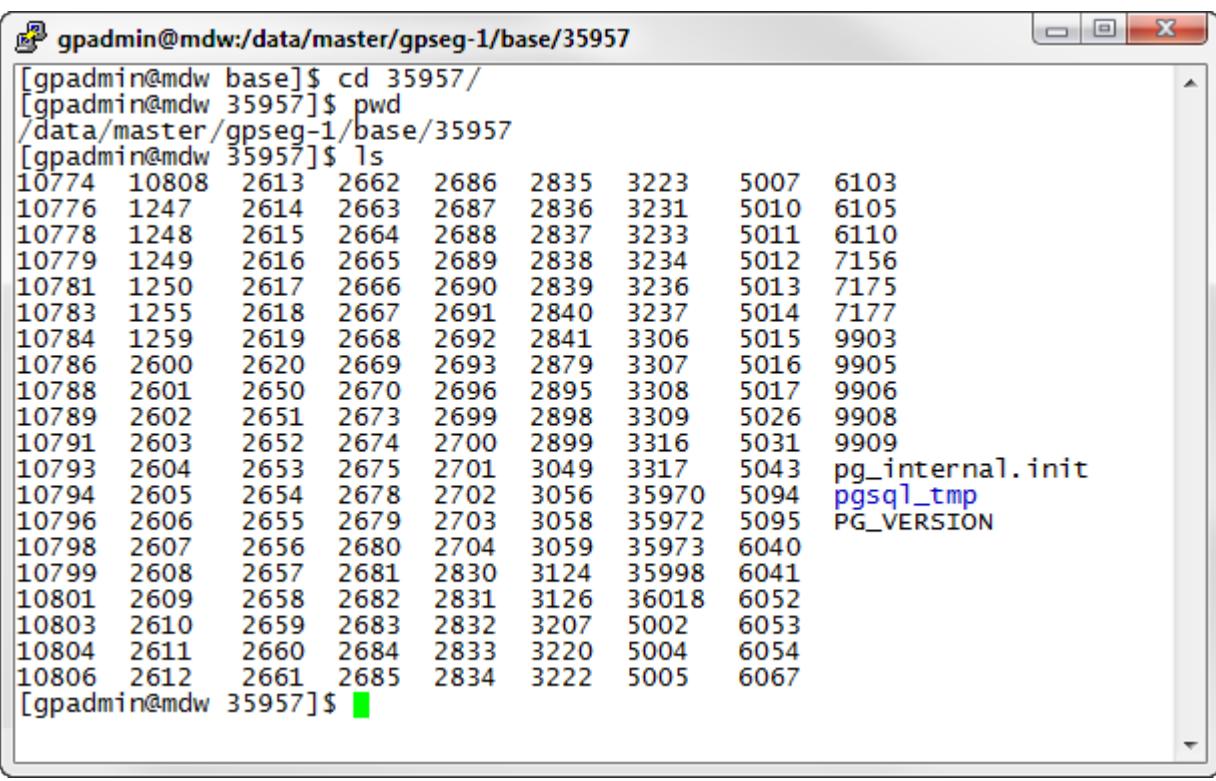
<b>Purpose:</b>	<p>Each instance in the Greenplum Database, the master and each segment, has a data directory where database files are stored on disk. In this lab, you examine the database files on disk and determine which database objects they correspond to by looking up their OID (object identifier) numbers in the system catalog tables.</p> <p>You will examine the database server processes running on a Greenplum Database host.</p>
<b>Tasks:</b>	<p>Students perform the following tasks:</p> <ul style="list-style-type: none"><li>• Explore database files on disk</li><li>• Examine server processes</li></ul>
<b>References:</b>	<p>Module 3 – Greenplum Database Tools, Utilities, and Internals</p> <ul style="list-style-type: none"><li>• Lesson: Database Internals</li></ul>

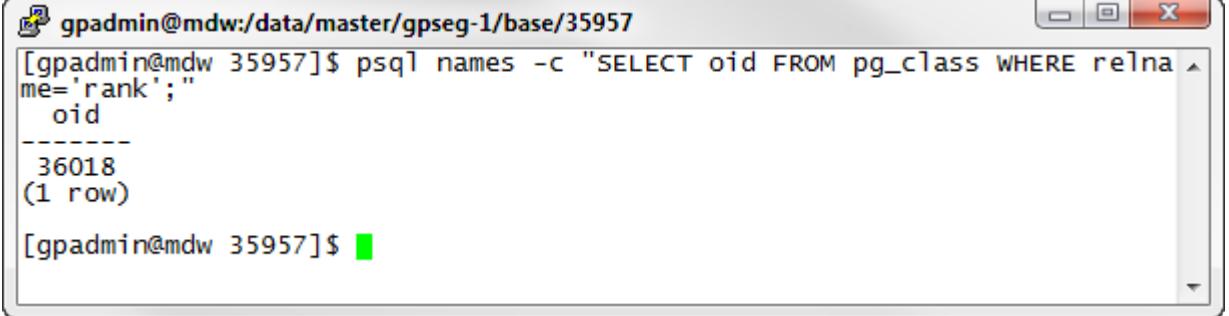
## Lab 6: Task 1 – Explore Database Files on Disk

Step	Action
1.	<p>You should be logged as gpadmin at <code>MASTER_DATA_DIRECTORY</code>. If you are not at the master data directory cd to it.</p> <pre>[gpadmin@mdw ~] \$ cd \$MASTER_DATA_DIRECTORY [gpadmin@mdw gpseg-1]\$ ls</pre>  <p>You should see several directories.</p>
2.	<p>Create a database called names.</p> <pre>[gpadmin@mdw gpseg-1]\$ createdb names</pre>  <p><b>Note:</b> The <code>createdb</code> command is a Greenplum utility that is executed from the command line. The equivalent SQL command is</p> <pre>CREATE DATABASE names;</pre>
3.	<p>Change to the <code>base</code> subdirectory. This is where database files are stored. You should see a subdirectory for each database that exists in your Greenplum Database system:</p> <pre>[gpadmin@mdw gpseg-1]\$ cd base [gpadmin@mdw base]\$ ls</pre>  <p>How many subdirectories are located in the <code>base</code> subdirectory? _____</p>

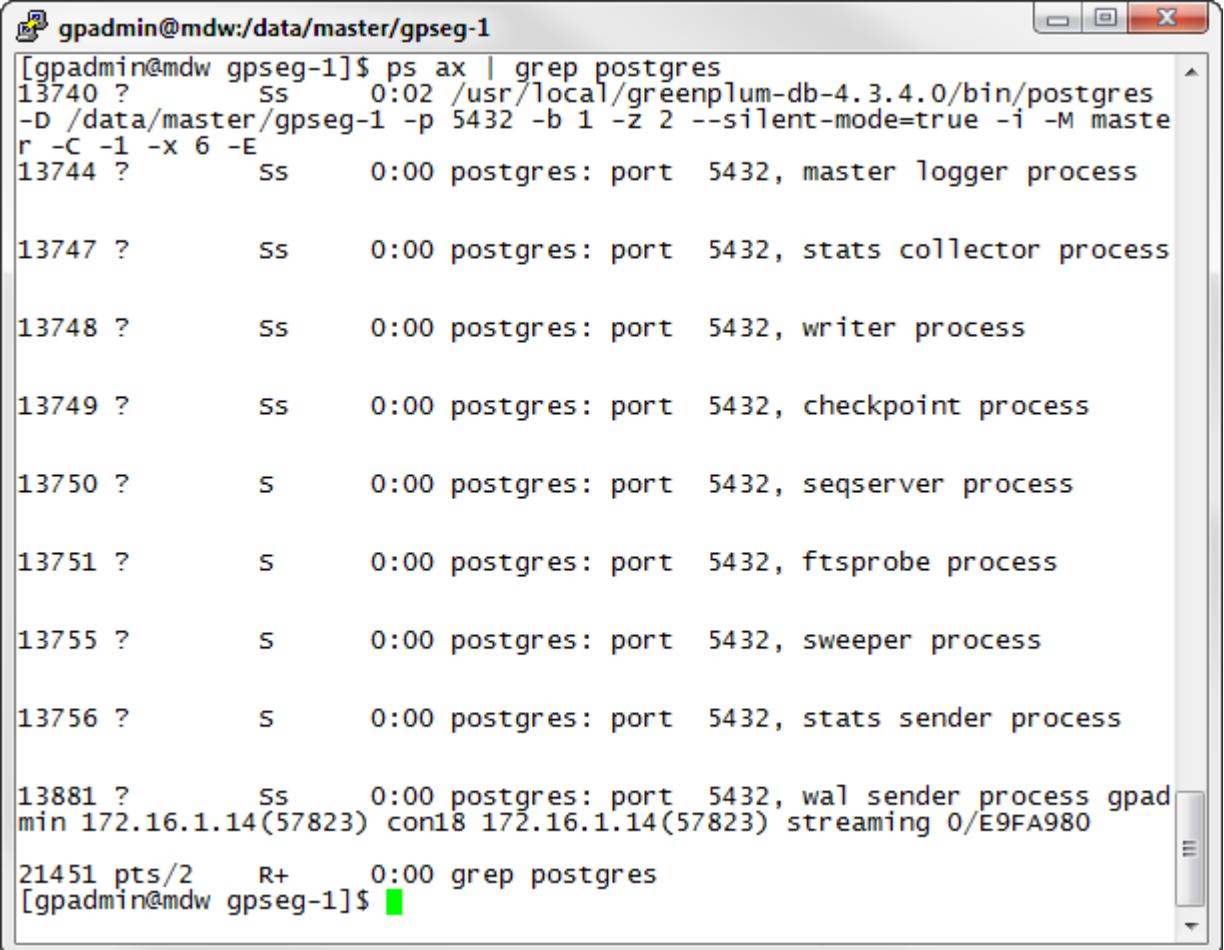
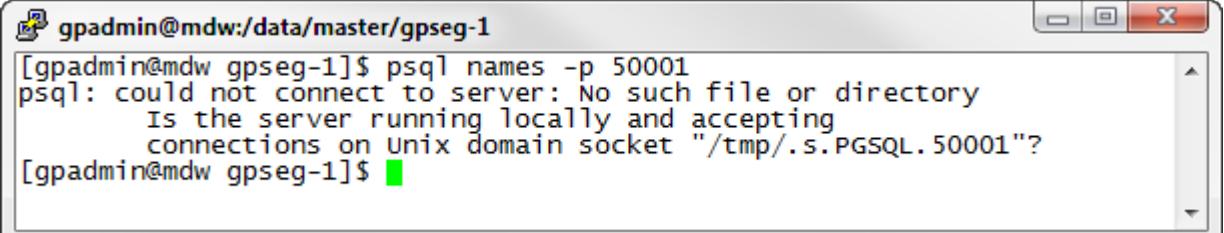
Step	Action
4.	<p>Find out which database each directory corresponds to by looking up the OID number in the system catalog table, pg_database:</p> <pre>[gpadmin@mdw base]\$ psql -c "SELECT datname, oid FROM pg_database;"</pre>  <p><b>Note:</b> Your output may differ from the screenshot shown here.</p> <p>Record the OID of the names database in the space provided.</p> <hr/>

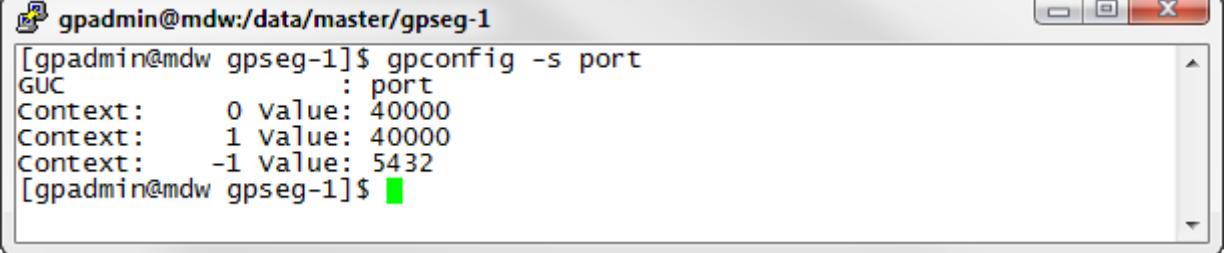
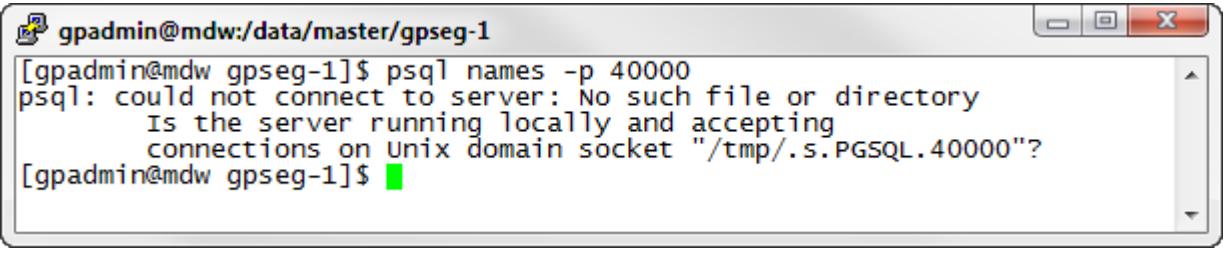
Step	Action
5.	<p>Using the SQL statement provided, create a schema and tables in the names database:</p> <pre>[gpadmin@mdw base]\$ psql -d names names=# CREATE SCHEMA baby;  names=# CREATE TABLE baby.names (id int PRIMARY KEY, name varchar(30), origin varchar(30), meaning text);  names=# CREATE TABLE baby.rank (id int, rank int, year int, gender char(1), count int) DISTRIBUTED BY (id, gender, year);  names=# \q</pre>  <p>You will examine the syntax in greater detail in upcoming lessons and labs.</p>

Step	Action																																																
6.	<p>Change to the names database directory on disk and examine the files in that subdirectory. You must know the OID of the names database.</p> <p>There are several files, identified by OID, that correspond to database objects. View the contents of the names database represented by its OID with the following commands:</p> <pre>[gpadmin@mdw base]\$ cd &lt;oid_of_names_db&gt; [gpadmin@mdw base]\$ ls</pre> <p>In this example, we use the OID for the names database obtained earlier in the lab.</p> <pre>[gpadmin@mdw base]\$ cd 35957 [gpadmin@mdw 35957]\$ pwd [/data/master/gpseg-1/base/35957] [gpadmin@mdw 35957]\$ ls</pre>  <table border="1"> <caption>Output of ls command for directory /data/master/gpseg-1/base/35957</caption> <thead> <tr> <th>OID</th> <th>Name</th> </tr> </thead> <tbody> <tr><td>10774</td><td>10808</td></tr> <tr><td>10776</td><td>1247</td></tr> <tr><td>10778</td><td>1248</td></tr> <tr><td>10779</td><td>1249</td></tr> <tr><td>10781</td><td>1250</td></tr> <tr><td>10783</td><td>1255</td></tr> <tr><td>10784</td><td>1259</td></tr> <tr><td>10786</td><td>2600</td></tr> <tr><td>10788</td><td>2601</td></tr> <tr><td>10789</td><td>2602</td></tr> <tr><td>10791</td><td>2603</td></tr> <tr><td>10793</td><td>2604</td></tr> <tr><td>10794</td><td>2605</td></tr> <tr><td>10796</td><td>2606</td></tr> <tr><td>10798</td><td>2607</td></tr> <tr><td>10799</td><td>2608</td></tr> <tr><td>10801</td><td>2609</td></tr> <tr><td>10803</td><td>2610</td></tr> <tr><td>10804</td><td>2611</td></tr> <tr><td>10806</td><td>2612</td></tr> <tr><td></td><td>pg_internal.init</td></tr> <tr><td></td><td>pgsql_tmp</td></tr> <tr><td></td><td>PG_VERSION</td></tr> </tbody> </table>	OID	Name	10774	10808	10776	1247	10778	1248	10779	1249	10781	1250	10783	1255	10784	1259	10786	2600	10788	2601	10789	2602	10791	2603	10793	2604	10794	2605	10796	2606	10798	2607	10799	2608	10801	2609	10803	2610	10804	2611	10806	2612		pg_internal.init		pgsql_tmp		PG_VERSION
OID	Name																																																
10774	10808																																																
10776	1247																																																
10778	1248																																																
10779	1249																																																
10781	1250																																																
10783	1255																																																
10784	1259																																																
10786	2600																																																
10788	2601																																																
10789	2602																																																
10791	2603																																																
10793	2604																																																
10794	2605																																																
10796	2606																																																
10798	2607																																																
10799	2608																																																
10801	2609																																																
10803	2610																																																
10804	2611																																																
10806	2612																																																
	pg_internal.init																																																
	pgsql_tmp																																																
	PG_VERSION																																																

Step	Action
7.	<p>Discover which OID file corresponds to the rank table by looking it up in the pg_class table:</p> <pre>[gpadmin@mdw 35957]\$ psql names -c "SELECT oid FROM pg_class WHERE relname='rank';"</pre>  <p>Which OID corresponds to the rank table? _____</p>
	<p><b>Summary</b></p> <p>Each master and segment instance in the Greenplum Database has a data directory where database files are stored on disk. Each database object on disk can be identified by looking up its OID in the system catalog tables. Database objects in the Greenplum Database are distributed across the segment instances. The OIDs are consistent across all instances in the array.</p>

## Lab 6: Task 2 – Examine Server Processes

Step	Action
1.	<p>You should be logged in to the master server, mdw, as gpadmin. Change to the \$MASTER_DATA_DIRECTORY directory. Find all of the postgres processes running on the system:</p> <pre>[gpadmin@mdw 35957]\$ cd \$MASTER_DATA_DIRECTORY [gpadmin@mdw gpseg-1]\$ ps ax   grep postgres</pre>  <p>How many are there? _____</p> <p>Can you identify the database listener process, also known as the postmaster?</p>
2.	<p>Connect directly to a segment instance using psql and a segment instance port:</p> <pre>[gpadmin@mdw gpseg-1]\$ psql names -p 50001</pre>  <p>What happened? _____</p>

Step	Action
3.	<p>Using the <code>gpconfig</code> command, view the defined ports for the database by examining the <code>port</code> parameter.</p> <pre>[gpadmin@mdw gpseg-1]\$ gpconfig -s port</pre> 
4.	<p>Attempt to connect to the one of the segment ports listed in the previous step.</p> <pre>[gpadmin@mdw gpseg-1]\$ psql names -p 40000</pre>  <p>Where you able to connect? Why or why not?</p> <hr/>
	<p><b>Summary</b></p> <p>The postgres database server process, known as the postmaster in prior releases accepts client connections. In the Greenplum Database, only the <code>postgres</code> listener process of the master instance can accept user client connections. The segments only allow connections from the master. The database listener process has several sub <code>postgres</code> processes running at any given time. Sub <code>postgres</code> processes are spawned as needed to handle client requests. In the Greenplum Database, the <code>ps</code> output for a <code>postgres</code> client process will have similar information as found in the Greenplum Database server logs:</p> <pre>postgres: &lt;user&gt;&lt;database&gt;&lt;host&gt;&lt;con#&gt;&lt;seg#&gt;&lt;cmd#&gt;&lt;slice#&gt; status&gt;</pre>

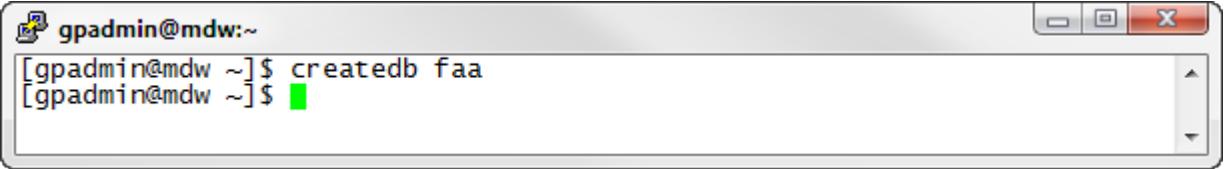
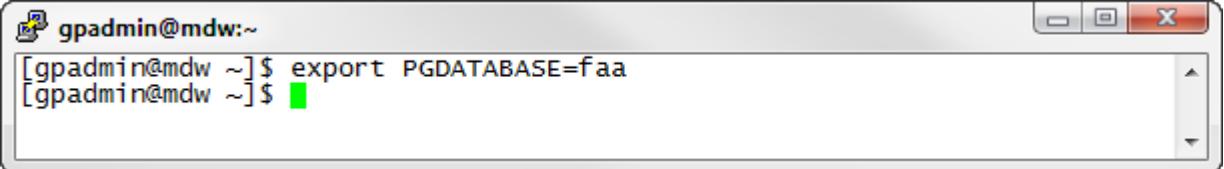
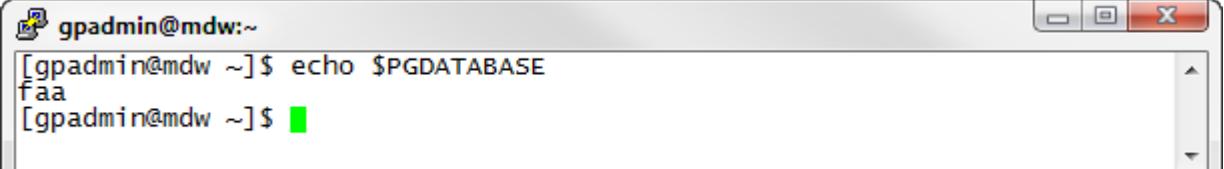
End of Lab Exercise

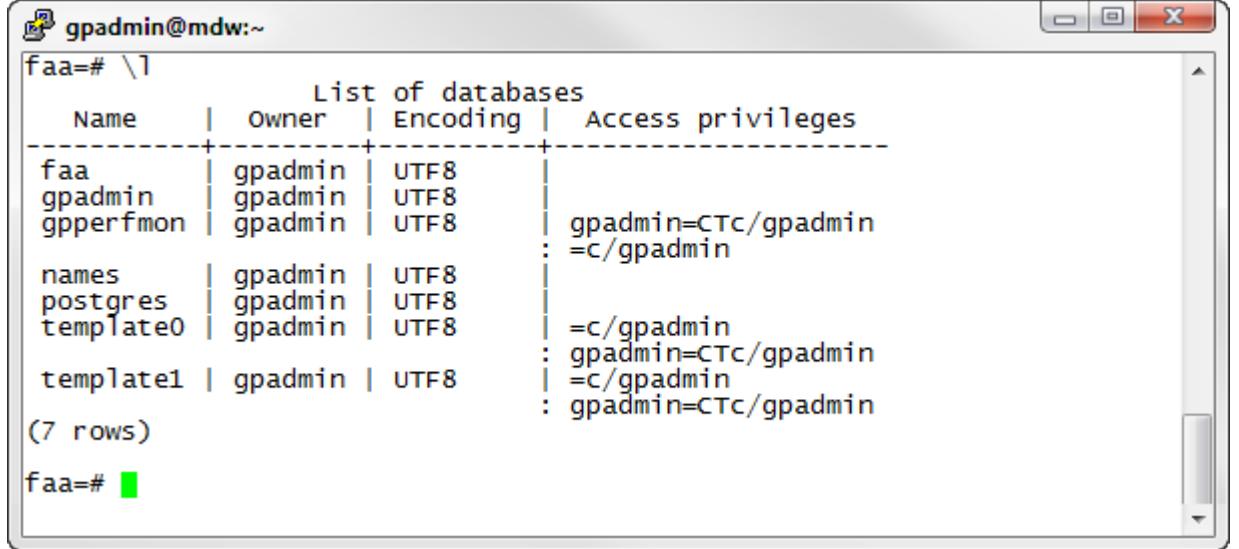
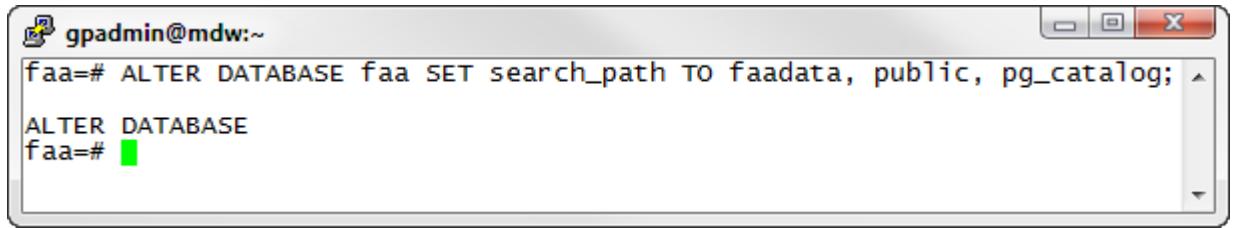
This page intentionally left blank

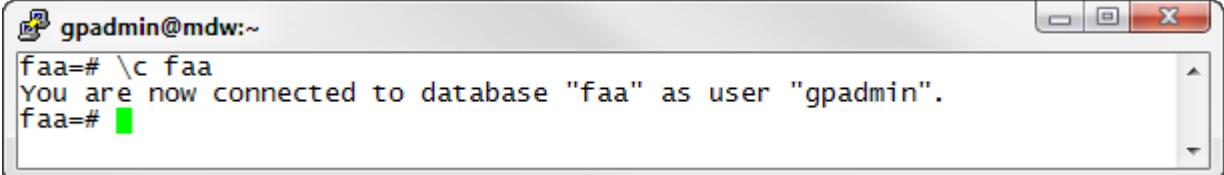
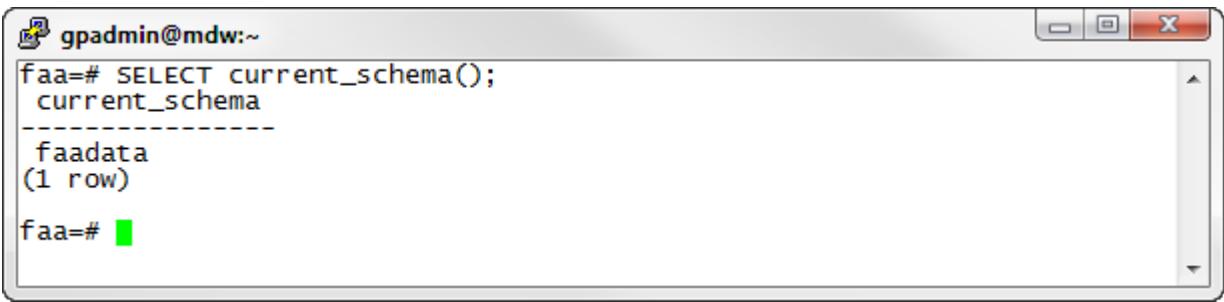
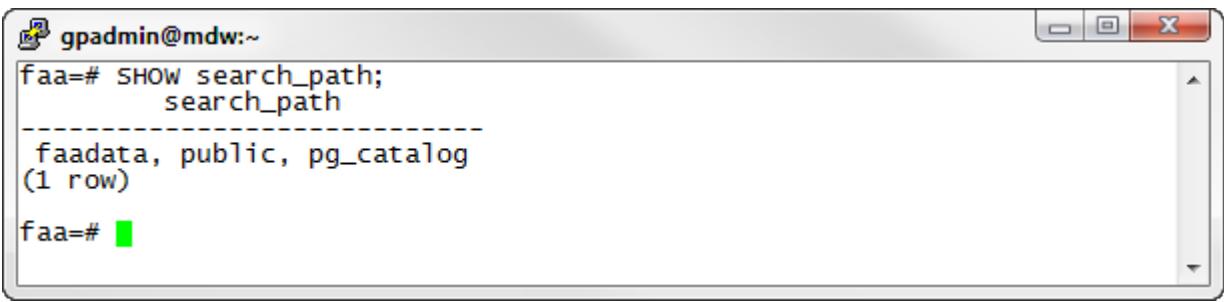
## Lab 7. Data Definition Language

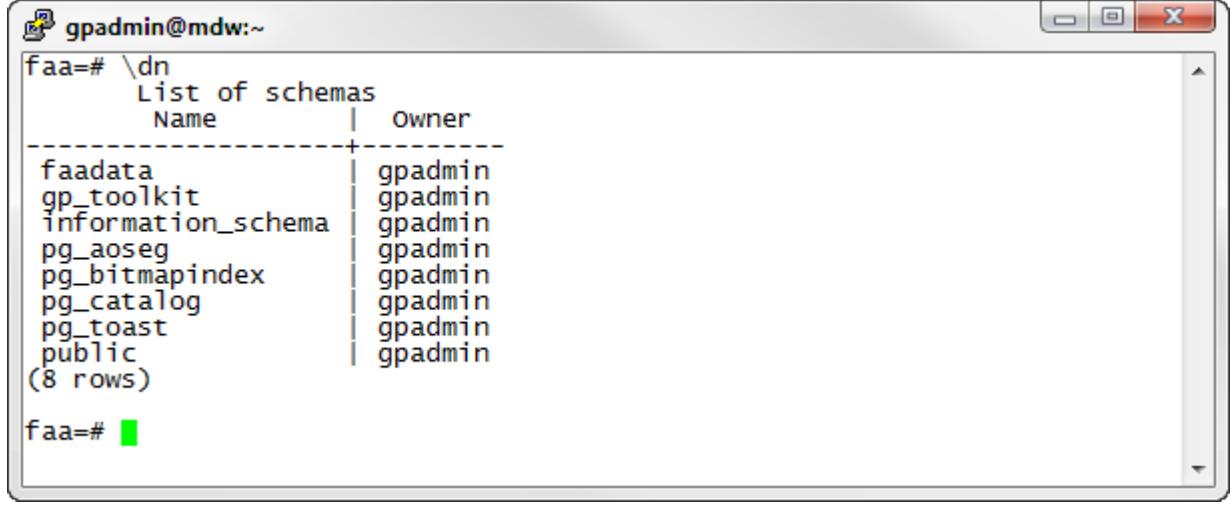
<b>Purpose:</b>	<p>In this lab, you will learn how to manage databases and schemas in the Greenplum Database. You will create a database called <code>faa</code> and a schema called <code>faadata</code>. You will also set the schema search path and learn how to verify which schema you are in.</p> <p>You will also create some tables in the Greenplum Database and learn how the Greenplum distribution key for a table is chosen.</p>
<b>Tasks:</b>	<p>Students perform the following tasks:</p> <ul style="list-style-type: none"><li>• Create databases and schemas</li><li>• Create tables</li><li>• View indexes and schemas</li></ul>
<b>References:</b>	<p>Module 4 – Defining and Securing the User Database</p> <ul style="list-style-type: none"><li>• Lesson: Data Definition Language</li></ul>

## Lab 7: Task 1 – Create Databases and Schemas

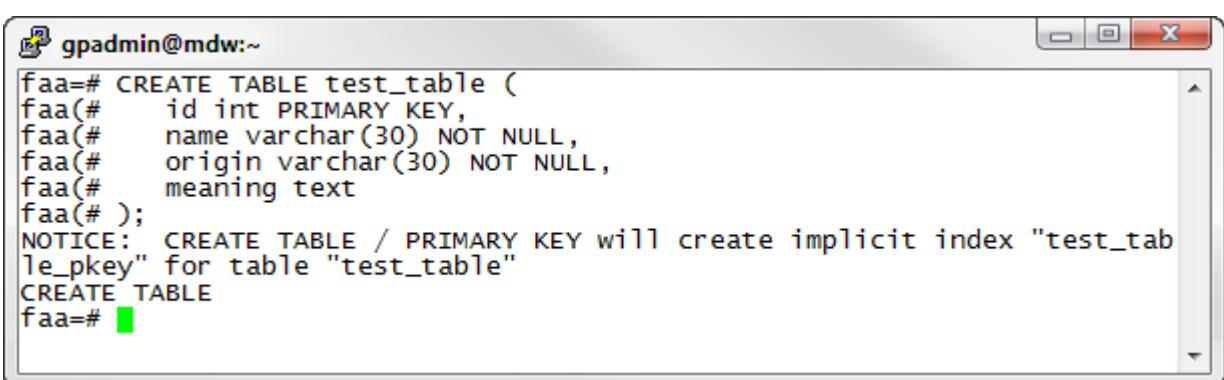
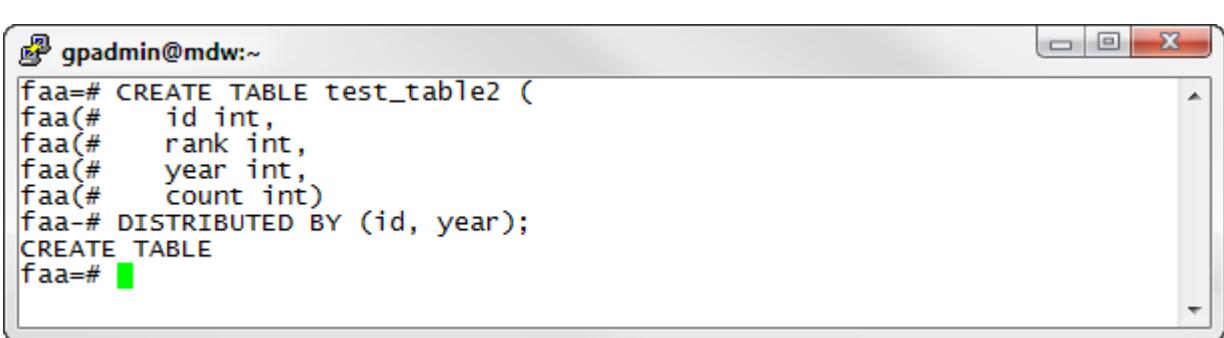
Step	Action
1.	<p>Log in as gpadmin, create a new database called faa using the createdb Greenplum client:</p> <pre>[gpadmin@mdw ~]\$ createdb faa</pre> 
2.	<p>Set the faa database to be the default database you connect to when you start a psql session without specifying the database name:</p> <pre>[gpadmin@mdw ~]\$ export PGDATABASE=faa</pre> 
3.	<p>Verify that the <b>PGDATABASE</b> environment variable is set correctly:</p> <pre>[gpadmin@mdw ~]\$ echo \$PGDATABASE</pre> <p>It should now be set to faa.</p> 
4.	<p>Start a psql session and connect to the faa database. If the PGDATABASE environment variable is set correctly, you do not need to supply the database name:</p> <pre>[gpadmin@mdw ~]\$ psql</pre> 

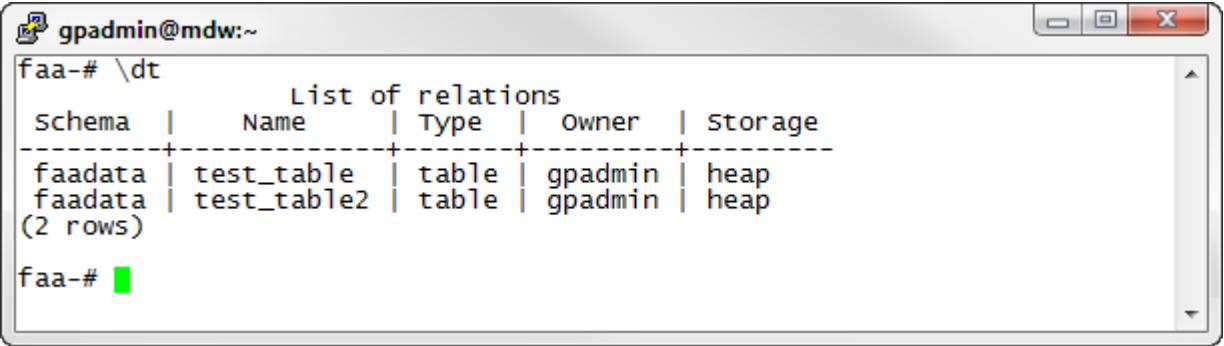
Step	Action
5.	<p>At the psql prompt, list all the databases in the system with the \l meta-command.</p> <pre data-bbox="306 219 453 249">faa=# \l</pre>  <p>Which ones do you see? Is the faa database there? _____</p>
6.	<p>At the psql prompt, create a schema named faadata using the CREATE SCHEMA SQL command:</p> <pre data-bbox="306 1030 812 1060">faa=# create schema faadata;</pre> 
7.	<p>Change the search path on the database, faa so that the new faadata schema is the default schema:</p> <pre data-bbox="306 1442 1367 1505">faa=# ALTER DATABASE faa SET search_path TO faadata, public, pg_catalog;</pre> 

Step	Action
8.	<p>The change to the search path will not become visible until the next client connection. Use the \c meta-command to reconnect to the faa database:</p> <pre data-bbox="311 270 523 297">faa=# \c faa</pre> 
9.	<p>Run the current_schema() function to verify that you are indeed in the faadata schema:</p> <pre data-bbox="311 635 833 663">faa=# SELECT current_schema();</pre> 
10.	<p>Examine the value of the search_path configuration parameter to verify it is correct:</p> <pre data-bbox="311 1110 714 1140">faa=# SHOW search_path;</pre> 

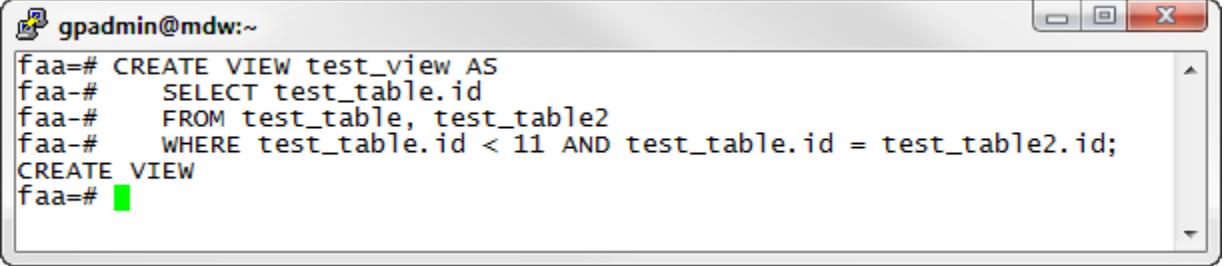
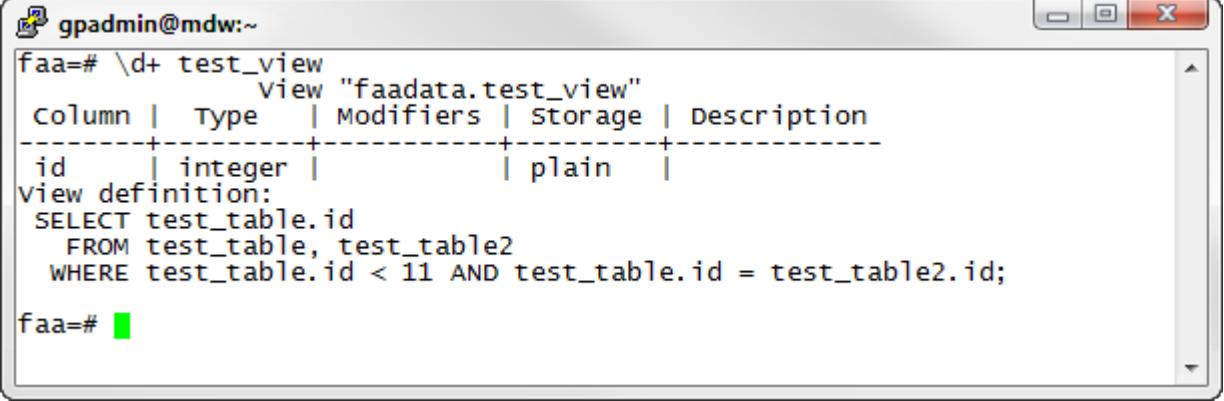
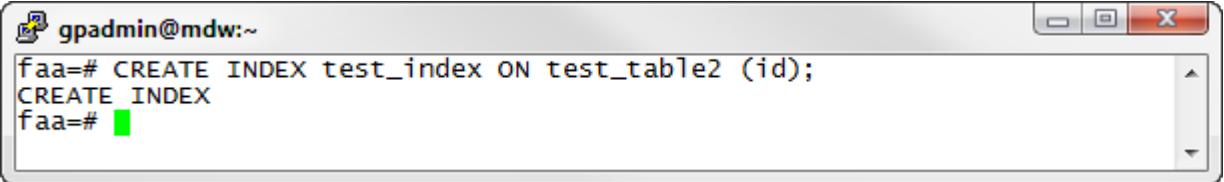
Step	Action
11. Use the <code>\dn</code> meta-command to list the schemas in the database:	<pre data-bbox="306 219 474 249">faa=# \dn</pre>  <p>Which ones do you see? _____</p> <p>Which ones are for system-level objects and which ones are for user-created objects? _____</p>
	<p><b>Summary</b></p> <p>You can create or drop databases using the <code>CREATE DATABASE</code> or <code>DROP DATABASE</code> commands or the <code>createdb</code> or <code>dropdb</code> client programs respectively.</p> <p>By default, every newly created database has a schema named <code>public</code>, which is where objects are created by default.</p> <p>If you create your own schemas and do not want to use qualified names all the time, you should set the <code>search_path</code> parameter to ensure your schema is first in the search path. You can use the <code>ALTER DATABASE</code> command to set the schema search path for the database.</p> <p>You can also use the <code>ALTER ROLE</code> command to set the schema search path for a particular role.</p>

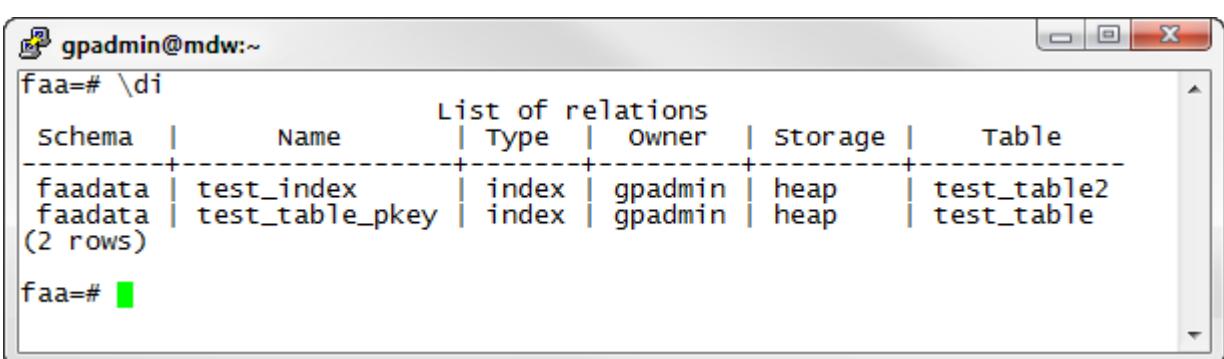
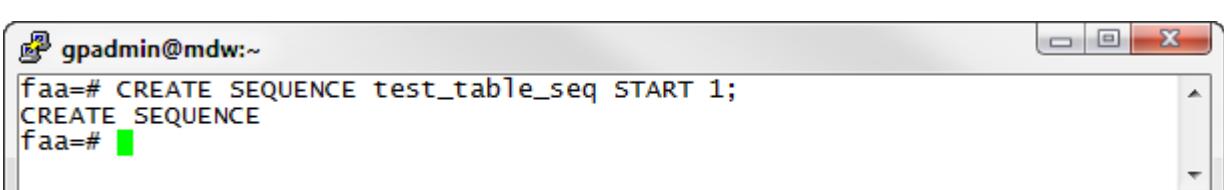
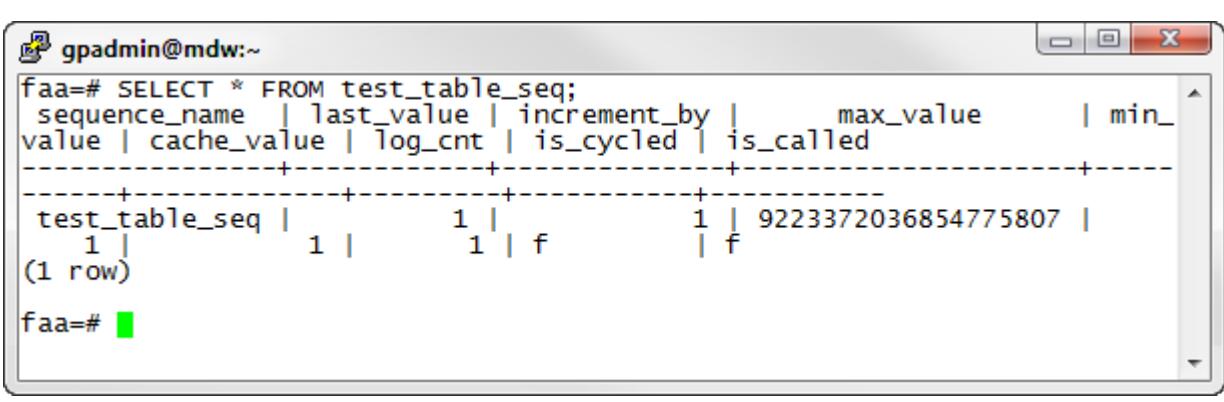
## Lab 7: Task 2 – Create Tables

Step	Action
1.	<p>You will create tables in the faa Greenplum database and learn how the Greenplum distribution key for a table is chosen.</p> <p>For the purpose of this lab exercises, the faa database will have two tables:</p> <ul style="list-style-type: none"> <li>• The test_table table will act as a fact data.</li> <li>• The test_table2 table will act as a dimension table.</li> </ul> <p>At the psql prompt, create a new table with the following definition:</p> <pre>faa=# CREATE TABLE test_table (     id int PRIMARY KEY,     name varchar(30) NOT NULL,     origin varchar(30) NOT NULL,     meaning text );</pre>  <p>Note that the table has a PRIMARY KEY and there is no explicit DISTRIBUTED BY clause given. The primary key is automatically chosen as the distribution key in this case.</p>
2.	<p>Create a table with the following definition and explicitly declare a distribution key using a DISTRIBUTED BY clause:</p> <pre>faa=# CREATE TABLE test_table2 (     id int,     rank int,     year int,     count int) DISTRIBUTED BY (id, year);</pre> 

Step	Action
3.	<p>Why is this distribution key a good choice for this table?</p> <hr/> <p>What would be the distribution key if you left out the DISTRIBUTED BY clause?</p> <hr/> <p>Would that be a good distribution key for this table? Is that a unique key?</p> <hr/>
4.	<p>Use the \dt meta-command to list the tables in the database:</p> <pre>faa=# \dt</pre>  <p>Which ones do you see? Are the test_table and test_table2 tables there?</p>
5.	<p>Exit psql:</p> <pre>faa=# \q</pre>
	<p><b>Summary</b></p> <p>When creating the table, there is an additional clause to declare the Greenplum distribution key column(s).</p> <p>If a DISTRIBUTED BY clause is not supplied, then either the PRIMARY KEY, if the table has one, or the first column of the table will be used. This may or may not be the desirable distribution key.</p> <p>To ensure an even distribution of data, choose a distribution key that is unique for each record. If that is not possible, the column(s) with the lowest selectivity can also be chosen. If a good choice of distribution columns is not available, choose DISTRIBUTED RANDOMLY as the distribution key.</p>

## Lab 7: Task 3 – View Indexes and Schemas

Step	Action
1.	<p>Start a psql session:  <code>[gpadmin@mdw ~] \$ psql</code></p> <p>Verify you are in the faa database. If not, issue the following meta-command: <code>\c faa</code>.</p>
2.	<p>At the psql prompt, create a new view on the test tables:</p> <pre>faa=# CREATE VIEW test_view AS     SELECT test_table.id     FROM test_table, test_table2     WHERE test_table.id &lt; 11 AND test_table.id = test_table2.id;</pre> 
3.	<p>Confirm the view definition is correct:</p> <pre>faa=# \d+ test_view</pre> 
4.	<p>Create an index on the <code>id</code> column of the <code>test_table2</code> table:</p> <pre>faa=# CREATE INDEX test_index ON test_table2 (id);</pre> 

Step	Action
5.	<p>Confirm the index was created:</p> <pre>faa=# \di</pre>  <pre>gpadmin@mdw:~ faa=# \di       List of relations  Schema       Name        Type   Owner   Storage     Table -----+---------------+-----+-----+-----+-----+  faadata   test_index    index   gpadmin   heap      test_table2  faadata   test_table_pkey   index   gpadmin   heap      test_table (2 rows)  faa=# </pre>
6.	<p>Create a sequence table that you can use to assign unique ids when inserting new records into the test_table table. The id value in the test_table table is currently 1, so the sequence starts at 1:</p> <pre>faa=# CREATE SEQUENCE test_table_seq START 1;</pre>  <pre>gpadmin@mdw:~ faa=# CREATE SEQUENCE test_table_seq START 1; CREATE SEQUENCE faa=# </pre>
7.	<p>Examine the sequence table you just created:</p> <pre>faa=# SELECT * FROM test_table_seq;</pre>  <pre>gpadmin@mdw:~ faa=# SELECT * FROM test_table_seq;       sequence_name   last_value   increment_by   max_value   min_  value   cache_value   log_cnt   is_cycled   is_called -----+-----+-----+-----+-----+-----+-----+-----+  test_table_seq         1           1          f          1   9223372036854775807    (1 row)  faa=# </pre>
8.	<p>Exit psql:</p> <pre>faa=# \q</pre>

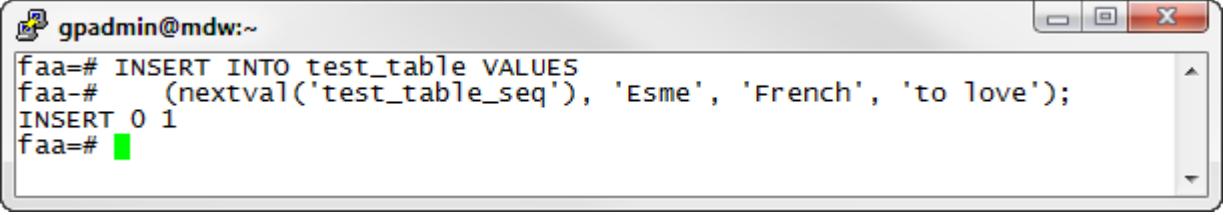
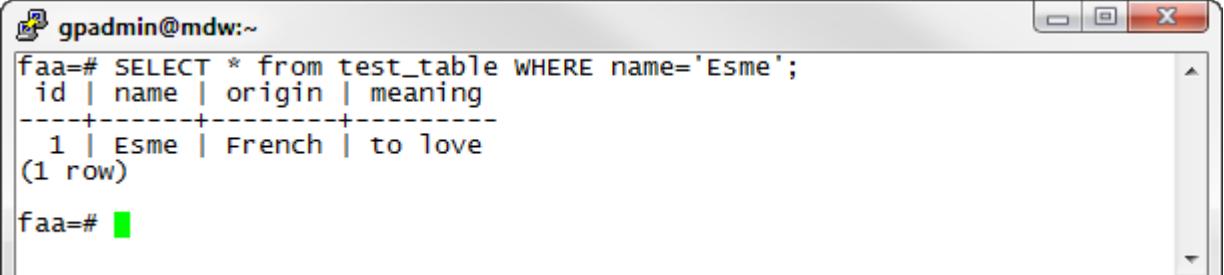
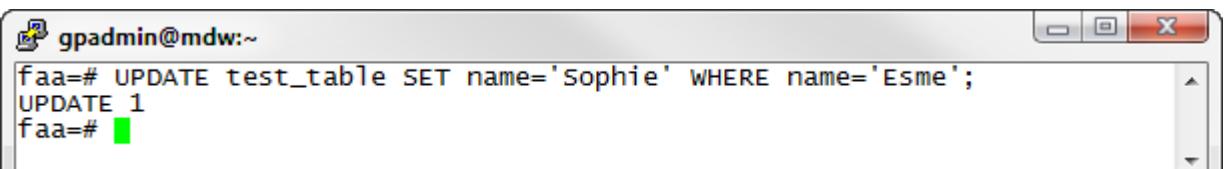
Step	Action
	<p><b>Summary</b></p> <p>Views allow you to store frequently used queries and then access them in SELECT statements as if they were a regular table. You created a view that selects the top ten baby names from your database.</p> <p>Indexes are not always the performance enhancer in the Greenplum Database as they are in traditional database management systems. In some cases indexes can improve query performance, and in some cases indexes have no effect or can slightly degrade performance. You have added an index on the rank table id column, since that column will be used often in your queries to join with the names table. When you get to query profiling, you will determine if this index is indeed being utilized.</p> <p>Sequences are used to generate numbers, helpful for incrementing unique id columns such as the id column of your names table. You can use the sequence when you insert new names into this table to generate a unique id number that won't conflict with the other id numbers already being used.</p>

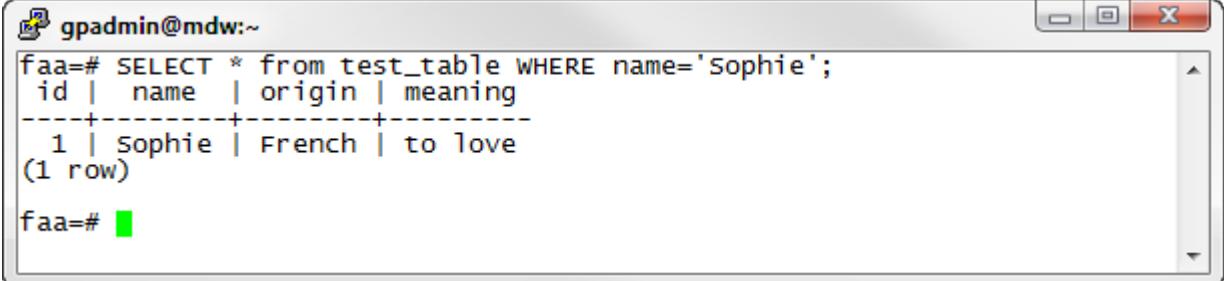
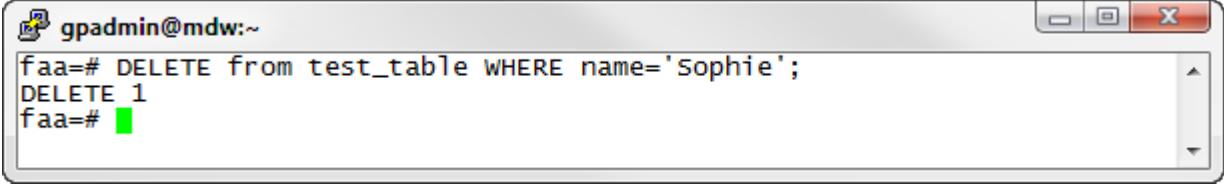
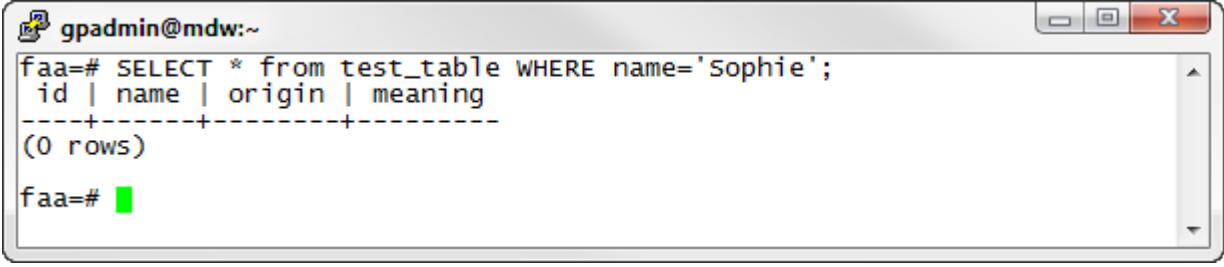
*End of Lab Exercise*

## Lab 8. Data Manipulation Language and Data Query

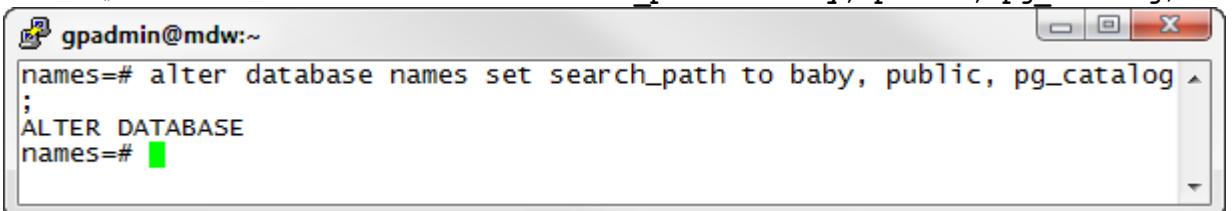
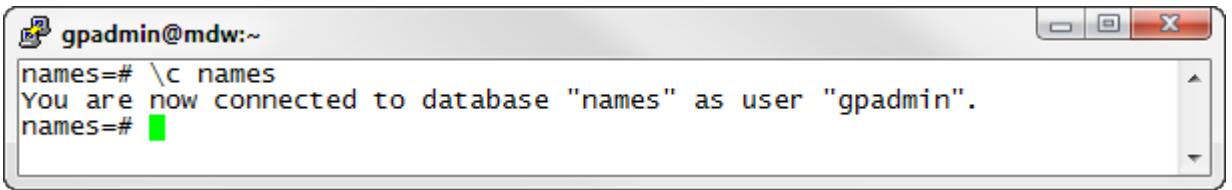
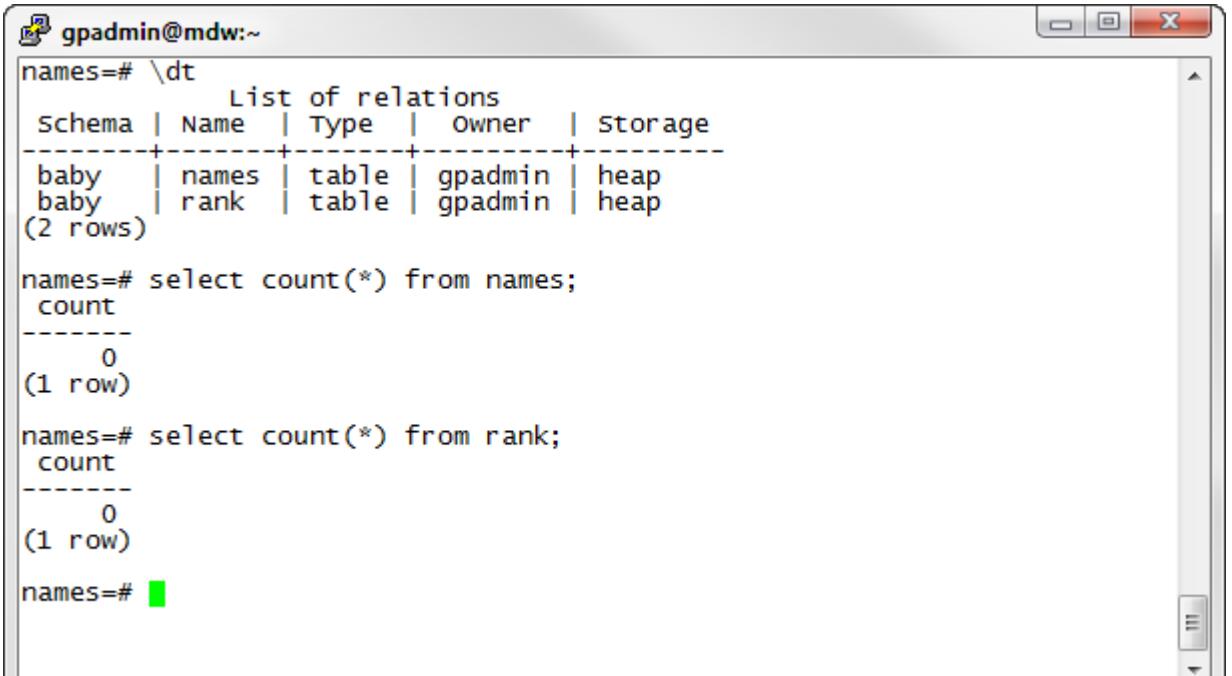
<b>Purpose:</b>	In this lab, you will familiarize yourself with the INSERT, UPDATE, and DELETE SQL commands.
<b>Tasks:</b>	Students perform the following task: Insert, update, and delete records.
<b>References:</b>	<p>Module 4 – Defining and Securing the User Database</p> <ul style="list-style-type: none"><li>• Lesson: Data Manipulation Language and Data Query Language</li></ul>

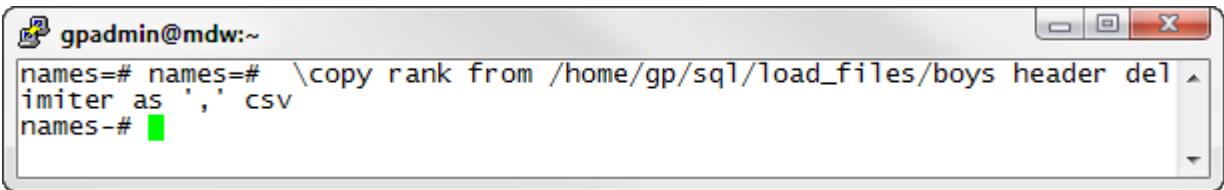
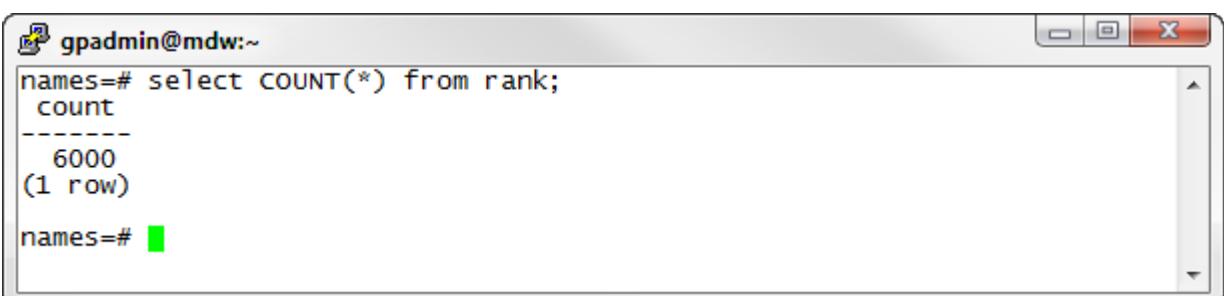
## Lab 8: Task 1 – Insert, Update, and Delete Records

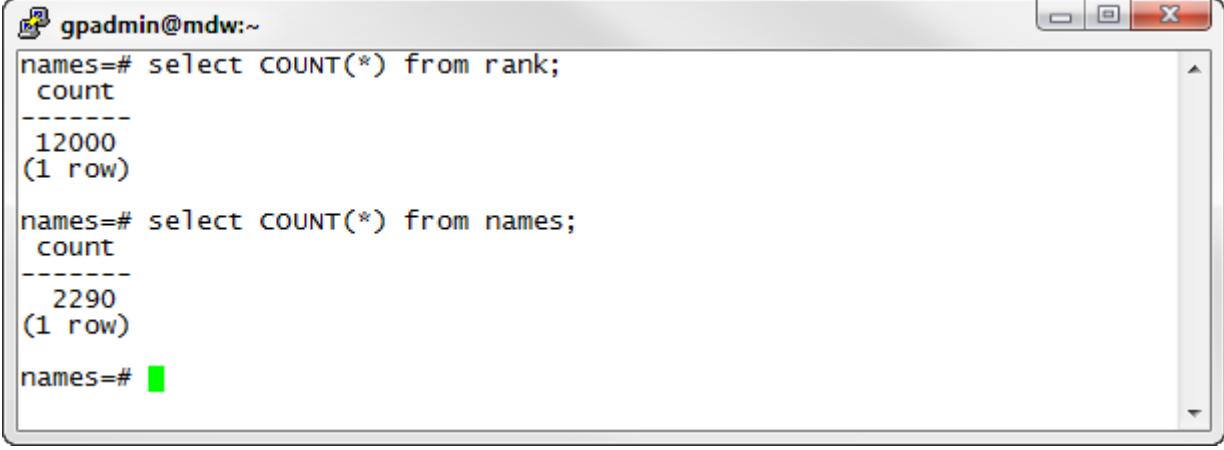
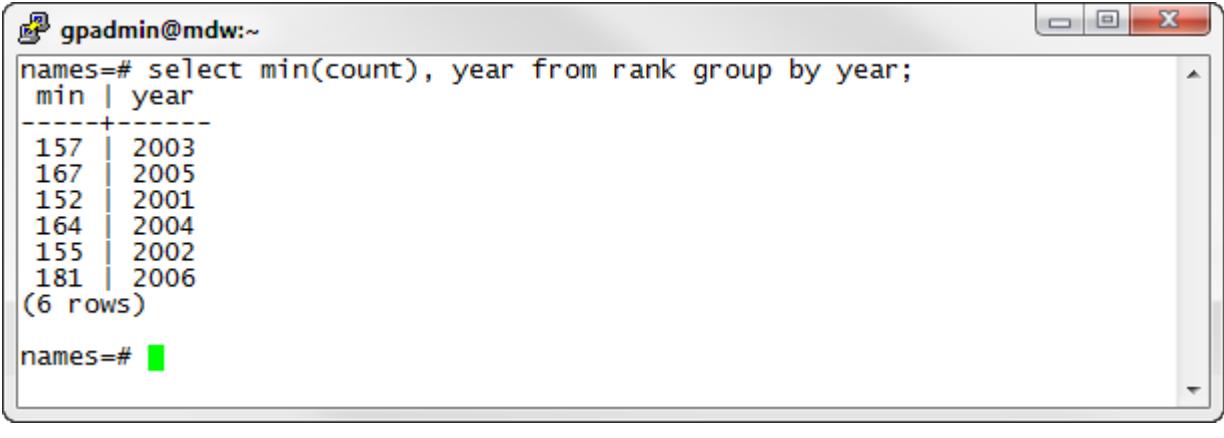
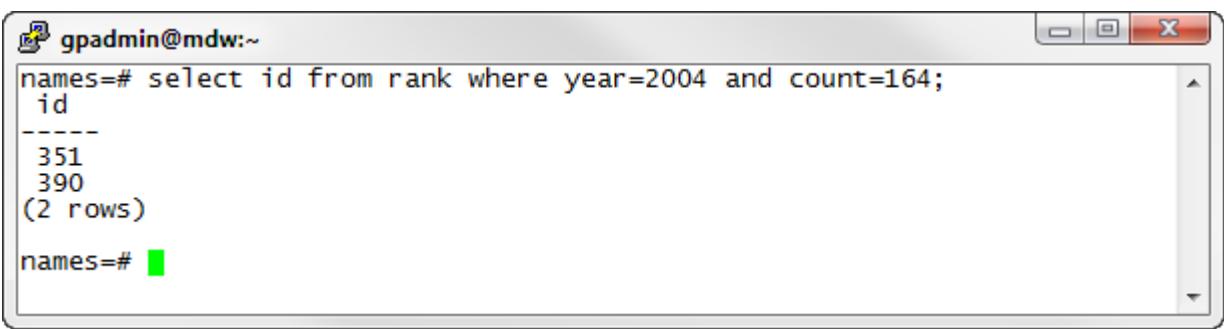
Step	Action
1.	<p>Connect to the faa database as the gpadmin user, if not already connected:</p> <pre>[gpadmin@mdw ~]\$ psql faa</pre>  <p>You can also specify the database using the <code>-d</code> option followed by the database name.</p>
2.	<p>At the psql prompt, add a new record to the test_table table.</p> <pre>faa=# INSERT INTO test_table VALUES       (nextval('test_table_seq'), 'Esme', 'French', 'to love');</pre> 
3.	<p>Check to see that the record you created is there:</p> <pre>faa=# SELECT * from test_table WHERE name='Esme';</pre> 
4.	<p>Update the record to change the name column to Sophie for any instances where the name column is Esme:</p> <pre>faa=# UPDATE test_table SET name='Sophie' WHERE name='Esme';</pre> 

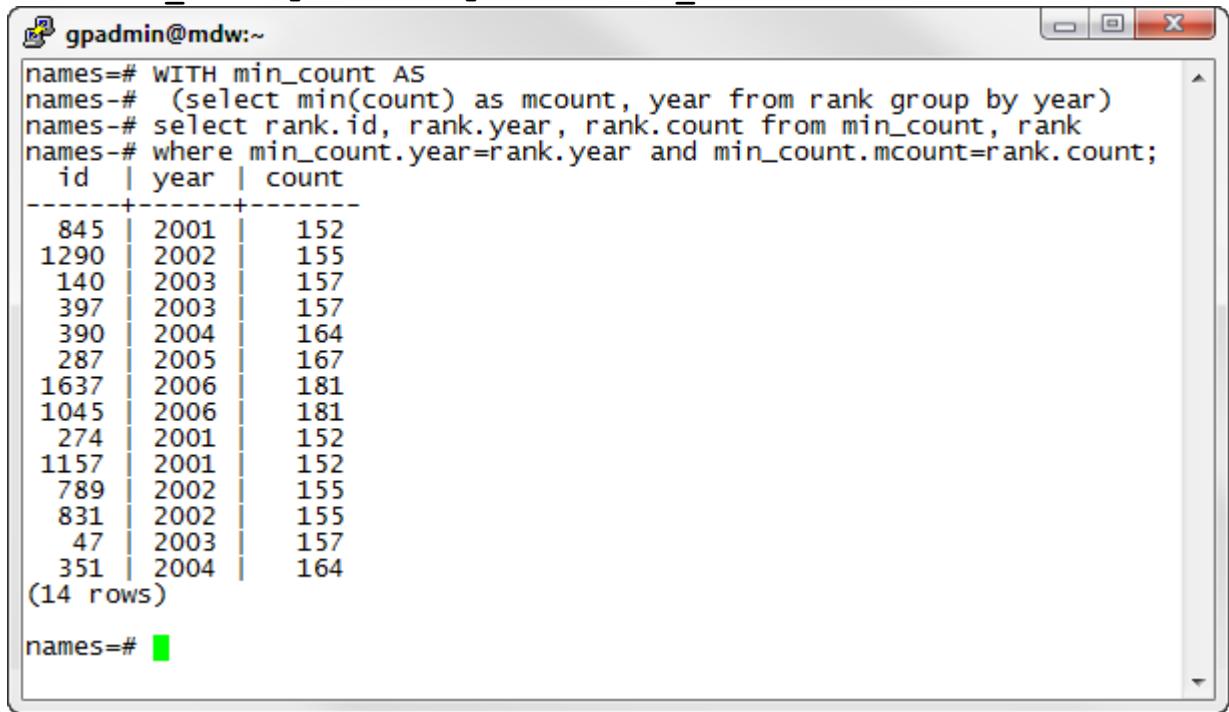
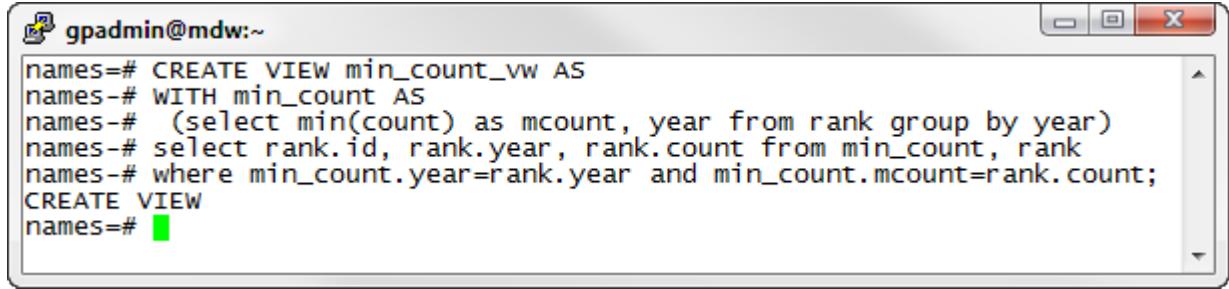
Step	Action
5.	<p>Check to see that the record was changed:</p> <pre data-bbox="311 219 1225 255">faa=# SELECT * from test_table WHERE name='Sophie';</pre>  <pre data-bbox="311 494 437 523">faa=#</pre>
6.	<p>Delete the record. Do not forget the WHERE clause or you will delete all of the rows:</p> <pre data-bbox="311 692 1192 728">faa=# DELETE from test_table WHERE name='Sophie';</pre>  <pre data-bbox="311 874 437 903">faa=#</pre>
7.	<p>Check to see that the record was deleted:</p> <pre data-bbox="311 1058 1225 1094">faa=# SELECT * from test_table WHERE name='Sophie';</pre>  <pre data-bbox="311 1311 437 1341">faa=#</pre> <p>Exit your psql session.</p> <pre data-bbox="311 1474 458 1510">faa=# \q</pre>

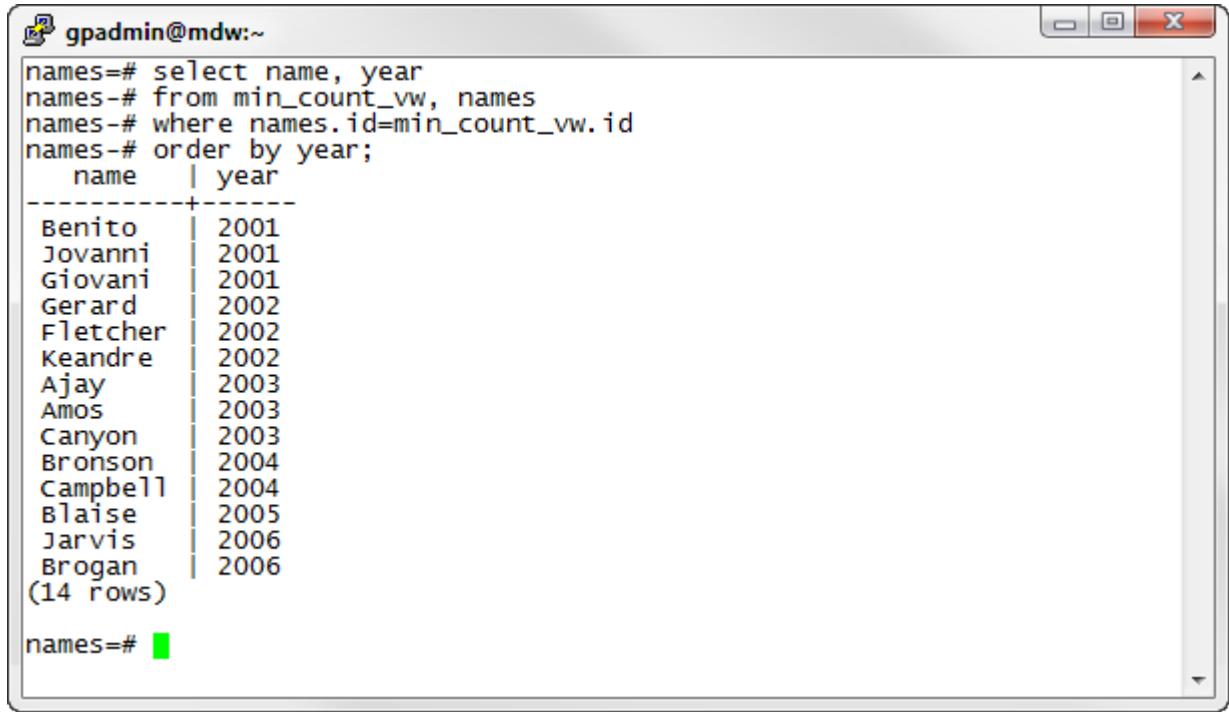
## Lab 8: Task 2 – Accessing Data to Generate Reports

Step	Action
	In this task, you use a variety of statements to generate a report. The report identifies the least popular baby names, across boys and girls, from data collected.
1.	Connect to the names database as the gpadmin user, if not already connected: <pre>[gpadmin@mdw ~]\$ psql names</pre>
2.	Update the search_path for the names database so that you can easily access the tables from the names database without needing to specify the schema name. <pre>names=# alter database names set search_path to baby, public, pg_catalog;</pre> 
3.	Reconnect to the database to access the settings. <pre>names=# \c names</pre> 
4.	Identify the tables in this database and use COUNT to verify the tables have no rows. <pre>names=# \dt names=# select count(*) from names; names=# select count(*) from rank;</pre> 

Step	Action						
5.	<p>Using the psql copy command, copy data from local files into the rank table. The data will be copied from the CSV file, /home/gp/sql/load_files/boys, and includes a header line. This command behaves like the SQL COPY command, which will be covered later in the course. For now, you are populating tables with values to be manipulated and accessed.</p> <pre>names=# \copy rank from /home/gp/sql/load_files/boys header delimiter as ',' csv</pre> 						
6.	<p>Using COUNT, verify data has now been loaded into the rank table.</p> <pre>names=# select COUNT(*) from rank;</pre> 						
7.	<p>Using the syntax shown earlier, load data from the files noted to the correlating tables. Use the up arrow (<math>\uparrow</math>) to scroll through your PSQL buffer until you find the \copy command and change the line as needed. Use the left (<math>\leftarrow</math>) and right (<math>\rightarrow</math>) arrows to move through the command to the required position.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="background-color: #cccccc; width: 50%;">File</th><th style="background-color: #cccccc; width: 50%;">Table</th></tr> </thead> <tbody> <tr> <td>/home/gp/sql/load_files/girls</td><td>rank</td></tr> <tr> <td>/home/gp/sql/load_files/name_ids</td><td>names</td></tr> </tbody> </table> <pre>names=# \copy rank from /home/gp/sql/load_files/girls header delimiter as ',' csv names=# \copy names from /home/gp/sql/load_files/name_ids header delimiter as ',' csv</pre> 	File	Table	/home/gp/sql/load_files/girls	rank	/home/gp/sql/load_files/name_ids	names
File	Table						
/home/gp/sql/load_files/girls	rank						
/home/gp/sql/load_files/name_ids	names						

Step	Action
8.	<p>Use COUNT to verify that the rows have been loaded into the respective tables.</p> <pre>names=# select COUNT(*) from rank; names=# select COUNT(*) from names;</pre> 
9.	<p>The rank table highlights the number of occurrences of babies given a specific name. Each row indicates the ID for the name, the name's ranking or weight, the year the information was collected, and the number of babies given the specific name. The names table associates the ID with the name and provides the origin and local meaning of the name. While the rank can be used to determine the least popular name as defined by the rank, this task will have you use the aggregate functions against the count associated with the names.</p> <p>Use the MIN function to find the lowest count of a name for each year. Remember, MIN is an aggregate function and requires the use of GROUP BY.</p> <pre>names=# select min(count), year from rank group by year;</pre> 
10.	<p>Examine the rank table to list the ID(s) that had the lowest count for the year 2004. Use the information returned from the previous example to complete the request.</p> <pre>names=# select id from rank where year=2004 and count=164;</pre> 

Step	Action																																													
11.	<p>This two-step process can easily be changed to a single step and would allow us to collect the information for all years involved. Use the WITH clause to define a common table expression that allows you to return the ID, year, and count for the lowest assigned name for each year.</p> <pre>names=# WITH min_count AS   (select min(count) as mcount, year from rank group by year) select rank.id, rank.year, rank.count from min_count, rank where min_count.year=rank.year and min_count.mcount=rank.count;</pre>  <table border="1"> <thead> <tr> <th>id</th> <th>year</th> <th>count</th> </tr> </thead> <tbody> <tr><td>845</td><td>2001</td><td>152</td></tr> <tr><td>1290</td><td>2002</td><td>155</td></tr> <tr><td>140</td><td>2003</td><td>157</td></tr> <tr><td>397</td><td>2003</td><td>157</td></tr> <tr><td>390</td><td>2004</td><td>164</td></tr> <tr><td>287</td><td>2005</td><td>167</td></tr> <tr><td>1637</td><td>2006</td><td>181</td></tr> <tr><td>1045</td><td>2006</td><td>181</td></tr> <tr><td>274</td><td>2001</td><td>152</td></tr> <tr><td>1157</td><td>2001</td><td>152</td></tr> <tr><td>789</td><td>2002</td><td>155</td></tr> <tr><td>831</td><td>2002</td><td>155</td></tr> <tr><td>47</td><td>2003</td><td>157</td></tr> <tr><td>351</td><td>2004</td><td>164</td></tr> </tbody> </table> <p>(14 rows)</p> <pre>names=#</pre>	id	year	count	845	2001	152	1290	2002	155	140	2003	157	397	2003	157	390	2004	164	287	2005	167	1637	2006	181	1045	2006	181	274	2001	152	1157	2001	152	789	2002	155	831	2002	155	47	2003	157	351	2004	164
id	year	count																																												
845	2001	152																																												
1290	2002	155																																												
140	2003	157																																												
397	2003	157																																												
390	2004	164																																												
287	2005	167																																												
1637	2006	181																																												
1045	2006	181																																												
274	2001	152																																												
1157	2001	152																																												
789	2002	155																																												
831	2002	155																																												
47	2003	157																																												
351	2004	164																																												
12.	<p>If this will be a frequently accessed statement, it would be best to save it as a view. Create a view called min_count_vw that contains the common table expression you defined in the previous step.</p> <pre>names=# CREATE VIEW min_count_vw AS WITH min_count AS   (select min(count) as mcount, year from rank group by year) select rank.id, rank.year, rank.count from min_count, rank where min_count.year=rank.year and min_count.mcount=rank.count;</pre>  <pre>names=# CREATE VIEW min_count_vw AS names-# WITH min_count AS names-#   (select min(count) as mcount, year from rank group by year) names-# select rank.id, rank.year, rank.count from min_count, rank names-# where min_count.year=rank.year and min_count.mcount=rank.count; CREATE VIEW names=# </pre>																																													

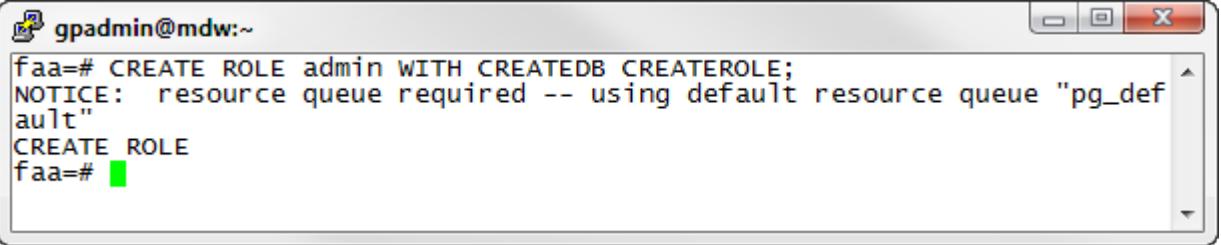
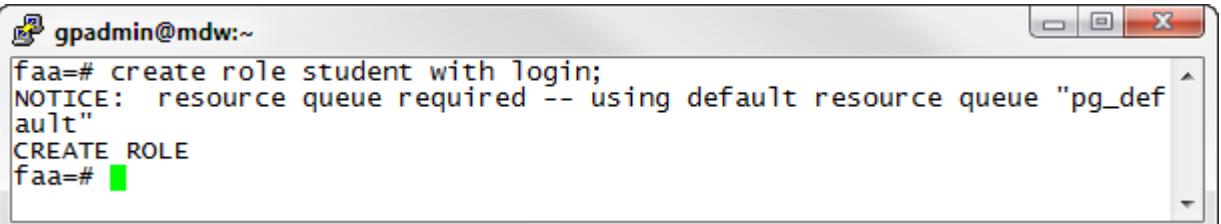
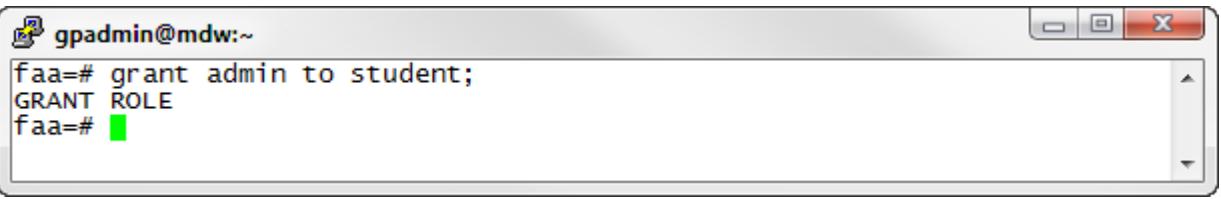
Step	Action
13.	<p>Using the view you created, list the least assigned baby names for the years collected. Make the report easier to read by sorting the results by year.</p> <pre>names=# select name, year from min_count_vw, names where names.id=min_count_vw.id order by year;</pre>  <pre>gpadmin@mdw:~ names=# select name, year names-# from min_count_vw, names names-# where names.id=min_count_vw.id names-# order by year;       name     year -----+-----     Benito   2001   Jovanni   2001   Giovani   2001    Gerard   2002  Fletcher   2002  Keandre   2002     Ajay     2003    Amos     2003   Canyon    2003  Bronson    2004 Campbell1   2004   Blaise    2005    Jarvis   2006   Brogan    2006 (14 rows)  names=# </pre>
	<p><b>Summary</b></p> <p>If you have many rows to insert, consider using COPY or external tables instead of INSERT.</p> <p>UPDATE is used to change one or more column values of an existing row. DELETE is used to delete rows from a table based on some qualifying condition.</p> <p>An unqualified delete on a table will delete all rows. A more efficient way to delete all rows from a table is the TRUNCATE command.</p> <p>A variety of tools are available to help generate information for reports. A portion of this lab exercise focused on using some aggregate functions to collect the information that was needed. Views and common table expressions let you simplify more complex statements and make these statements available for future use.</p>

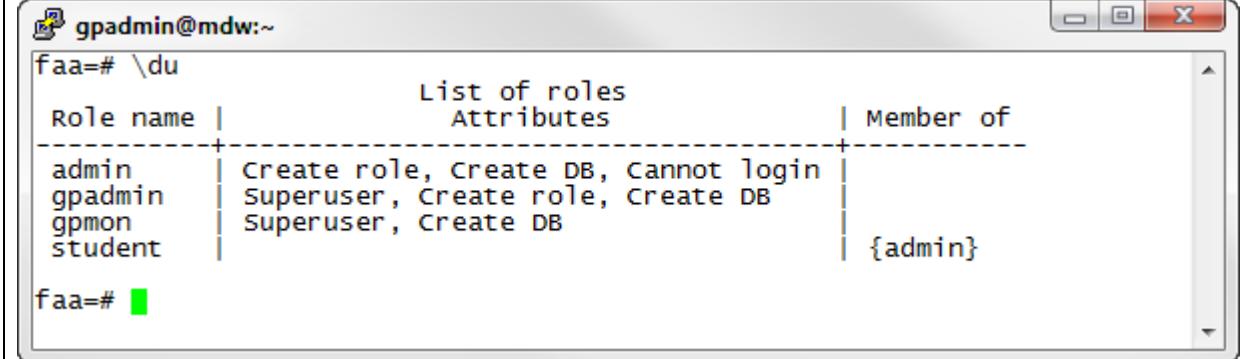
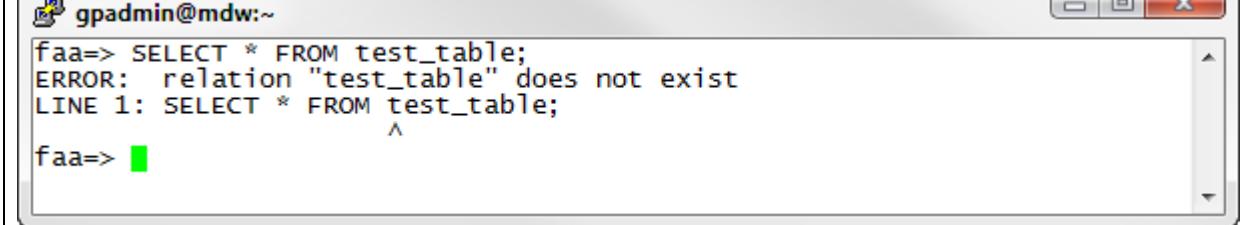
End of Lab Exercise

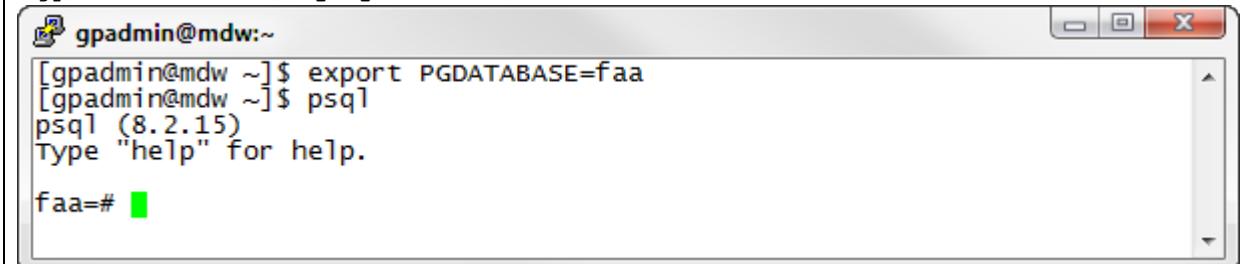
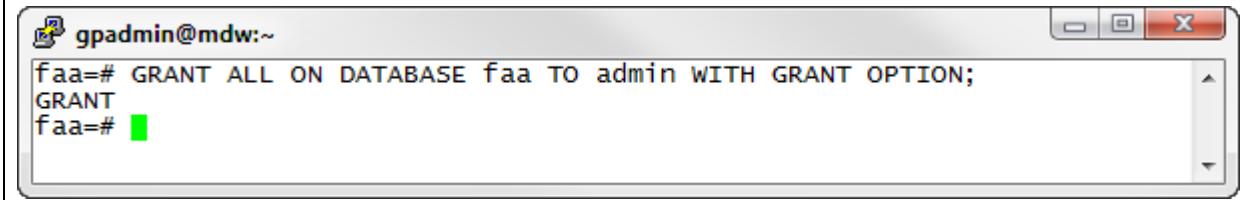
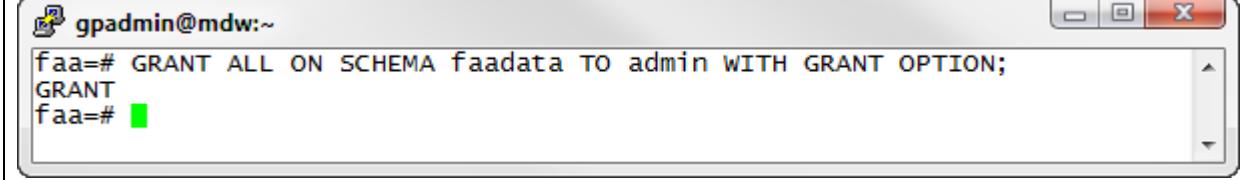
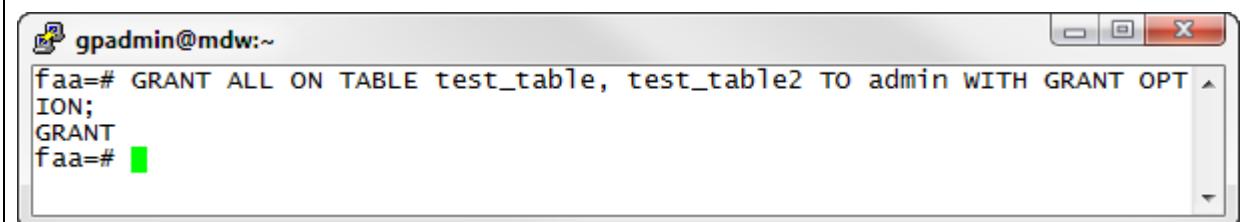
## Lab 9. Roles, Privileges, and Resources

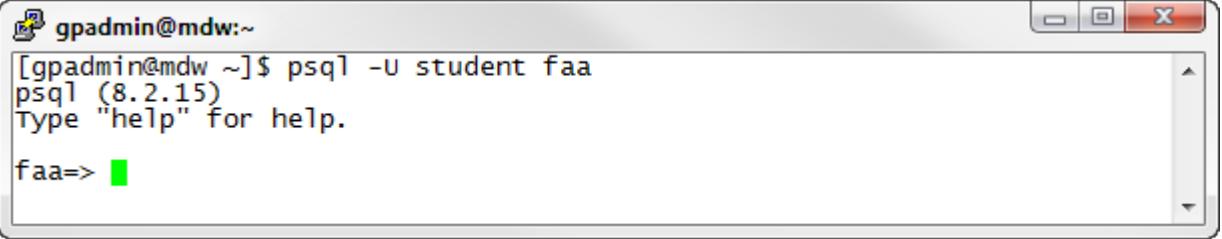
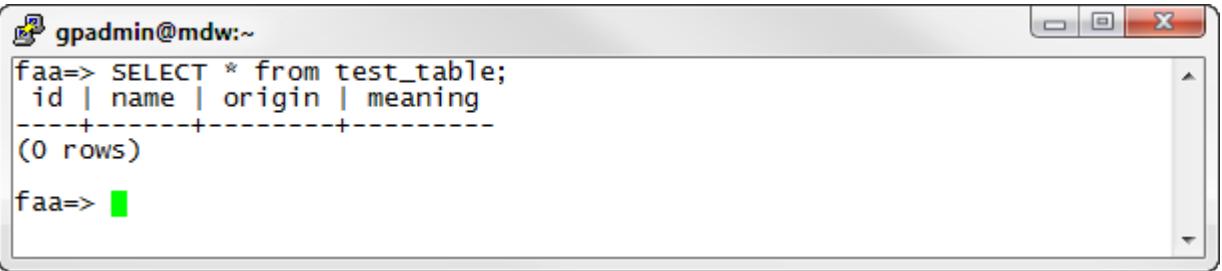
<b>Purpose:</b>	<p>In this lab you will learn how to create roles that are users and roles that are groups.</p> <p>User-level roles can log in to a database. Group-level roles are useful for managing permissions and access privileges to database objects.</p> <p>You will learn how to grant privileges on database objects to a group-level role. Members, or user-level roles, of the group inherit those access privileges.</p> <p>You will create a resource queue and assign a user role to this resource queue. The user name you use is the name you added in the pg_hba.conf file, your first name. For testing purposes, the resource queue you create will have an ACTIVE_THRESHOLD of 1, meaning that only one active statement submitted through the resource queue will be allowed to run at any given time.</p>
<b>Tasks:</b>	<p>Students perform the following tasks:</p> <ul style="list-style-type: none"><li>• Manage user and group roles</li><li>• Manage resource queues</li></ul>
<b>References:</b>	<p>Module 4 – Defining and Securing the User Database</p> <ul style="list-style-type: none"><li>• Lesson: Roles, Privileges, and Resources</li></ul>

## Lab 9: Task 1 – Manage User and Group Roles

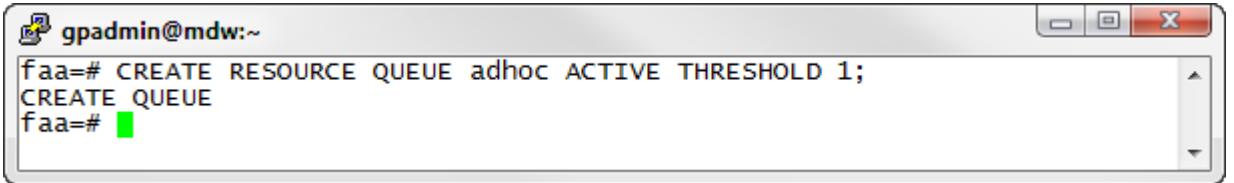
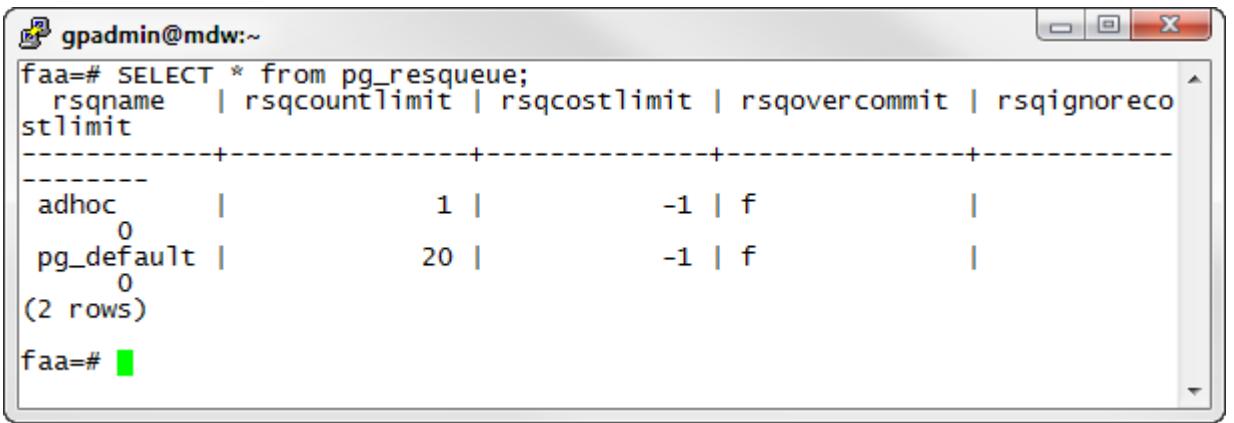
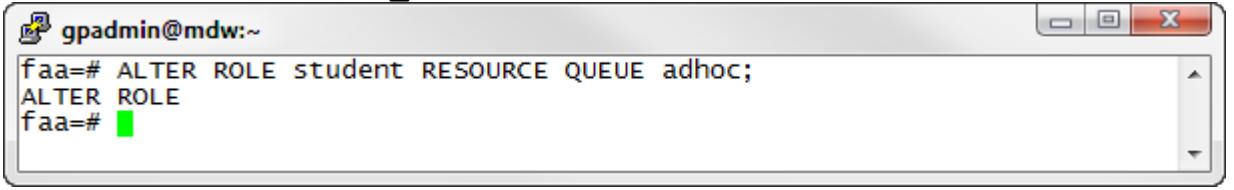
Step	Action
1.	<p>Start a psql session: [gpadmin@mdw ~]\$ <b>psql</b></p> <p>If you are not in the <code>faa</code> database, issue the following meta-command to connect to the database: <code>\c faa</code>.</p>
2.	<p>Create a role named <code>admin</code> and give it appropriate permission attributes for an administrator group-level role using the following command</p> <p><code>faa=# CREATE ROLE admin WITH CREATEDB CREATEROLE;</code></p> 
3.	<p>Create a user login role for yourself and give it appropriate permission attributes:</p> <p><code>faa=# CREATE ROLE &lt;your_name&gt; WITH LOGIN;</code></p> <p>Replace <code>&lt;your_name&gt;</code> with the name you used in Lab 5: Greenplum Database Server Configuration.</p> 
4.	<p>Add the user role you just created as a member of the <code>admin</code> role using the GRANT command:</p> <p><code>faa=# GRANT admin TO &lt;your_name&gt;;</code></p> 

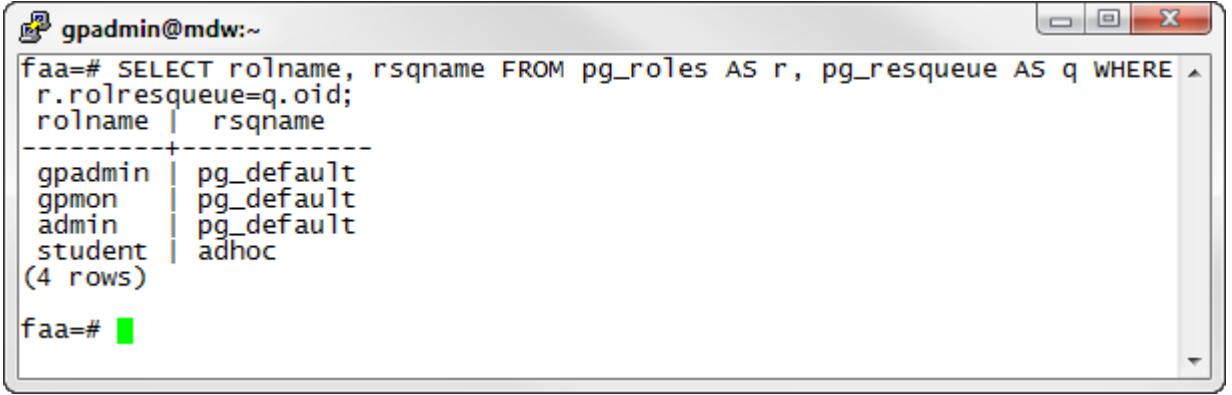
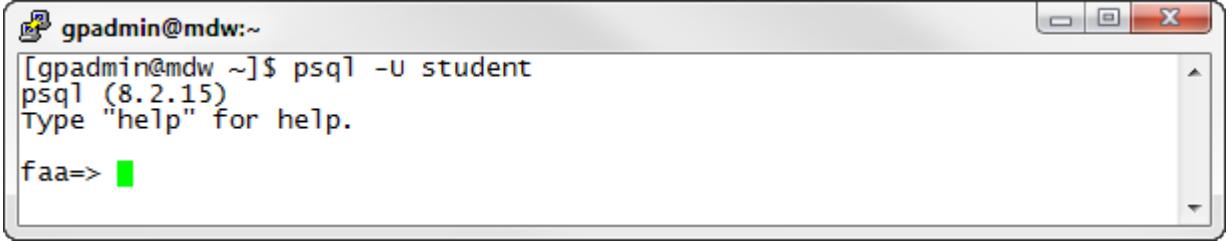
Step	Action
5.	<p>Confirm that your roles are configured as expected by using the \du meta-command in psql:</p> <pre data-bbox="311 185 1553 221">faa=# \du</pre> <p>This shows a list of all roles in the system.</p>  <pre data-bbox="311 633 1553 669">faa=#</pre>
6.	<p>Change to the user role you just created using the SET ROLE command:</p> <pre data-bbox="311 792 1553 827">faa=# SET ROLE TO &lt;your_name&gt;;</pre>  <pre data-bbox="311 960 1553 996">faa=&gt;</pre> <p>Note that your prompt has changed to reflect the level of the user you have connected into the database with.</p>
7.	<p>Run a query as this role. Select all columns from the test_table table:</p> <pre data-bbox="311 1256 1553 1292">faa=&gt; SELECT * from test_table;</pre>  <pre data-bbox="311 1446 1553 1482">faa=&gt;</pre> <p>The command did not succeed. Why not? _____</p>
8.	<p>Exit psql:</p> <pre data-bbox="311 1721 1553 1757">faa=&gt; \q</pre>
9.	<p>Set the PGDATABASE variable to faa.</p> <pre data-bbox="311 1826 1553 1862">[gpadmin@mdw ~]\$ export PGDATABASE=faa</pre>

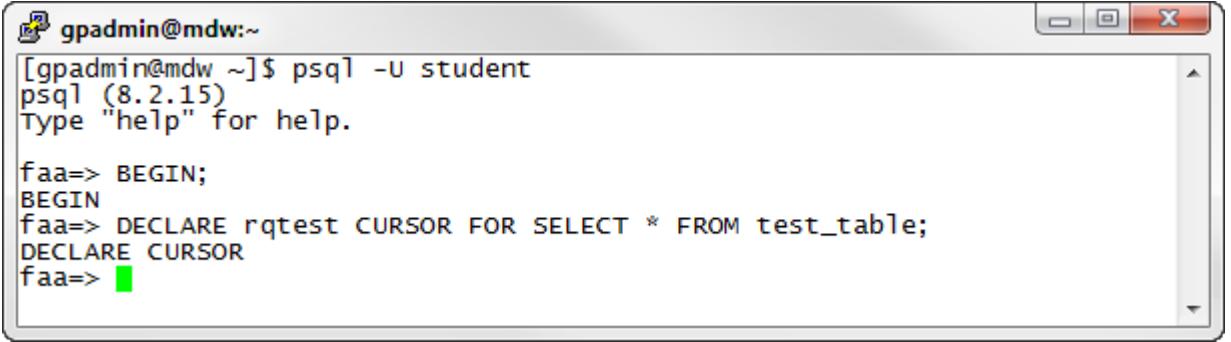
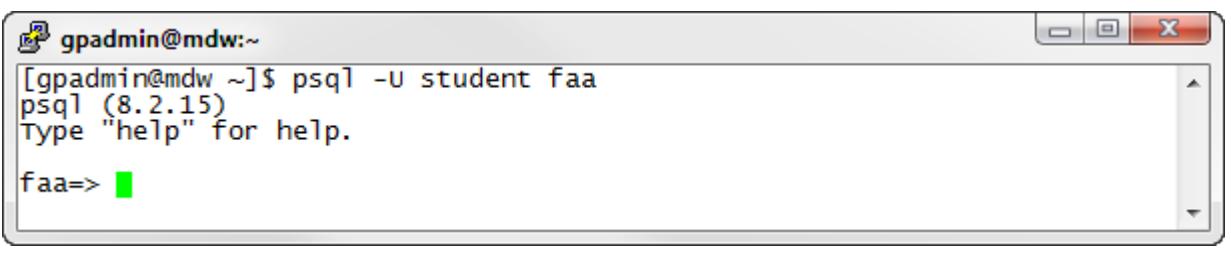
Step	Action
10.	<p>Log in to the database as <b>gpadmin</b>.</p> <pre>[gpadmin@mdw ~]\$ psql gpadmin@mdw:~</pre> 
11.	<p>Grant privileges to the faa database, faadata schema, and test_table and test_table2 tables to the admin role. Grant all privileges on the faa database:</p> <pre>faa=# GRANT ALL ON DATABASE faa TO admin WITH GRANT OPTION;</pre> 
12.	<p>Grant all privileges on the schema, faadata using the GRANT OPTION clause:</p> <pre>faa=# GRANT ALL ON SCHEMA faadata TO admin WITH GRANT OPTION;</pre> 
13.	<p>Grant all privileges to admin on the tables, test_table and test_table2 with the GRANT OPTION clause:</p> <pre>faa=# GRANT ALL ON TABLE test_table, test_table2 TO admin WITH GRANT OPTION;</pre> 
14.	<p>Exit psql:</p> <pre>faa=# \q</pre>

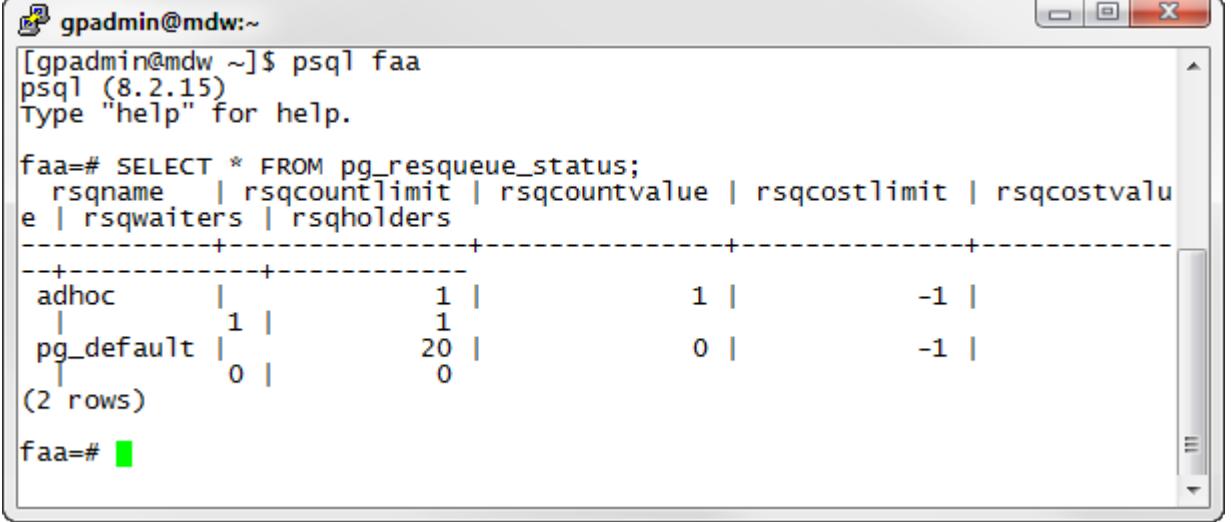
Step	Action
15.	<p>Start a psql session as the user role you created earlier and connect to the faa database:</p> <pre data-bbox="331 228 964 259">[gpadmin@mdw ~]\$ psql -U student faa</pre> 
16.	<p>Run the SELECT query you ran earlier to test access to the tables. You should now have permissions to see this view (and its schema):</p> <pre data-bbox="331 724 871 756">faa=&gt; SELECT * from test_table;</pre> 
17.	<p>Exit psql:</p> <pre data-bbox="331 1178 463 1210">faa=&gt; \q</pre>
	<p><b>Summary</b></p> <p>Greenplum manages database access permissions using the concept of roles. The concept of roles subsumes the concepts of users and groups. A role can be a database user, a group, or both. Roles can own database objects (for example, tables) and can assign privileges on those objects to other roles to control access to the objects. Roles can be members of other roles, thus a member role can inherit the attributes and privileges of its parent role.</p> <p>Note that if you are using table inheritance and partitioning, table privileges are not passed down from parent tables to child tables. You must explicitly set privileges on each child table. If you use the Greenplum management utilities to partition a table, the utility takes care of passing down the table permissions for you.</p> <p>When you create a new login role, you must be sure that the <code>pg_hba.conf</code> configuration file of the master allows the role to connect to the Greenplum Database. Otherwise the role will be denied access.</p>

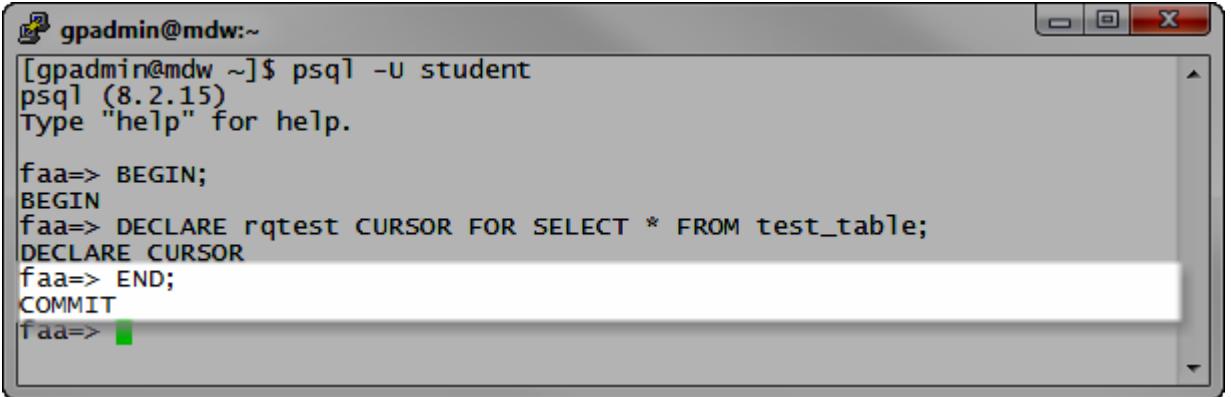
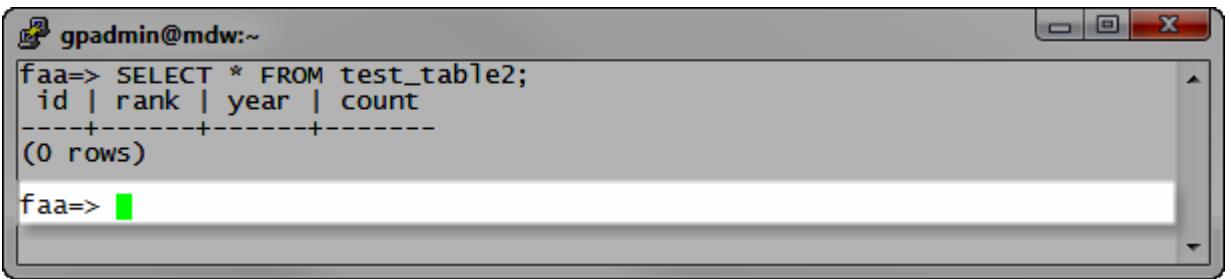
## Lab 9: Task 2 – Manage Resource Queues

Step	Action
1.	<p>At the <code>psql</code> prompt, create a resource queue named <code>adhoc</code> and give it an ACTIVE THRESHOLD limit of 1. Connect to the <code>faa</code> database as <code>gpadmin</code>:</p> <pre>[gpadmin@mdw ~]\$ psql</pre>
2.	<p>Create the resource queue, <code>adhoc</code>, with ACTIVE THRESHOLD equal to 1:</p> <pre>faa=# CREATE RESOURCE QUEUE adhoc ACTIVE THRESHOLD 1;</pre> 
3.	<p>Confirm the resource queue was created correctly by checking the <code>pg_resqueue</code> system table:</p> <pre>faa=# SELECT * from pg_resqueue;</pre>  <p>Note: The <code>pg_default</code> resource queue is the default queue to which all roles are assigned if the resource queue is not specified when the role is created.</p>
4.	<p>Assign the <code>&lt;your_name&gt;</code> user-level role that you created to the <code>adhoc</code> resource queue:</p> <pre>faa=# ALTER ROLE &lt;your_name&gt; RESOURCE QUEUE adhoc;</pre> 

Step	Action
5.	<p>Confirm that the role was assigned the adhoc resource queue by querying the pg_resqueue and pg_roles system tables:</p> <pre data-bbox="313 270 1481 333">faa=# SELECT rolname, rsqname FROM pg_roles AS r, pg_resqueue AS q WHERE r.rolresqueue=q.oid;</pre>  <pre data-bbox="313 650 442 677">faa=#</pre>
6.	<p>Exit the psql session:</p> <pre data-bbox="313 840 458 868">faa=# \q</pre>
7.	<p>Start a new psql session as the &lt;your_name&gt; user-level role:</p> <pre data-bbox="313 1009 882 1036">[gpadmin@mdw ~]\$ psql -U student</pre>  <pre data-bbox="313 1220 442 1248">faa=&gt;</pre>

Step	Action
8.	<p>Verify that the resource queue is working.</p> <p>To hold a query open, open a cursor within a transaction. This action holds the one active query slot you are allowed in the adhoc resource queue. The cursor will remain open until the transaction is closed, which will give you the time required to test the resource queue's limits:</p> <pre data-bbox="314 375 1318 445">faa=&gt; BEGIN; faa=&gt; DECLARE rqtest CURSOR FOR SELECT * FROM test_table;</pre> 
9.	<p>Leave the current session open, and open a new terminal session to the master server. Connect as root and switch to the gpadmin user.</p> <p>To test the resource queue, you must run a query in another session to see if it is allowed to run while the other session is holding open the resource queue's active query slot.</p> <p>From the second terminal window:</p> <p>Switch users to gpadmin:</p> <pre data-bbox="331 1167 556 1195">su - gpadmin</pre> 
10.	<p>Log in to psql with &lt;your_name&gt;:</p> <pre data-bbox="314 1480 948 1507">[gpadmin@mdw ~]\$ psql -U student faa</pre> 

Step	Action																												
11.	<p>Run a query in the second terminal session:</p> <pre data-bbox="314 217 882 255">faa=&gt; SELECT * FROM test_table2;</pre>  <p>Did the query run? What happened?</p>																												
12.	<p>Check the resource queue status:</p> <p>Open a third terminal session. Log in as <b>gpadmin</b>.</p>																												
13.	<p>Connect to the <b>faa</b> database as <b>gpadmin</b>. Issue the following command to view the state of the resource queues:</p> <pre data-bbox="331 830 1024 868">faa=# SELECT * FROM pg_resqueue_status;</pre>  <table border="1"> <thead> <tr> <th data-bbox="331 1083 470 1142">rsqname</th> <th data-bbox="470 1083 600 1142">rsqcountlimit</th> <th data-bbox="600 1083 731 1142">rsqcountvalue</th> <th data-bbox="731 1083 861 1142">rsqcostlimit</th> <th data-bbox="861 1083 992 1142">rsqcostvalue</th> <th data-bbox="992 1083 1122 1142">rsqwaiters</th> <th data-bbox="1122 1083 1253 1142">rsqholders</th> </tr> </thead> <tbody> <tr> <td data-bbox="331 1184 470 1243">adhoc</td> <td data-bbox="470 1184 600 1243">1</td> <td data-bbox="600 1184 731 1243">1</td> <td data-bbox="731 1184 861 1243">1</td> <td data-bbox="861 1184 992 1243">-1</td> <td data-bbox="992 1184 1122 1243">1</td> <td data-bbox="1122 1184 1253 1243">-1</td> </tr> <tr> <td data-bbox="331 1243 470 1303">pg_default</td> <td data-bbox="470 1243 600 1303">0</td> <td data-bbox="600 1243 731 1303">20</td> <td data-bbox="731 1243 861 1303">0</td> <td data-bbox="861 1243 992 1303">-1</td> <td data-bbox="992 1243 1122 1303">0</td> <td data-bbox="1122 1243 1253 1303">-1</td> </tr> <tr> <td data-bbox="331 1303 470 1341">(2 rows)</td> <td data-bbox="470 1303 600 1341"></td> <td data-bbox="600 1303 731 1341"></td> <td data-bbox="731 1303 861 1341"></td> <td data-bbox="861 1303 992 1341"></td> <td data-bbox="992 1303 1122 1341"></td> <td data-bbox="1122 1303 1253 1341"></td> </tr> </tbody> </table> <p>The <b>rsqwaiters</b> column shows the number of statements waiting in a queue. The <b>rsqholders</b> column shows the number of queries currently running in a queue.</p>	rsqname	rsqcountlimit	rsqcountvalue	rsqcostlimit	rsqcostvalue	rsqwaiters	rsqholders	adhoc	1	1	1	-1	1	-1	pg_default	0	20	0	-1	0	-1	(2 rows)						
rsqname	rsqcountlimit	rsqcountvalue	rsqcostlimit	rsqcostvalue	rsqwaiters	rsqholders																							
adhoc	1	1	1	-1	1	-1																							
pg_default	0	20	0	-1	0	-1																							
(2 rows)																													

Step	Action
14.	<p>Leaving the second terminal session open, return to the first terminal session, the one with the open cursor, and end the transaction:</p> <pre data-bbox="314 238 486 270">faa=&gt; END ;</pre> 
15.	<p>Examine the second terminal session. The waiting query should have executed immediately after the transaction in the first session was closed.</p> <p>Exit <code>psql</code> in all three windows. Close all terminal sessions.</p> 
16.	<p><b>Summary</b></p> <p>Administrators can create resource queues for the various types of workloads in their organization. The administrator would then set limits on the resource queue based on his/her estimate of how resource intensive the queries associated with that workload are likely to be.</p> <p>Database roles (users) are then assigned to the appropriate resource queue. A resource queue can support multiple roles.</p>

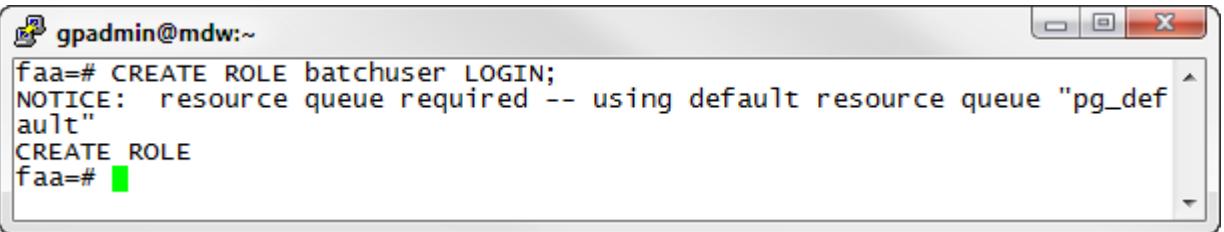
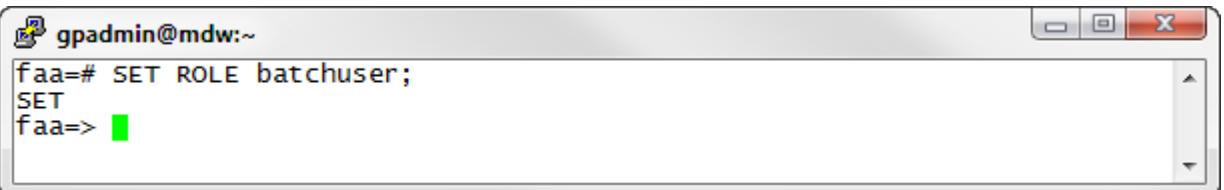
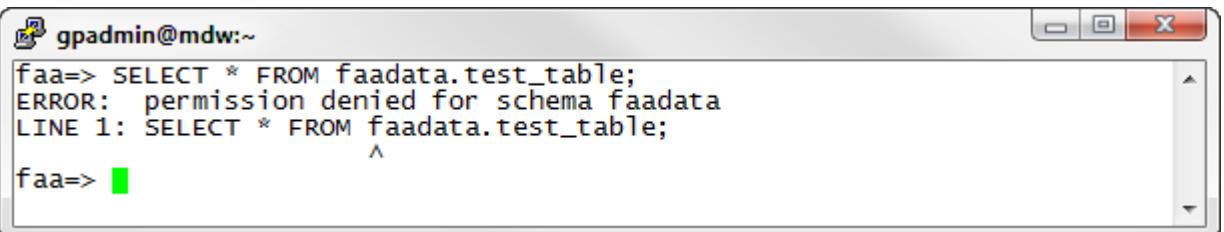
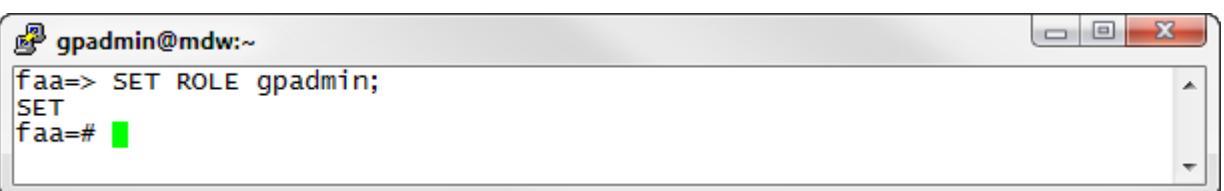
End of Lab Exercise

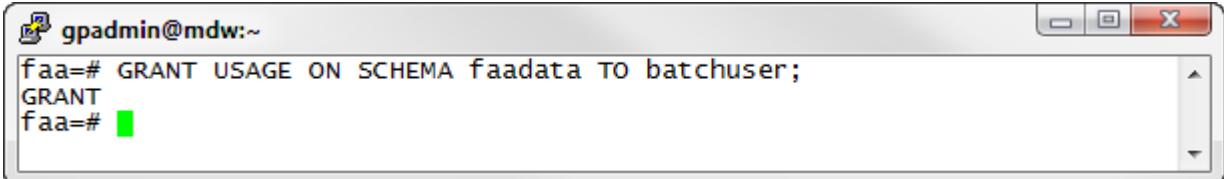
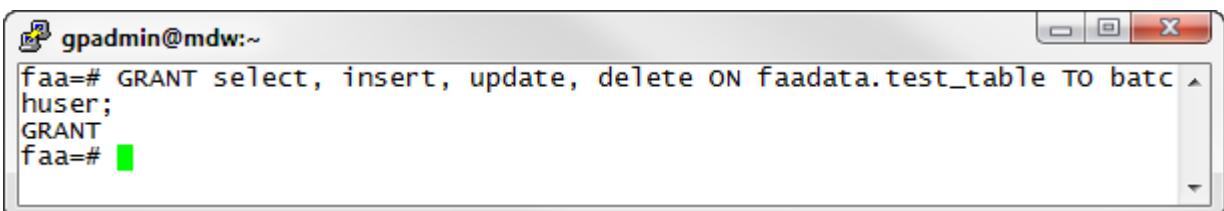
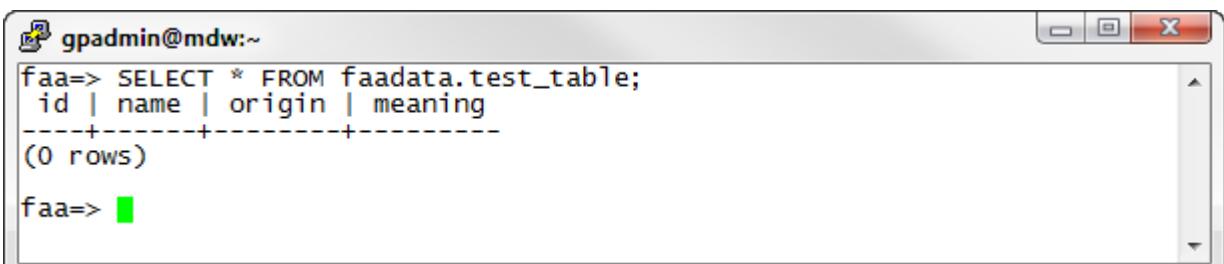
This page intentionally left blank

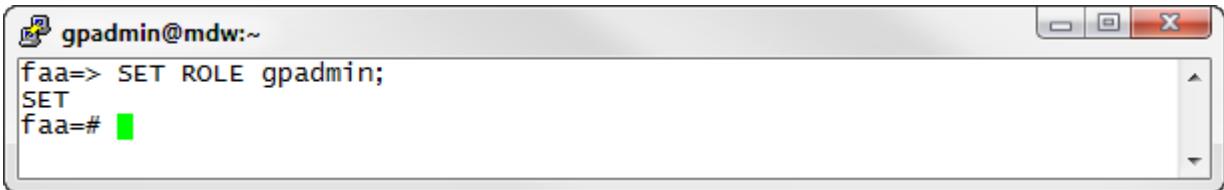
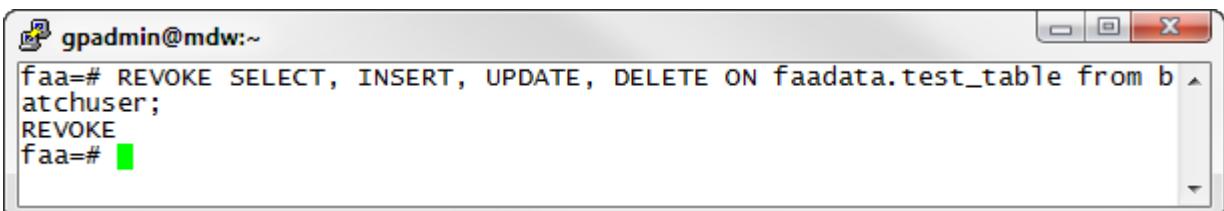
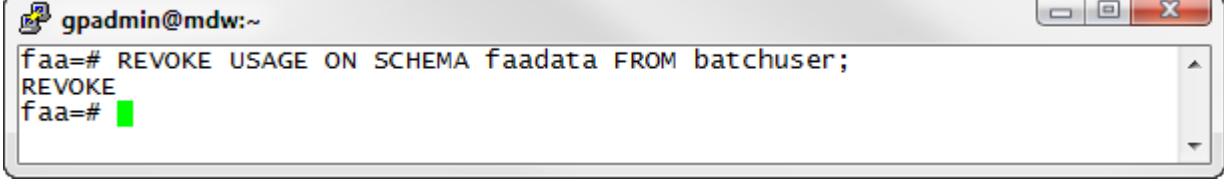
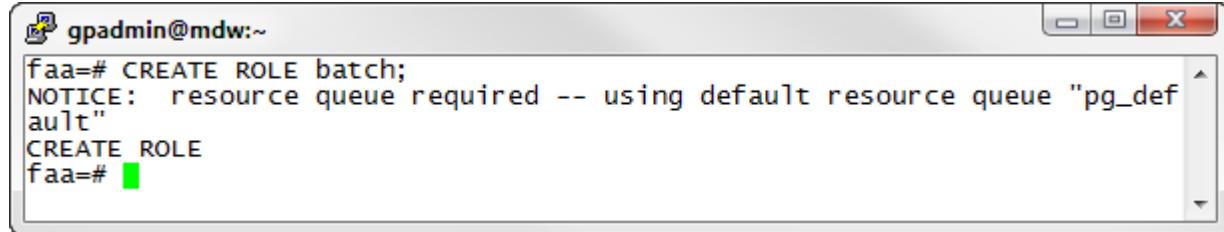
## Lab 10. Controlling Access

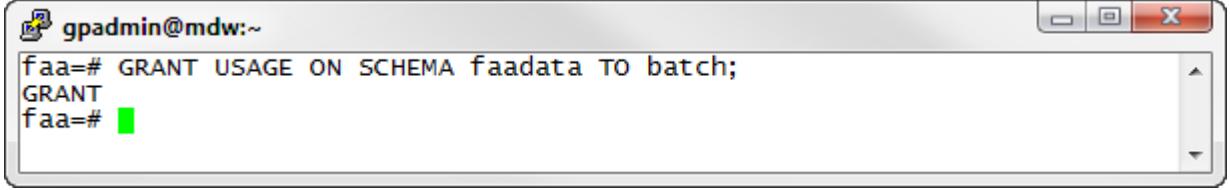
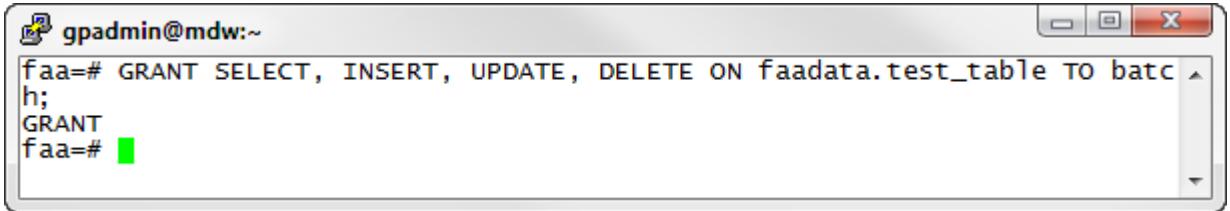
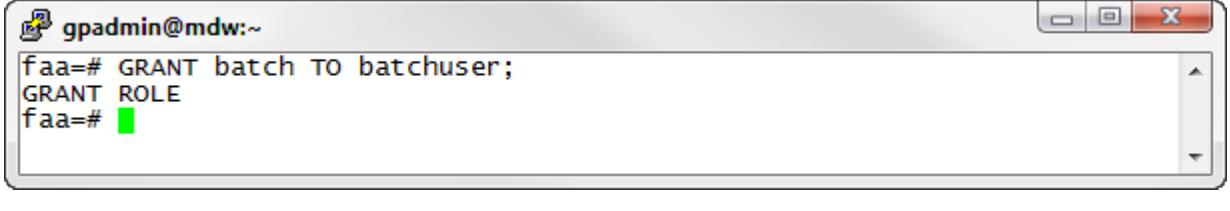
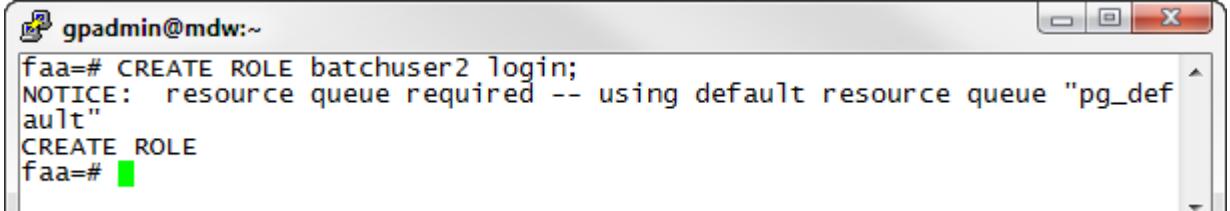
<b>Purpose:</b>	In this lab, you design and implement the basic security architecture for your data mart. You will create two new users and use groups to control the level of access that users, who will be members of the group, receive. This method makes it easier to administer privileges by working with groups instead of working with individual user accounts.
<b>Tasks:</b>	Students perform the following task: Implement basic security at the group level.
<b>References:</b>	Module 4 – Defining and Securing the User Database <ul style="list-style-type: none"><li>• Lesson: Roles, Privileges, and Resources</li></ul>

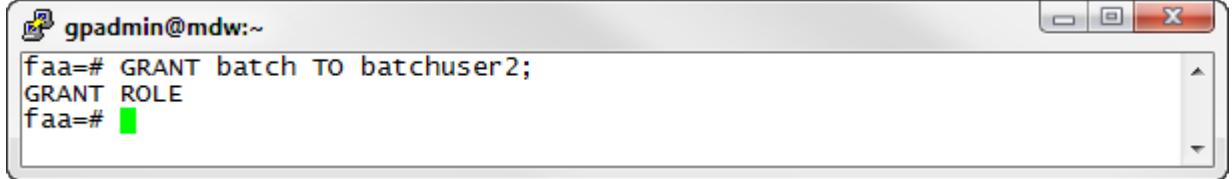
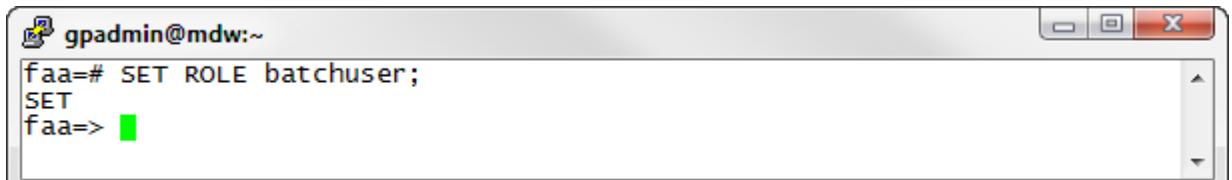
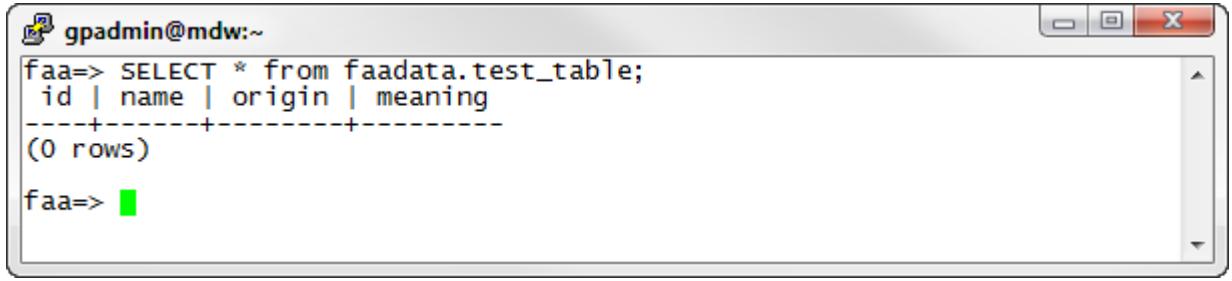
## Lab 10: Task – Implement Basic Security at the Group Level

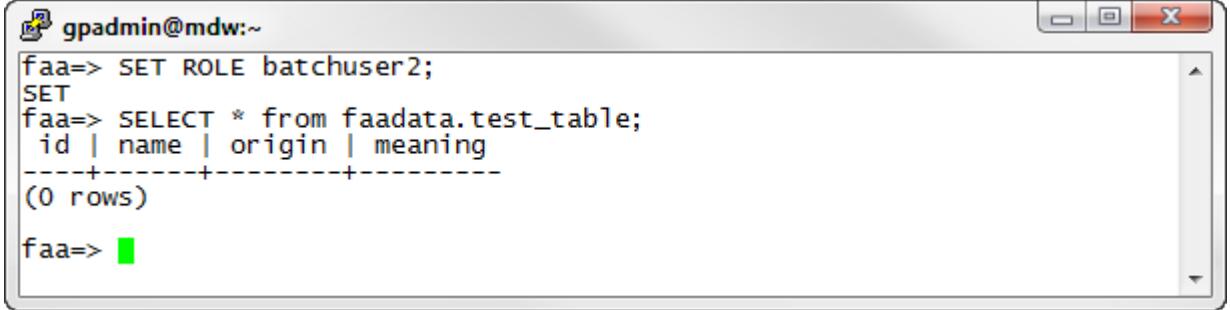
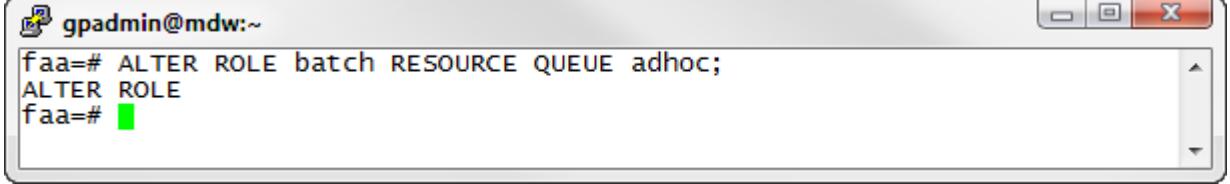
Step	Action
1.	<p>Open a terminal session to the master server using the gpadmin user account, if one is not already available.      Connect to the faa database as gpadmin.</p> <pre>[gpadmin@mdw ~]\$ psql faa</pre>
2.	<p>Create one of the user roles that will be used to verify the security implementation:</p> <pre>faa=# CREATE ROLE batchuser LOGIN;</pre> 
3.	<p>Change over to the new role you created, <b>batchuser</b>:</p> <pre>faa=# SET ROLE batchuser;</pre> 
4.	<p>Query the test_table table:</p> <pre>faa=&gt; SELECT * FROM faadata.test_table;</pre>  <p>Did it succeed? If not, why not? _____</p>
5.	<p>Change back to gpadmin:</p> <pre>faa=&gt; SET ROLE gpadmin;</pre> 

Step	Action
6.	<p>Grant permissions to the faadata schema:</p> <pre>faa=# GRANT USAGE ON SCHEMA faadata TO batchuser;</pre>  <p>While you have been granted USAGE access on the schema, you still do not have access to the existing tables that do not specifically grant access to all.</p>
7.	<p>Grant access of our test_table to the group:</p> <pre>faa=# GRANT select, insert, update, delete ON faadata.test_table TO batchuser;</pre> 
8.	<p>Try accessing the faadata.test_table again as batchuser. Change over to the new role:</p> <pre>faa=# SET ROLE batchuser;</pre> 
9.	<p>Query the test_table table:</p> <pre>faa=&gt; SELECT * FROM faadata.test_table;</pre>  <p>You should now have SELECT, INSERT, DELETE, and UPDATE access to the test_table table in the faadata schema.</p>

Step	Action
10.	Create a group role and assign privileges indirectly. First, change back to the gpadmin role:  faa=> <b>SET ROLE gpadmin;</b>  
11.	Revoke privilege to the faadata.test_table table as gpadmin:  faa=# <b>REVOKE SELECT, INSERT, UPDATE, DELETE ON faadata.test_table from batchuser;</b>  
12.	Revoke privileges to the <b>faadata</b> schema:  faa=# <b>REVOKE USAGE ON SCHEMA faadata FROM batchuser;</b> 
13.	Create the batch group role that you will assign privileges to:  faa=# <b>CREATE ROLE batch;</b>  

Step	Action
14.	<p>Grant permissions to the faadata schema to the batch role you created:</p> <pre data-bbox="300 219 1095 249">faa=# GRANT USAGE ON SCHEMA faadata TO batch;</pre> 
15.	<p>Grant access on the test_table to the group, batch:</p> <pre data-bbox="300 593 1486 650">faa=# GRANT SELECT, INSERT, UPDATE, DELETE ON faadata.test_table TO batch;</pre>  <p>Members of the group, batch, will now have SELECT, INSERT, UPDATE, and DELETE privileges on the faadata.test_table table.</p>
16.	<p>Assign a user role to the newly created group role:</p> <pre data-bbox="300 1132 850 1159">faa=# GRANT batch TO batchuser;</pre> 
17.	<p>Create a second user role, batchuser2:</p> <pre data-bbox="300 1501 920 1529">faa=# CREATE ROLE batchuser2 login;</pre> 

Step	Action
18.	<p>Assign the batchuser2 user role to the batch group role:</p> <pre data-bbox="298 228 866 259">faa=# GRANT batch TO batchuser2;</pre> 
19.	<p>Verify that both user roles have access the faadata.test_table table. First, change to the batch user role:</p> <pre data-bbox="298 629 740 661">faa=# SET ROLE batchuser;</pre> 
20.	<p>Query the test_table table:</p> <pre data-bbox="298 1003 985 1034">faa=&gt; SELECT * from faadata.test_table;</pre>  <p>You should have access to the table because of the role's relationship to the batch role.</p>

Step	Action
21.	<p>Change to the batchuser2 role and verify this user has access to the same table:</p> <pre data-bbox="300 219 992 287">faa=&gt; SET ROLE batchuser2; faa=&gt; SELECT * from faadata.test_table;</pre>  <p>This role has also been granted access to the table due to its association with the batch role.</p>
22.	<p>Tie all users in the batch group into the adhoc resource queue you created earlier:</p> <p>Change to the gpadmin role:</p> <pre data-bbox="300 889 708 920">faa=&gt; SET ROLE gpadmin;</pre> 
23.	<p>Add the batch role to the resource queue, adhoc:</p> <pre data-bbox="300 1262 1078 1290">faa=# ALTER ROLE batch RESOURCE QUEUE adhoc;</pre>  <p>Exit psql.</p> <pre data-bbox="300 1613 447 1641">faa=# \q</pre>

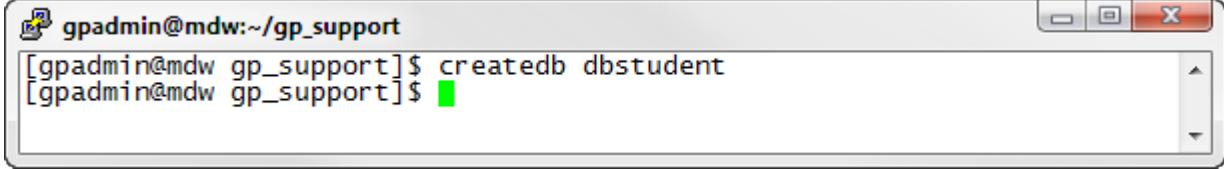
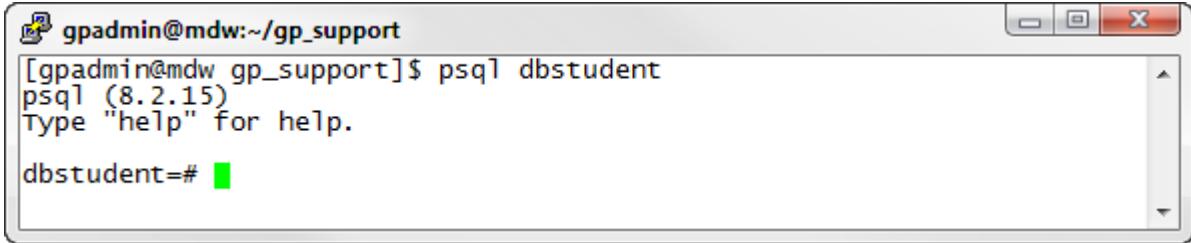
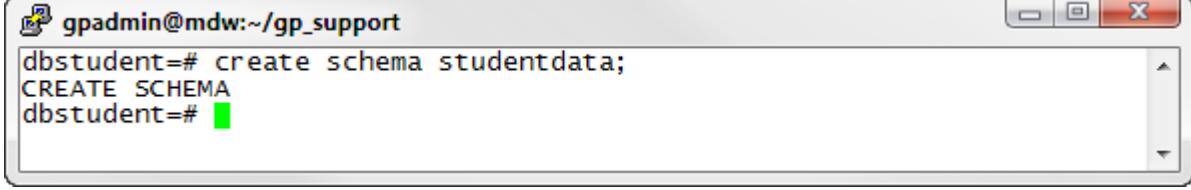
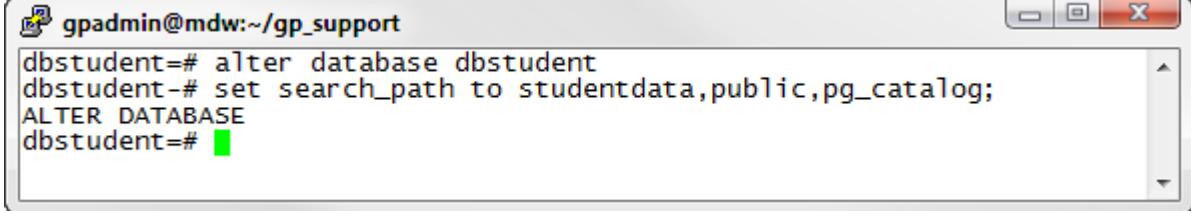
Step	Action
	<p><b>Summary</b></p> <p>You should work with the user community and the administrators to come up with a security architecture that will fit your access requirements.</p> <p>Table and schema level access is done through the use of roles.</p> <p>Row level access is best accomplished with a combination of roles and views against the data that has all of the security logic built into the view. While this may inhibit performance for some queries, it will also ensure that your data is secure and only “need to know” access is given to the end users.</p> <p>Row level security may also be performed by reporting engines. This may suffice if your data security level is low.</p>

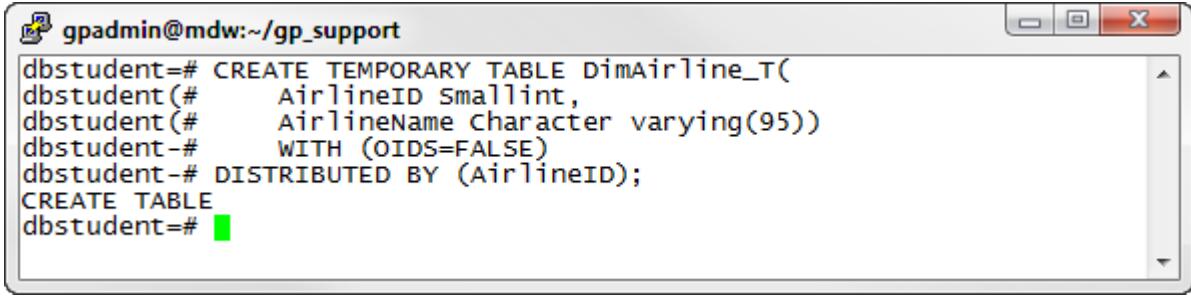
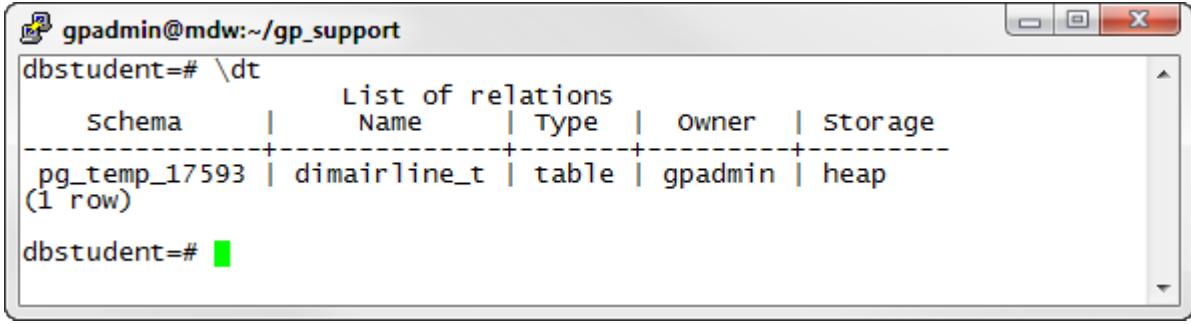
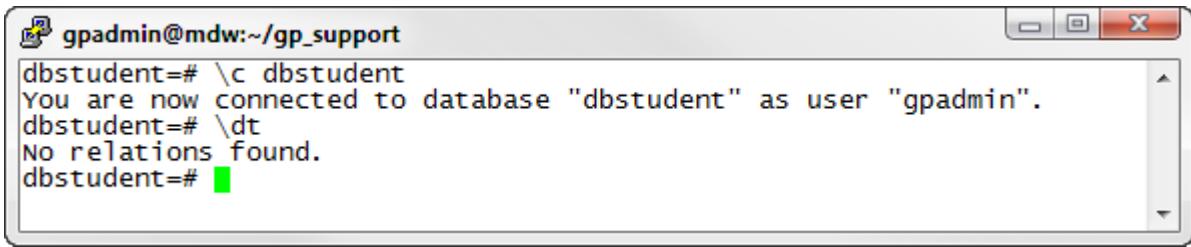
*End of Lab Exercise*

## Lab 11. Table Management

<b>Purpose:</b>	The purpose of this lab is to create various types of supported Greenplum Database tables. You will create tables of varying types, load data to these tables, and discover which table type offers better performance under specific conditions.
<b>Tasks:</b>	Students perform the following tasks: <ul style="list-style-type: none"><li>• Create a temporary table</li><li>• Verify the size of two identical tables where one is compressed and the other is not</li><li>• Verify the performance on a column-oriented table</li></ul>
<b>References:</b>	Module 5 – Data Loading and Distribution <ul style="list-style-type: none"><li>• Lesson: Implementing Table Storage Models, Compression, and Tablespaces</li></ul>

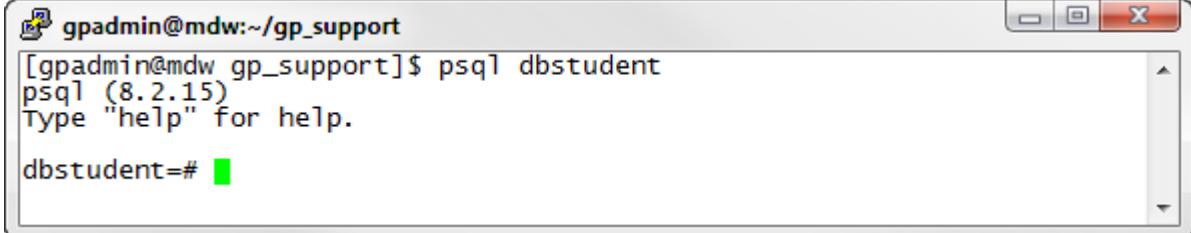
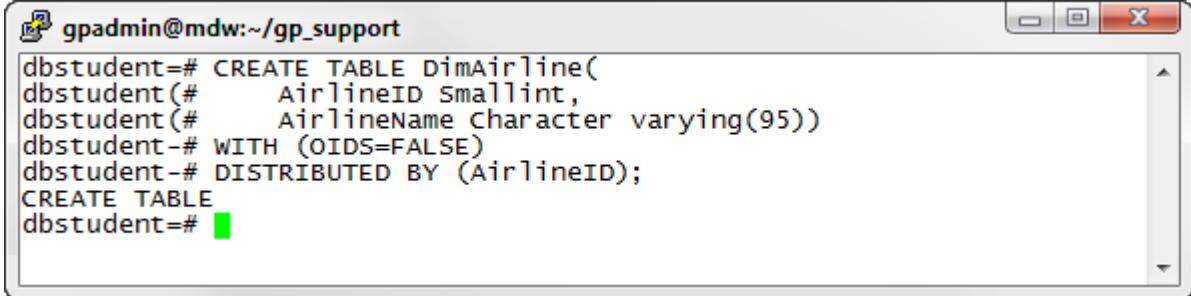
## Lab 11: Task 1 – Creating Temporary Tables

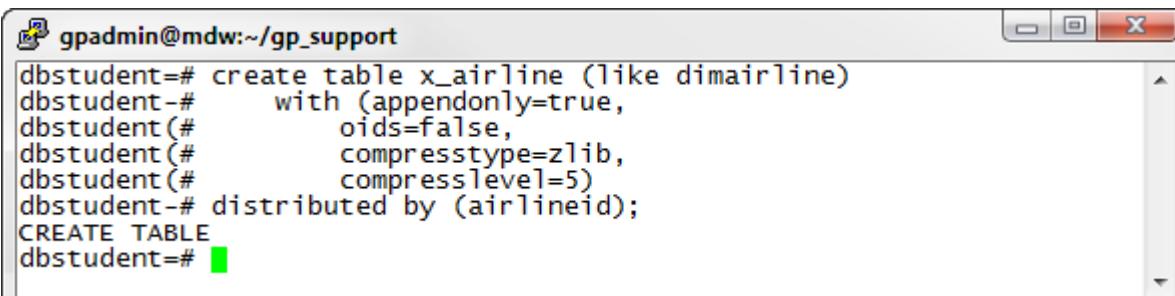
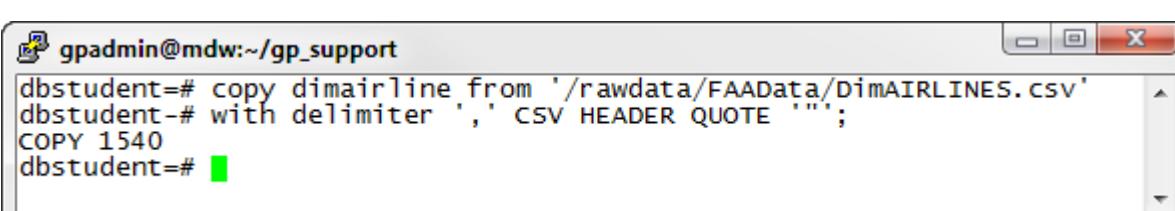
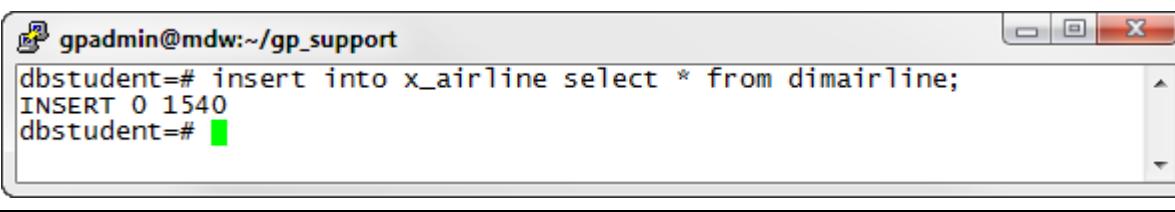
Step	Action
1.	<p>Create the dbstudent database which will be used to support your tables.</p> <pre>[gpadmin@mdw ~] \$ createdb dbstudent</pre>  A screenshot of a terminal window titled "gpadmin@mdw:~/gp_support". It shows the command "createdb dbstudent" being run and its output. The window has standard operating system window controls (minimize, maximize, close) at the top right. <pre>gpadmin@mdw:~/gp_support [gpadmin@mdw gp_support]\$ createdb dbstudent [gpadmin@mdw gp_support]\$</pre>
2.	<p>Connect to the dbstudent database:</p> <pre>[gpadmin@mdw ~] \$ psql dbstudent</pre>  A screenshot of a terminal window titled "gpadmin@mdw:~/gp_support". It shows the command "psql dbstudent" being run and its output. The window has standard operating system window controls (minimize, maximize, close) at the top right. <pre>gpadmin@mdw:~/gp_support [gpadmin@mdw gp_support]\$ psql dbstudent psql (8.2.15) Type "help" for help.  dbstudent=#</pre>
3.	<p>Create a new schema called studentdata.</p> <pre>dbstudent=# create schema studentdata;</pre>  A screenshot of a terminal window titled "gpadmin@mdw:~/gp_support". It shows the command "create schema studentdata;" being run and its output. The window has standard operating system window controls (minimize, maximize, close) at the top right. <pre>gpadmin@mdw:~/gp_support dbstudent=# create schema studentdata; CREATE SCHEMA dbstudent=# </pre>
4.	<p>Change the search_path parameter on the database dbstudent so that the new studentdata schema is the default schema.</p> <pre>dbstudent=# alter database dbstudent set search_path to studentdata,public,pg_catalog;</pre>  A screenshot of a terminal window titled "gpadmin@mdw:~/gp_support". It shows the command "alter database dbstudent set search_path to studentdata,public,pg_catalog;" being run and its output. The window has standard operating system window controls (minimize, maximize, close) at the top right. <pre>gpadmin@mdw:~/gp_support dbstudent=# alter database dbstudent dbstudent-# set search_path to studentdata,public,pg_catalog; ALTER DATABASE dbstudent=# </pre> <p>Reconnect to the database to re-read the search_path settings.</p> <pre>dbstudent=# \c dbstudent</pre>

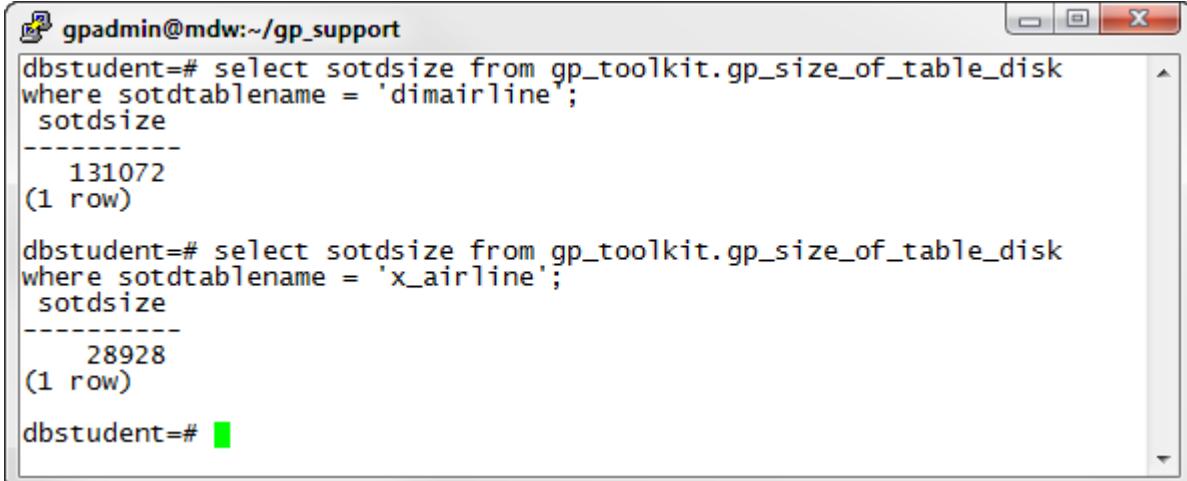
Step	Action
5.	<p>Create a temporary table called dimairline_t using the following SQL syntax:</p> <pre data-bbox="192 228 1041 386">dbstudent=# CREATE TEMPORARY TABLE DimAirline_T(     AirlineID Smallint,     AirlineName Character varying(95))     WITH (OIDS=FALSE) DISTRIBUTED BY (AirlineID);</pre> 
6.	<p>List all the tables in the dbstudent database.</p> <pre data-bbox="192 804 458 834">dbstudent=# \dt</pre>  <p>Note that the table is listed in a temporary schema. The schema name may vary from what is displayed in this step. Any other temporary tables created in this session will be saved to the same temporary schema.</p>
7.	<p>Reconnect to the dbstudent database.</p> <pre data-bbox="192 1417 621 1446">dbstudent=# \c dbstudent</pre>
8.	<p>List all the tables in the dbstudent database.</p> <pre data-bbox="192 1558 458 1588">dbstudent=# \dt</pre>  <p>The temporary schema that was created when you created the temporary table is automatically dropped when the session ends. Any tables in the schema are also dropped. Reconnecting to the database creates another temporary schema which will be used should you create temporary tables in the new session.</p>

Step	Action
	<p><b>Summary</b></p> <p>Temporary tables exist in a special schema, so a schema name may not be given when creating a temporary table. When creating temporary tables, the table name must be unique to the session.</p> <p>Different sessions can use the same name as the schema name will differ from session to session.</p> <p>Temporary tables are automatically dropped at the end of the session or optionally, at the end of a transaction with the <code>ON COMMIT</code> clause. This will occur when dropping the session or reconnecting to it, something that occurs often with business intelligence applications and reporting tools.</p> <p>Temporary tables are a good way of handling complex and intensive SQL statements used for generating reports. They can be used to reduce performance impacts for queries that may have a tendency to generate computational skew.</p>

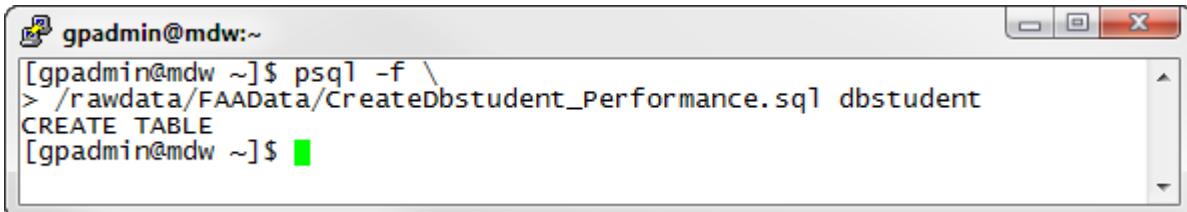
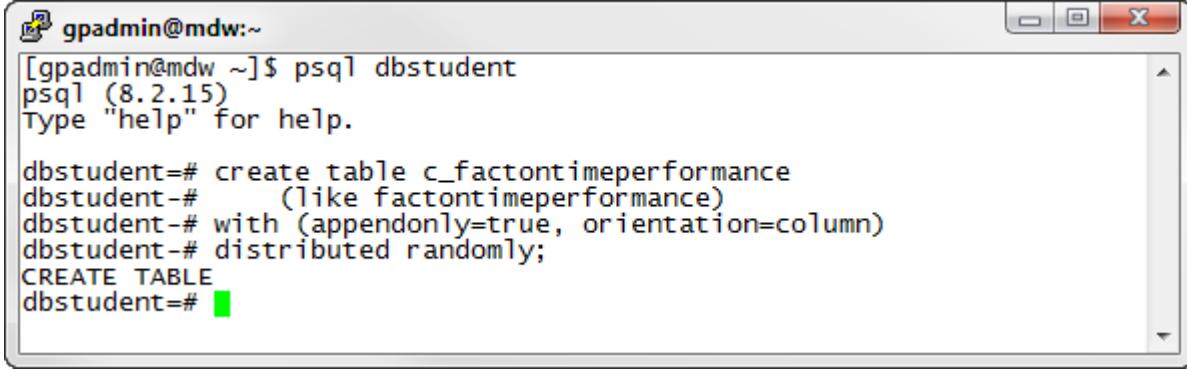
## Lab 11: Task 2 – Creating Compressed Tables

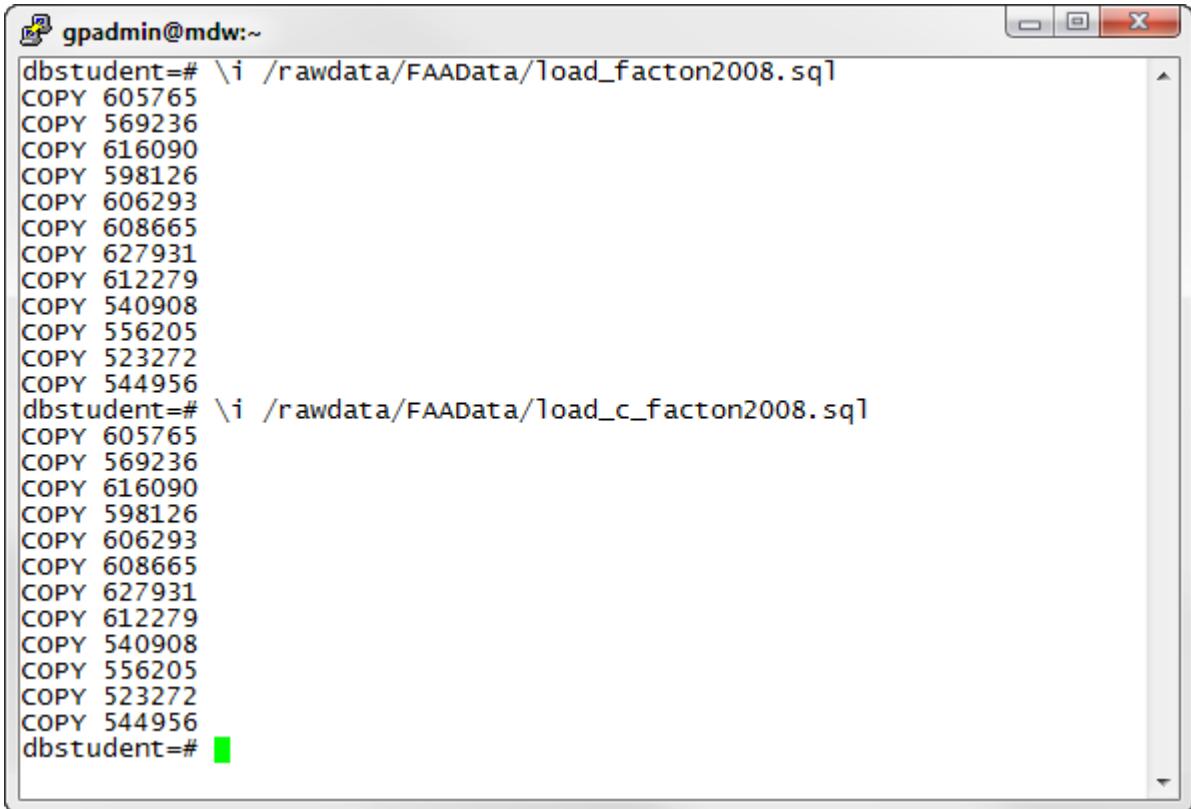
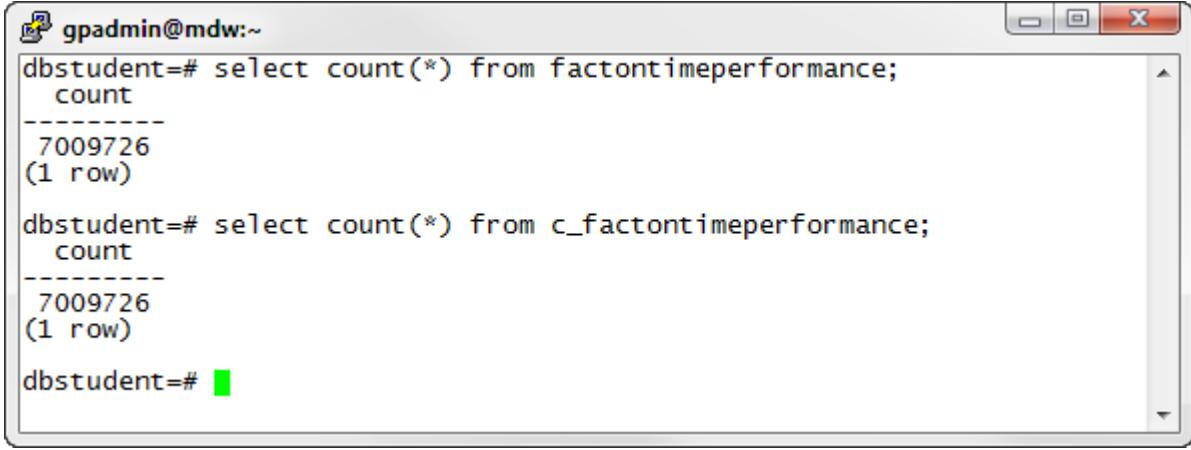
Step	Action
1.	<p>If not already connected to the database, connect to the dbstudent database.</p> <pre>[gpadmin@mdw ~] \$ psql dbstudent</pre> 
2.	<p>In this task, you will create two tables: a compressed and a regular heap table. These two tables both have an equal number of fields and records.</p> <p>Create a regular heap table called dimairline as shown below:</p> <pre>dbstudent=# CREATE TABLE DimAirline(     AirlineID Smallint,     AirlineName Character varying(95)) WITH (OIDS=FALSE) DISTRIBUTED BY (AirlineID);</pre> 

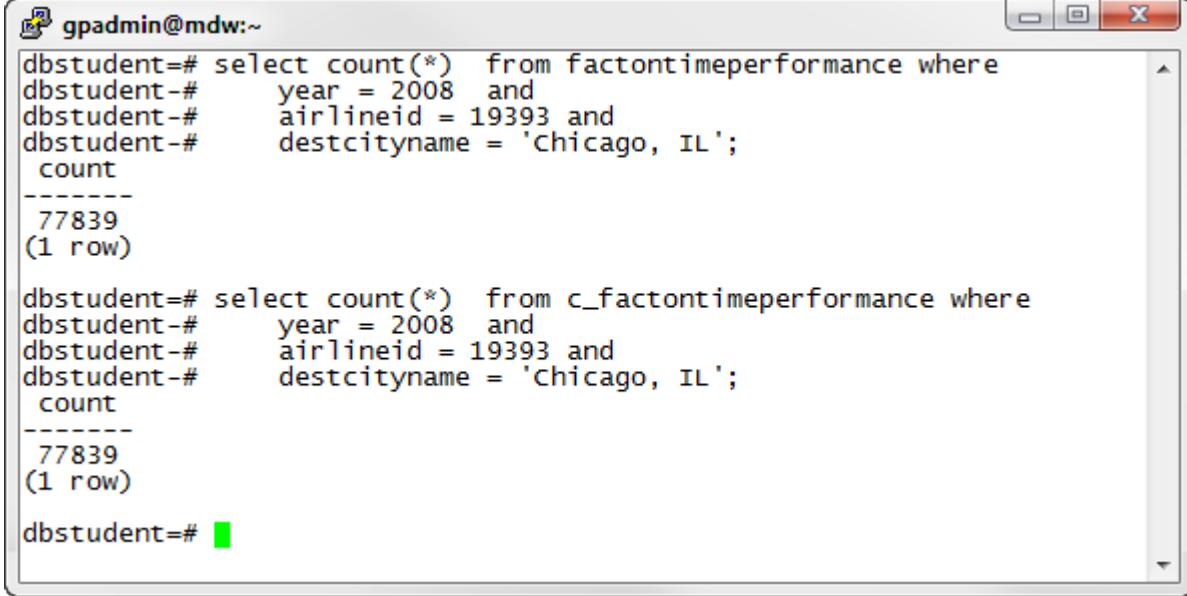
Step	Action
3.	<p>Create a compressed table called <code>x_airline</code> as shown below:</p> <pre>dbstudent=# create table x_airline (like dimairline)   with (appendonly=true,         oids=false,         compressstype=zlib,         compresslevel=5) distributed by (airlineid);</pre> 
4.	<p>Populate the heap table as shown.</p> <pre>dbstudent=# copy dimairline from '/rawdata/FAAData/DimAIRLINES.csv' with delimiter ',' CSV HEADER QUOTE ''';</pre> 
5.	<p>Populate the compressed table with records from <code>dimairline</code> table as shown.</p> <pre>dbstudent=# insert into x_airline select * from dimairline;</pre> 

Step	Action
6.	<p>Verify the size of both tables by viewing the sotdsize column of the gp_toolkit(gp_size_of_table_disk) table. This table stores information on all tables within the database. The size column, sotdsize, is displayed in bytes.</p> <pre data-bbox="192 312 1346 460">dbstudent=# select sotdsize from gp_toolkit.gp_size_of_table_disk where sotdtablename = 'dimairline'; dbstudent=# select sotdsize from gp_toolkit.gp_size_of_table_disk where sotdtablename = 'x_airline';</pre>  <p>Note that the listed size of the tables differ greatly. The compressed AO table, x_airline, is less than 25% of the size of the heap table, dimairline.</p>
7.	<p>Exit the database once you have completed the steps.</p> <pre data-bbox="192 1189 448 1220">dbstudent=# \q</pre>
	<p><b>Summary</b></p> <p>Both tables have the same number of fields and records. However, the compressed table is considerably smaller in terms of the amount of its disk size than the uncompressed table. Tables that use compression must be an append-only table. You can only add new rows to an append-only table. You cannot update or delete rows from an append-only table. This allows for more compact storage on disk. It saves 20 bytes per row.</p> <p>Choosing greater levels of compression result in more compact storage. Increasing compression requires more CPU cycles to access the data when required, either on reads or writes. For data that is not often accessed, it may be applicable to apply strong compression to the table, or partition. This ensures that older data that is not often accessed do not consume large amounts of storage and should balance well against how often you need to access them.</p>

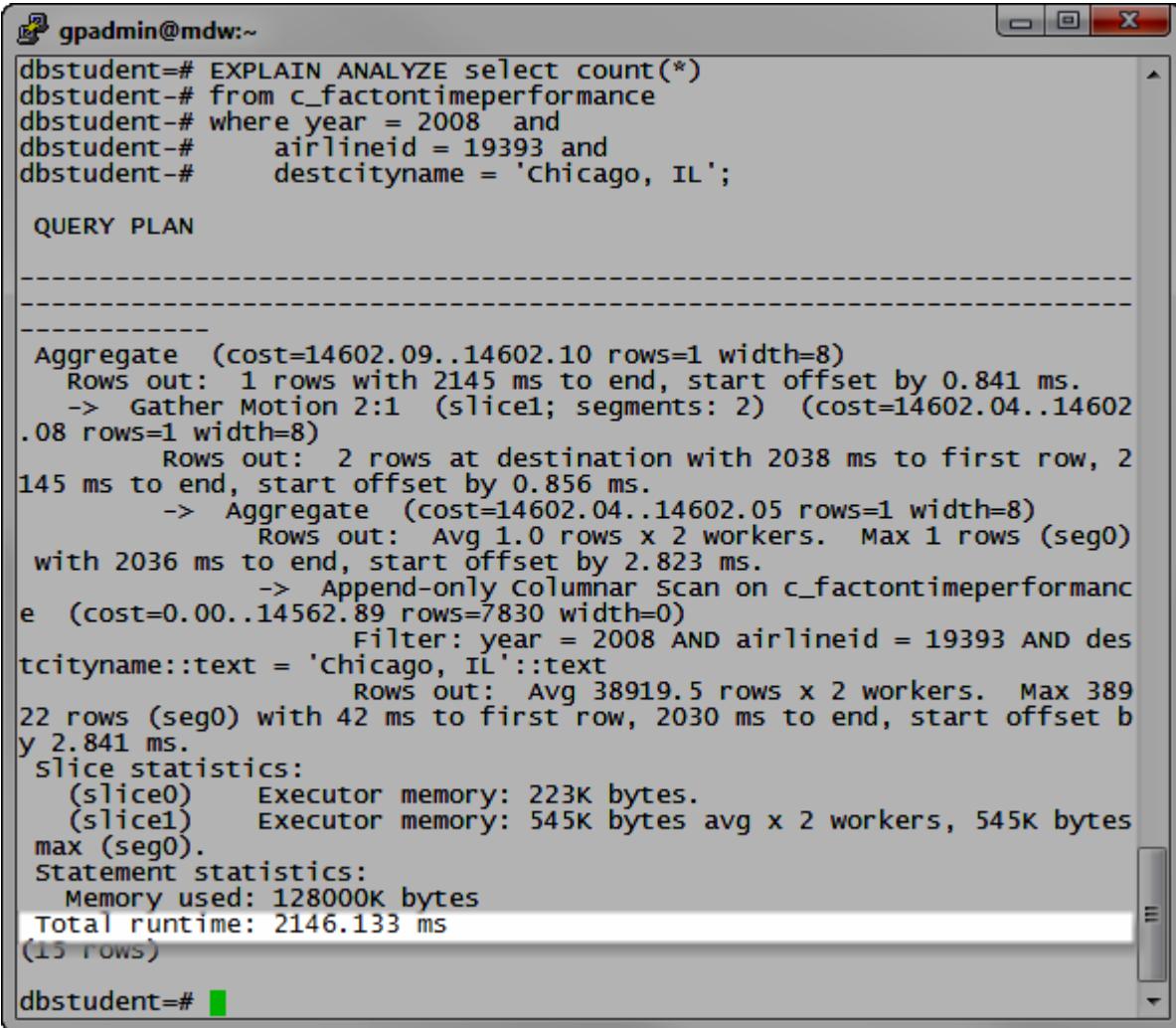
## Lab 11: Task 3 – Creating a Column-Wise Storage Table

Step	Action
1.	<p>Create two tables in the dbstudent database. The tables will be called factontimeperformance and c_factontimeperformance. The factontimeperformance table is a regular heap table and c_factontimeperformance is a column-wise storage oriented table. Run the script below to create the heap table.</p> <pre>[gpadmin@mdw ~]\$ psql -f \ /rawdata/FAAData/CreateDbstudent_Performance.sql dbstudent</pre>  A screenshot of a terminal window titled "gpadmin@mdw:~". The window contains the command "psql -f \> /rawdata/FAAData/CreateDbstudent_Performance.sql dbstudent" followed by the output "CREATE TABLE". <p>The heap table factontimeperformance has been created.</p> <p>Connect to the dbstudent database and create the compressed table, c_factontimeperformance, based factontimeperformance as shown.</p> <pre>[gpadmin@mdw ~]\$ psql dbstudent dbstudent=# create table c_factontimeperformance           (like factontimeperformance)       with (appendonly=true, orientation=column)       distributed randomly;</pre>  A screenshot of a terminal window titled "gpadmin@mdw:~". The window contains the command "psql dbstudent" followed by the command "create table c_factontimeperformance (like factontimeperformance) with (appendonly=true, orientation=column) distributed randomly;". The output shows the table being created with the message "CREATE TABLE".
2.	

Step	Action
3.	<p>Use the \i command to execute scripts that will populate the tables you have just created. It may take a few minutes to load all of the data.</p> <pre>dbstudent=# \i /rawdata/FAAData/load_facton2008.sql dbstudent=# \i /rawdata/FAAData/load_c_facton2008.sql</pre>  <p>After running these scripts, each table should contain 7 million records. The data loaded into both tables represent the entire year of 2008 for FAA flight data.</p>
4.	<p>Verify you have the same number of records on both tables.</p> <pre>dbstudent=# select count(*) from factontimeperformance; dbstudent=# select count(*) from c_factontimeperformance;</pre> 

Step	Action
5.	<p>Calculate how many times Southwest Airlines flew to Chicago in the year 2008 running the query below. Run the query on both tables: factontimeperformance and c_factontimeperformance. You should see the same results for both tables.</p> <pre data-bbox="192 304 1312 599">dbstudent=# select count(*)  from factontimeperformance where   year = 2008  and   airlineid = 19393 and   destcityname = 'Chicago, IL'; dbstudent=# select count(*)  from c_factontimeperformance where   year = 2008  and   airlineid = 19393 and   destcityname = 'Chicago, IL';</pre>  <pre data-bbox="208 699 1225 1205">gpadmin@mdw:~\$ dbstudent=# select count(*)  from factontimeperformance where dbstudent-#      year = 2008  and dbstudent-#      airlineid = 19393 and dbstudent-#      destcityname = 'Chicago, IL'; count ----- 77839 (1 row)  dbstudent=# select count(*)  from c_factontimeperformance where dbstudent-#      year = 2008  and dbstudent-#      airlineid = 19393 and dbstudent-#      destcityname = 'Chicago, IL'; count ----- 77839 (1 row)  dbstudent=# gpadmin@mdw:~\$</pre>

Step	Action
6.	<p>To verify which query is faster, analyze the queries using <code>explain analyze</code> command. This command will be taught in more detail in later labs. For now, pay close attention to the <b>Total runtime</b> line.</p> <p>Run the command for the <code>factontimeperformance</code> table, which is a row-oriented heap table.</p> <pre>dbstudent=# EXPLAIN ANALYZE select count(*) from factontimeperformance where year = 2008 and       airlineid = 19393 and       destcityname = 'Chicago, IL';</pre> <pre>gpadmin@mdw:~ dbstudent=# EXPLAIN ANALYZE select count(*) dbstudent-# from factontimeperformance dbstudent-# where year = 2008 and dbstudent-#     airlineid = 19393 and dbstudent-#     destcityname = 'Chicago, IL';  QUERY PLAN  ----- Aggregate  (cost=15755.21..15755.22 rows=1 width=8)   Rows out: 1 rows with 9746 ms to end, start offset by 0.963 ms.     -&gt; Gather Motion 2:1 (slice1; segments: 2)  (cost=15755.16..15755 .20 rows=1 width=8)       Rows out: 2 rows at destination with 9597 ms to first row, 9 746 ms to end, start offset by 0.980 ms.       -&gt; Aggregate (cost=15755.16..15755.17 rows=1 width=8)         Rows out: Avg 1.0 rows x 2 workers. Max 1 rows (seg0)         with 9744 ms to end, start offset by 2.813 ms.         -&gt; Seq Scan on factontimeperformance  (cost=0.00..1571 5.82 rows=7867 width=0)           Filter: year = 2008 AND airlineid = 19393 AND des tcityname::text = 'chicago, IL'::text           Rows out: Avg 38919.5 rows x 2 workers. Max 389 22 rows (seg0) with 25 ms to first row, 9733 ms to end, start offset b y 2.842 ms.           Slice statistics:             (slice0) Executor memory: 223K bytes.             (slice1) Executor memory: 163K bytes avg x 2 workers, 163K bytes max (seg0).           Statement statistics:             Memory used: 128000K bytes             Total runtime: 9751.207 ms (15 rows)  dbstudent=# </pre> <p>In this example, the query took 9751.207ms to complete. Your time may vary from what is displayed here. Compare these results against the append-only column-oriented table.</p>

Step	Action
7.	<p>Execute the same command for the c_factontimeperformance table, which is an append-only column oriented table.</p> <pre>dbstudent=# EXPLAIN ANALYZE select count(*) from c_factontimeperformance where year = 2008 and       airlineid = 19393 and       destcityname = 'Chicago, IL';</pre>  <p>In this example, the query took 2146.133 ms to complete. Your time may vary from what is displayed here. You should note that the completion time on the column-oriented table is significantly less than the heap table. As there are few columns that are being accessed, column-oriented tables offer an advantage in this query.</p>
8.	Exit the database.

Step	Action
	<p><b>Summary</b></p> <p>For most general purpose or mixed workloads, row-oriented storage offers the best combination of flexibility and performance. However, there are certain specific use cases where a column-oriented storage model provides more efficient I/O and storage.</p> <p>Column-oriented tables offer strong performance gains over row-oriented tables when the table is being used mostly for reads and there are few columns being selected against or aggregated over. For data that is not updated often or is limited to a single column, column-oriented tables may offer a distinct advantage for gathering data from the table over row-oriented tables. A negative impact can be seen if you need to select a majority of the columns on a column-oriented table. Understanding the data and how it will be used will help you choose the appropriate storage model for the table.</p>

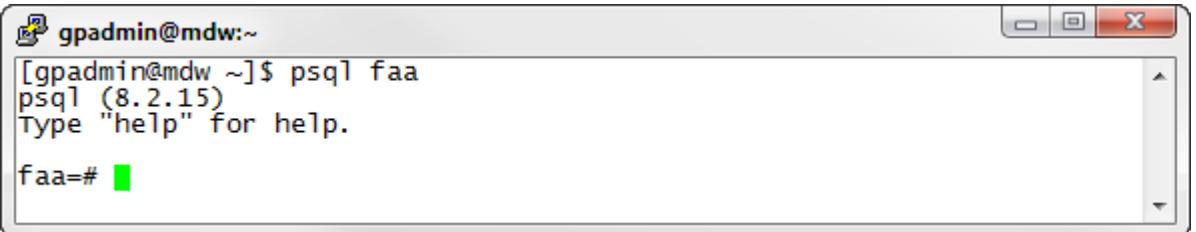
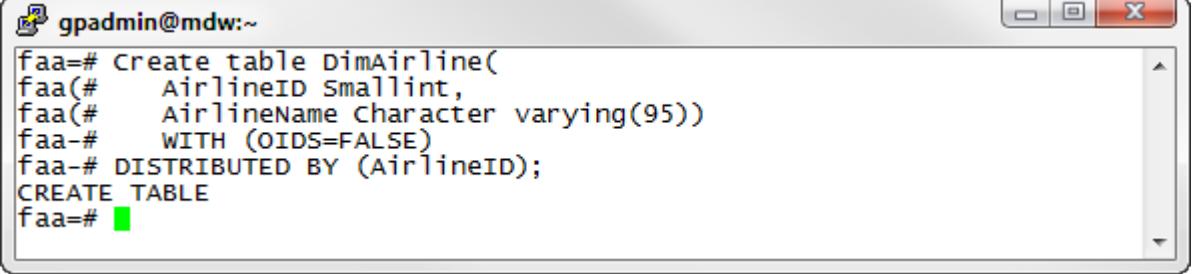
*End of Lab Exercise*

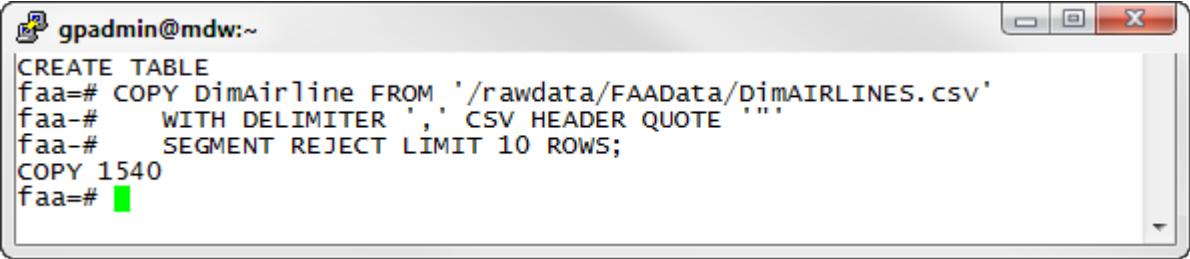
This page intentionally left blank

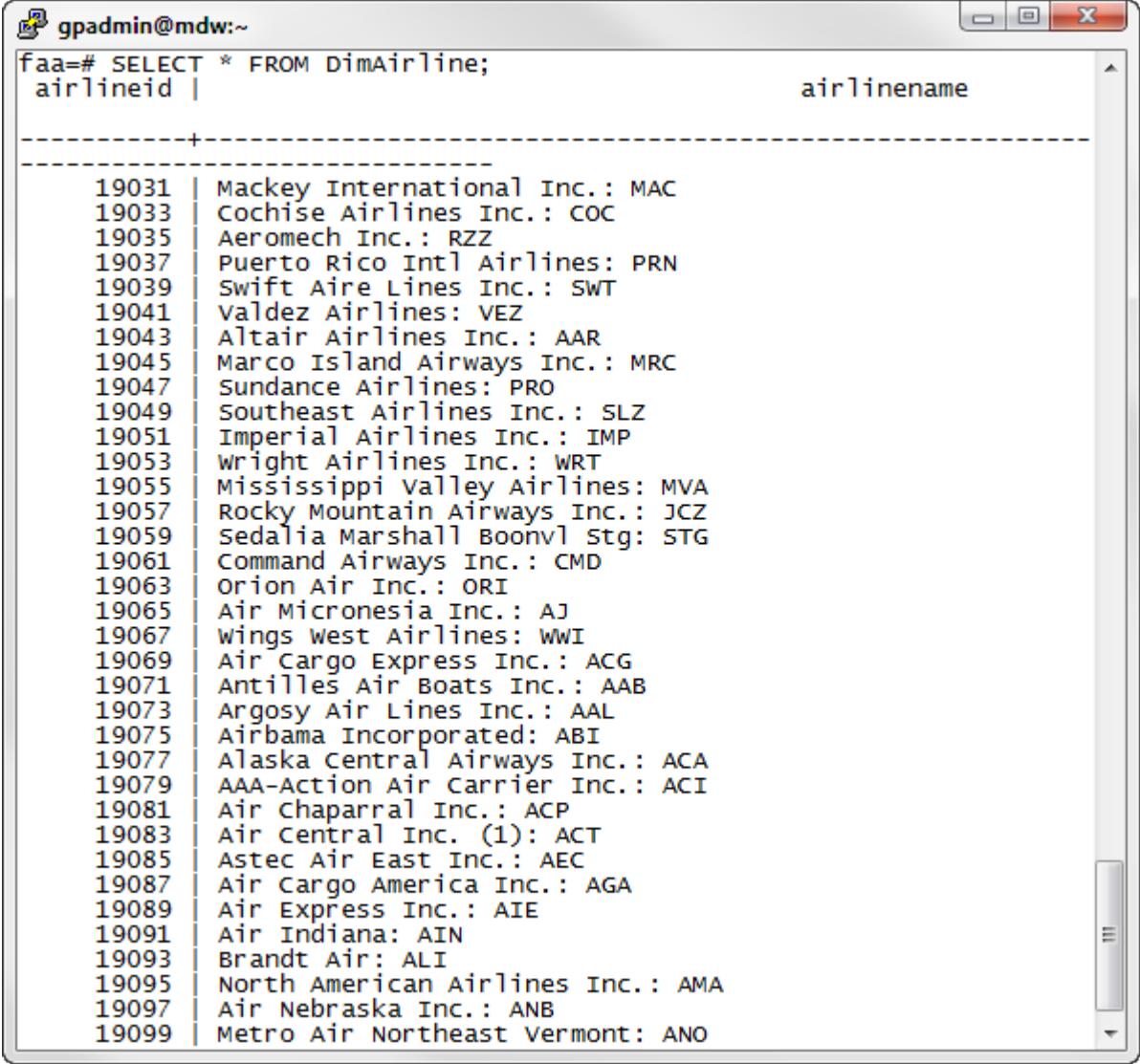
## Lab 12. Data Loading

<b>Purpose:</b>	<p>The purpose of this lab is to learn the different techniques for loading data into a Greenplum system.</p> <p>You will start by using the <code>COPY</code> command. The <code>COPY</code> command is well suited for importing small amounts of data in text or CSV format.</p> <p>The second technique you will learn for loading data is creating external tables and using an insert statement from the external table into an internal Greenplum table.</p> <p>The third technique that you will learn for loading data is using the <code>gpfdist</code> utility.</p>
<b>Tasks:</b>	<p>Students perform the following tasks:</p> <ul style="list-style-type: none"><li>• Create a dimension table and load data using <code>COPY</code></li><li>• Create a dimension table and load data using external tables</li><li>• Create a fact table and load data using <code>gpfdist</code></li><li>• Create a writeable external table to populate a different database</li><li>• Load data from a compressed file with <code>gpfdist</code></li></ul>
<b>References:</b>	<p>Module 5 – Data Loading and Distribution</p> <ul style="list-style-type: none"><li>• Lesson: Data Loading</li></ul>

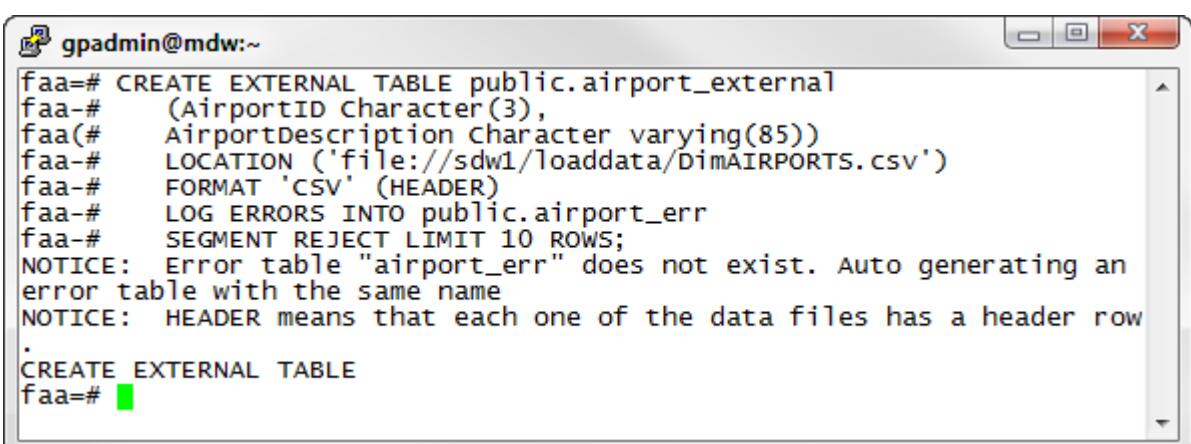
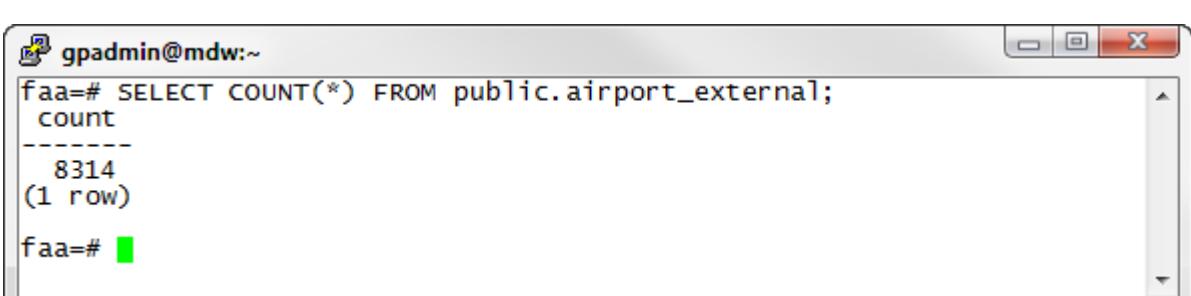
## Lab 12: Task 1 – Create a Dimension Table and Load Data Using COPY

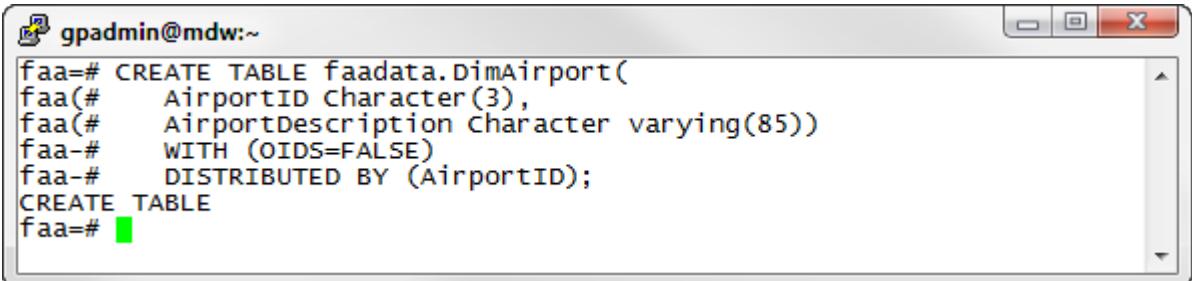
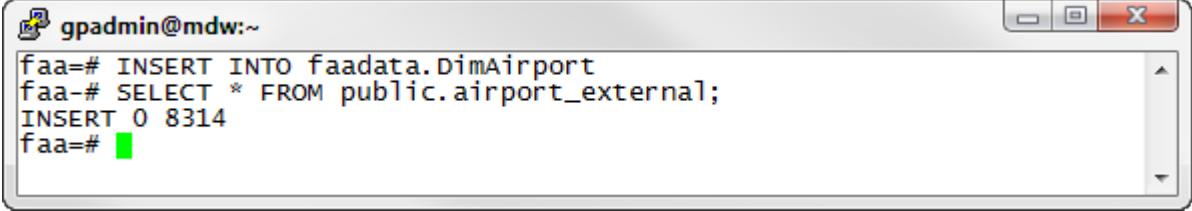
Step	Action
1.	<p>Connect to the faa database as the gpadmin user:</p> <pre>[gpadmin@mdw ~]\$ psql faa</pre> 
2.	<p>In a separate terminal session where you have a psql session started, create the first Dimension table DimAirline:</p> <pre>faa=# Create table DimAirline(     AirlineID Smallint,     AirlineName Character varying(95))     WITH (OIDS=FALSE) DISTRIBUTED BY (AirlineID);</pre>  <p>Note: You can also copy and paste this command from the script DDLScript.sql located in /rawdata/FAAData.</p>

Step	Action
3.	<p>From the <code>psql</code> session, use the <code>COPY</code> command to populate the <code>DimAirline</code> table using the data from the CSV file:</p> <pre data-bbox="313 270 1383 361">faa=# COPY DimAirline FROM '/rawdata/FAAData/DimAIRLINES.csv'       WITH DELIMITER ',' CSV HEADER QUOTE """       SEGMENT REJECT LIMIT 10 ROWS;</pre>  <p>Use single quotes to refer to the file in the <code>COPY</code> command. The delimiter used in this file is the comma. The data file is a CSV file with a header. Quoted items are surrounded with double quotes.</p> <p>In this example, a maximum of ten (10) row failures are allowed before the copy fails.</p> <p>Note: You can also copy and paste the above command from the script <code>CopyData.sql</code> located on <code>/rawdata/FAAData</code>.</p>

Step	Action																																																																								
4.	<p>Verify that the table is now populated with data:</p> <pre data-bbox="311 206 861 236">faa=# SELECT * FROM DimAirline;</pre>  <table border="1"> <thead> <tr> <th>airlineid</th> <th>airlinename</th> </tr> </thead> <tbody> <tr><td>19031</td><td>Mackey International Inc.: MAC</td></tr> <tr><td>19033</td><td>Cochise Airlines Inc.: COC</td></tr> <tr><td>19035</td><td>Aeromech Inc.: RZZ</td></tr> <tr><td>19037</td><td>Puerto Rico Intl Airlines: PRN</td></tr> <tr><td>19039</td><td>Swift Aire Lines Inc.: SWT</td></tr> <tr><td>19041</td><td>Valdez Airlines: VEZ</td></tr> <tr><td>19043</td><td>Altair Airlines Inc.: AAR</td></tr> <tr><td>19045</td><td>Marco Island Airways Inc.: MRC</td></tr> <tr><td>19047</td><td>Sundance Airlines: PRO</td></tr> <tr><td>19049</td><td>Southeast Airlines Inc.: SLZ</td></tr> <tr><td>19051</td><td>Imperial Airlines Inc.: IMP</td></tr> <tr><td>19053</td><td>Wright Airlines Inc.: WRT</td></tr> <tr><td>19055</td><td>Mississippi Valley Airlines: MVA</td></tr> <tr><td>19057</td><td>Rocky Mountain Airways Inc.: JCZ</td></tr> <tr><td>19059</td><td>Sedalia Marshall Boonvl Stg: STG</td></tr> <tr><td>19061</td><td>Command Airways Inc.: CMD</td></tr> <tr><td>19063</td><td>Orion Air Inc.: ORI</td></tr> <tr><td>19065</td><td>Air Micronesia Inc.: AJ</td></tr> <tr><td>19067</td><td>Wings West Airlines: WWI</td></tr> <tr><td>19069</td><td>Air Cargo Express Inc.: ACG</td></tr> <tr><td>19071</td><td>Antilles Air Boats Inc.: AAB</td></tr> <tr><td>19073</td><td>Argosy Air Lines Inc.: AAL</td></tr> <tr><td>19075</td><td>Airbama Incorporated: ABI</td></tr> <tr><td>19077</td><td>Alaska Central Airways Inc.: ACA</td></tr> <tr><td>19079</td><td>AAA-Action Air Carrier Inc.: ACI</td></tr> <tr><td>19081</td><td>Air Chaparral Inc.: ACP</td></tr> <tr><td>19083</td><td>Air Central Inc. (1): ACT</td></tr> <tr><td>19085</td><td>Astec Air East Inc.: AEC</td></tr> <tr><td>19087</td><td>Air Cargo America Inc.: AGA</td></tr> <tr><td>19089</td><td>Air Express Inc.: AIE</td></tr> <tr><td>19091</td><td>Air Indiana: AIN</td></tr> <tr><td>19093</td><td>Brandt Air: ALI</td></tr> <tr><td>19095</td><td>North American Airlines Inc.: AMA</td></tr> <tr><td>19097</td><td>Air Nebraska Inc.: ANB</td></tr> <tr><td>19099</td><td>Metro Air Northeast Vermont: ANO</td></tr> </tbody> </table> <p>Hit <b>q</b> to quit scrolling through the data or use the spacebar to scroll through all of the data.</p>	airlineid	airlinename	19031	Mackey International Inc.: MAC	19033	Cochise Airlines Inc.: COC	19035	Aeromech Inc.: RZZ	19037	Puerto Rico Intl Airlines: PRN	19039	Swift Aire Lines Inc.: SWT	19041	Valdez Airlines: VEZ	19043	Altair Airlines Inc.: AAR	19045	Marco Island Airways Inc.: MRC	19047	Sundance Airlines: PRO	19049	Southeast Airlines Inc.: SLZ	19051	Imperial Airlines Inc.: IMP	19053	Wright Airlines Inc.: WRT	19055	Mississippi Valley Airlines: MVA	19057	Rocky Mountain Airways Inc.: JCZ	19059	Sedalia Marshall Boonvl Stg: STG	19061	Command Airways Inc.: CMD	19063	Orion Air Inc.: ORI	19065	Air Micronesia Inc.: AJ	19067	Wings West Airlines: WWI	19069	Air Cargo Express Inc.: ACG	19071	Antilles Air Boats Inc.: AAB	19073	Argosy Air Lines Inc.: AAL	19075	Airbama Incorporated: ABI	19077	Alaska Central Airways Inc.: ACA	19079	AAA-Action Air Carrier Inc.: ACI	19081	Air Chaparral Inc.: ACP	19083	Air Central Inc. (1): ACT	19085	Astec Air East Inc.: AEC	19087	Air Cargo America Inc.: AGA	19089	Air Express Inc.: AIE	19091	Air Indiana: AIN	19093	Brandt Air: ALI	19095	North American Airlines Inc.: AMA	19097	Air Nebraska Inc.: ANB	19099	Metro Air Northeast Vermont: ANO
airlineid	airlinename																																																																								
19031	Mackey International Inc.: MAC																																																																								
19033	Cochise Airlines Inc.: COC																																																																								
19035	Aeromech Inc.: RZZ																																																																								
19037	Puerto Rico Intl Airlines: PRN																																																																								
19039	Swift Aire Lines Inc.: SWT																																																																								
19041	Valdez Airlines: VEZ																																																																								
19043	Altair Airlines Inc.: AAR																																																																								
19045	Marco Island Airways Inc.: MRC																																																																								
19047	Sundance Airlines: PRO																																																																								
19049	Southeast Airlines Inc.: SLZ																																																																								
19051	Imperial Airlines Inc.: IMP																																																																								
19053	Wright Airlines Inc.: WRT																																																																								
19055	Mississippi Valley Airlines: MVA																																																																								
19057	Rocky Mountain Airways Inc.: JCZ																																																																								
19059	Sedalia Marshall Boonvl Stg: STG																																																																								
19061	Command Airways Inc.: CMD																																																																								
19063	Orion Air Inc.: ORI																																																																								
19065	Air Micronesia Inc.: AJ																																																																								
19067	Wings West Airlines: WWI																																																																								
19069	Air Cargo Express Inc.: ACG																																																																								
19071	Antilles Air Boats Inc.: AAB																																																																								
19073	Argosy Air Lines Inc.: AAL																																																																								
19075	Airbama Incorporated: ABI																																																																								
19077	Alaska Central Airways Inc.: ACA																																																																								
19079	AAA-Action Air Carrier Inc.: ACI																																																																								
19081	Air Chaparral Inc.: ACP																																																																								
19083	Air Central Inc. (1): ACT																																																																								
19085	Astec Air East Inc.: AEC																																																																								
19087	Air Cargo America Inc.: AGA																																																																								
19089	Air Express Inc.: AIE																																																																								
19091	Air Indiana: AIN																																																																								
19093	Brandt Air: ALI																																																																								
19095	North American Airlines Inc.: AMA																																																																								
19097	Air Nebraska Inc.: ANB																																																																								
19099	Metro Air Northeast Vermont: ANO																																																																								
	<p><b>Summary</b></p> <p>The <code>COPY</code> command can be used to load data from a file (or from standard input) into a table. <code>COPY</code> can be run as a single command in non-interactive mode or in an interactive <code>psql</code> session. <code>COPY</code> does not load data in parallel, so it is better suited for loading smaller amounts of data. If loading large amounts of data, external tables offer better load performance.</p> <p>If your data load file contains mixed data, you can use <code>COPY</code> to quickly load the data into a temporary table and then insert it from there through the parent table.</p>																																																																								

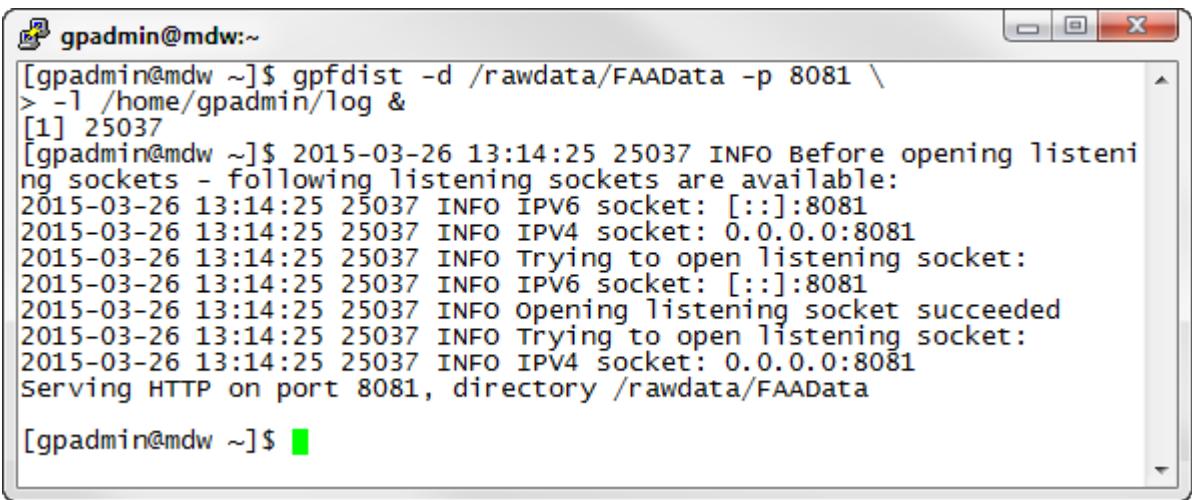
## Lab 12: Task 2 – Create a Dimension Table and Load Data Using External Tables

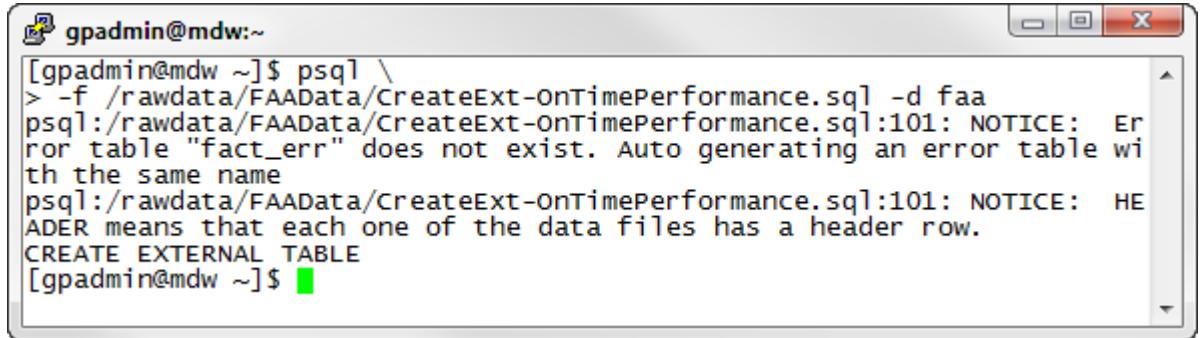
Step	Action
1.	<p>Define an external table called public.airport_external that points to a CSV file located on another server. The file can be accessed as file://sdw1/loaddata/DimAIRPORTS.csv. To create the table, use the CREATE EXTERNAL TABLE command:</p> <pre>faa=# CREATE EXTERNAL TABLE public.airport_external     (AirportID Character(3),      AirportDescription Character varying(85))    LOCATION ('file://sdw1/loaddata/DimAIRPORTS.csv')   FORMAT 'CSV' (HEADER)  LOG ERRORS INTO public.airport_err SEGMENT REJECT LIMIT 10 ROWS;</pre>  <p>Note: When using the file:// protocol, the external data file(s) must reside on a segment host in a location accessible by the Greenplum super user (gpadmin).</p>
2.	<p>Verify that you are able to access data from the external table by counting the number of records:</p> <pre>faa=# SELECT COUNT(*) FROM public.airport_external;</pre>  <p>If you receive an error, verify the syntax you used to create the external table, the host name for the file, and the file name. Verify that the file is in the location you specified in the external table syntax. If necessary, drop the external table using <code>DROP EXTERNAL TABLE public.airport_external</code> and recreate the external table again.</p>

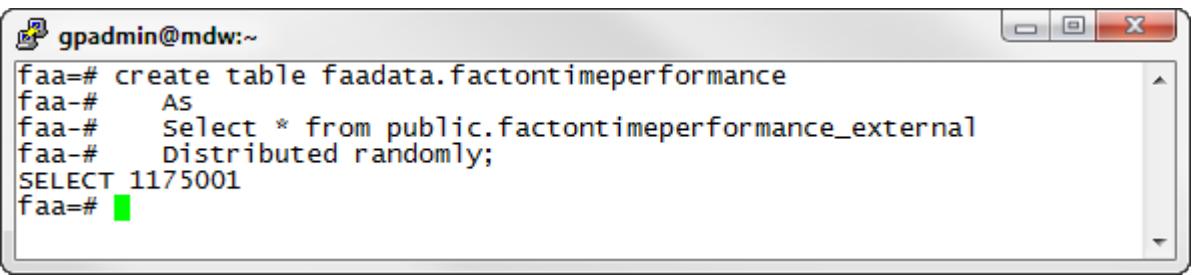
Step	Action
3.	<p>Create a standard table to copy the external data into:</p> <pre>faa=# CREATE TABLE faadata.DimAirport(     AirportID Character(3),     AirportDescription Character varying(85)) WITH (OIDS=FALSE) DISTRIBUTED BY (AirportID);</pre>  <p><b>Note:</b> You can also copy and paste this command from the DDLScript.sql script.</p>
4.	<p>Insert the data into the DimAirport from the external table:</p> <pre>faa=# INSERT INTO faadata.DimAirport SELECT * FROM public.airport_external;</pre> 
5.	<p>Verify that the dimension table now contains data:</p> <pre>faa=# SELECT * FROM faadata.DimAirport LIMIT 10;</pre>  <p><b>Note:</b> The LIMIT clause returns the specified number of rows to the user, reducing the amount of memory required by the client to handle the number of rows returned by Greenplum. This can be helpful if your query returns a large number of rows but you wish to only view a subset of these rows.</p>

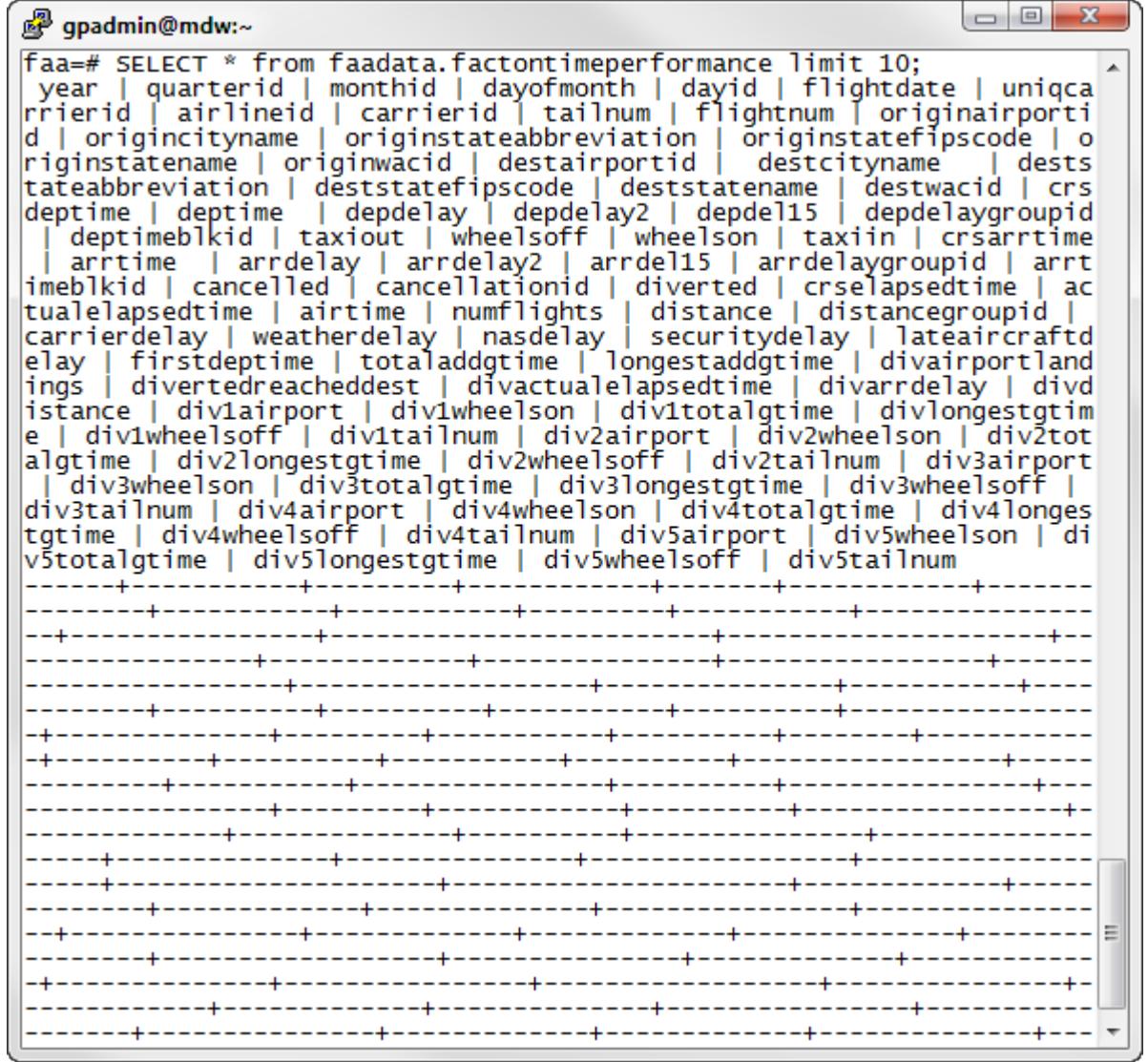
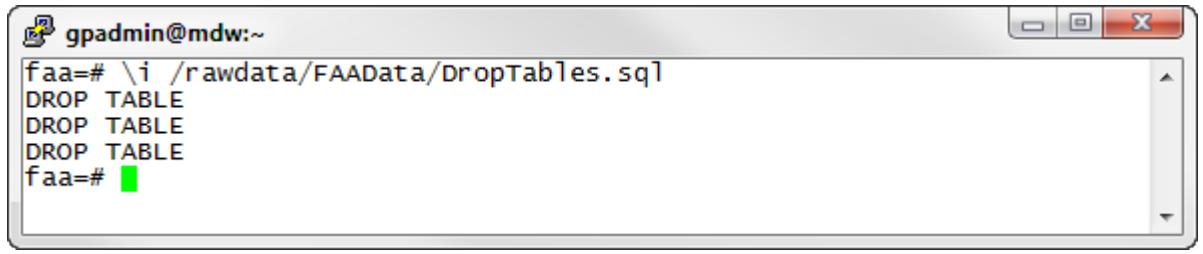
Step	Action
6.	Exit the database before proceeding.
	<p><b>Summary</b></p> <p>Both <code>CREATE EXTERNAL TABLE</code> and <code>COPY</code> operations can be run using the single row error isolation feature. This feature allows you to load good rows while filtering out error rows. Error rows can be logged in to an error table for further examination. External tables offer additional flexibility over <code>COPY</code> because you can use regular SQL commands to select and move the data prior to inserting it into its final destination tables. This is useful for ETL processing common in data warehousing applications.</p> <p>It is good practice to automatically run <code>ANALYZE</code> after any data load so that the query planner has the most up-to-date statistics. If you had any errors during your data loads, it is also a good idea to run <code>VACUUM</code> to reclaim any wasted space.</p>

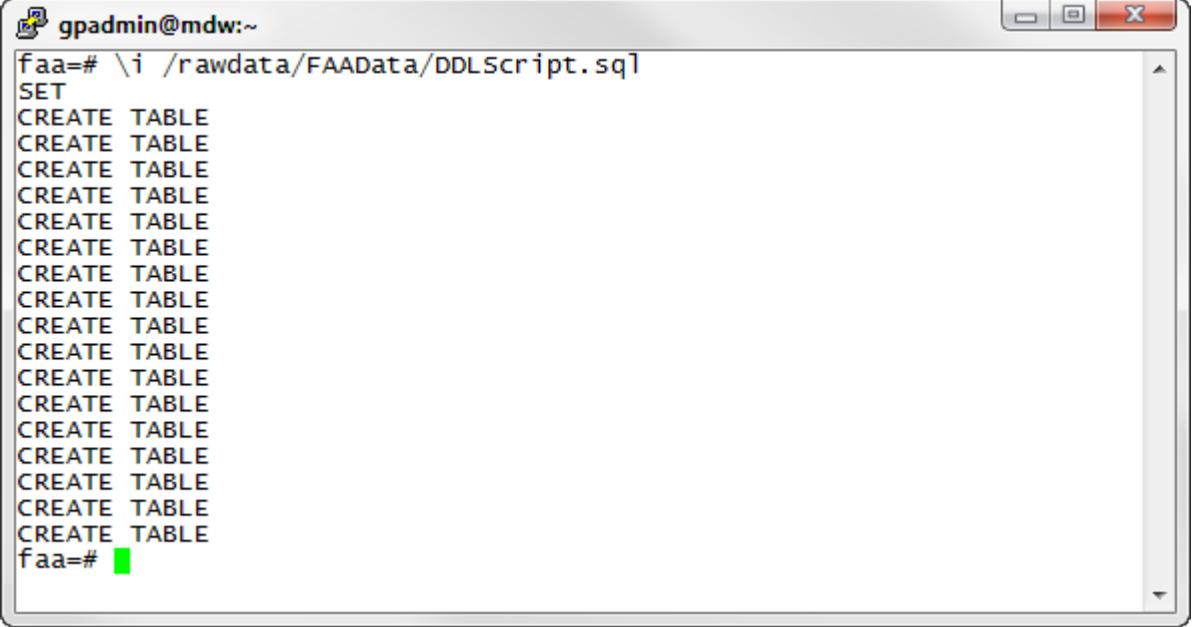
## Lab 12: Task 3 – Create a Fact Table and Load Data using gpfdist

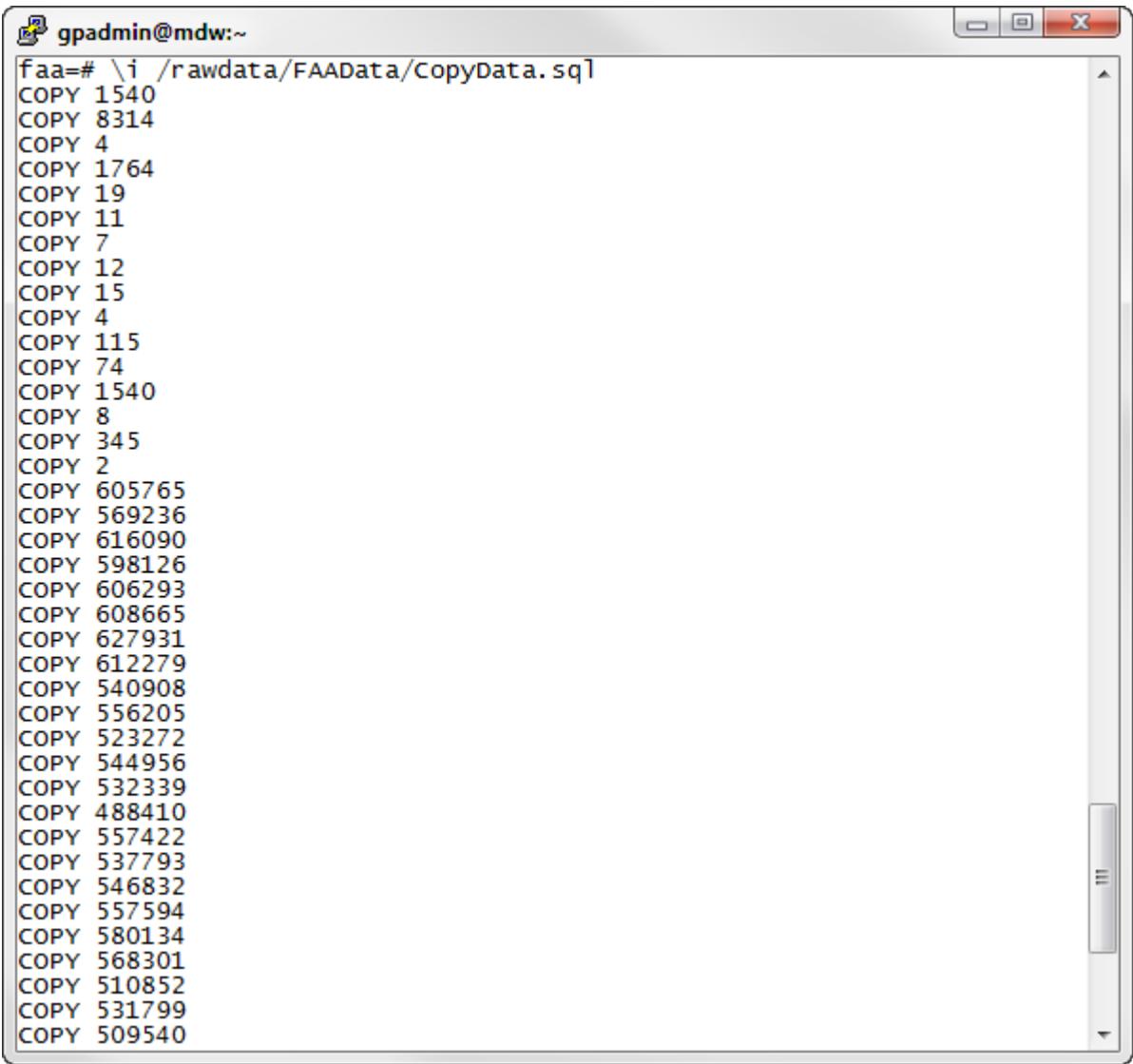
Step	Action
1.	<p>If not already connected to the master server, open a terminal session to the master server and connect as gpadmin.</p>
2.	<p>Start a gpfdist session.</p> <pre>[gpadmin@mdw ~]\$ gpfdist -d /rawdata/FAAData -p 8081 \ -l /home/gpadmin/log &amp;</pre>  <p>This terminal window shows the command being run and its output. The output includes logs from March 26, 2015, at 13:14:25, detailing the opening of listening sockets on port 8081. It shows both IPv6 and IPv4 socket attempts, with the message "Opening listening socket succeeded". The final line indicates "Serving HTTP on port 8081, directory /rawdata/FAAData".</p> <p>This starts a parallel file distribution session on port 8081 for files in the directory, /rawdata/FAAData. These files can now be read into the database from an external table.</p>
3.	<p>Verify that gpfdist is running:</p> <pre>[gpadmin@mdw ~]\$ pgrep gpfdist</pre>  <p>This terminal window shows the pgrep command being run to find processes named gpfdist. The output shows a single process ID: 25037.</p>
4.	<p>Examine the contents of the /rawdata/FAAData/CreateExt-OnTimePerformance.sql file and go to the end of it. You will not be making any changes to this file.</p>

Step	Action
	<p>This file will create an external table as shown below.</p> <pre data-bbox="306 185 1383 481">CREATE EXTERNAL TABLE public.FactOnTimePerformance_external ( ) LOCATION ('gpfdist://mdw:8081/On_Time_On_Time_Performance_2008_1.csv',  'gpfdist://mdw:8081/On_Time_On_Time_Performance_2008_2.csv') FORMAT 'CSV' (HEADER DELIMITER ',') LOG ERRORS INTO public.fact_err SEGMENT REJECT LIMIT 10 ROWS;</pre>
5.	<p>Execute the script as shown.</p> <pre data-bbox="306 572 1334 629">[gpadmin@mdw ~]\$ psql \ -f /rawdata/FAAData/CreateExt-OnTimePerformance.sql -d faa</pre>  <p>The output highlights that the <code>fact_err</code> table did not previously exist. One will be created to capture any failures. The last portion of the output highlights that a header is expected of all files loaded through this external table.</p>
6.	<p>Connect to the <code>faa</code> database.</p> <pre data-bbox="306 1157 752 1193">[gpadmin@mdw ~]\$ psql faa</pre> 

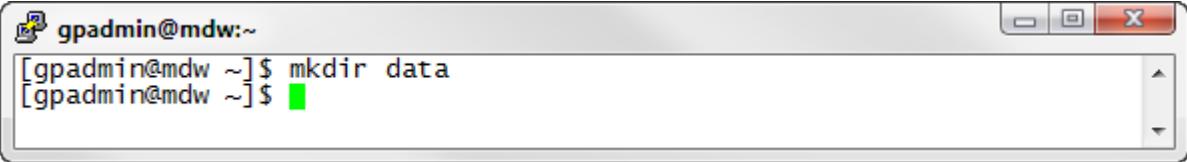
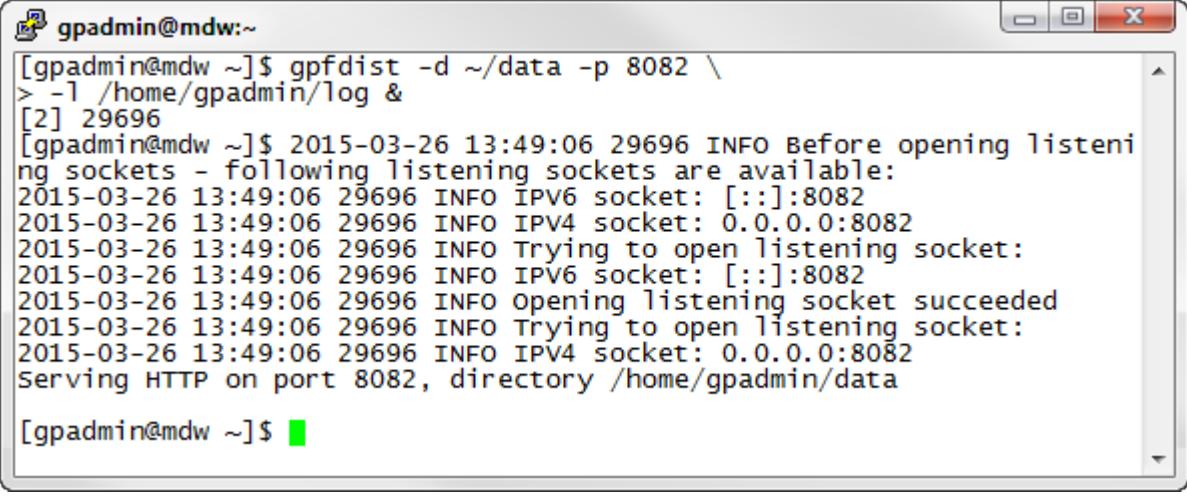
Step	Action
7.	<p>Confirm that the contents of the files can now be accessed through the external table created:</p> <pre>faa=# SELECT * from public.factontimeperformance_external limit 10;</pre> 
8.	<p>Create a regular heap table based on the external table and populate it with the data from the external table:</p> <pre>faa=# create table faadata.factontimeperformance       As       Select * from public.factontimeperformance_external       Distributed randomly;</pre> 

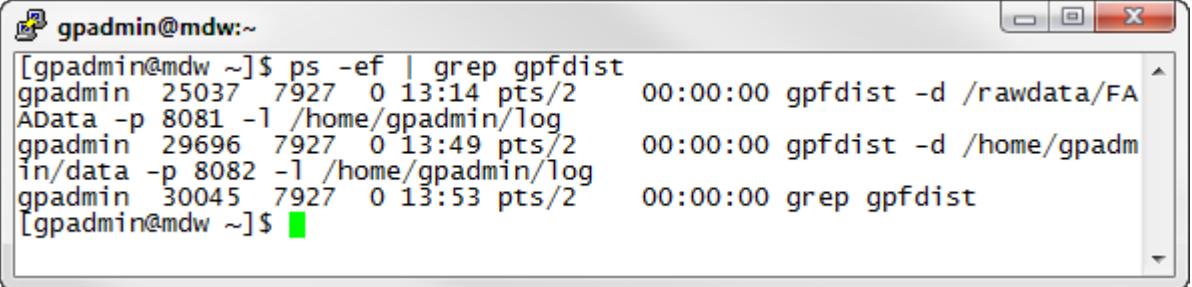
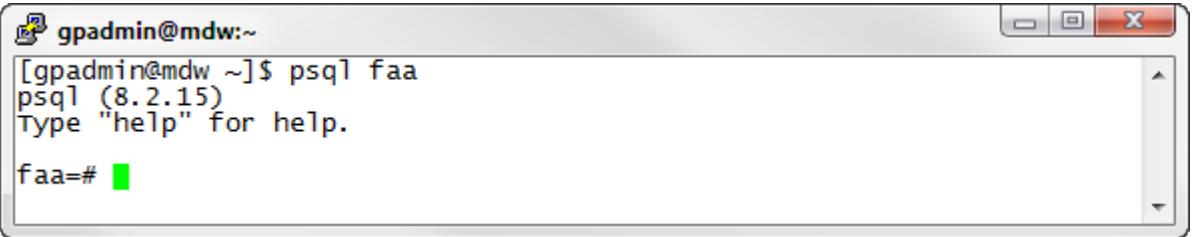
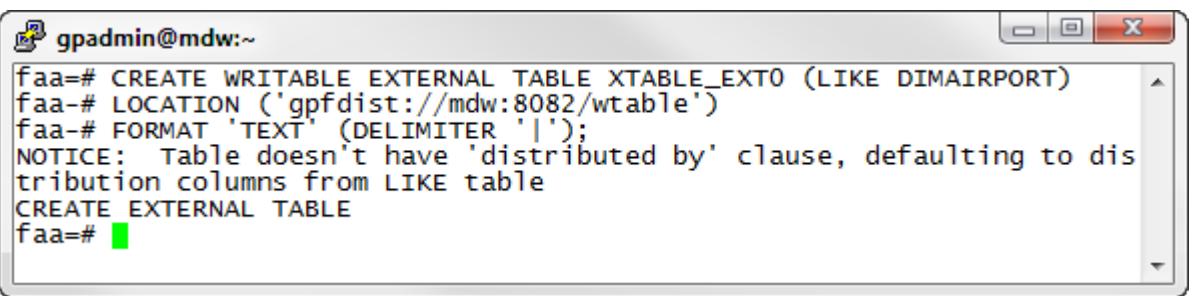
Step	Action
9.	<p>Confirm that the faadata.factontimeperformance table has been created and populated:</p> <pre>faa=# SELECT * from faadata.factontimeperformance limit 10;</pre>  <p>The LIMIT 10 clause lists only 10 of the possible rows.</p>
10.	<p>Execute the /rawdata/FAAData/DropTables.sql script to drop the tables created. This script will drop the dimairline, dimairport, and factontimeperformance tables.</p> <pre>faa=# \i /rawdata/FAAData/DropTables.sql</pre> 

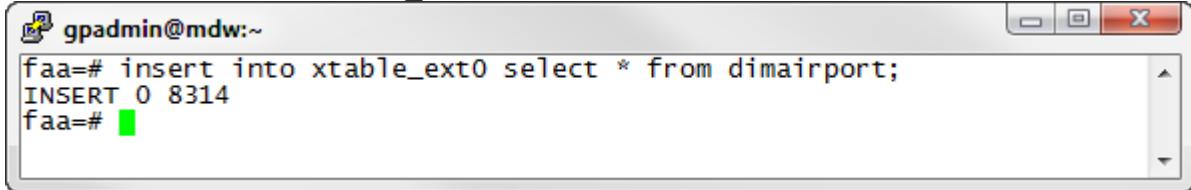
Step	Action
11.	<p>The next two steps will create and populate the tables that will be used in future labs.</p> <p>Create the tables by executing the script DDLScript.sql.</p> <pre>faa=# \i /rawdata/FAAData/DDLScript.sql</pre> 

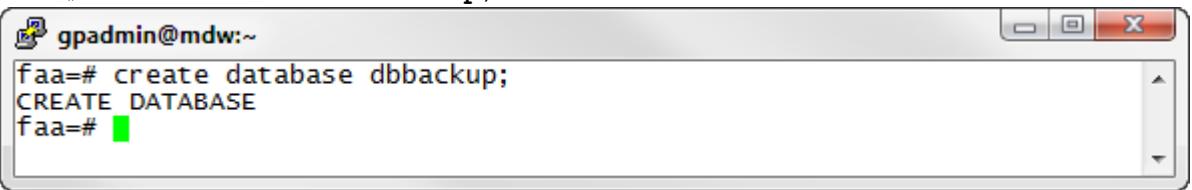
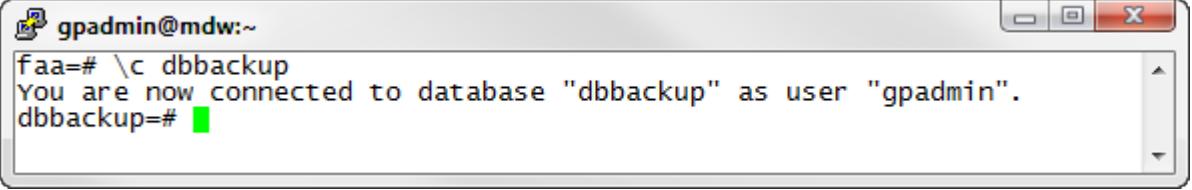
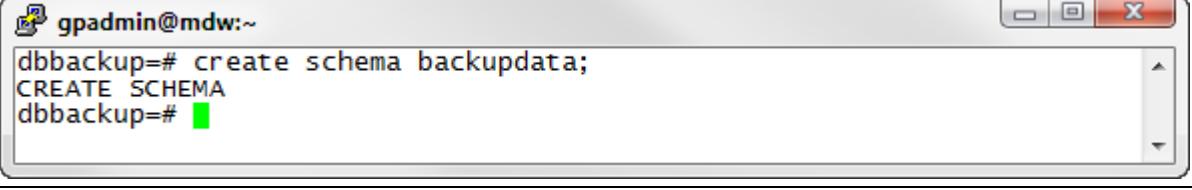
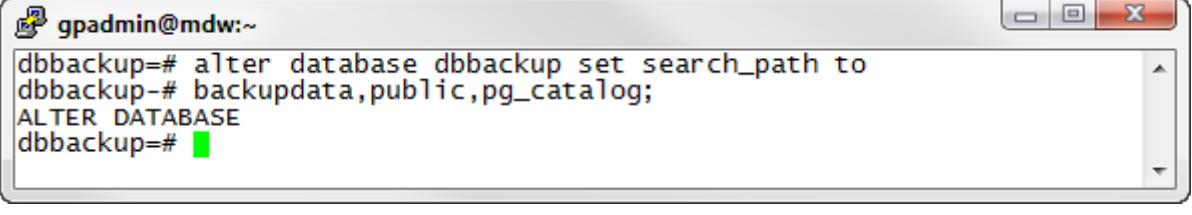
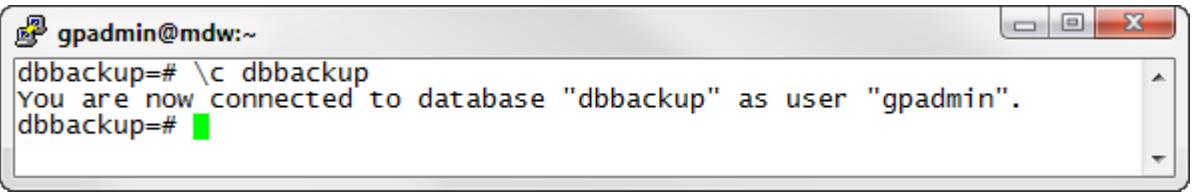
Step	Action
12.	<p>Load data for the tables you just created in the faa schema by executing the script, CopyData.sql. The data is approximately a total of 20M rows and may take 5-10 minutes to populate.</p> <pre data-bbox="311 276 980 308">faa=# \i /rawdata/FAAData/CopyData.sql</pre>  <pre data-bbox="311 392 943 1438"> gpadmin@mdw:~ faa=# \i /rawdata/FAAData/CopyData.sql COPY 1540 COPY 8314 COPY 4 COPY 1764 COPY 19 COPY 11 COPY 7 COPY 12 COPY 15 COPY 4 COPY 115 COPY 74 COPY 1540 COPY 8 COPY 345 COPY 2 COPY 605765 COPY 569236 COPY 616090 COPY 598126 COPY 606293 COPY 608665 COPY 627931 COPY 612279 COPY 540908 COPY 556205 COPY 523272 COPY 544956 COPY 532339 COPY 488410 COPY 557422 COPY 537793 COPY 546832 COPY 557594 COPY 580134 COPY 568301 COPY 510852 COPY 531799 COPY 509540 </pre>
13.	Exit the database.
	<p><b>Summary</b></p> <p>The parallel file distribution program can be used in conjunction with gpload or other ETL tools to take advantage of the parallel load. It starts a light-weight session on the port specified and can be used to load multiple files from the same directory.</p> <p>You can start multiple gpfdist processes to maximize the transfer of data.</p>

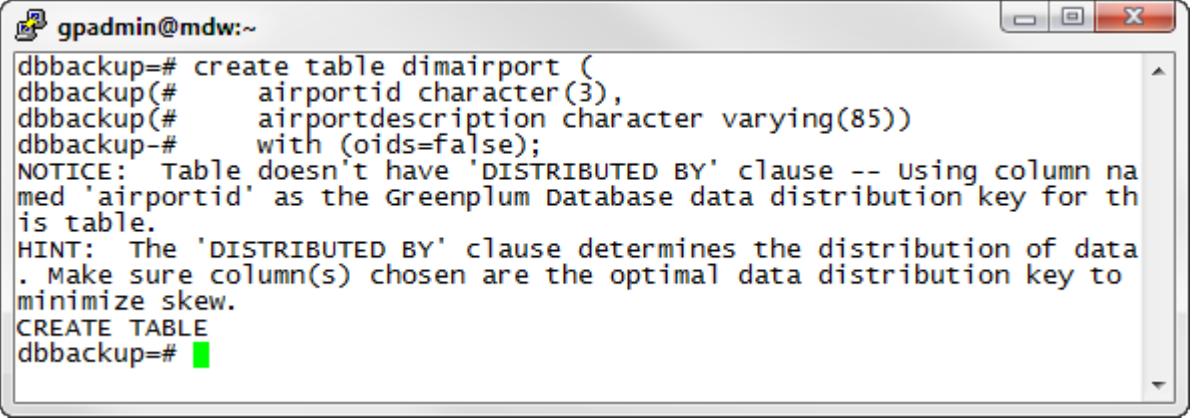
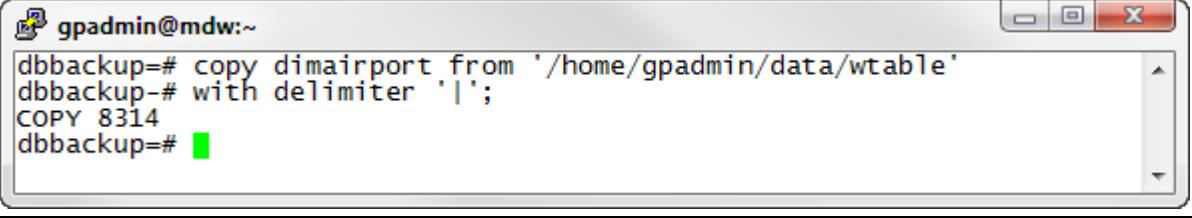
## Lab 12: Task 4 – Create a Writeable External Table to Load Data into a Different Database using gpfdist

Step	Action
1.	<p>If not already connected to the master server, open a terminal session to the master server and connect as gpadmin.</p>
2.	<p>Create a directory called data in your home directory that you will use in this lab. This directory will be used by writable tables to create files with data pushed from the database.</p> <pre>[gpadmin@mdw ~]\$ mkdir data</pre> 
3.	<p>Start a gpfdist session that will access files in the ~/data directory you created. Start this gpfdist session on port 8082. You must have write permission on the directory you are using for this session.</p> <pre>[gpadmin@mdw ~]\$ gpfdist -d ~/data -p 8082 \ &gt; -l /home/gpadmin/log &amp;</pre>  <p>This starts a parallel file distribution session on port 8082 for files in the directory, /home/gpadmin/data.</p>

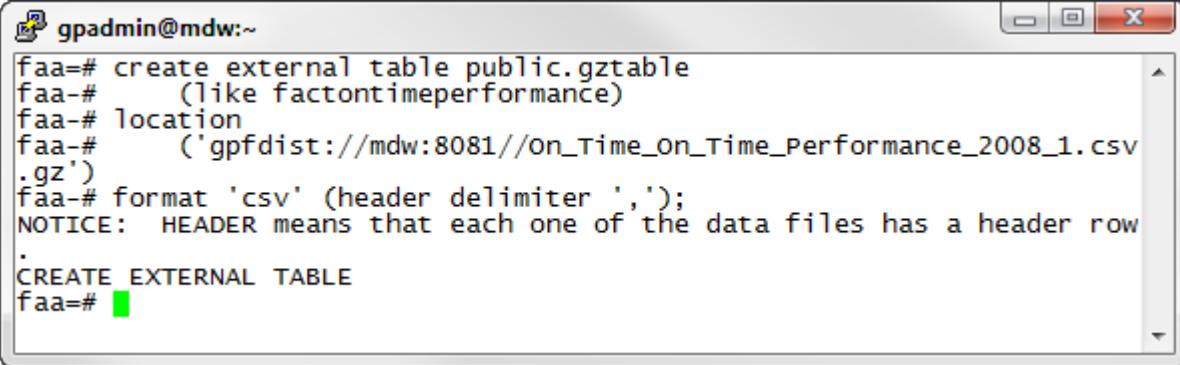
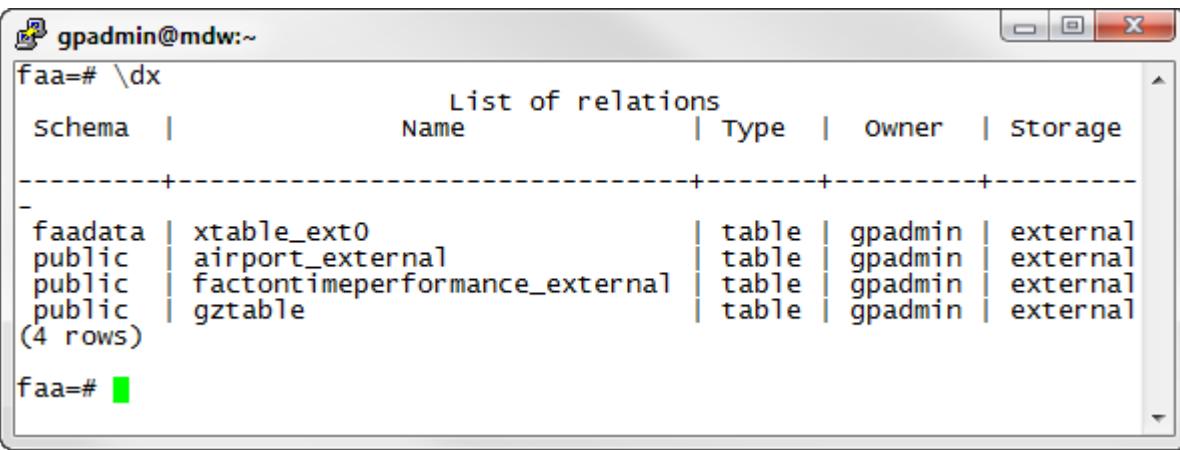
Step	Action
4.	<p>Verify that gpfdist is running for the newly created session:</p> <pre>[gpadmin@mdw ~]\$ ps -ef   grep gpfdist</pre>  <p>You may see a second gpfdist session running for port 8081 created earlier in the lab.</p>
5.	<p>Connect to the faa database as the gpadmin user.</p> <pre>[gpadmin@mdw ~]\$ psql faa</pre> 
6.	<p>Create a writable external called XTABLE_EXT0. The table will be a copy of the structure of the dimairport table. The table should point to the wtable file through the gpfdist session running on port 8082.</p> <pre>faa=# CREATE WRITABLE EXTERNAL TABLE XTABLE_EXT0 (LIKE DIMAIRPORT) LOCATION ('gpfdist://mdw:8082/wtable') FORMAT 'TEXT' (DELIMITER ' ');</pre>  <p>The gpfdist process is pointing to /home/gpadmin/data on the master server, mdw. This writeable table is therefore pointing to the file, /home/gpadmin/data/wtable. Data with multiple columns will be separated with the pipe ( ) symbol.</p>

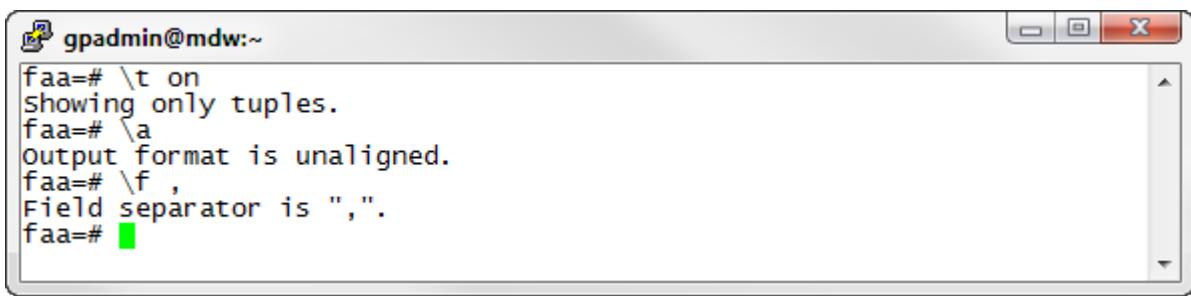
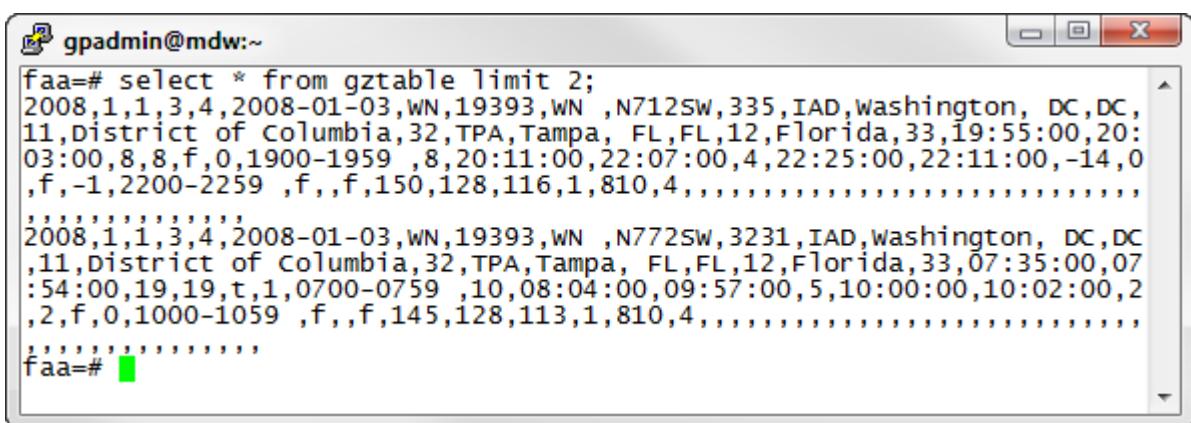
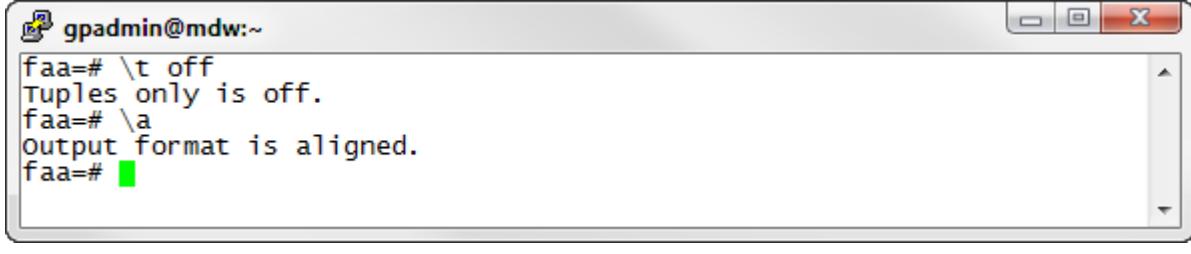
Step	Action
7.	<p>Populate the wtable file with content from the dimairport table. This is known as unloading data. It does not, however, remove data from the dimairport table.</p> <pre data-bbox="251 255 1148 285">faa=# insert into xtable_ext0 select * from dimairport;</pre> 
8.	<p>Verify that the contents of the file have been populated with the data contained in the dimairport table. Use the escape shell (!) to execute the more command for the /home/gpadmin/data/wtable file.</p> <pre data-bbox="251 599 948 629">faa=# \! more /home/gpadmin/data/wtable</pre>  <p>Hit q to exit.</p>

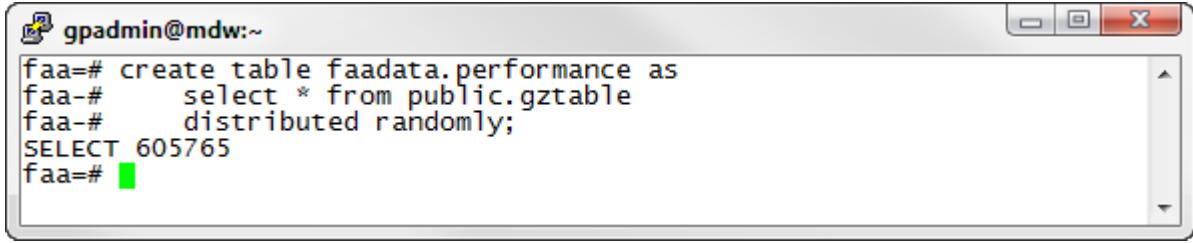
Step	Action
9.	<p>Create a new database called dbbackup.</p> <pre data-bbox="257 181 752 213">faa=# create database dbbackup;</pre> 
10.	<p>Connect to the dbbackup database.</p> <pre data-bbox="257 456 548 487">faa=# \c dbbackup</pre> 
11.	<p>Create a new schema for the dbbackup database called backupdata.</p> <pre data-bbox="257 730 833 762">dbbackup=# create schema backupdata;</pre> 
12.	<p>Change the search_path parameter on the dbbackup database so that the backupdata schema is the default schema.</p> <pre data-bbox="257 1068 1192 1142">dbbackup=# alter database dbbackup set search_path to backupdata,public,pg_catalog;</pre> 
13.	<p>Reconnect to the dbbackup database. This step is necessary for reloading the new search_path parameter.</p> <pre data-bbox="257 1533 662 1564">dbbackup=# \c dbbackup</pre> 

Step	Action
14.	<p>Create dimairport table in the dbbackup database using the following syntax:</p> <pre>dbbackup=# create table dimairport (     airportid character(3),     airportdescription character varying(85)) with (oids=false);</pre> 
15.	<p>Populate the dimairport table with the content from the /home/gpadmin/data/wtable file.</p> <pre>dbbackup=# copy dimairport from '/home/gpadmin/data/wtable' with delimiter ' ';</pre> 
	<p><b>Summary</b></p> <p>Writable external tables are used to select rows from database tables and output the rows to files, named pipes, or to other executable programs. You could unload data from a Greenplum Database table and send that data to a text file. Once the text file is created, you can use it to populate another Greenplum Database table. This technique, shown in the exercise you just completed, is an example of migrating data from one database to another. If necessary, you can perform transforms on the data when inserting it into the database.</p> <p>Writable external tables only allow INSERT operations.</p>

## Lab 12: Task 5 – Loading a Compressed File using gpfdist

Step	Action																														
1.	<p>Connect to the faa database as gpadmin.</p> <pre>dbbackup=# \c faa</pre>																														
2.	<p>Create an external table called public.gztable based on the DDL of the factontimeperformance table. Populate public.gztable with the contents from a compressed file without first decompressing it.</p> <pre>faa=# create external table public.gztable       (like factontimeperformance) location       ('gpfdist://mdw:8081//On_Time_On_Time_Performance_2008_1.csv.gz') format 'csv' (header delimiter ',');</pre> 																														
3.	<p>Verify the external table has been created listing external tables with the \dx command.</p> <pre>faa=# \dx</pre>  <table border="1"> <thead> <tr> <th style="text-align: left;">Schema</th> <th style="text-align: left;">Name</th> <th style="text-align: left;">Type</th> <th style="text-align: left;">Owner</th> <th style="text-align: left;">Storage</th> </tr> </thead> <tbody> <tr> <td>faadata</td> <td>xtable_ext0</td> <td>table</td> <td>gpadmin</td> <td>external</td> </tr> <tr> <td>public</td> <td>airport_external</td> <td>table</td> <td>gpadmin</td> <td>external</td> </tr> <tr> <td>public</td> <td>factontimeperformance_external</td> <td>table</td> <td>gpadmin</td> <td>external</td> </tr> <tr> <td>public</td> <td>gztable</td> <td>table</td> <td>gpadmin</td> <td>external</td> </tr> <tr> <td colspan="5">(4 rows)</td></tr> </tbody> </table>	Schema	Name	Type	Owner	Storage	faadata	xtable_ext0	table	gpadmin	external	public	airport_external	table	gpadmin	external	public	factontimeperformance_external	table	gpadmin	external	public	gztable	table	gpadmin	external	(4 rows)				
Schema	Name	Type	Owner	Storage																											
faadata	xtable_ext0	table	gpadmin	external																											
public	airport_external	table	gpadmin	external																											
public	factontimeperformance_external	table	gpadmin	external																											
public	gztable	table	gpadmin	external																											
(4 rows)																															

Step	Action
4.	<p>To make the output easier to see, complete the following to change the output mode in PSQL:</p> <ul style="list-style-type: none"> <li>Issue the following command to eliminate the header from the output: faa=# \t on</li> <li>Issue the following command to change to unaligned output. The default separator for unaligned output is the pipe ( ) symbol: faa=# \a</li> <li>Issue the following command to change the field separator to a comma (,): faa=# \f ,</li> </ul>  <pre>gpadmin@mdw:~\$ faa=# \t on Showing only tuples. faa=# \a output format is unaligned. faa=# \f , Field separator is ",". faa=# </pre>
5.	<p>List two records from this external table as shown.</p> <pre>faa=# select * from gztable limit 2;</pre>  <pre>gpadmin@mdw:~\$ faa=# select * from gztable limit 2; 2008,1,1,3,4,2008-01-03,WN,19393,WN ,N712SW,335,IAD,Washington, DC,DC, 11,District of Columbia,32,TPA,Tampa, FL,FL,12,Florida,33,19:55:00,20: 03:00,8,8,f,0,1900-1959 ,8,20:11:00,22:07:00,4,22:25:00,22:11:00,-14,0 ,f,-1,2200-2259 ,f,,f,150,128,116,1,810,4,,,,,,,,,,, ,,,,,,,,,,, 2008,1,1,3,4,2008-01-03,WN,19393,WN ,N772SW,3231,IAD,Washington, DC,DC ,11,District of Columbia,32,TPA,Tampa, FL,FL,12,Florida,33,07:35:00,07 :54:00,19,19,t,1,0700-0759 ,10,08:04:00,09:57:00,5,10:00:00,10:02:00,2 ,2,f,0,1000-1059 ,f,,f,145,128,113,1,810,4,,,,,,,,,,, faa=# </pre>
6.	<p>Reset your output by issuing the following PSQL meta commands:</p> <pre>faa=# \t off faa=# \a</pre>  <pre>gpadmin@mdw:~\$ faa=# \t off Tuples only is off. faa=# \a output format is aligned. faa=# </pre>

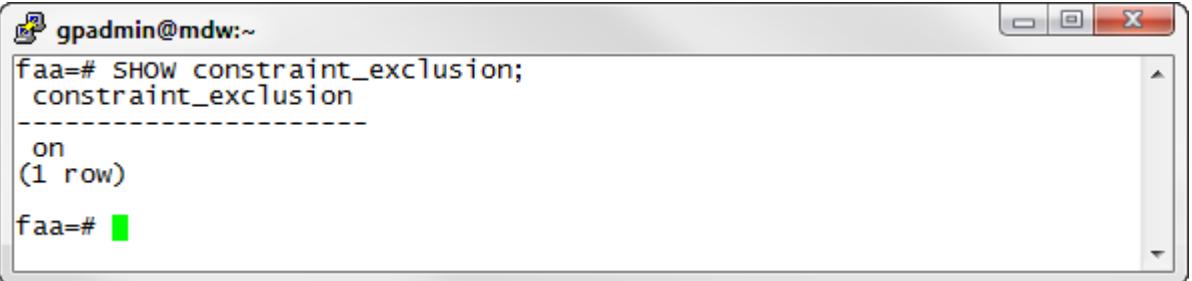
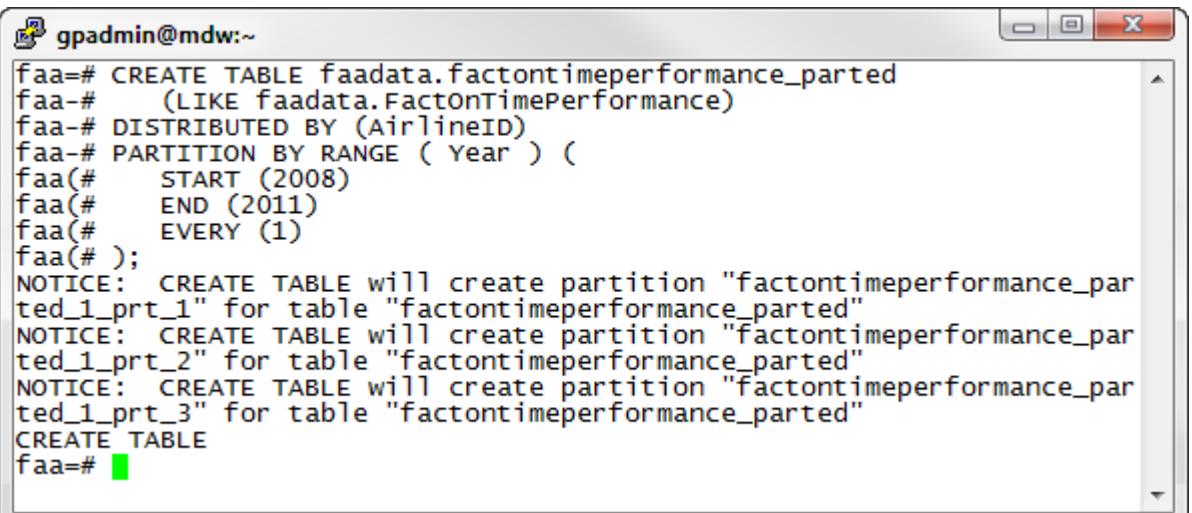
Step	Action
7.	<p>Create a new table called faadata.performance and populate it using the data from the external table, public.gztable.</p> <pre data-bbox="251 270 979 375">faa=# create table faadata.performance as       select * from public.gztable       distributed randomly;</pre> 
8.	Exit the database.
	<p><b>Summary</b>  Data in files compressed with gzip or bzip2, can be loaded into Greenplum Database tables without uncompressing the files. This saves storage space on the host serving these files.</p>

*End of Lab Exercise*

## Lab 13. Table Partitioning

<b>Purpose:</b>	The purpose of this lab is to learn about Greenplum's partitioning capabilities. Partitioning is typically used on large fact tables to improve performance and manageability. Greenplum supports single level range and list partitioning. You will learn how to create and manage partitioned tables.
<b>Tasks:</b>	Students perform the following task: create and manage table partitions.
<b>References:</b>	Module 5 – Data Loading and Distribution <ul style="list-style-type: none"><li>• Lesson: Table Partitioning</li></ul>

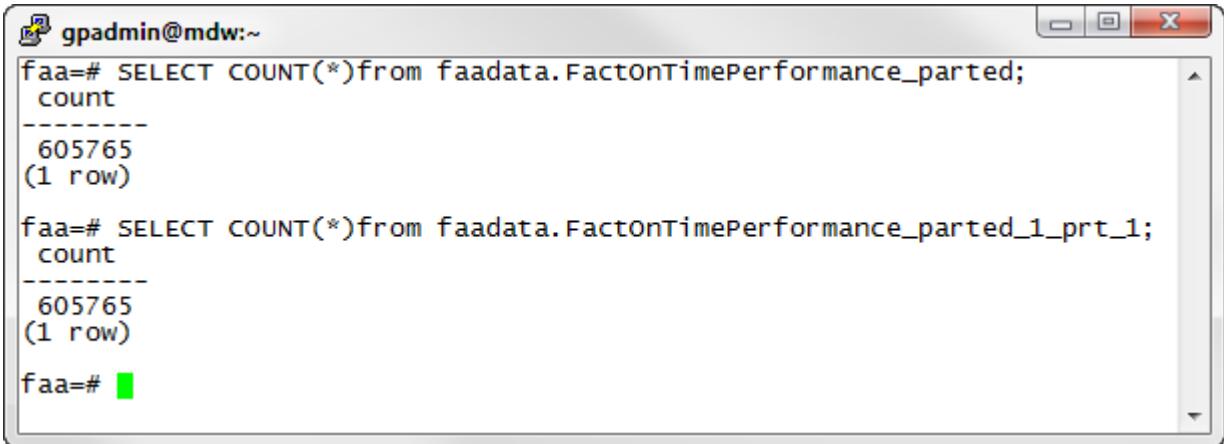
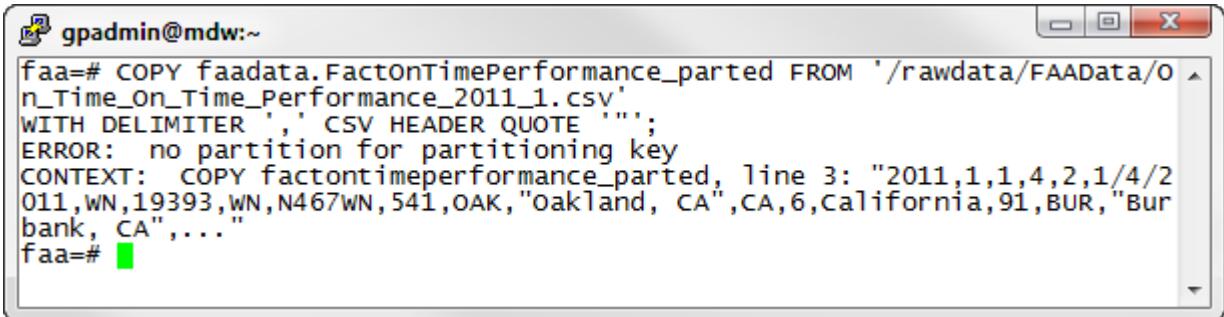
## Lab 13: Task – Create and Manage Table Partitions

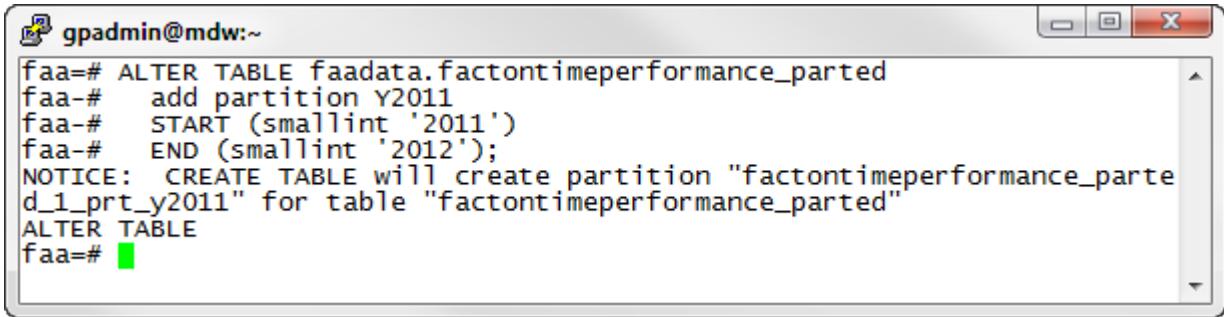
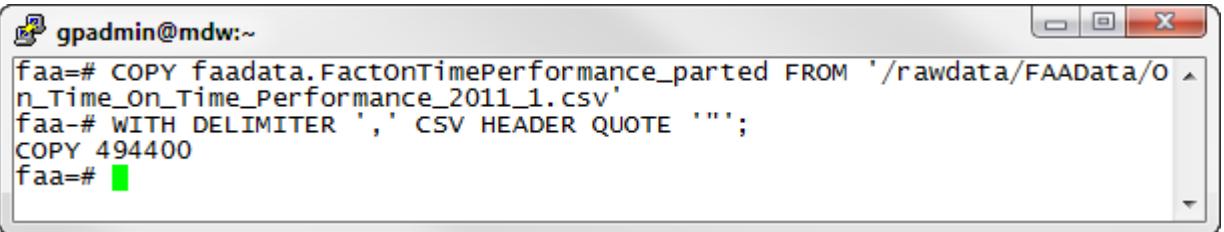
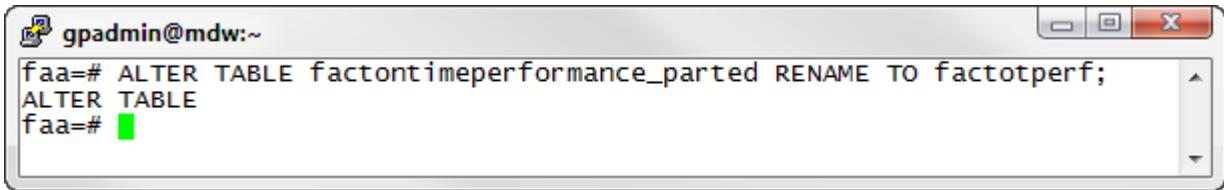
Step	Action
1.	<p>Connect to the faa database as the gpadmin user, if not already connected:</p> <pre>[gpadmin@mdw ~]\$ psql faa</pre>
2.	<p>Make sure that the partitioned tables feature is turned on. This allows the query planner to selectively scan table partitions at runtime. Confirm that constraint_exclusion has a value of <b>on</b>:</p> <pre>faa=# SHOW constraint_exclusion;</pre>  <pre>gpadmin@mdw:~\$ faa=# SHOW constraint_exclusion; constraint_exclusion ----- on (1 row)  faa=#</pre>
3.	<p>Create a partitioned version of the fact table. Each month will be stored in a separate partition. The partitioned table will be based on the faadata.FactOnTimePerformance table created in an earlier lab.</p> <pre>faa=# CREATE TABLE faadata.factontimeperformance_parted       (LIKE faadata.FactOnTimePerformance) DISTRIBUTED BY (AirlineID) PARTITION BY RANGE ( Year ) (     START (2008)     END (2011)     EVERY (1) );</pre>  <pre>gpadmin@mdw:~\$ faa=# CREATE TABLE faadata.factontimeperformance_parted faa-#   (LIKE faadata.FactOnTimePerformance) faa-# DISTRIBUTED BY (AirlineID) faa-# PARTITION BY RANGE ( Year ) ( faa(#   START (2008) faa(#   END (2011) faa(#   EVERY (1) faa(# ); NOTICE: CREATE TABLE will create partition "factontimeperformance_parted_1_prt_1" for table "factontimeperformance_parted" NOTICE: CREATE TABLE will create partition "factontimeperformance_parted_1_prt_2" for table "factontimeperformance_parted" NOTICE: CREATE TABLE will create partition "factontimeperformance_parted_1_prt_3" for table "factontimeperformance_parted" CREATE TABLE faa=#</pre>

Step	Action
4.	<p>List the tables:</p> <pre>faa=# \dt</pre> <pre>gpadmin@mdw:~ faa=# \dt               List of relations  Schema             Name             Type   Owner   Storage -----+---------------------+-----+-----+-----  faadata   dimairline            table   gpadmin   heap  faadata   dimairport            table   gpadmin   heap  faadata   dimcancellation       table   gpadmin   heap  faadata   dimcarrier_history    table   gpadmin   heap  faadata   dimday                 table   gpadmin   heap  faadata   dimdepart_arrive_block   table   gpadmin   heap  faadata   dimdistance_group     table   gpadmin   heap  faadata   dimdiversions          table   gpadmin   heap  faadata   dimmonth                table   gpadmin   heap  faadata   dimontime_delay_groups   table   gpadmin   heap  faadata   dimquarter              table   gpadmin   heap  faadata   dimstate_abr            table   gpadmin   heap  faadata   dimstate_fips           table   gpadmin   heap  faadata   dimunique_carriers      table   gpadmin   heap  faadata   dimworld_area_code      table   gpadmin   heap  faadata   dimyesno_resp            table   gpadmin   heap  faadata   factontimeperformance   table   gpadmin   heap  faadata   factontimeperformance_parted   table   gpadmin   heap  faadata   factontimeperformance_parted_1_prt_1   table   gpadmin   heap  faadata   factontimeperformance_parted_1_prt_2   table   gpadmin   heap  faadata   factontimeperformance_parted_1_prt_3   table   gpadmin   heap  faadata   performance               table   gpadmin   heap  faadata   test_table                table   gpadmin   heap  faadata   test_table2               table   gpadmin   heap  public    airport_err              table   gpadmin   heap  public    fact_err                  table   gpadmin   heap (26 rows)  faa#</pre>

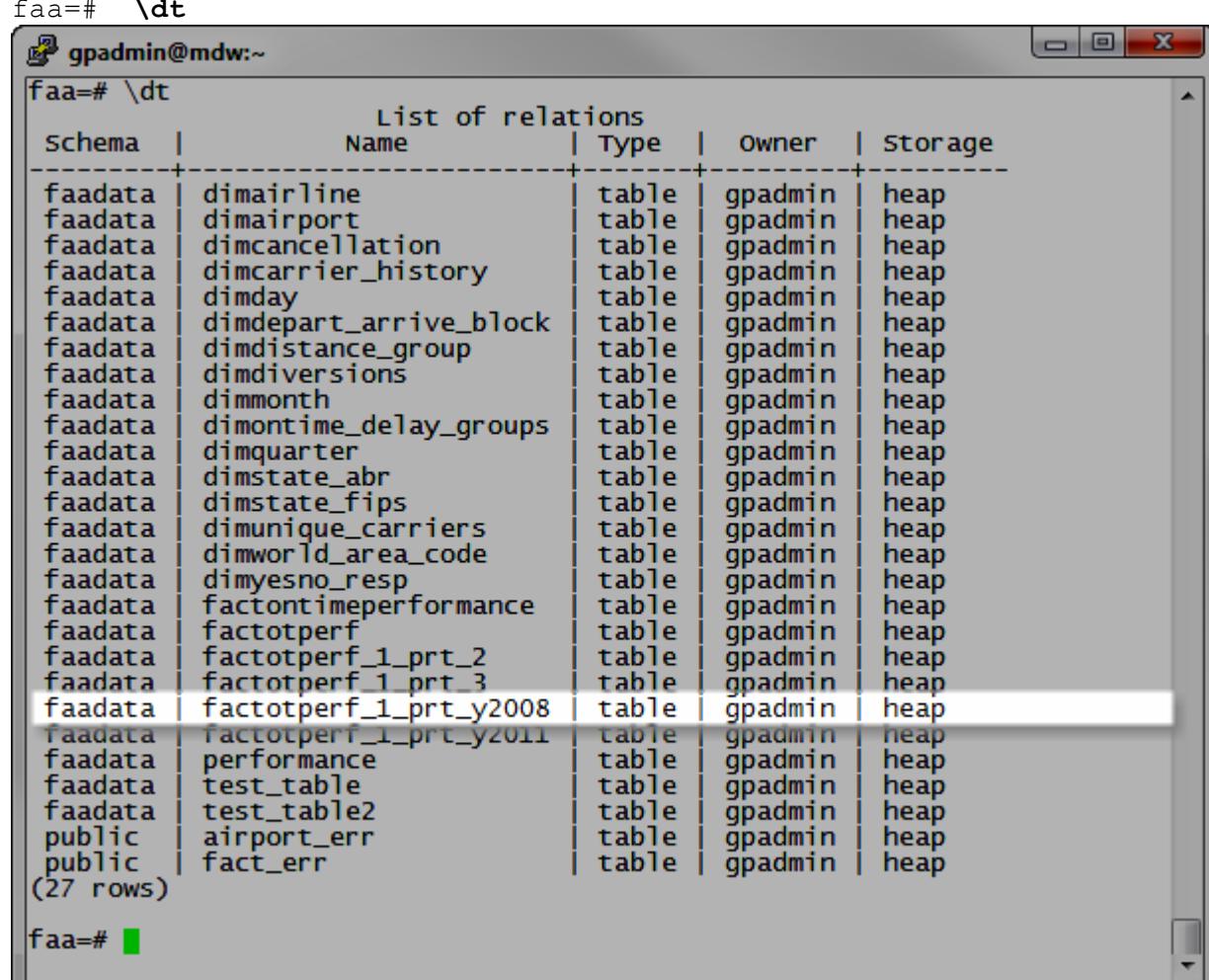
You should note that the `faadata.factontimeperformance_parted` table was created along with 3 child tables.

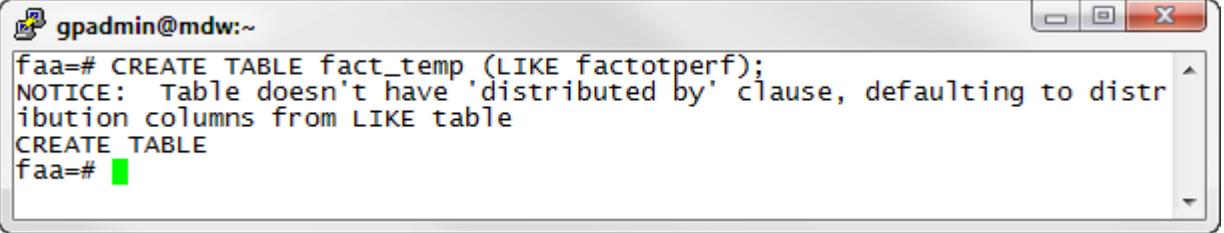
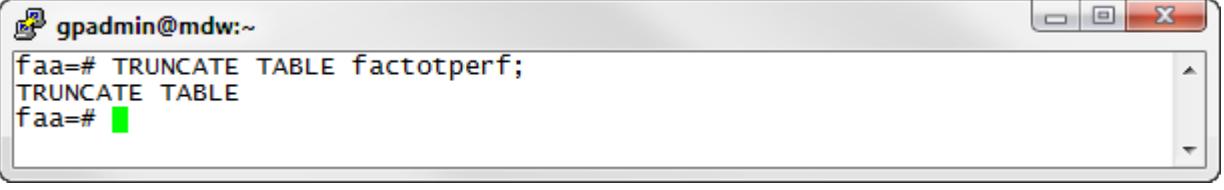
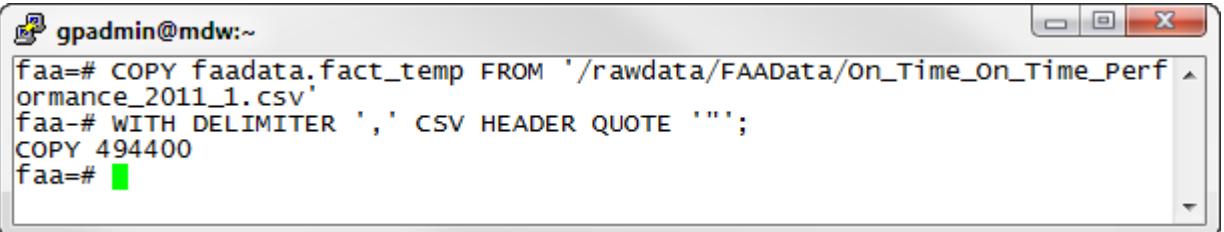
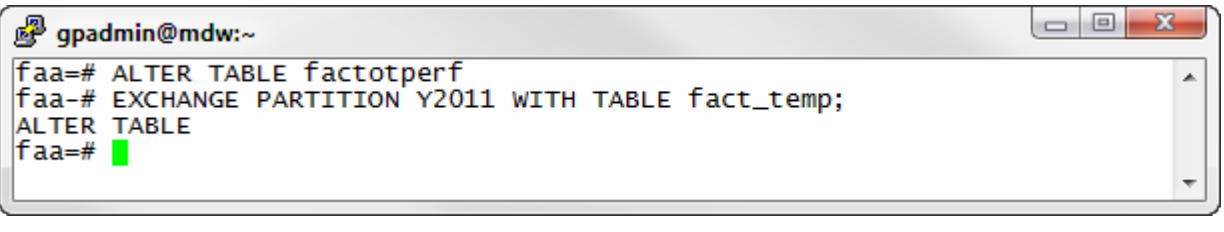
Step	Action
5.	<p>Check the definition of one of the child tables:</p> <pre>faa=# \d+ faadata.factontimeperformance_parted_1_prt_1</pre> <pre>gpadmin@mdw:~</pre> <pre>Table "faadata.factontimeperformance_parted_1_prt_1"   column            type        modifiers   storage   Description -----+-----+-----+-----+ year        smallint               plain      quarterid   smallint               plain      monthid     smallint               plain      dayofmonth   smallint               plain      dayid       smallint               plain      flightdate   date                   plain      uniqcarrierid   character varying(7)               extended   airlineid    smallint               plain      carrierid    character(3)               extended   tailnum      character(6)               extended   flightnum    character varying(5)               extended      div5longesttime   smallint               plain        div5wheeloff     time without time zone               plain        div5tailnum      character(6)               extended    Check constraints:   "factontimeperformance_parted_1_prt_1_check" CHECK (year &gt;= 2008::smallint AND year &lt; 2009::smallint) Inherits: factontimeperformance_parted Has OIDs: no Distributed by: (airlineid)  faa=#</pre>
6.	<p>Load data into the partitioned table to verify that the partition scheme is working:</p> <pre>faa=# COPY faadata.FactOnTimePerformance_parted FROM '/rawdata/FAAData/On_Time_On_Time_Performance_2008_1.csv' WITH DELIMITER ',' CSV HEADER QUOTE ''';</pre> <pre>gpadmin@mdw:~</pre> <pre>faa=# COPY faadata.FactOnTimePerformance_parted FROM '/rawdata/FAAData/On_Time_On_Time_Performance_2008_1.csv' faa-# WITH DELIMITER ',' CSV HEADER QUOTE ''; COPY 605765 faa=#</pre> <p><b>Note:</b> There is a double quote in the middle of two single quotes at the end of the line. The filename specified in the command is case sensitive.</p> <p><b>Note:</b> You are loading data into the parent table, not the child table. If the partitioning is working correctly, Greenplum will place the data into the appropriate partition when data is inserted into the parent table.</p>

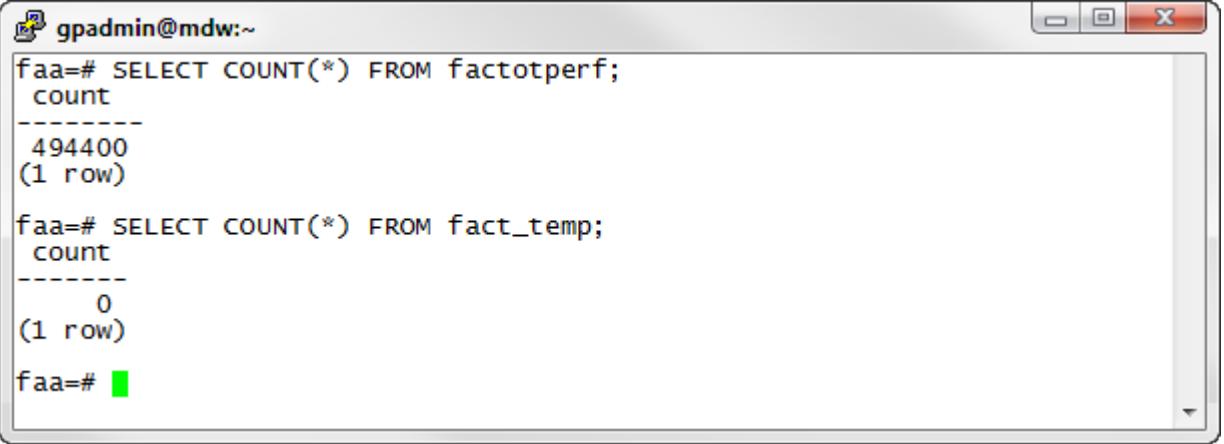
Step	Action
7.	<p>Verify that the data was copied into the correct partition:</p> <pre>faa=# SELECT COUNT(*) from faadata.FactOnTimePerformance_parted; faa=# SELECT COUNT(*) from faadata.FactOnTimePerformance_parted_1_prt_1;</pre>  <p><b>Note:</b> All of the records were placed into the first child partition table. If you query against any of the other child tables, you will find that they do not contain any rows.</p>
8.	<p>Load the January 2011 data into the same table:</p> <pre>faa=# COPY faadata.FactOnTimePerformance_parted FROM '/rawdata/FAADData/On_Time_On_Time_Performance_2011_1.csv' WITH DELIMITER ',' CSV HEADER QUOTE ''';</pre>  <p>This operation should fail since the partition for 2011 has not been created yet. There is also no default partition to capture any data that does not fall into the range of any defined partitions.</p>

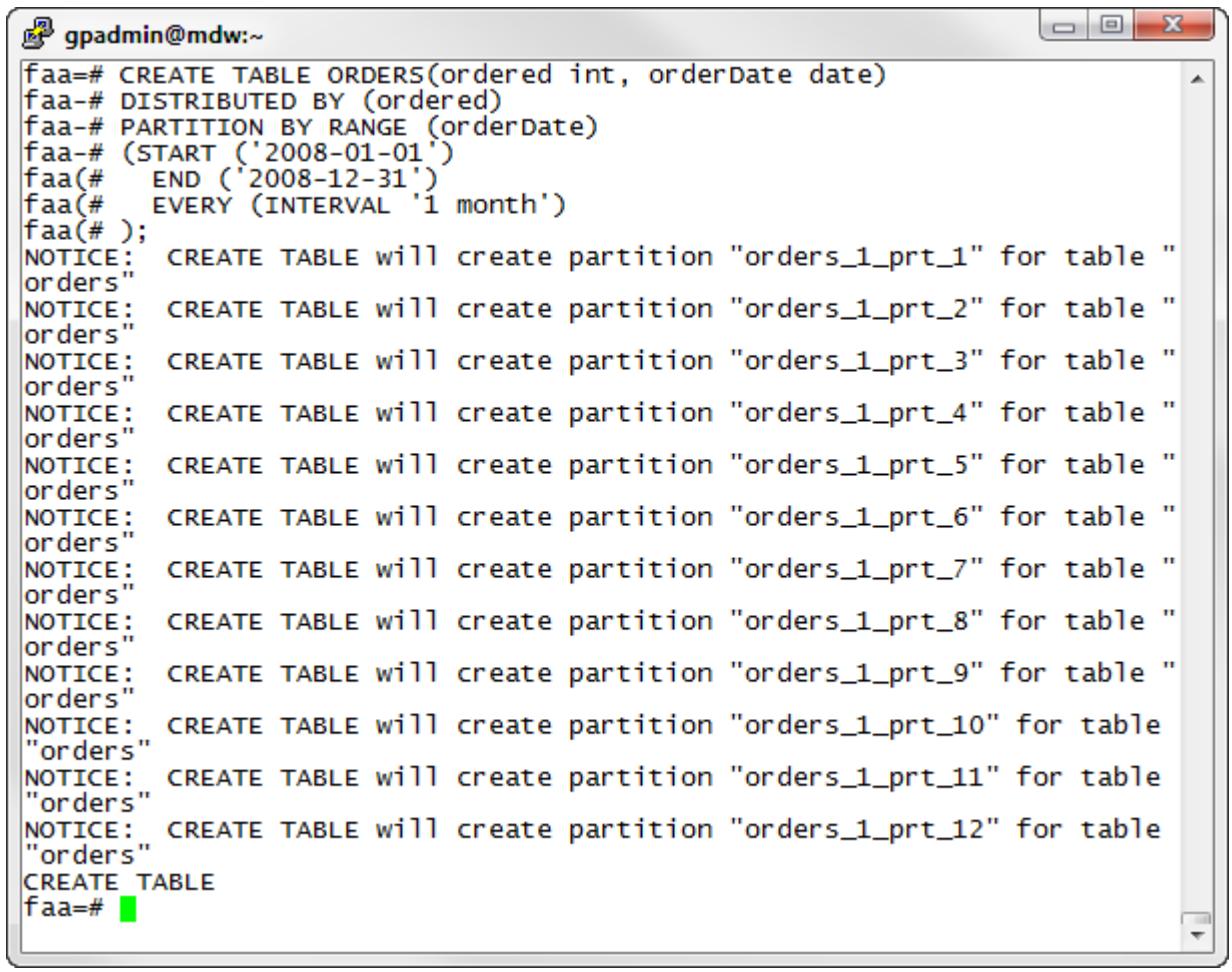
Step	Action
9.	<p>Create a new partition for the 2011 data:</p> <pre>ALTER TABLE faadata.factontimeperformance_parted add partition Y2011 START (smallint '2011') END (smallint '2012');</pre> 
10.	<p>Load the January 2011 data again:</p> <pre>COPY faadata.FactOnTimePerformance_parted FROM '/rawdata/FAAData/On_Time_On_Time_Performance_2011_1.csv' WITH DELIMITER ',' CSV HEADER QUOTE ''';</pre>  <p>Note: The operation should now succeed.</p>
11.	<p>The table and partition names can be long and non-intuitive. Rename the table to factotperf:</p> <pre>ALTER TABLE factontimeperformance_parted RENAME TO factotperf;</pre> 

Step	Action
12.	<p>Verify that the parent table and the child tables have been renamed:</p> <pre>faa=# \dt</pre> <pre>gpadmin@mdw:~ faa=# \dt       List of relations  Schema          Name           Type   Owner   Storage -----+----------------+-----+-----+-----+  faadata   dimairline        table   gpadmin   heap  faadata   dimairport         table   gpadmin   heap  faadata   dimcancellation    table   gpadmin   heap  faadata   dimcarrier_history  table   gpadmin   heap  faadata   dimday             table   gpadmin   heap  faadata   dimdepart_arrive_block   table   gpadmin   heap  faadata   dimdistance_group   table   gpadmin   heap  faadata   dimdiversions      table   gpadmin   heap  faadata   dimmonth            table   gpadmin   heap  faadata   dimontime_delay_groups   table   gpadmin   heap  faadata   dimquarter           table   gpadmin   heap  faadata   dimstate_abr         table   gpadmin   heap  faadata   dimstate_fips        table   gpadmin   heap  faadata   dimunique_carriers   table   gpadmin   heap  faadata   dimworld_area_code   table   gpadmin   heap  faadata   dimyesno_resp        table   gpadmin   heap  faadata   factontimeperformance   table   gpadmin   heap  faadata   factotperf            table   gpadmin   heap  faadata   factotperf_1_prt_1     table   gpadmin   heap  faadata   factotperf_1_prt_2     table   gpadmin   heap  faadata   factotperf_1_prt_3     table   gpadmin   heap  faadata   factotperf_1_prt_y2011   table   gpadmin   heap  faadata   performance            table   gpadmin   heap  faadata   test_table             table   gpadmin   heap  faadata   test_table2            table   gpadmin   heap  public    airport_err            table   gpadmin   heap  public    fact_err                table   gpadmin   heap (27 rows)  faa=# </pre>
13.	<p>Rename the 2008 partition so that it has a more intuitive name:</p> <pre>faa=# ALTER TABLE factotperf RENAME PARTITION for (2008) to Y2008;</pre> <pre>gpadmin@mdw:~ faa=# ALTER TABLE factotperf RENAME PARTITION for (2008) to Y2008; faa=# </pre>

Step	Action																																																																																																																																																	
14.	<p>Verify that the partition has been renamed:</p> <pre>faa=# \dt</pre>  <table border="1"> <thead> <tr> <th>Schema</th> <th>Name</th> <th>Type</th> <th>Owner</th> <th>Storage</th> </tr> </thead> <tbody> <tr><td>faadata</td><td>dimairline</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>dimairport</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>dimcancellation</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>dimcarrier_history</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>dimday</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>dimdepart_arrive_block</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>dimdistance_group</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>dimdiversions</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>dimmonth</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>dimontime_delay_groups</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>dimquarter</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>dimstate_abr</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>dimstate_fips</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>dimunique_carriers</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>dimworld_area_code</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>dimyesno_resp</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>factontimeperformance</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>factotperf</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>factotperf_1_prt_2</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>factotperf_1_prt_3</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>factotperf_1_prt_y2008</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>factotperf_1_prt_y2011</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>performance</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>test_table</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>test_table2</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>public</td><td>airport_err</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>public</td><td>fact_err</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td></td><td>(27 rows)</td><td></td><td></td><td></td></tr> </tbody> </table> <pre>faa=# </pre>	Schema	Name	Type	Owner	Storage	faadata	dimairline	table	gpadmin	heap	faadata	dimairport	table	gpadmin	heap	faadata	dimcancellation	table	gpadmin	heap	faadata	dimcarrier_history	table	gpadmin	heap	faadata	dimday	table	gpadmin	heap	faadata	dimdepart_arrive_block	table	gpadmin	heap	faadata	dimdistance_group	table	gpadmin	heap	faadata	dimdiversions	table	gpadmin	heap	faadata	dimmonth	table	gpadmin	heap	faadata	dimontime_delay_groups	table	gpadmin	heap	faadata	dimquarter	table	gpadmin	heap	faadata	dimstate_abr	table	gpadmin	heap	faadata	dimstate_fips	table	gpadmin	heap	faadata	dimunique_carriers	table	gpadmin	heap	faadata	dimworld_area_code	table	gpadmin	heap	faadata	dimyesno_resp	table	gpadmin	heap	faadata	factontimeperformance	table	gpadmin	heap	faadata	factotperf	table	gpadmin	heap	faadata	factotperf_1_prt_2	table	gpadmin	heap	faadata	factotperf_1_prt_3	table	gpadmin	heap	faadata	factotperf_1_prt_y2008	table	gpadmin	heap	faadata	factotperf_1_prt_y2011	table	gpadmin	heap	faadata	performance	table	gpadmin	heap	faadata	test_table	table	gpadmin	heap	faadata	test_table2	table	gpadmin	heap	public	airport_err	table	gpadmin	heap	public	fact_err	table	gpadmin	heap		(27 rows)			
Schema	Name	Type	Owner	Storage																																																																																																																																														
faadata	dimairline	table	gpadmin	heap																																																																																																																																														
faadata	dimairport	table	gpadmin	heap																																																																																																																																														
faadata	dimcancellation	table	gpadmin	heap																																																																																																																																														
faadata	dimcarrier_history	table	gpadmin	heap																																																																																																																																														
faadata	dimday	table	gpadmin	heap																																																																																																																																														
faadata	dimdepart_arrive_block	table	gpadmin	heap																																																																																																																																														
faadata	dimdistance_group	table	gpadmin	heap																																																																																																																																														
faadata	dimdiversions	table	gpadmin	heap																																																																																																																																														
faadata	dimmonth	table	gpadmin	heap																																																																																																																																														
faadata	dimontime_delay_groups	table	gpadmin	heap																																																																																																																																														
faadata	dimquarter	table	gpadmin	heap																																																																																																																																														
faadata	dimstate_abr	table	gpadmin	heap																																																																																																																																														
faadata	dimstate_fips	table	gpadmin	heap																																																																																																																																														
faadata	dimunique_carriers	table	gpadmin	heap																																																																																																																																														
faadata	dimworld_area_code	table	gpadmin	heap																																																																																																																																														
faadata	dimyesno_resp	table	gpadmin	heap																																																																																																																																														
faadata	factontimeperformance	table	gpadmin	heap																																																																																																																																														
faadata	factotperf	table	gpadmin	heap																																																																																																																																														
faadata	factotperf_1_prt_2	table	gpadmin	heap																																																																																																																																														
faadata	factotperf_1_prt_3	table	gpadmin	heap																																																																																																																																														
faadata	factotperf_1_prt_y2008	table	gpadmin	heap																																																																																																																																														
faadata	factotperf_1_prt_y2011	table	gpadmin	heap																																																																																																																																														
faadata	performance	table	gpadmin	heap																																																																																																																																														
faadata	test_table	table	gpadmin	heap																																																																																																																																														
faadata	test_table2	table	gpadmin	heap																																																																																																																																														
public	airport_err	table	gpadmin	heap																																																																																																																																														
public	fact_err	table	gpadmin	heap																																																																																																																																														
	(27 rows)																																																																																																																																																	
15.	<p>You may need to remove a partition after a period of time to age out data. Drop the 2008 partition with the following command:</p> <pre>faa=# ALTER TABLE factotperf DROP PARTITION FOR (2008);</pre> <img alt="Screenshot of a terminal window showing the execution of the ALTER TABLE command. The window title is 'gpadmin@mdw:~'. The command 'faa=# ALTER TABLE factotperf DROP PARTITION FOR (2008);' is entered. A notice message 'NOTICE: dropped partition "y2008" for relation "factotperf"' is displayed. The command 'ALTER TABLE' is also shown. The prompt 'faa=' is at the bottom." data-bbox="180 690 930 790"/> <pre>faa=# ALTER TABLE factotperf DROP PARTITION FOR (2008); NOTICE: dropped partition "y2008" for relation "factotperf" ALTER TABLE faa=# </pre>																																																																																																																																																	

Step	Action
16.	<p>Exchanging a partition allows you to bring data from a table into a partition. Create a table named fact_temp that has the same structure as factotperf. This table will contain the data that will eventually be placed into the partitioned table:</p> <pre data-bbox="306 312 1148 344">faa=# CREATE TABLE fact_temp (LIKE factotperf);</pre> 
17.	<p>Remove any existing data from the factotperf table by truncating the table. This will truncate all partitions within the table.</p> <pre data-bbox="306 745 882 777">faa=# TRUNCATE TABLE factotperf;</pre> 
18.	<p>Copy the January 2011 data into fact_temp:</p> <pre data-bbox="306 1110 1307 1205">faa=# COPY faadata.fact_temp FROM '/rawdata/FAAData/On_Time_On_Time_Performance_2011_1.csv' WITH DELIMITER ',' CSV HEADER QUOTE ''';</pre> 
19.	<p>Take the data found in the factotperf table and push it to the Y2011 partition using the EXCHANGE clause on the partition. This will also move the data that was in the partition to the fact_temp table:</p> <pre data-bbox="306 1649 1432 1712">faa=# ALTER TABLE factotperf EXCHANGE PARTITION Y2011 WITH TABLE fact_temp;</pre> 

Step	Action
20.	<p>Verify that the data is now in the partitioned table and that the fact_temp table has no rows:</p> <pre>faa=# SELECT COUNT(*) FROM factotperf; faa=# SELECT COUNT(*) FROM fact_temp;</pre> 

Step	Action
21.	<p>Sometimes a partition may become too large and it might be worthwhile to split the partition. Create a new partitioned table:</p> <pre data-bbox="298 264 1253 487"> faa=# CREATE TABLE ORDERS(ordered int, orderDate date) DISTRIBUTED BY (ordered) PARTITION BY RANGE (orderDate) (START ('2008-01-01') END ('2008-12-31') EVERY (INTERVAL '1 month') ); </pre> 

Step	Action
22.	<p>Split the January partition into two partitions:</p> <pre>faa=# ALTER TABLE orders SPLIT PARTITION FOR ('2008-01-01') AT ('2008-01-16') INTO (PARTITION jan20081to15, PARTITION jan200816to31);</pre>
23.	<p>Verify that the new partitions have been created:</p> <pre>faa-# \dt</pre> <p>Exit the database before proceeding.</p>

Step	Action
	<p><b>Summary</b></p> <p>Table partitioning addresses the problem of supporting very large tables, such as fact tables, by allowing you to logically divide them into smaller and more manageable pieces. Partitioning is used to improve performance by scanning only the relevant data needed to satisfy a query. It can also facilitate database loading and maintenance.</p> <p>In the Greenplum Database, partitioning is a procedure that creates multiple sub-tables (or child tables) from a single large table (or parent table) and setting exclusion constraints on the child tables. Table data resides in the child tables only; no data should be loaded in the parent tables. Note that partitioned tables in the Greenplum Database are also physically distributed across the segment instances just as are non-partitioned tables.</p> <p>The <code>ALTER TABLE</code> syntax for handling partitions is very flexible. It allows almost as much control over the structure of a table as the <code>CREATE TABLE</code> syntax. With this one command you are given the ability to add partitions, drop partitions, and rename partitions. Remember that if you drop a partition, it will remove the partition and all dependents.</p> <p>Exchanging a partition is an easy way to facilitate tricky operations that with other databases would require locking out the users or running late at night so as not to interfere with work. By running a load on a table that is not being used, and then quickly exchanging it once the load is finished, you can see more data security and less downtime. The advantages of exchanging a partition do not end there.</p> <p>The alter table syntax can be used to split a partition if the partition is growing too large. To use this command, you must split your partition on the partitioning key. The value specified will land in the latter of the two partition tables created.</p>

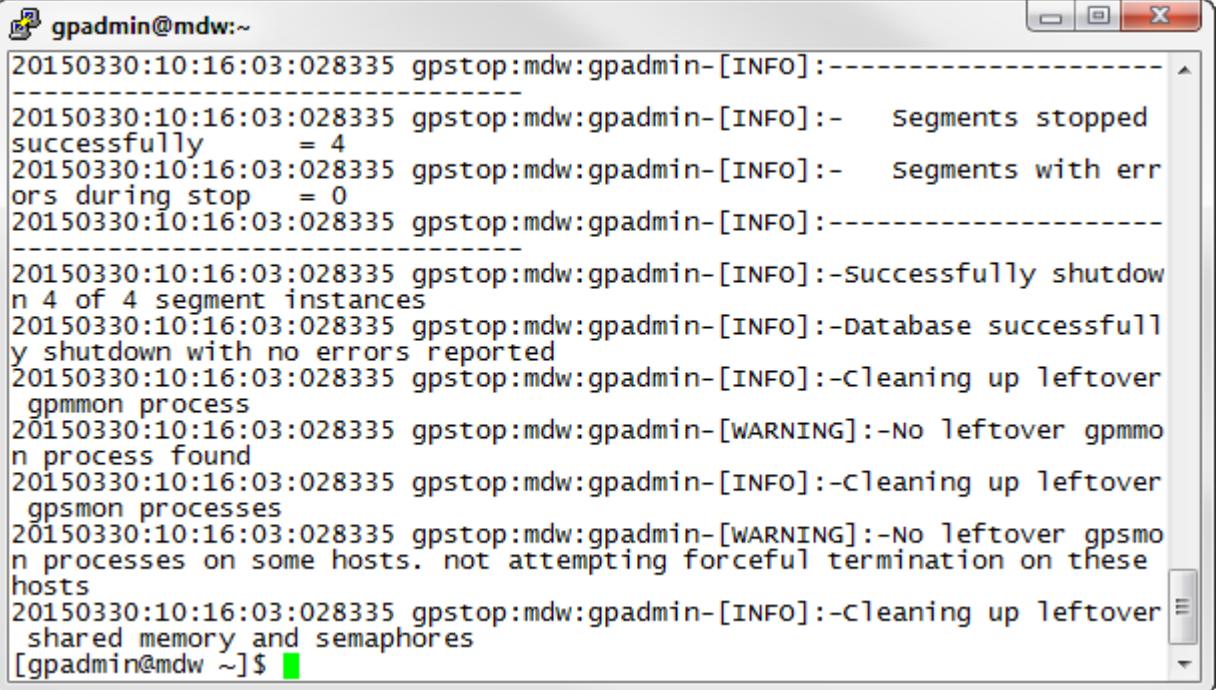
*End of Lab Exercise*

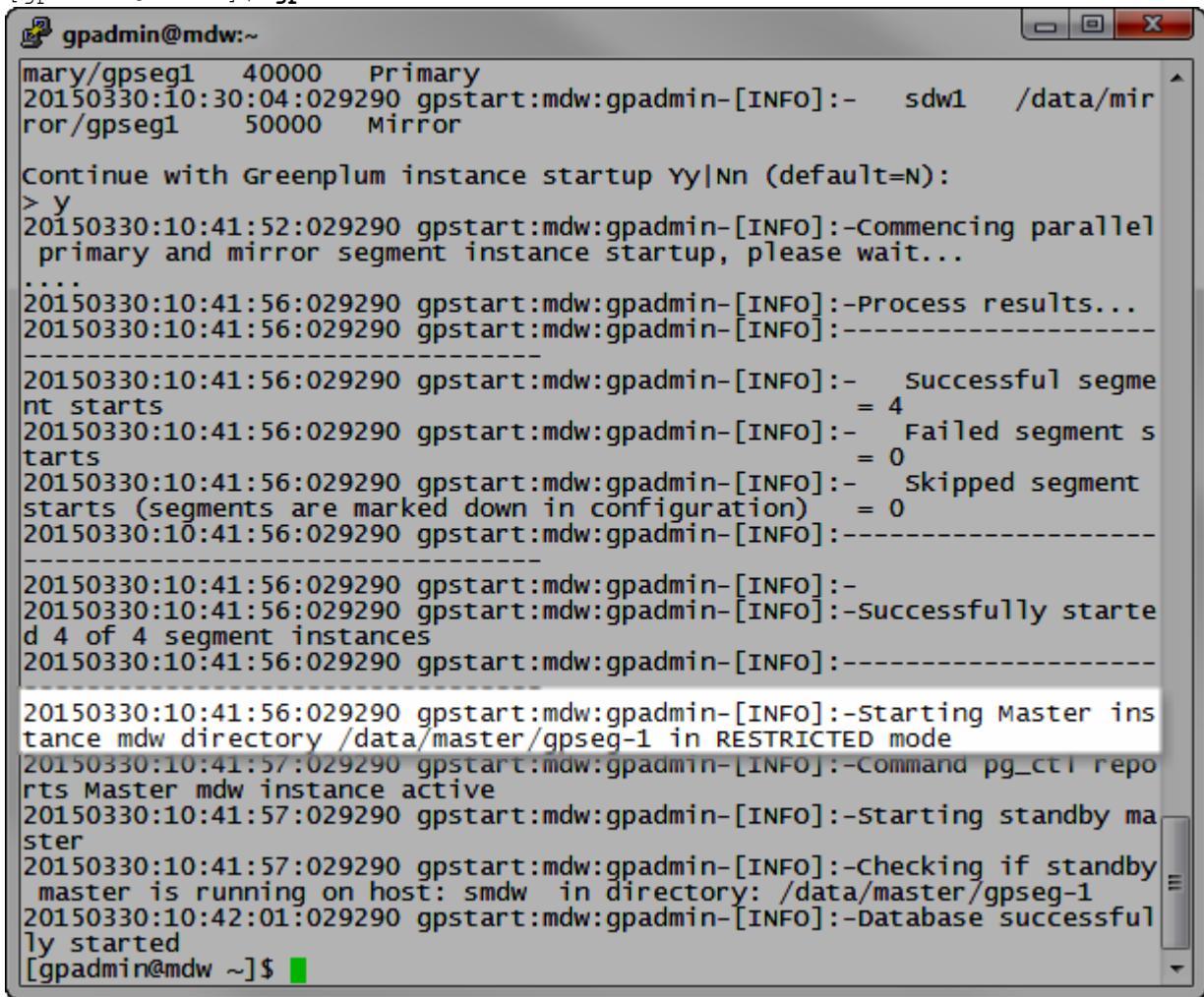
This page intentionally left blank

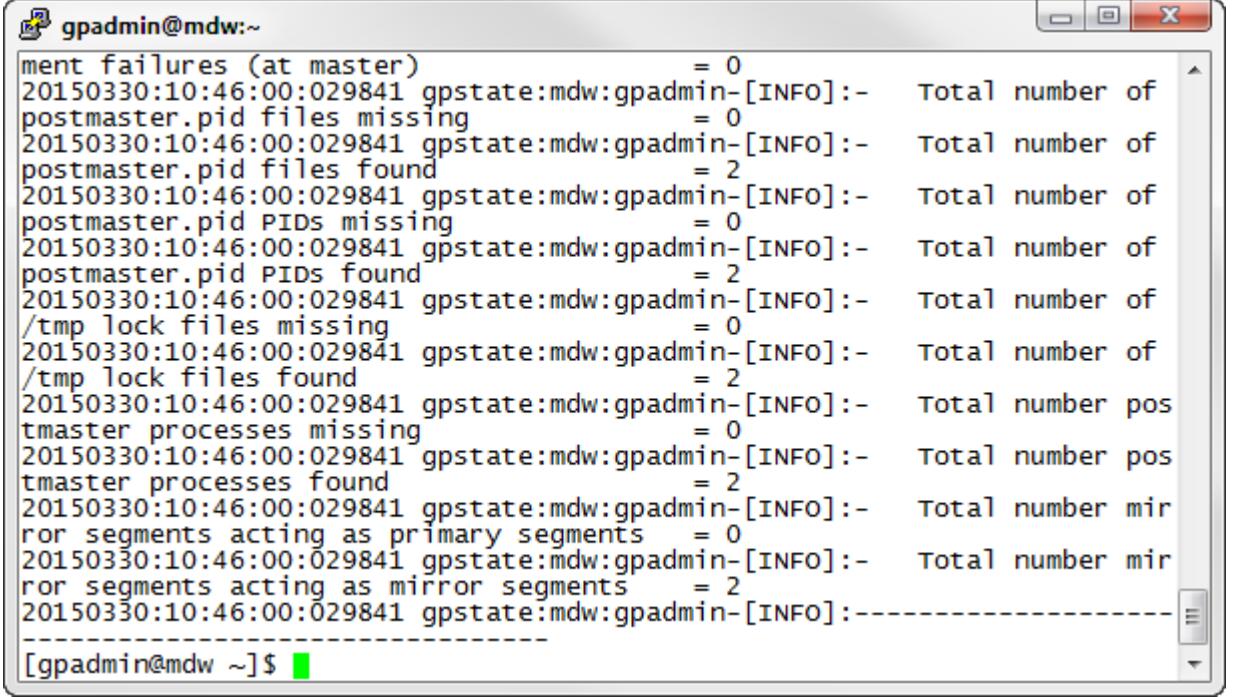
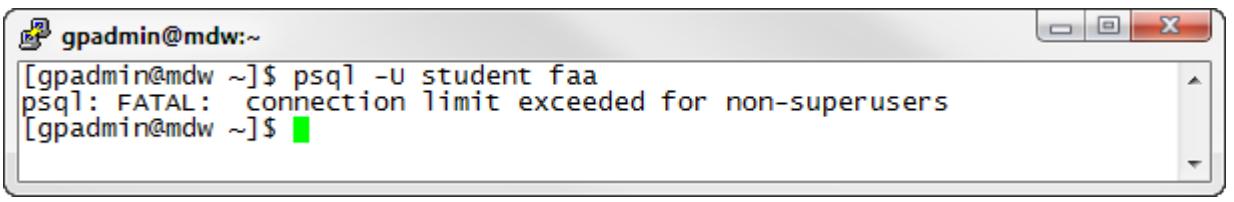
## Lab 14. Managing the Greenplum Database

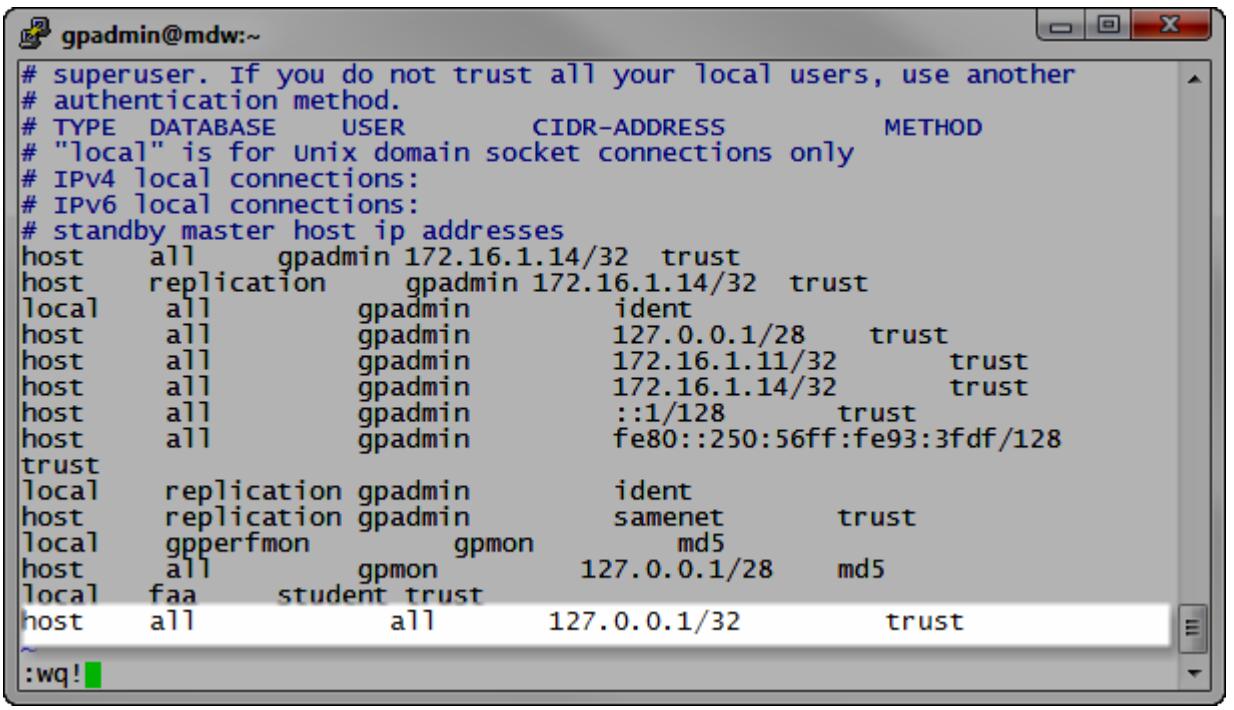
<b>Purpose:</b>	In this lab, you will start, stop, and restart the Greenplum Database. You will also execute commands that provide information on the state of Greenplum.  Greenplum clients must be executed from the master instance while logged in as a Greenplum superuser account.
<b>Tasks:</b>	Students perform the following task: Perform system administration tasks on the Greenplum environment.
<b>References:</b>	Module 6: Database Management and Archiving <ul style="list-style-type: none"><li>• Lesson: Managing the Greenplum Database</li></ul>

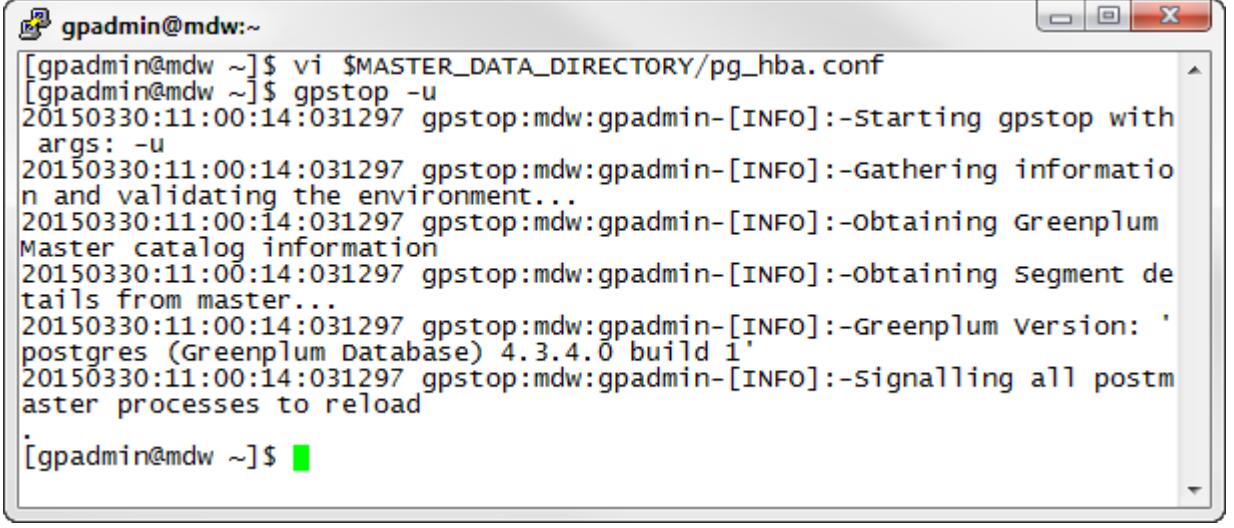
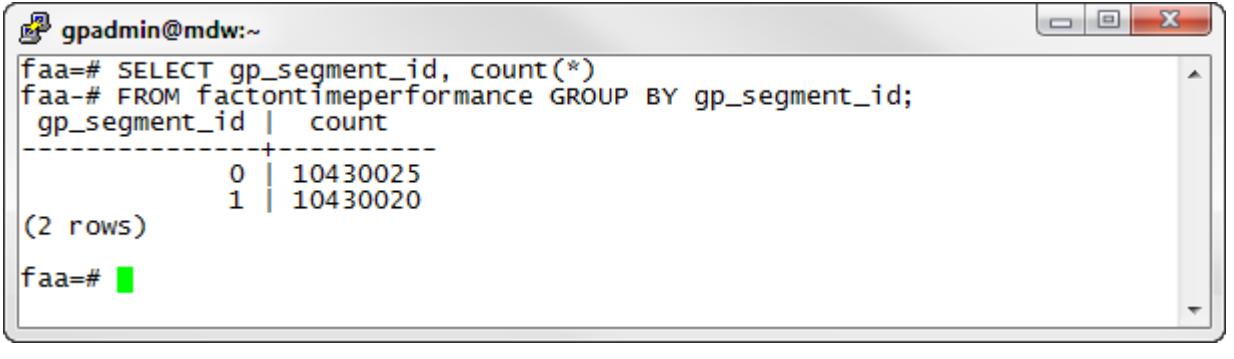
## Lab 14: Task – Perform System Administration Tasks on the Greenplum Environment

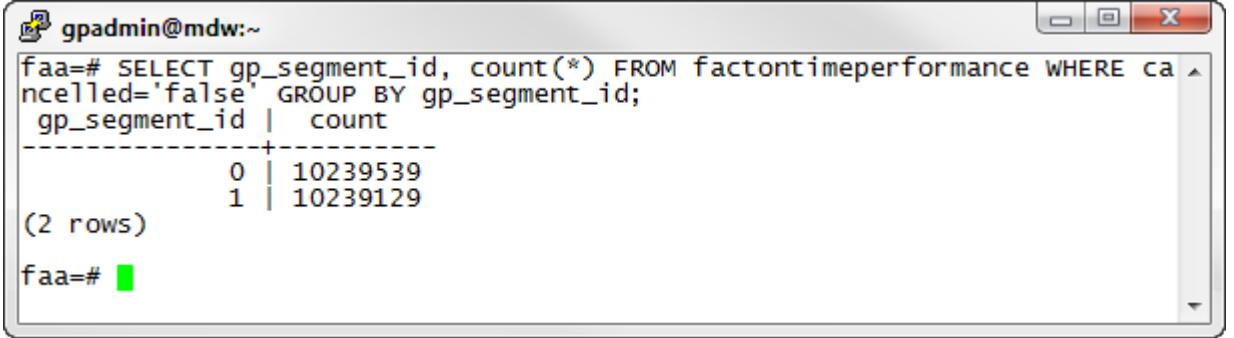
Step	Action
1.	<p>If not already connected to mdw, log in as gpadmin on your master server and stop the Greenplum database with the following command :</p> <pre>[gpadmin@mdw ~]\$ gpstop</pre>  <p>When prompted, answer <b>y</b> to proceed with shutting down the instances.</p>

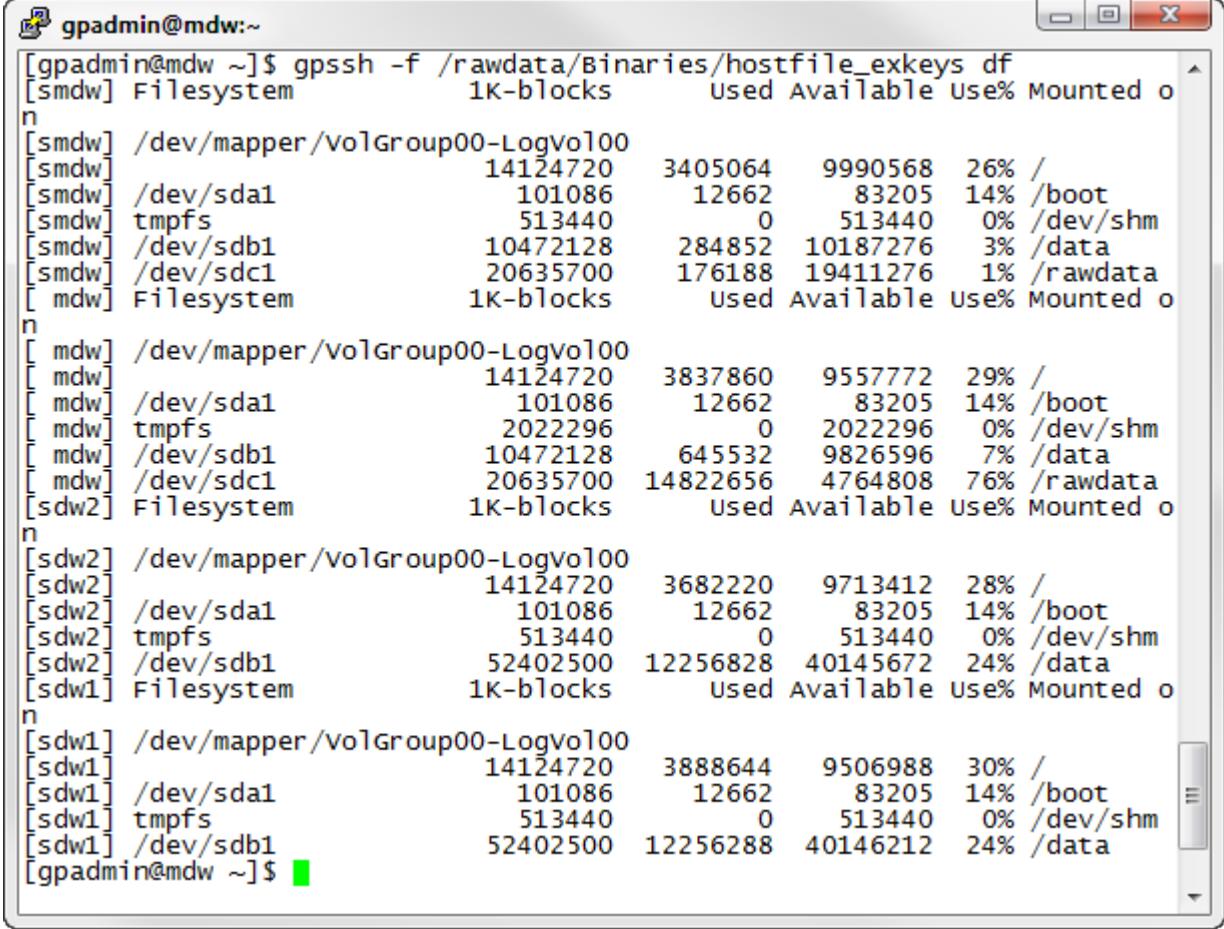
Step	Action
2.	<p>The gpstart command is used to start the Greenplum database defined by the \$MASTER_DATA_DIRECTORY parameter or as specified with the -d option. To start the Greenplum database in restrictive mode where only the Greenplum superuser can connect, issue the following command in UNIX as gpadmin:</p> <pre>[gpadmin@mdw ~] \$ gpstart -R</pre>  <p>When prompted, answer <b>y</b> to proceed with starting the instances. You can skip any prompts by including the <b>-a</b> option as a part of the command.</p> <p>This mode is useful if you need to perform some isolated work in the environment and need to ensure other users cannot connect to the Greenplum Database.</p>

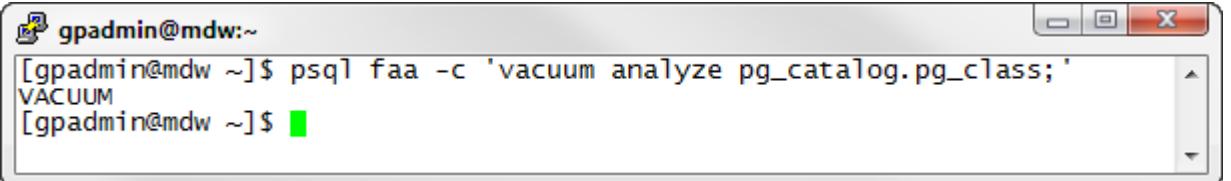
Step	Action
3.	<p>Verify the state of the cluster to ensure that the database is running:</p> <pre>[gpadmin@mdw ~]\$ gpstate</pre>  <pre>gpadmin@mdw:~\$ gpstate ment failures (at master) = 0 Total number of 20150330:10:46:00:029841 gpstate:mdw:gpadmin-[INFO]:- postmaster.pid files missing = 0 Total number of 20150330:10:46:00:029841 gpstate:mdw:gpadmin-[INFO]:- postmaster.pid files found = 2 Total number of 20150330:10:46:00:029841 gpstate:mdw:gpadmin-[INFO]:- postmaster.pid PIDs missing = 0 Total number of 20150330:10:46:00:029841 gpstate:mdw:gpadmin-[INFO]:- postmaster.pid PIDs found = 2 Total number of 20150330:10:46:00:029841 gpstate:mdw:gpadmin-[INFO]:- /tmp lock files missing = 0 Total number of 20150330:10:46:00:029841 gpstate:mdw:gpadmin-[INFO]:- /tmp lock files found = 2 Total number of 20150330:10:46:00:029841 gpstate:mdw:gpadmin-[INFO]:- tmaster processes missing = 0 Total number pos 20150330:10:46:00:029841 gpstate:mdw:gpadmin-[INFO]:- tmaster processes found = 2 Total number pos 20150330:10:46:00:029841 gpstate:mdw:gpadmin-[INFO]:- ror segments acting as primary segments = 0 Total number mir 20150330:10:46:00:029841 gpstate:mdw:gpadmin-[INFO]:- ror segments acting as mirror segments = 2 Total number mir 20150330:10:46:00:029841 gpstate:mdw:gpadmin-[INFO]:-----[gpadmin@mdw ~]\$</pre>
4.	<p>Verify the student account that you created in earlier labs cannot connect to the database.</p> <pre>[gpadmin@mdw ~]\$ psql -U student faa</pre>  <pre>gpadmin@mdw:~\$ psql -U student faa psql: FATAL:  connection limit exceeded for non-superusers gpadmin@mdw ~\$</pre>

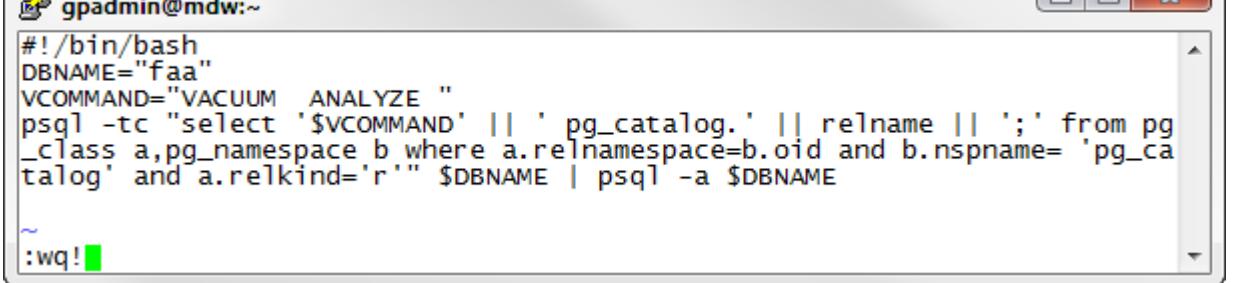
Step	Action
5.	<p>There are times when you want to change parameters of the database by editing either pg_hba.conf or postgresql.conf, but do not want to shut down the database to make it aware of the changes. The gpstop -u command lets the master re-read the configuration files without shutting down services.</p> <p>Using vi, edit the pg_hba.conf file and add the following line at the end:</p> <pre data-bbox="306 418 1356 447">host      all           all      127.0.0.1/32      trust</pre> <p>The new entry allows all users from 127.0.0.1 to connect to all of the databases within the Greenplum Database cluster.</p> <pre data-bbox="313 587 1269 616">[gpadmin@mdw ~]\$ vi \$MASTER_DATA_DIRECTORY/pg_hba.conf</pre> 

Step	Action
6.	<p>Read the new configuration into the database by issuing the following command:</p> <pre>[gpadmin@mdw ~]\$ gpstop -u</pre>  <p>This will reload all the configuration files without shutting the database down.</p>
7.	<p>Log in to PSQL and check for skew on the main fact table, factontimeperformance:</p> <pre>[gpadmin@mdw ~]\$ psql faa</pre>
8.	<pre>faa=# SELECT gp_segment_id, count(*) FROM factontimeperformance GROUP BY gp_segment_id;</pre> 

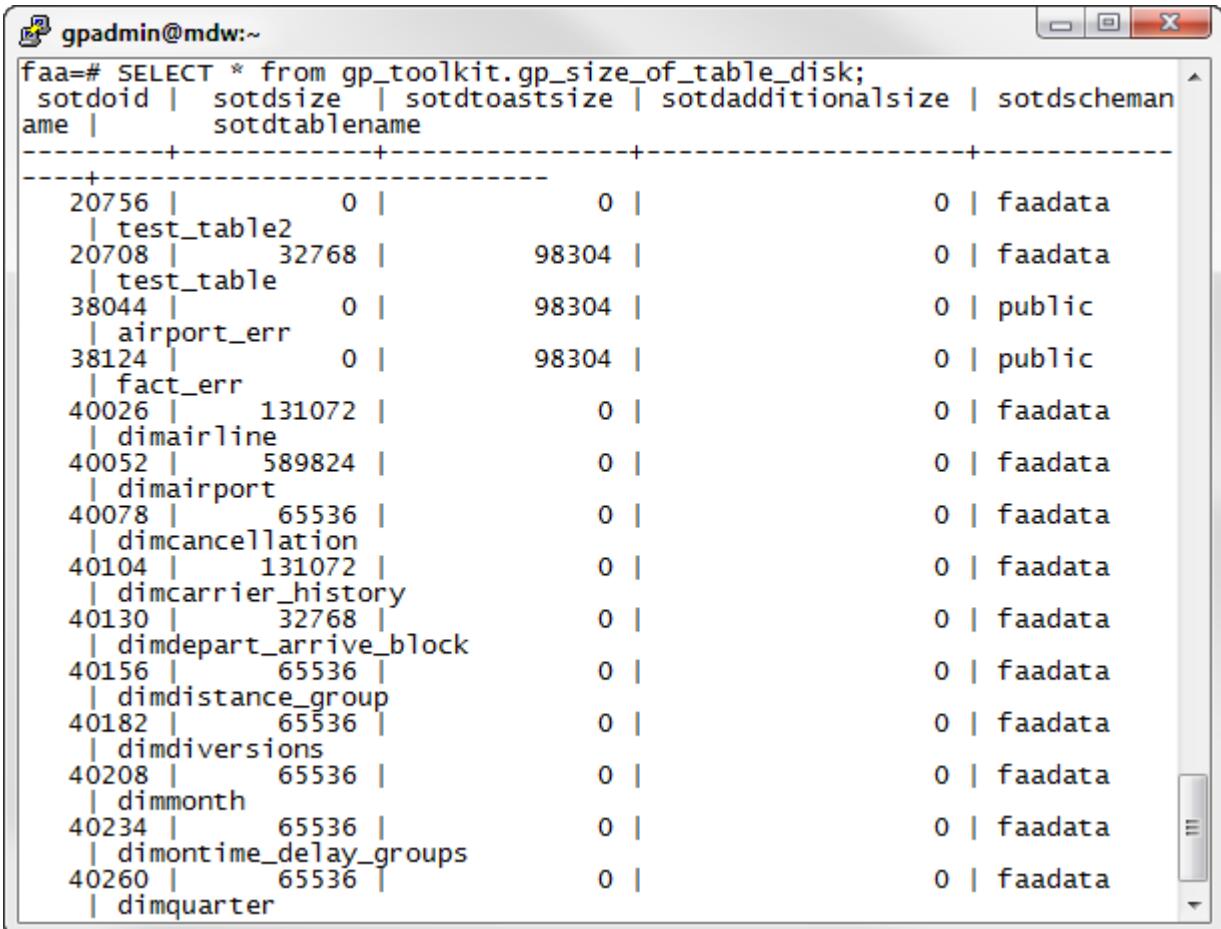
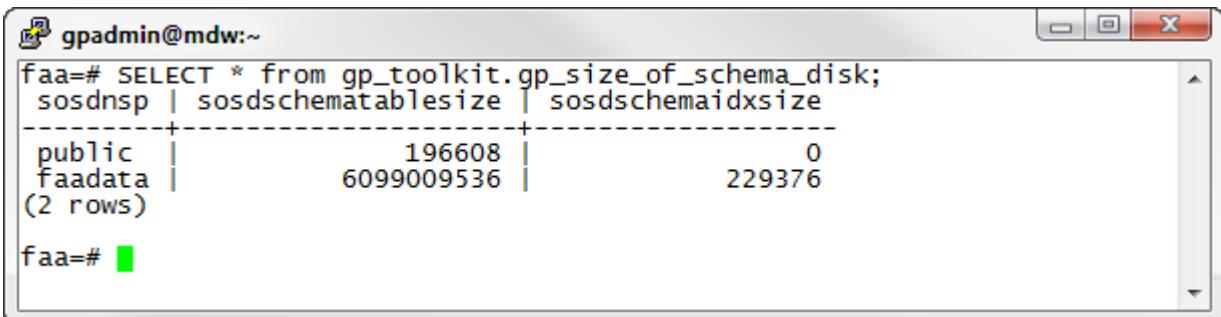
Step	Action
9.	<p>Check for processing skew:</p> <pre>faa=# SELECT gp_segment_id, count(*) FROM factontimeperformance WHERE cancelled='false' GROUP BY gp_segment_id;</pre>  <pre>gpadmin@mdw:~\$ faa=# SELECT gp_segment_id, count(*) FROM factontimeperformance WHERE cancelled='false' GROUP BY gp_segment_id; gp_segment_id   count -----+-----           0   10239539           1   10239129 (2 rows)  faa=#</pre>
10.	<p>Exit the database.</p> <pre>faa=# \q</pre>

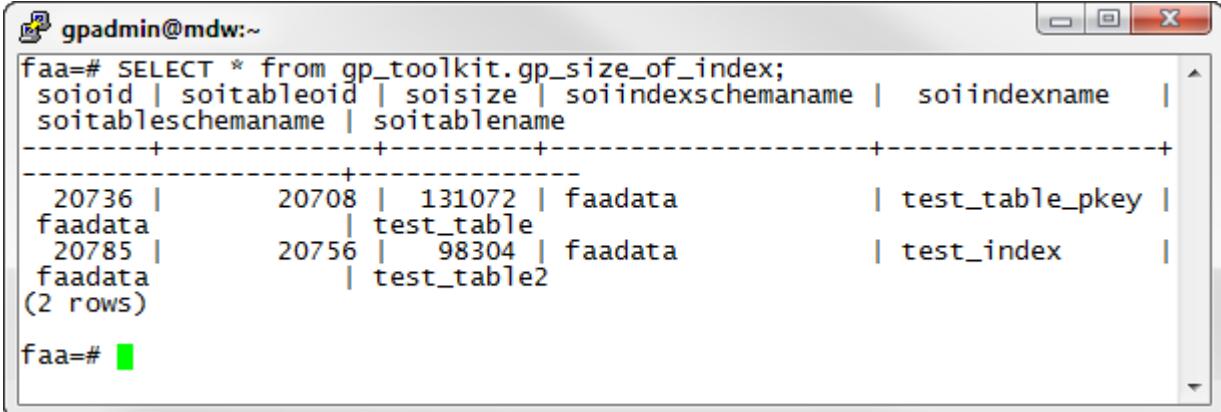
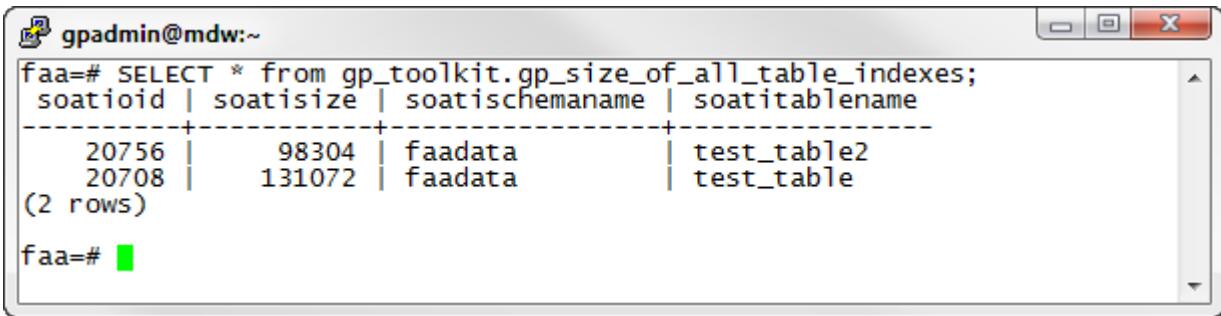
Step	Action																																																																																																																																				
11.	<p>There may be times when you need to perform administrative duties on the segment hosts from the master. You can use the <code>gpssh -f</code> command from UNIX as gpadmin. The <code>/rawdata/Binaries/hostfile_exkeys</code> file contains the names of all hosts in this cluster. You can use this file to query the hosts.</p> <pre>[gpadmin@mdw ~]\$ gpssh -f /rawdata/Binaries/hostfile_exkeys df</pre>  <table border="1"> <thead> <tr> <th>Filesystem</th> <th>1K-blocks</th> <th>Used</th> <th>Available</th> <th>Use%</th> <th>Mounted on</th> </tr> </thead> <tbody> <tr> <td>[smdw] /dev/mapper/volGroup00-LogVol00</td> <td>14124720</td> <td>3405064</td> <td>9990568</td> <td>26%</td> <td>/</td> </tr> <tr> <td>[smdw] /dev/sda1</td> <td>101086</td> <td>12662</td> <td>83205</td> <td>14%</td> <td>/boot</td> </tr> <tr> <td>[smdw] tmpfs</td> <td>513440</td> <td>0</td> <td>513440</td> <td>0%</td> <td>/dev/shm</td> </tr> <tr> <td>[smdw] /dev/sdb1</td> <td>10472128</td> <td>284852</td> <td>10187276</td> <td>3%</td> <td>/data</td> </tr> <tr> <td>[smdw] /dev/sdc1</td> <td>20635700</td> <td>176188</td> <td>19411276</td> <td>1%</td> <td>/rawdata</td> </tr> <tr> <td>[mdw] Filesystem</td> <td>1K-blocks</td> <td>Used</td> <td>Available</td> <td>Use%</td> <td>Mounted on</td> </tr> <tr> <td>[ mdw] /dev/mapper/volGroup00-LogVol00</td> <td>14124720</td> <td>3837860</td> <td>9557772</td> <td>29%</td> <td>/</td> </tr> <tr> <td>[ mdw] /dev/sda1</td> <td>101086</td> <td>12662</td> <td>83205</td> <td>14%</td> <td>/boot</td> </tr> <tr> <td>[ mdw] tmpfs</td> <td>2022296</td> <td>0</td> <td>2022296</td> <td>0%</td> <td>/dev/shm</td> </tr> <tr> <td>[ mdw] /dev/sdb1</td> <td>10472128</td> <td>645532</td> <td>9826596</td> <td>7%</td> <td>/data</td> </tr> <tr> <td>[ mdw] /dev/sdc1</td> <td>20635700</td> <td>14822656</td> <td>4764808</td> <td>76%</td> <td>/rawdata</td> </tr> <tr> <td>[sdw2] Filesystem</td> <td>1K-blocks</td> <td>Used</td> <td>Available</td> <td>Use%</td> <td>Mounted on</td> </tr> <tr> <td>[sdw2] /dev/mapper/volGroup00-LogVol00</td> <td>14124720</td> <td>3682220</td> <td>9713412</td> <td>28%</td> <td>/</td> </tr> <tr> <td>[sdw2] /dev/sda1</td> <td>101086</td> <td>12662</td> <td>83205</td> <td>14%</td> <td>/boot</td> </tr> <tr> <td>[sdw2] tmpfs</td> <td>513440</td> <td>0</td> <td>513440</td> <td>0%</td> <td>/dev/shm</td> </tr> <tr> <td>[sdw2] /dev/sdb1</td> <td>52402500</td> <td>12256828</td> <td>40145672</td> <td>24%</td> <td>/data</td> </tr> <tr> <td>[sdw1] Filesystem</td> <td>1K-blocks</td> <td>Used</td> <td>Available</td> <td>Use%</td> <td>Mounted on</td> </tr> <tr> <td>[sdw1] /dev/mapper/volGroup00-LogVol00</td> <td>14124720</td> <td>3888644</td> <td>9506988</td> <td>30%</td> <td>/</td> </tr> <tr> <td>[sdw1] /dev/sda1</td> <td>101086</td> <td>12662</td> <td>83205</td> <td>14%</td> <td>/boot</td> </tr> <tr> <td>[sdw1] tmpfs</td> <td>513440</td> <td>0</td> <td>513440</td> <td>0%</td> <td>/dev/shm</td> </tr> <tr> <td>[sdw1] /dev/sdb1</td> <td>52402500</td> <td>12256288</td> <td>40146212</td> <td>24%</td> <td>/data</td> </tr> </tbody> </table>	Filesystem	1K-blocks	Used	Available	Use%	Mounted on	[smdw] /dev/mapper/volGroup00-LogVol00	14124720	3405064	9990568	26%	/	[smdw] /dev/sda1	101086	12662	83205	14%	/boot	[smdw] tmpfs	513440	0	513440	0%	/dev/shm	[smdw] /dev/sdb1	10472128	284852	10187276	3%	/data	[smdw] /dev/sdc1	20635700	176188	19411276	1%	/rawdata	[mdw] Filesystem	1K-blocks	Used	Available	Use%	Mounted on	[ mdw] /dev/mapper/volGroup00-LogVol00	14124720	3837860	9557772	29%	/	[ mdw] /dev/sda1	101086	12662	83205	14%	/boot	[ mdw] tmpfs	2022296	0	2022296	0%	/dev/shm	[ mdw] /dev/sdb1	10472128	645532	9826596	7%	/data	[ mdw] /dev/sdc1	20635700	14822656	4764808	76%	/rawdata	[sdw2] Filesystem	1K-blocks	Used	Available	Use%	Mounted on	[sdw2] /dev/mapper/volGroup00-LogVol00	14124720	3682220	9713412	28%	/	[sdw2] /dev/sda1	101086	12662	83205	14%	/boot	[sdw2] tmpfs	513440	0	513440	0%	/dev/shm	[sdw2] /dev/sdb1	52402500	12256828	40145672	24%	/data	[sdw1] Filesystem	1K-blocks	Used	Available	Use%	Mounted on	[sdw1] /dev/mapper/volGroup00-LogVol00	14124720	3888644	9506988	30%	/	[sdw1] /dev/sda1	101086	12662	83205	14%	/boot	[sdw1] tmpfs	513440	0	513440	0%	/dev/shm	[sdw1] /dev/sdb1	52402500	12256288	40146212	24%	/data
Filesystem	1K-blocks	Used	Available	Use%	Mounted on																																																																																																																																
[smdw] /dev/mapper/volGroup00-LogVol00	14124720	3405064	9990568	26%	/																																																																																																																																
[smdw] /dev/sda1	101086	12662	83205	14%	/boot																																																																																																																																
[smdw] tmpfs	513440	0	513440	0%	/dev/shm																																																																																																																																
[smdw] /dev/sdb1	10472128	284852	10187276	3%	/data																																																																																																																																
[smdw] /dev/sdc1	20635700	176188	19411276	1%	/rawdata																																																																																																																																
[mdw] Filesystem	1K-blocks	Used	Available	Use%	Mounted on																																																																																																																																
[ mdw] /dev/mapper/volGroup00-LogVol00	14124720	3837860	9557772	29%	/																																																																																																																																
[ mdw] /dev/sda1	101086	12662	83205	14%	/boot																																																																																																																																
[ mdw] tmpfs	2022296	0	2022296	0%	/dev/shm																																																																																																																																
[ mdw] /dev/sdb1	10472128	645532	9826596	7%	/data																																																																																																																																
[ mdw] /dev/sdc1	20635700	14822656	4764808	76%	/rawdata																																																																																																																																
[sdw2] Filesystem	1K-blocks	Used	Available	Use%	Mounted on																																																																																																																																
[sdw2] /dev/mapper/volGroup00-LogVol00	14124720	3682220	9713412	28%	/																																																																																																																																
[sdw2] /dev/sda1	101086	12662	83205	14%	/boot																																																																																																																																
[sdw2] tmpfs	513440	0	513440	0%	/dev/shm																																																																																																																																
[sdw2] /dev/sdb1	52402500	12256828	40145672	24%	/data																																																																																																																																
[sdw1] Filesystem	1K-blocks	Used	Available	Use%	Mounted on																																																																																																																																
[sdw1] /dev/mapper/volGroup00-LogVol00	14124720	3888644	9506988	30%	/																																																																																																																																
[sdw1] /dev/sda1	101086	12662	83205	14%	/boot																																																																																																																																
[sdw1] tmpfs	513440	0	513440	0%	/dev/shm																																																																																																																																
[sdw1] /dev/sdb1	52402500	12256288	40146212	24%	/data																																																																																																																																

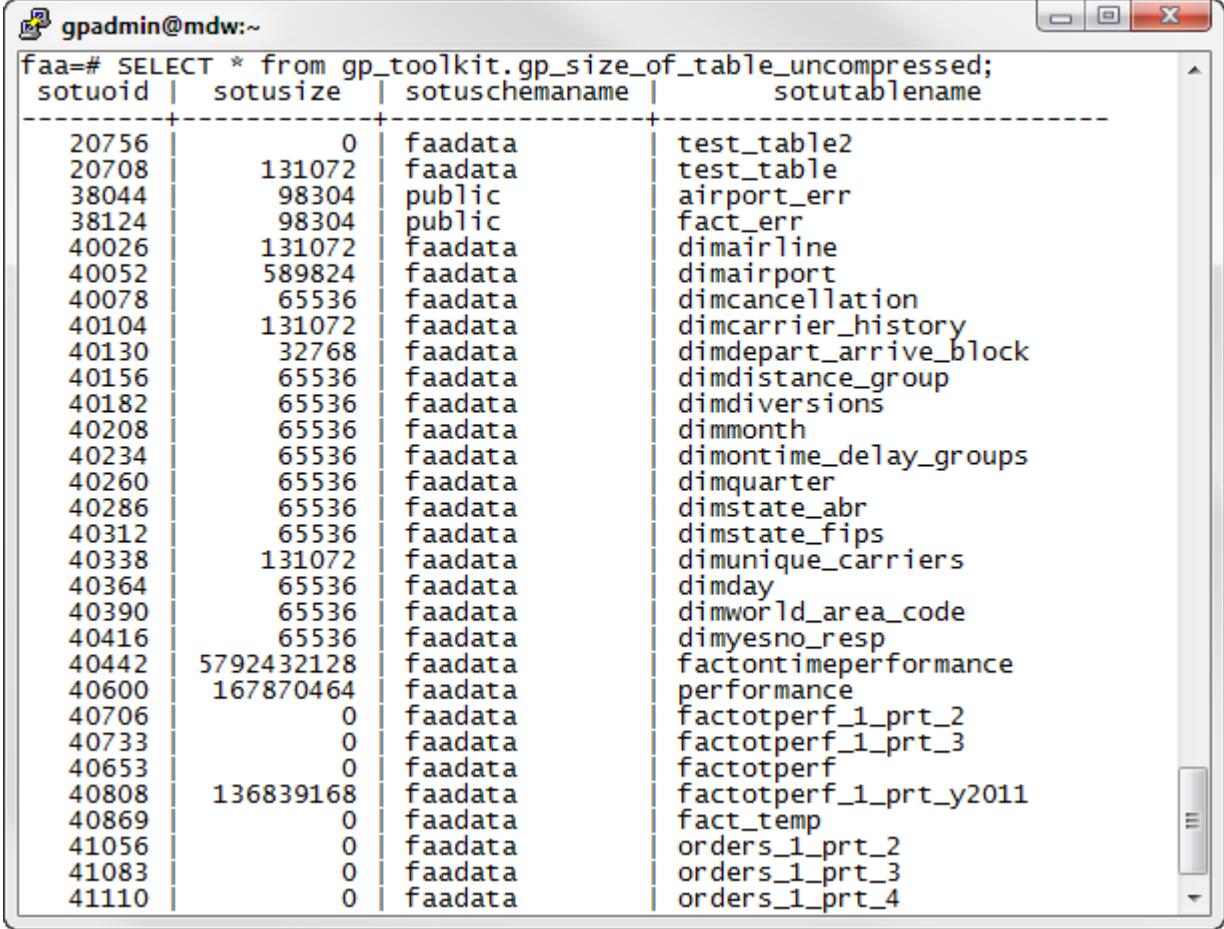
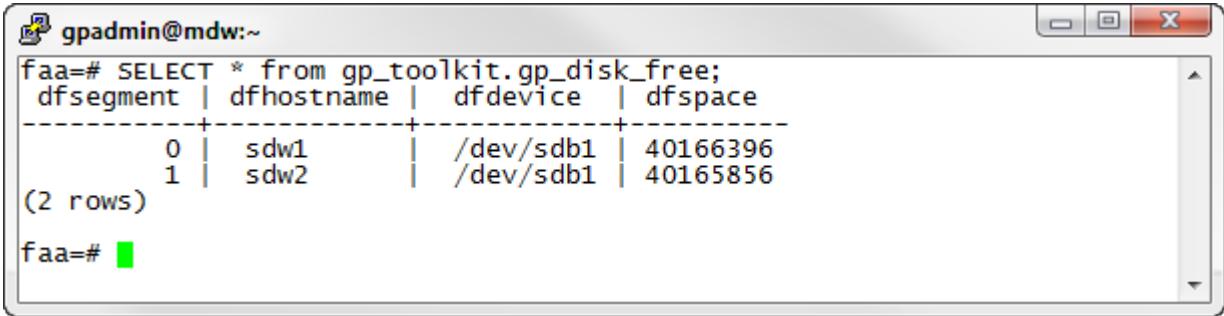
Step	Action
12.	<p>After a database has been running for a long time, its catalog can become bloated, especially if there are a large number of UPDATEs and DELETEs to the database environment. A bloated system requires regular maintenance consisting of scheduled VACUUMs. These systems may periodically require a full VACUUM ANALYZE to clean up space.</p> <p>It is safer to run a regular VACUUM. A VACUUM FULL can slow down the performance of the system dramatically. It should be executed when in maintenance mode. To run a vacuum on only one catalog table, for example pg_class, execute the following command:</p> <pre data-bbox="316 502 1475 536">[gpadmin@mdw ~]\$ psql faa -c 'vacuum analyze pg_catalog.pg_class;'</pre>  <p>The screenshot shows a terminal window titled 'gpadmin@mdw:~'. The command entered is 'psql faa -c 'vacuum analyze pg_catalog.pg_class;''. The output shows 'VACUUM' being executed. The terminal window has a standard Windows-style title bar and scroll bars.</p> <p>The pg_class table keeps the object names in the database. If there are a lot of updates to the table, such as with creating and dropping objects, the table can become bloated and negatively impact the system.</p> <p><b>Note:</b> You do not directly make changes to this table. This table is affected by updates in your environment, such as when you create or delete a table.</p>

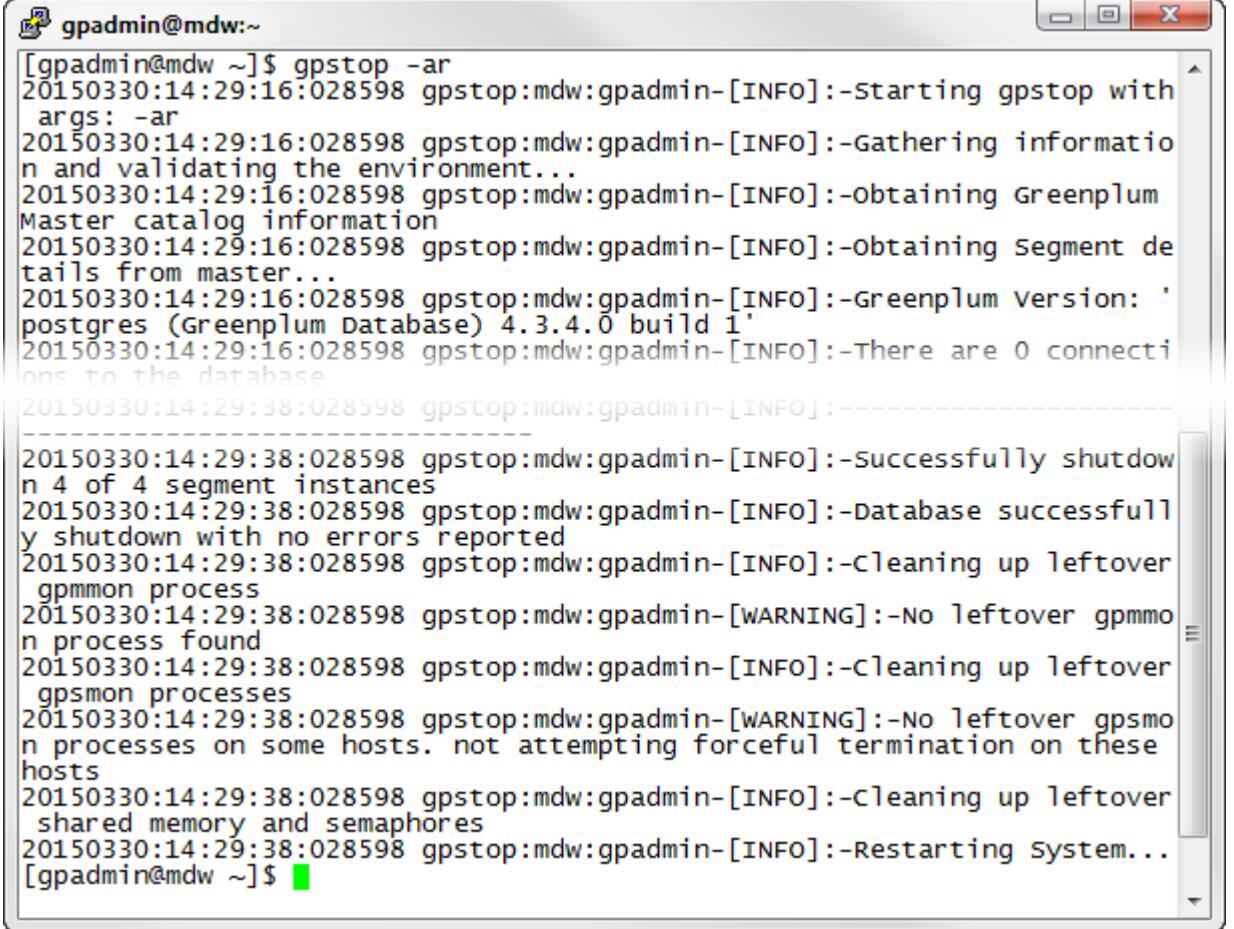
Step	Action
13.	<p>To run a VACUUM ANALYZE on all catalog tables create the following script and execute it.</p> <ol style="list-style-type: none"> <li>Using vi, create a new file called <b>gp_vacuum_analyze</b> with the following content:</li> </ol> <pre>#!/bin/bash DBNAME="faa" VCOMMAND="VACUUM ANALYZE " psql -tc "select '\$VCOMMAND'    ' pg_catalog.'    relname    ';' from pg_class a,pg_namespace b where a.relnamespace=b.oid and b.nspname= 'pg_catalog' and a.relkind='r'" \$DBNAME   psql -a \$DBNAME</pre> <ol style="list-style-type: none"> <li>Save the file and exit vi.</li> </ol>  <pre>gpadmin@mdw:~ #!/bin/bash DBNAME="faa" VCOMMAND="VACUUM ANALYZE " psql -tc "select '\$VCOMMAND'    ' pg_catalog.'    relname    ';' from pg_class a,pg_namespace b where a.relnamespace=b.oid and b.nspname= 'pg_catalog' and a.relkind='r'" \$DBNAME   psql -a \$DBNAME ~:wq!</pre> <ol style="list-style-type: none"> <li>Change the permissions on the file so that it is executable:</li> </ol> <pre>chmod 755 gp_vacuum_analyze</pre> <ol style="list-style-type: none"> <li>Execute the script by typing the following:</li> </ol> <pre>./gp_vacuum_analyze</pre>

Step	Action
	 gpadmin@mdw:~ [gpadmin@mdw ~]\$ vi gp_vacuum_analyze [gpadmin@mdw ~]\$ chmod 755 gp_vacuum_analyze [gpadmin@mdw ~]\$ ./gp_vacuum_analyze VACUUM ANALYZE pg_catalog.pg_san_configuration; VACUUM ANALYZE pg_catalog.pg_verification_history; VACUUM ANALYZE pg_catalog.pg_window; VACUUM ANALYZE pg_catalog.pg_exttable; VACUUM ANALYZE pg_catalog.pg_appendonly; VACUUM ANALYZE pg_catalog.pg_appendonly_alter_column; VACUUM ANALYZE pg_catalog.pg_fastsequence; VACUUM ANALYZE pg_catalog.pg_extprotocol; VACUUM ANALYZE pg_catalog.pg_partition; VACUUM ANALYZE pg_catalog.pg_partition_rule; VACUUM ANALYZE pg_catalog.pg_filespace; VACUUM ANALYZE pg_catalog.pg_filespace_entry; VACUUM ANALYZE pg_catalog.pg_global_sequence; VACUUM ANALYZE pg_catalog.pg_persistent_relation_node; VACUUM ANALYZE pg_catalog.pg_relation_node; VACUUM
	<p><b>Note:</b> Recall that it is recommended that you execute VACUUM on a regular schedule and ANALYZE separately on its own schedule.</p>
14.	Open a PSQL session and log in as <b>gpadmin</b> to the <b>faa</b> database. <pre>[gpadmin@mdw ~]\$ psql faa</pre>
15.	Check the size of the database using the following command: <pre>faa=# SELECT * from gp_toolkit.gp_size_of_database;</pre>  gpadmin@mdw:~ <pre>faa=# SELECT * from gp_toolkit.gp_size_of_database; sodddatname   sodddatsize -----+----- gpadmin        29851660 dbstudent      3484062006 gpperfmon      262602926 dbbackup       30474252 names          35651806 faa            6133416110 (6 rows)</pre>

Step	Action																																																																																										
16.	<p>Retrieve the size of each relational table in user schemas of the user database.</p> <pre>faa=# SELECT * from gp_toolkit.gp_size_of_table_disk;</pre>  <table border="1"> <thead> <tr> <th>sotdoid</th> <th>sotdsize</th> <th>sotdtoastsize</th> <th>sotdadditionalsize</th> <th>sotdschemaname</th> <th>sotdtblname</th> </tr> </thead> <tbody> <tr><td>20756</td><td>0</td><td>0</td><td>0</td><td>faadata</td><td>  test_table2</td></tr> <tr><td>20708</td><td>32768</td><td>98304</td><td>0</td><td>faadata</td><td>  test_table</td></tr> <tr><td>38044</td><td>0</td><td>98304</td><td>0</td><td>public</td><td>  airport_err</td></tr> <tr><td>38124</td><td>0</td><td>98304</td><td>0</td><td>public</td><td>  fact_err</td></tr> <tr><td>40026</td><td>131072</td><td>0</td><td>0</td><td>faadata</td><td>  dimairline</td></tr> <tr><td>40052</td><td>589824</td><td>0</td><td>0</td><td>faadata</td><td>  dimairport</td></tr> <tr><td>40078</td><td>65536</td><td>0</td><td>0</td><td>faadata</td><td>  dimcancellation</td></tr> <tr><td>40104</td><td>131072</td><td>0</td><td>0</td><td>faadata</td><td>  dimcarrier_history</td></tr> <tr><td>40130</td><td>32768</td><td>0</td><td>0</td><td>faadata</td><td>  dimdepart_arrive_block</td></tr> <tr><td>40156</td><td>65536</td><td>0</td><td>0</td><td>faadata</td><td>  dimdistance_group</td></tr> <tr><td>40182</td><td>65536</td><td>0</td><td>0</td><td>faadata</td><td>  dimdiversions</td></tr> <tr><td>40208</td><td>65536</td><td>0</td><td>0</td><td>faadata</td><td>  dimmonth</td></tr> <tr><td>40234</td><td>65536</td><td>0</td><td>0</td><td>faadata</td><td>  dimontime_delay_groups</td></tr> <tr><td>40260</td><td>65536</td><td>0</td><td>0</td><td>faadata</td><td>  dimquarter</td></tr> </tbody> </table>	sotdoid	sotdsize	sotdtoastsize	sotdadditionalsize	sotdschemaname	sotdtblname	20756	0	0	0	faadata	test_table2	20708	32768	98304	0	faadata	test_table	38044	0	98304	0	public	airport_err	38124	0	98304	0	public	fact_err	40026	131072	0	0	faadata	dimairline	40052	589824	0	0	faadata	dimairport	40078	65536	0	0	faadata	dimcancellation	40104	131072	0	0	faadata	dimcarrier_history	40130	32768	0	0	faadata	dimdepart_arrive_block	40156	65536	0	0	faadata	dimdistance_group	40182	65536	0	0	faadata	dimdiversions	40208	65536	0	0	faadata	dimmonth	40234	65536	0	0	faadata	dimontime_delay_groups	40260	65536	0	0	faadata	dimquarter
sotdoid	sotdsize	sotdtoastsize	sotdadditionalsize	sotdschemaname	sotdtblname																																																																																						
20756	0	0	0	faadata	test_table2																																																																																						
20708	32768	98304	0	faadata	test_table																																																																																						
38044	0	98304	0	public	airport_err																																																																																						
38124	0	98304	0	public	fact_err																																																																																						
40026	131072	0	0	faadata	dimairline																																																																																						
40052	589824	0	0	faadata	dimairport																																																																																						
40078	65536	0	0	faadata	dimcancellation																																																																																						
40104	131072	0	0	faadata	dimcarrier_history																																																																																						
40130	32768	0	0	faadata	dimdepart_arrive_block																																																																																						
40156	65536	0	0	faadata	dimdistance_group																																																																																						
40182	65536	0	0	faadata	dimdiversions																																																																																						
40208	65536	0	0	faadata	dimmonth																																																																																						
40234	65536	0	0	faadata	dimontime_delay_groups																																																																																						
40260	65536	0	0	faadata	dimquarter																																																																																						
17.	<p>Retrieve the size of all user schemas in the user database.</p> <pre>faa=# SELECT * from gp_toolkit.gp_size_of_schema_disk;</pre>  <table border="1"> <thead> <tr> <th>sosdnsp</th> <th>sosdschematablesize</th> <th>sosdschemaidxsize</th> </tr> </thead> <tbody> <tr><td>public</td><td>196608</td><td>0</td></tr> <tr><td>faadata</td><td>6099009536</td><td>229376</td></tr> </tbody> </table>	sosdnsp	sosdschematablesize	sosdschemaidxsize	public	196608	0	faadata	6099009536	229376																																																																																	
sosdnsp	sosdschematablesize	sosdschemaidxsize																																																																																									
public	196608	0																																																																																									
faadata	6099009536	229376																																																																																									

Step	Action
18.	<p>Retrieve the size of all user indexes in the user database.</p> <pre>faa=# SELECT * from gp_toolkit(gp_size_of_index);</pre>  <pre>Faa=# SELECT * from gp_toolkit(gp_size_of_index); soioid   soitableoid   soisize   soiindexschemaname   soiindexname soitableschemaname   soitablelename -----+-----+-----+-----+ 20736   20708   131072   faadata   test_table_pkey faadata   test_table 20785   20756   98304   faadata   test_index faadata   test_table2 (2 rows)  faa=#</pre>
19.	<p>Retrieve the cumulative size of all user indexes in a table, list this total size for each table in the user database.</p> <pre>faa=# SELECT * from gp_toolkit(gp_size_of_all_table_indexes);</pre>  <pre>Faa=# SELECT * from gp_toolkit(gp_size_of_all_table_indexes); soatioid   soatisize   soatischemaname   soatitablename -----+-----+-----+ 20756   98304   faadata   test_table2 20708   131072   faadata   test_table (2 rows)  faa=#</pre>

Step	Action																																																																																																																												
20.	<p>Retrieve the uncompressed table size.</p> <pre>faa=# SELECT * from gp_toolkit(gp_size_of_table_uncompressed;</pre>  <table border="1"> <thead> <tr> <th>sotuoid</th> <th>sotusize</th> <th>sotuschemaname</th> <th>sotutablename</th> </tr> </thead> <tbody> <tr><td>20756</td><td>0</td><td>faadata</td><td>test_table2</td></tr> <tr><td>20708</td><td>131072</td><td>faadata</td><td>test_table</td></tr> <tr><td>38044</td><td>98304</td><td>public</td><td>airport_err</td></tr> <tr><td>38124</td><td>98304</td><td>public</td><td>fact_err</td></tr> <tr><td>40026</td><td>131072</td><td>faadata</td><td>dimairline</td></tr> <tr><td>40052</td><td>589824</td><td>faadata</td><td>dimairport</td></tr> <tr><td>40078</td><td>65536</td><td>faadata</td><td>dimcancellation</td></tr> <tr><td>40104</td><td>131072</td><td>faadata</td><td>dimcarrier_history</td></tr> <tr><td>40130</td><td>32768</td><td>faadata</td><td>dimdepart_arrive_block</td></tr> <tr><td>40156</td><td>65536</td><td>faadata</td><td>dimdistance_group</td></tr> <tr><td>40182</td><td>65536</td><td>faadata</td><td>dimdiversions</td></tr> <tr><td>40208</td><td>65536</td><td>faadata</td><td>dimmonth</td></tr> <tr><td>40234</td><td>65536</td><td>faadata</td><td>dimontime_delay_groups</td></tr> <tr><td>40260</td><td>65536</td><td>faadata</td><td>dimquarter</td></tr> <tr><td>40286</td><td>65536</td><td>faadata</td><td>dimstate_abr</td></tr> <tr><td>40312</td><td>65536</td><td>faadata</td><td>dimstate_fips</td></tr> <tr><td>40338</td><td>131072</td><td>faadata</td><td>dimunique_carriers</td></tr> <tr><td>40364</td><td>65536</td><td>faadata</td><td>dimday</td></tr> <tr><td>40390</td><td>65536</td><td>faadata</td><td>dimworld_area_code</td></tr> <tr><td>40416</td><td>65536</td><td>faadata</td><td>dimyesno_resp</td></tr> <tr><td>40442</td><td>5792432128</td><td>faadata</td><td>factontimeperformance</td></tr> <tr><td>40600</td><td>167870464</td><td>faadata</td><td>performance</td></tr> <tr><td>40706</td><td>0</td><td>faadata</td><td>factotperf_1_prt_2</td></tr> <tr><td>40733</td><td>0</td><td>faadata</td><td>factotperf_1_prt_3</td></tr> <tr><td>40653</td><td>0</td><td>faadata</td><td>factotperf</td></tr> <tr><td>40808</td><td>136839168</td><td>faadata</td><td>factotperf_1_prt_y2011</td></tr> <tr><td>40869</td><td>0</td><td>faadata</td><td>fact_temp</td></tr> <tr><td>41056</td><td>0</td><td>faadata</td><td>orders_1_prt_2</td></tr> <tr><td>41083</td><td>0</td><td>faadata</td><td>orders_1_prt_3</td></tr> <tr><td>41110</td><td>0</td><td>faadata</td><td>orders_1_prt_4</td></tr> </tbody> </table>	sotuoid	sotusize	sotuschemaname	sotutablename	20756	0	faadata	test_table2	20708	131072	faadata	test_table	38044	98304	public	airport_err	38124	98304	public	fact_err	40026	131072	faadata	dimairline	40052	589824	faadata	dimairport	40078	65536	faadata	dimcancellation	40104	131072	faadata	dimcarrier_history	40130	32768	faadata	dimdepart_arrive_block	40156	65536	faadata	dimdistance_group	40182	65536	faadata	dimdiversions	40208	65536	faadata	dimmonth	40234	65536	faadata	dimontime_delay_groups	40260	65536	faadata	dimquarter	40286	65536	faadata	dimstate_abr	40312	65536	faadata	dimstate_fips	40338	131072	faadata	dimunique_carriers	40364	65536	faadata	dimday	40390	65536	faadata	dimworld_area_code	40416	65536	faadata	dimyesno_resp	40442	5792432128	faadata	factontimeperformance	40600	167870464	faadata	performance	40706	0	faadata	factotperf_1_prt_2	40733	0	faadata	factotperf_1_prt_3	40653	0	faadata	factotperf	40808	136839168	faadata	factotperf_1_prt_y2011	40869	0	faadata	fact_temp	41056	0	faadata	orders_1_prt_2	41083	0	faadata	orders_1_prt_3	41110	0	faadata	orders_1_prt_4
sotuoid	sotusize	sotuschemaname	sotutablename																																																																																																																										
20756	0	faadata	test_table2																																																																																																																										
20708	131072	faadata	test_table																																																																																																																										
38044	98304	public	airport_err																																																																																																																										
38124	98304	public	fact_err																																																																																																																										
40026	131072	faadata	dimairline																																																																																																																										
40052	589824	faadata	dimairport																																																																																																																										
40078	65536	faadata	dimcancellation																																																																																																																										
40104	131072	faadata	dimcarrier_history																																																																																																																										
40130	32768	faadata	dimdepart_arrive_block																																																																																																																										
40156	65536	faadata	dimdistance_group																																																																																																																										
40182	65536	faadata	dimdiversions																																																																																																																										
40208	65536	faadata	dimmonth																																																																																																																										
40234	65536	faadata	dimontime_delay_groups																																																																																																																										
40260	65536	faadata	dimquarter																																																																																																																										
40286	65536	faadata	dimstate_abr																																																																																																																										
40312	65536	faadata	dimstate_fips																																																																																																																										
40338	131072	faadata	dimunique_carriers																																																																																																																										
40364	65536	faadata	dimday																																																																																																																										
40390	65536	faadata	dimworld_area_code																																																																																																																										
40416	65536	faadata	dimyesno_resp																																																																																																																										
40442	5792432128	faadata	factontimeperformance																																																																																																																										
40600	167870464	faadata	performance																																																																																																																										
40706	0	faadata	factotperf_1_prt_2																																																																																																																										
40733	0	faadata	factotperf_1_prt_3																																																																																																																										
40653	0	faadata	factotperf																																																																																																																										
40808	136839168	faadata	factotperf_1_prt_y2011																																																																																																																										
40869	0	faadata	fact_temp																																																																																																																										
41056	0	faadata	orders_1_prt_2																																																																																																																										
41083	0	faadata	orders_1_prt_3																																																																																																																										
41110	0	faadata	orders_1_prt_4																																																																																																																										
21.	<p>Retrieve the total free disk space in Kbytes for each segment server and the file systems included as part of the Greenplum Database.</p> <pre>faa=# SELECT * from gp_toolkit(gp_disk_free;</pre>  <table border="1"> <thead> <tr> <th>dfsegment</th> <th>dfhostname</th> <th>dfdevice</th> <th>dfspace</th> </tr> </thead> <tbody> <tr><td>0</td><td>sdw1</td><td>/dev/sdb1</td><td>40166396</td></tr> <tr><td>1</td><td>sdw2</td><td>/dev/sdb1</td><td>40165856</td></tr> </tbody> </table>	dfsegment	dfhostname	dfdevice	dfspace	0	sdw1	/dev/sdb1	40166396	1	sdw2	/dev/sdb1	40165856																																																																																																																
dfsegment	dfhostname	dfdevice	dfspace																																																																																																																										
0	sdw1	/dev/sdb1	40166396																																																																																																																										
1	sdw2	/dev/sdb1	40165856																																																																																																																										
22.	Exit the database.																																																																																																																												

Step	Action
23.	<p>Restart the database in regular mode so that all users can connect to it. The <code>-r</code> option is used to restart the database.</p> <pre>[gpadmin@mdw ~]\$ gpstop -ar</pre>  <pre>[gpadmin@mdw ~]\$ gpstop -ar 20150330:14:29:16:028598 gpstop:mdw:gpadmin-[INFO]:-Starting gpstop with args: -ar 20150330:14:29:16:028598 gpstop:mdw:gpadmin-[INFO]:-Gathering information and validating the environment... 20150330:14:29:16:028598 gpstop:mdw:gpadmin-[INFO]:-obtaining Greenplum Master catalog information 20150330:14:29:16:028598 gpstop:mdw:gpadmin-[INFO]:-obtaining Segment details from master... 20150330:14:29:16:028598 gpstop:mdw:gpadmin-[INFO]:-Greenplum version: 'postgres (Greenplum Database) 4.3.4.0 build 1' 20150330:14:29:16:028598 gpstop:mdw:gpadmin-[INFO]:-There are 0 connections to the database 20150330:14:29:38:028598 gpstop:mdw:gpadmin-[INFO]:----- 20150330:14:29:38:028598 gpstop:mdw:gpadmin-[INFO]:-successfully shutdown 4 of 4 segment instances 20150330:14:29:38:028598 gpstop:mdw:gpadmin-[INFO]:-Database successfully shutdown with no errors reported 20150330:14:29:38:028598 gpstop:mdw:gpadmin-[INFO]:-cleaning up leftover gpmmon process 20150330:14:29:38:028598 gpstop:mdw:gpadmin-[WARNING]:-No leftover gpmmon process found 20150330:14:29:38:028598 gpstop:mdw:gpadmin-[INFO]:-cleaning up leftover gpsmon processes 20150330:14:29:38:028598 gpstop:mdw:gpadmin-[WARNING]:-No leftover gpsmon processes on some hosts. not attempting forceful termination on these hosts 20150330:14:29:38:028598 gpstop:mdw:gpadmin-[INFO]:-cleaning up leftover shared memory and semaphores 20150330:14:29:38:028598 gpstop:mdw:gpadmin-[INFO]:-Restarting system... [gpadmin@mdw ~]\$</pre>
	<p><b>Summary</b></p> <p>Checking table skew after loading is one of the most important ways to validate the efficiency of your distribution keys. Check the skew often on large tables that are frequently loaded, as data demographics may have changed. This is important on all Greenplum systems regardless of size. On smaller systems, there are fewer segments involved so the distribution is not over as many units of parallelism. On larger systems the hashing algorithm, particularly on non-unique distribution keys, may hash many more rows to a single segment.</p> <p>Check the size of databases and regularly VACUUM the system to reduce the chances of bloat.</p>

*End of Lab Exercise*

This page intentionally left blank

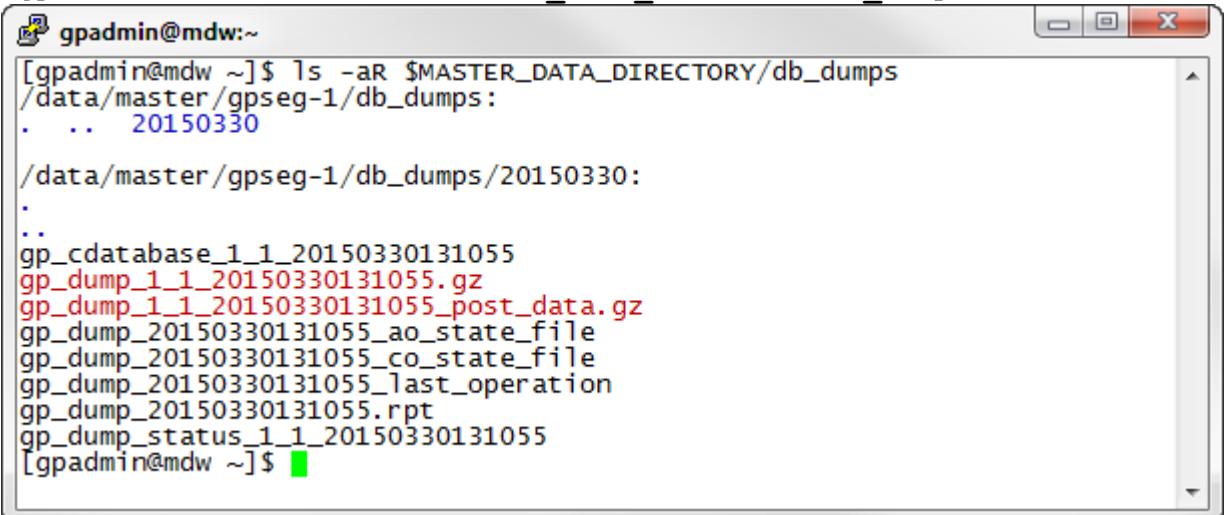
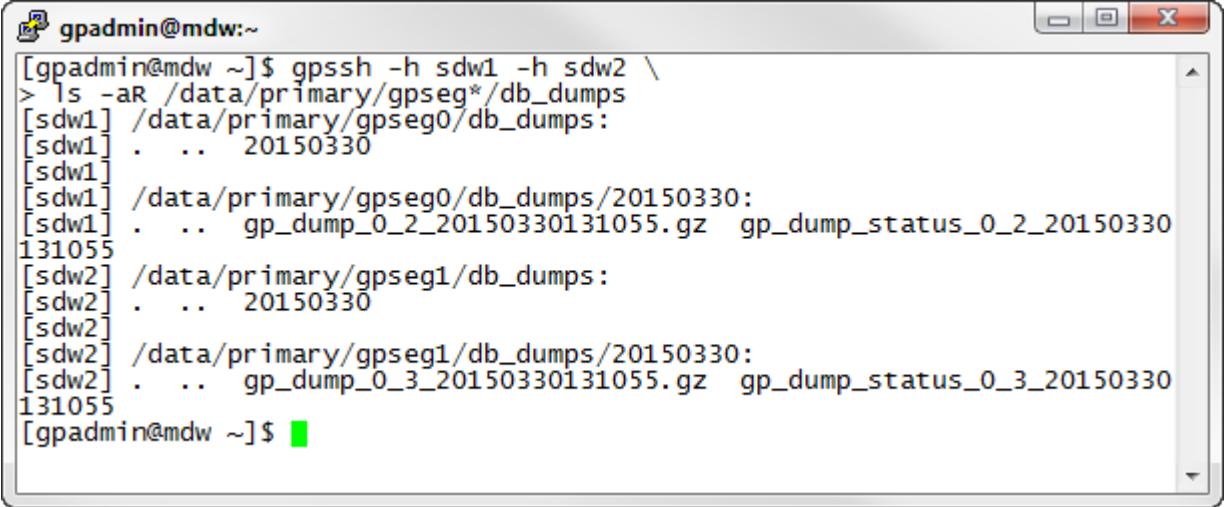
## Lab 15. Backups and Restores

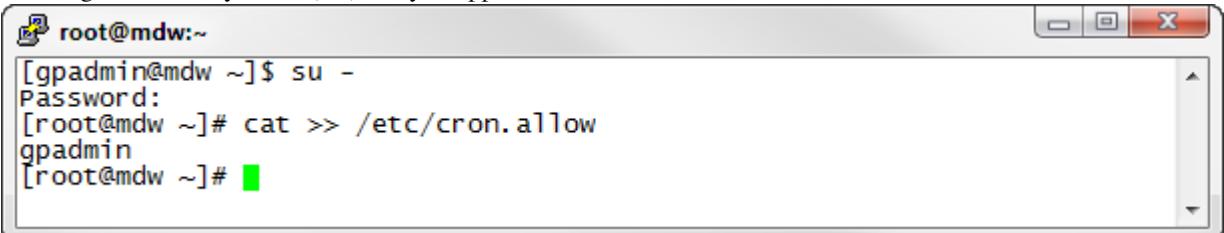
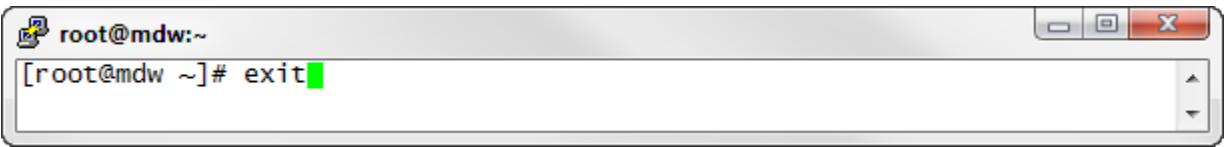
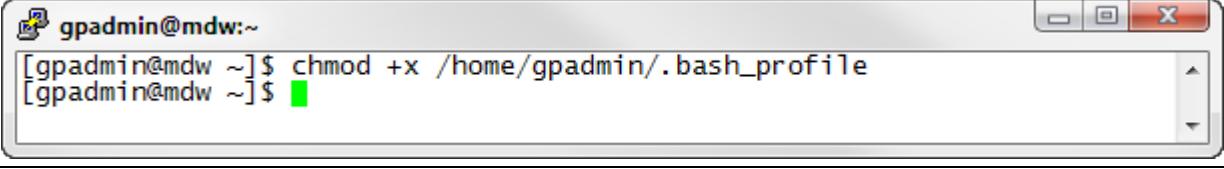
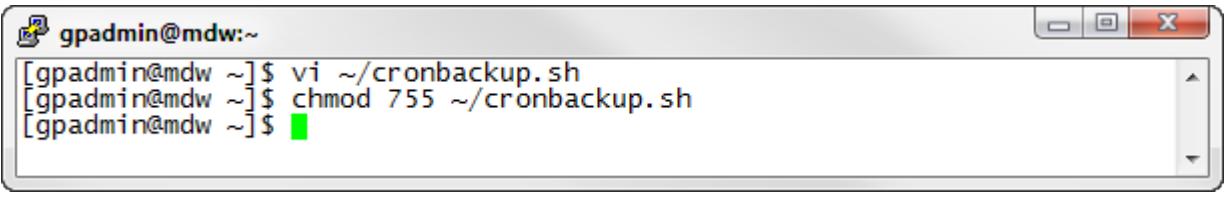
<b>Purpose:</b>	<p>In this lab, you will schedule a parallel backup operation with the <code>gpcrondump</code> command to dump the <code>faa</code> database. This will create a backup file of the master instance and each active segment instance in the Greenplum Database system.</p> <p>By default, the dump files are created in the data directory of their respective segment instance or master instance. In this exercise, you will redirect the dump files to one location to make it easier to collect and analyze them.</p> <p>You will then restore the database using the <code>gprestoredb</code> command.</p>
<b>Tasks:</b>	<p>Students perform the following task:</p> <ul style="list-style-type: none"><li>• Create and retrieve backups</li><li>• Perform incremental backups and restores</li><li>• Recover from a failed master</li></ul>
<b>References:</b>	<p>Module 6: Database Management and Archiving</p> <ul style="list-style-type: none"><li>• Lesson: Backups and Restores</li></ul>

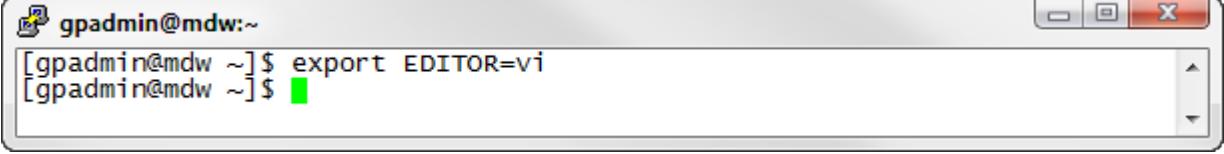
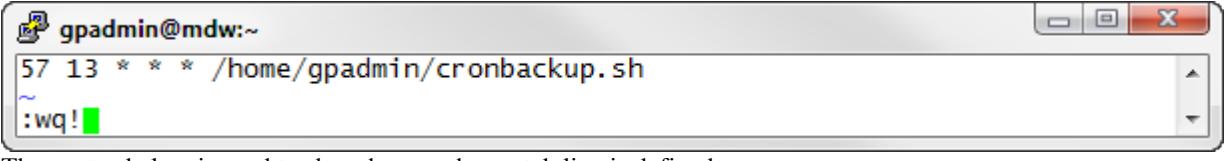
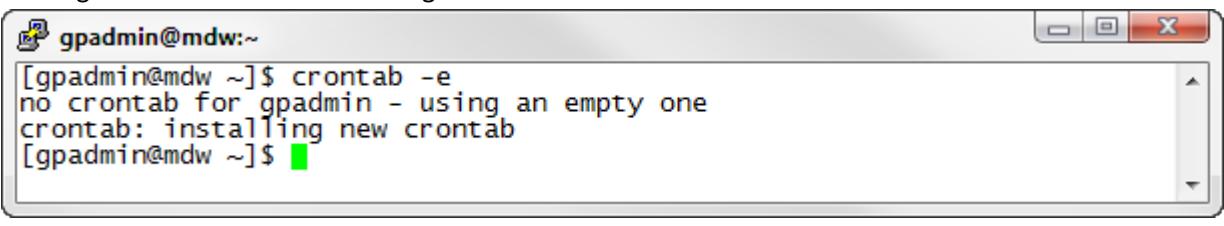
## Lab 15: Task 1 – Backups and Restores

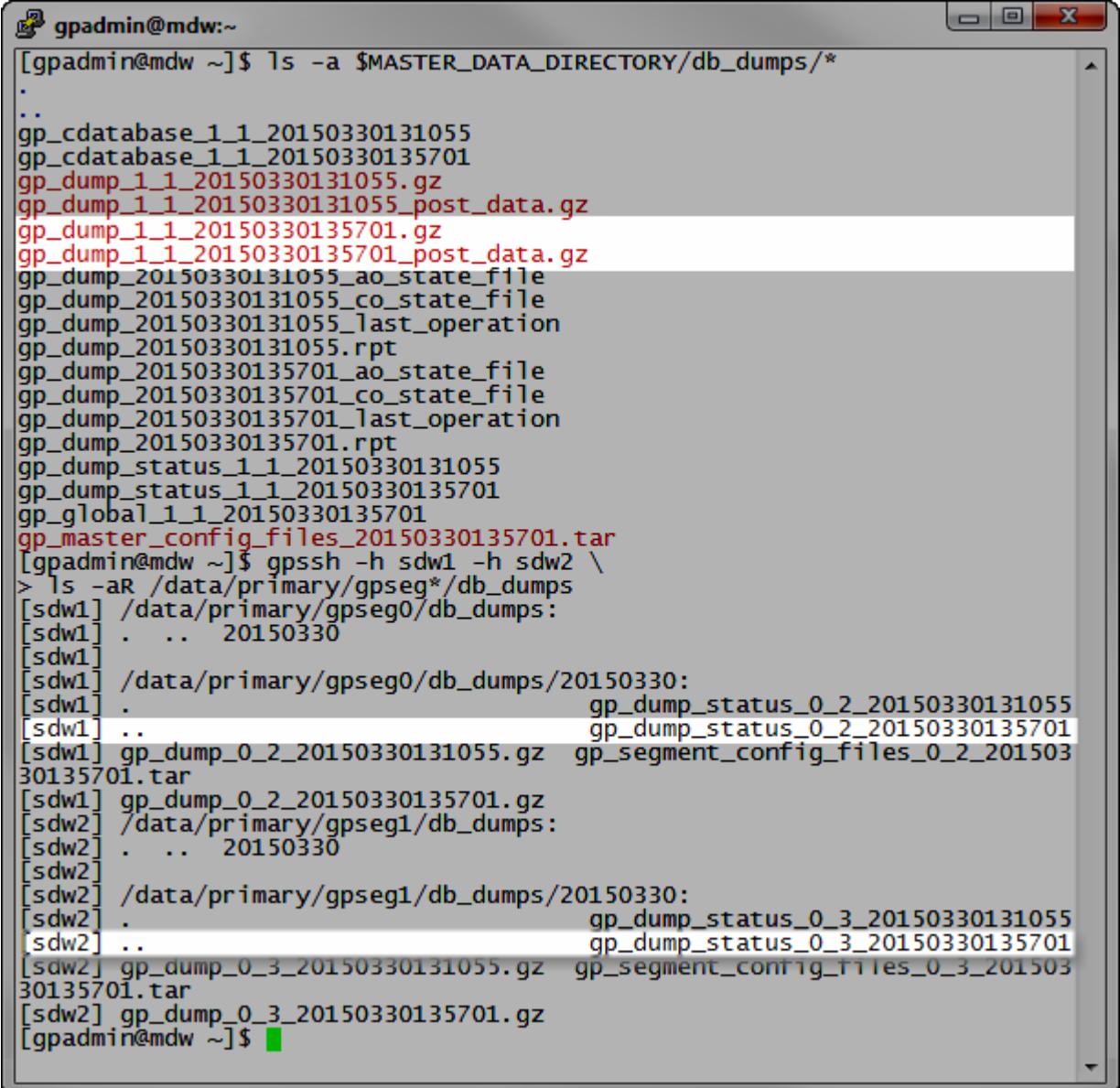
Step	Action
	<p>To create backups you will need a place to keep them. Sometimes it is a network share to a backup devise. Normally all segments will need to have this space. You can perform backups as parallel or non-parallel backups using the following:</p> <ul style="list-style-type: none"> <li>• <code>gpcrondump</code> for automatic scheduled parallel backups.</li> <li>• <code>pg_dump</code> non parallel backup that has to go through the master. (Not recommended because of slow performance) <code>pg_dump</code> and <code>pg_restore</code> is available for compatibility with standard postgres databases.</li> </ul>
1.	<p>Start an ad-hoc backup from the UNIX prompt as gpadmin:</p> <pre>[gpadmin@mdw ~]\$ gpcrondump faa</pre> <p>You will be prompted to continue. Type <code>y</code> and press Enter to proceed.</p>

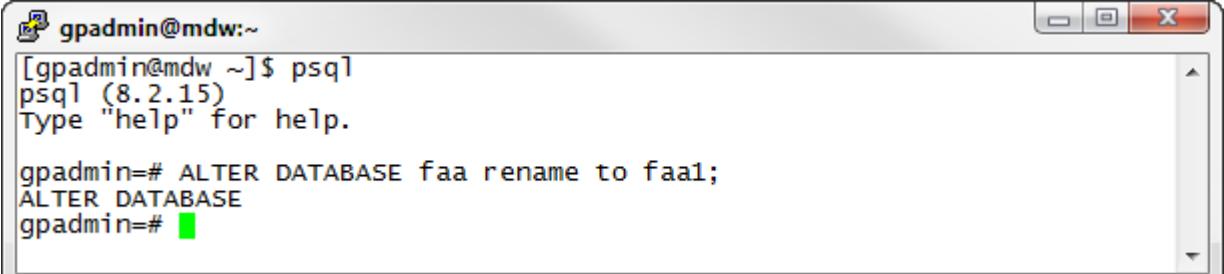
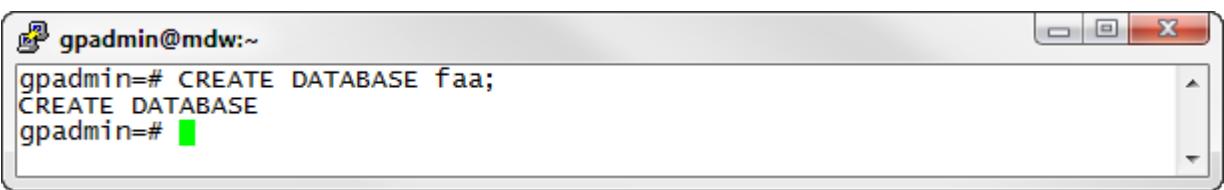
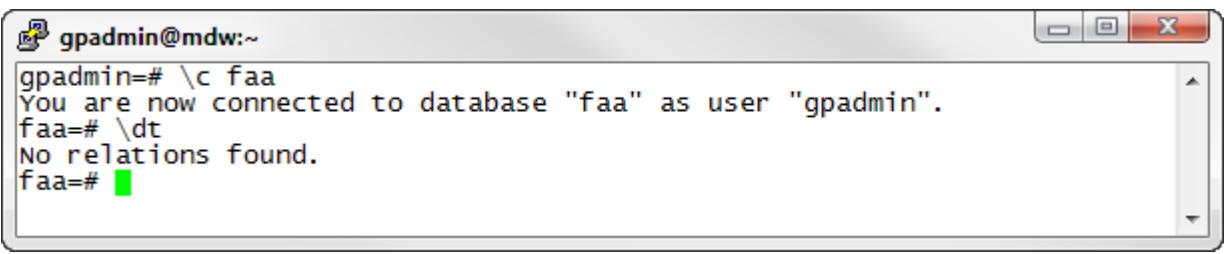
Step	Action
	 gpadmin@mdw:~ > y 20150330:13:11:55:017737 gpcrondump:mdw:gpadmin-[INFO]:-Directory /data/master/gpseg-1/db_dumps/20150330 not found, will try to create 20150330:13:11:55:017737 gpcrondump:mdw:gpadmin-[INFO]:-Created /data/master/gpseg-1/db_dumps/20150330 20150330:13:11:55:017737 gpcrondump:mdw:gpadmin-[INFO]:-Checked /data/master/gpseg-1 on master 20150330:13:11:56:017737 gpcrondump:mdw:gpadmin-[INFO]:-Configuring for single database dump 20150330:13:11:56:017737 gpcrondump:mdw:gpadmin-[INFO]:-Validating disk space 20150330:13:11:57:017737 gpcrondump:mdw:gpadmin-[INFO]:-Adding compression parameter 20150330:13:11:57:017737 gpcrondump:mdw:gpadmin-[INFO]:-Adding --no-expand-children 20150330:13:11:57:017737 gpcrondump:mdw:gpadmin-[INFO]:-Dump process command line gp_dump -p 5432 -U gpadmin --gp-d=db_dumps/20150330 --gp-r=/data/master/gpseg-1/db_dumps/20150330 --gp-s=p --gp-k=20150330131055 --no-lock --gp-c --no-expand-children gpadmin 20150330:13:11:57:017737 gpcrondump:mdw:gpadmin-[INFO]:-Starting Dump process 20150330:13:11:59:017737 gpcrondump:mdw:gpadmin-[INFO]:-Dump process returned exit code 0 20150330:13:11:59:017737 gpcrondump:mdw:gpadmin-[INFO]:-Timestamp key = 20150330131055 20150330:13:11:59:017737 gpcrondump:mdw:gpadmin-[INFO]:-Checked master status file and master dump file. 20150330:13:12:00:017737 gpcrondump:mdw:gpadmin-[INFO]:-Dump status report 20150330:13:12:00:017737 gpcrondump:mdw:gpadmin-[INFO]:----- ----- 20150330:13:12:00:017737 gpcrondump:mdw:gpadmin-[INFO]:-Target database = gpadmin 20150330:13:12:00:017737 gpcrondump:mdw:gpadmin-[INFO]:-Dump subdirectory = 20150330 20150330:13:12:00:017737 gpcrondump:mdw:gpadmin-[INFO]:-Dump type = Full database 20150330:13:12:00:017737 gpcrondump:mdw:gpadmin-[INFO]:-Clear old dump directories = off 20150330:13:12:00:017737 gpcrondump:mdw:gpadmin-[INFO]:-Dump start time = 13:10:55 20150330:13:12:00:017737 gpcrondump:mdw:gpadmin-[INFO]:-Dump end time = 13:11:59 20150330:13:12:00:017737 gpcrondump:mdw:gpadmin-[INFO]:-Status = COMPLETED 20150330:13:12:00:017737 gpcrondump:mdw:gpadmin-[INFO]:-Dump key = 20150330131055 20150330:13:12:00:017737 gpcrondump:mdw:gpadmin-[INFO]:-Dump file compression = On 20150330:13:12:00:017737 gpcrondump:mdw:gpadmin-[INFO]:-Vacuum mode type = off 20150330:13:12:00:017737 gpcrondump:mdw:gpadmin-[INFO]:-Exit code zero, no warnings generated 20150330:13:12:00:017737 gpcrondump:mdw:gpadmin-[INFO]:----- ----- 20150330:13:12:00:017737 gpcrondump:mdw:gpadmin-[WARNING]:-Found neither /usr/local/greenplum-db/.bin/mail_contacts nor /home/gpadmin/mail_contacts 20150330:13:12:00:017737 gpcrondump:mdw:gpadmin-[WARNING]:-Unable to send dump email notification 20150330:13:12:00:017737 gpcrondump:mdw:gpadmin-[INFO]:-To enable email notification, create /usr/local/greenplum-db/.bin/mail_contacts or /home/gpadmin/mail_contacts containing required email addresses [gpadmin@mdw ~]\$

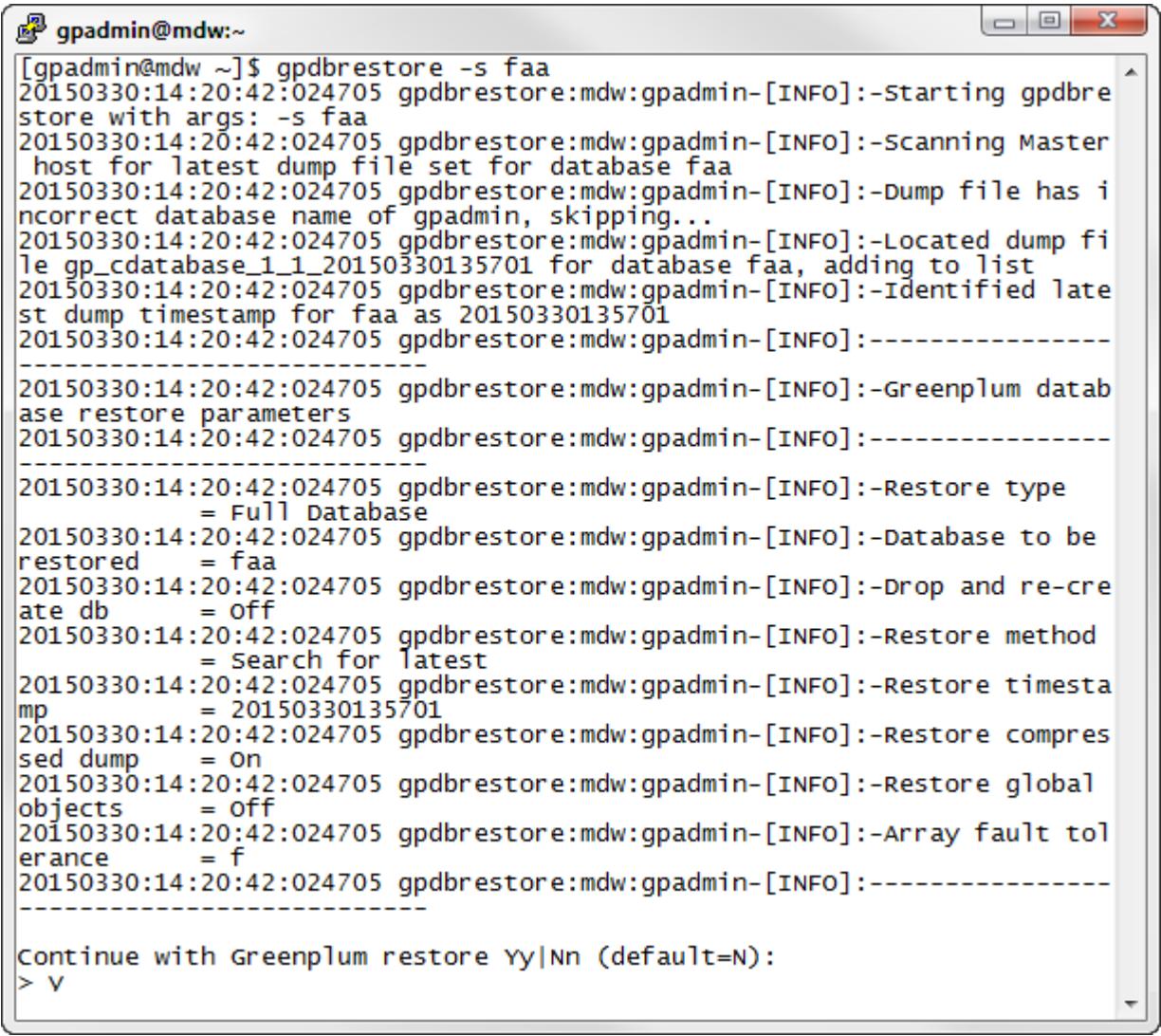
Step	Action
2.	<p>The backup files will be in the <code>data/db_dumps</code> directory on the master and all primary segments.</p> <pre>[gpadmin@mdw ~]\$ ls -aR \$MASTER_DATA_DIRECTORY/db_dumps</pre>  <p>All backup files are stored in a timestamped directory for the day that the backup was started. In this example, the directory, 20150330, is created when the first backup is executed on March 30<sup>th</sup>, 2015. Any subsequent backups performed on that day are stored to the same directory.</p>
3.	<p>Verify the corresponding backup files exist for the segments.</p> <pre>[gpadmin@mdw ~]\$ gpssh -h sdw1 -h sdw2 \ ls -aR /data/primary/gpseg*/db_dumps</pre> 

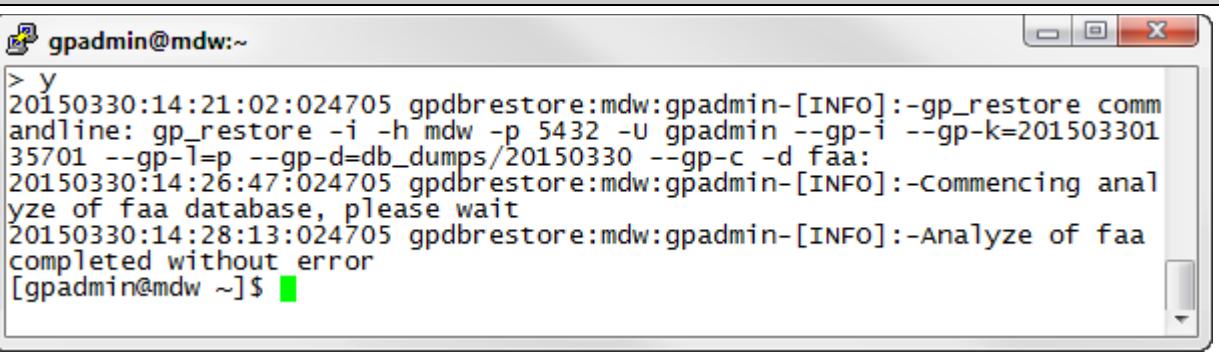
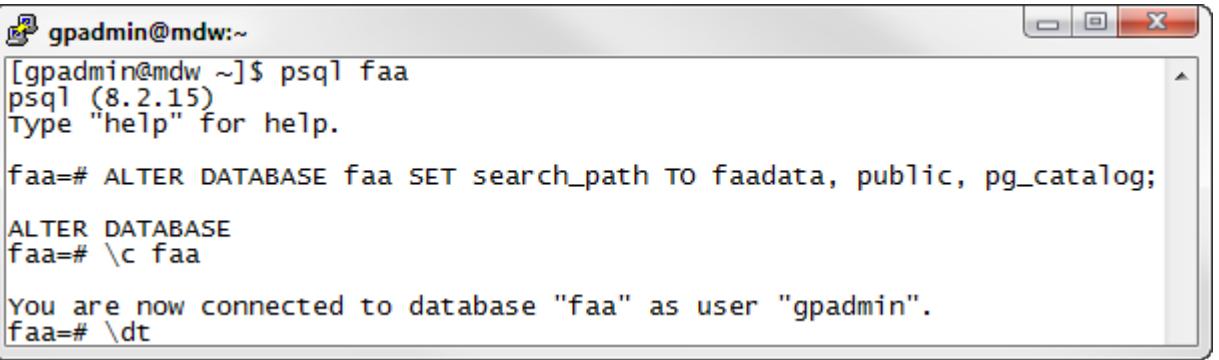
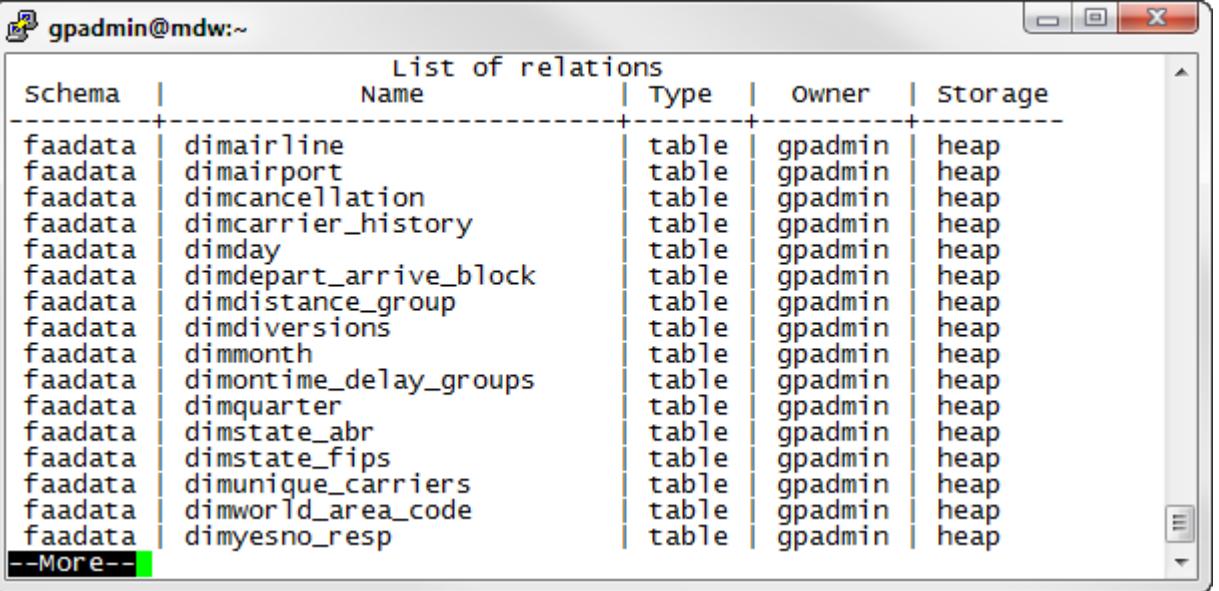
Step	Action
4.	<p>Over the next few steps, you will configure the environment so that gpcrondump is automated through cron. The crontab utility determines whether or not a user has appropriate permission to run a program at a particular point in time by checking the file /etc/cron.allow. A user must be explicitly included to this file to be able to use the crontab.</p> <p>As root, add the gpadmin account to the /etc/cron.allow file in a line by itself. In this example, the cat command is used to add a line to the end of the file. If the file does not exist, it will be created. If it does exist, the double greater than symbols (&gt;&gt;) lets you append to the file.</p>  <pre data-bbox="279 375 1503 608">root@mdw:~ [gpadmin@mdw ~]\$ su - Password: [root@mdw ~]# cat &gt;&gt; /etc/cron.allow gpadmin [root@mdw ~]#</pre> <p>To finish modifying the file with the cat command, hit CTRL-D.</p>
5.	<p>Exit from the root account, back to the gpadmin account.</p> <pre data-bbox="279 671 1503 903">[root@mdw ~]# exit</pre>  <pre data-bbox="279 756 1503 903">root@mdw:~ [root@mdw ~]# exit</pre>
6.	<p>Ensure that the permissions of .bash_profile file for the gpadmin user include the execute permission so that cron can properly access and execute the login script:</p> <pre data-bbox="279 988 1503 1220">[gpadmin@mdw ~]\$ chmod +x /home/gpadmin/.bash_profile</pre>  <pre data-bbox="279 1051 1503 1220">gpadmin@mdw:~ [gpadmin@mdw ~]\$ chmod +x /home/gpadmin/.bash_profile [gpadmin@mdw ~]\$</pre>
7.	<p>Using vi, add the following lines to the file /home/gpadmin/cronbackup.sh:</p> <pre data-bbox="372 1262 1503 1326">source /home/gpadmin/.bash_profile gpcrondump -x faa -c -g -G -a -q &gt;&gt; /tmp/gpcrondump.log</pre>  <pre data-bbox="279 1326 1503 1537">gpadmin@mdw:~ source /home/gpadmin/.bash_profile gpcrondump -x faa -c -g -G -a -q &gt;&gt; /tmp/gpcrondump.log ~ :wq!</pre>
8.	<p>Change the permissions on cronbackup.sh to 755 so that it is readable and executable by cron.</p> <pre data-bbox="279 1579 1503 1643">[gpadmin@mdw ~]\$ chmod 755 cronbackup.sh</pre>  <pre data-bbox="279 1643 1503 1854">gpadmin@mdw:~ [gpadmin@mdw ~]\$ vi ~/cronbackup.sh [gpadmin@mdw ~]\$ chmod 755 ~/cronbackup.sh [gpadmin@mdw ~]\$</pre>

Step	Action
9.	<p>Update the <b>EDITOR</b> environment variable to <b>vi</b> and export it. When modifying your cron jobs, the editor defined by the <b>EDITOR</b> variable will be used. If you are more familiar with and prefer to use Emacs, you can replace <b>vi</b> shown here with <b>emacs</b>:</p> <pre data-bbox="279 255 861 285">[gpadmin@mdw ~]\$ export EDITOR=vi</pre> 
10.	<p>You will set <b>cron</b> to execute the script you created, <b>cronbackup.sh</b>, five (5) minutes from the time you record in this step. Use the <b>date</b> command to obtain the current time:</p> <pre data-bbox="279 608 652 637">[gpadmin@mdw ~]\$ date</pre> 
11.	<p>Edit the crontab and add a line to execute the script you created, <b>cronbackup.sh</b>, five minutes from the time you recorded earlier:</p> <pre data-bbox="279 973 758 1003">[gpadmin@mdw ~]\$ crontab -e</pre>  <p>The syntax below is used to show how each crontab line is defined:</p> <pre data-bbox="279 1199 1111 1516"> 17 6 * * * /home/gpadmin/cronbackup.sh *   *   *   *   command to be executed -   -   -   -                                     +---- day of week (0 - 7) (Sunday=0 or 7)         +----- month (1 - 12)     +----- day of month (1 - 31) +----- hour (0 - 23) +----- min (0 - 59) </pre> <p>Save your changes and exit the editor. Once you have modified your crontab, you should receive a message that a new crontab is being installed.</p> 

Step	Action
12.	<p>Wait for the time to pass for the job to execute and look for the backup files. Execute the following commands where you should see a second set of backups with a new timestamp:</p> <pre>[gpadmin@mdw ~]\$ ls -a \$MASTER_DATA_DIRECTORY/db_dumps/* [gpadmin@mdw ~]\$ gpssh -h sdw1 -h sdw2 \ ls -aR /data/primary/gpseg*/db_dumps</pre>  <p>If the backup did not execute because time passed before you saved the crontab, you can execute the /home/gpadmin/cronbackup.sh script manually with the command, /bin/bash /home/gpadmin/cronbackup.sh.</p> <p>Do not move to the next step until the backup process has completed.</p>

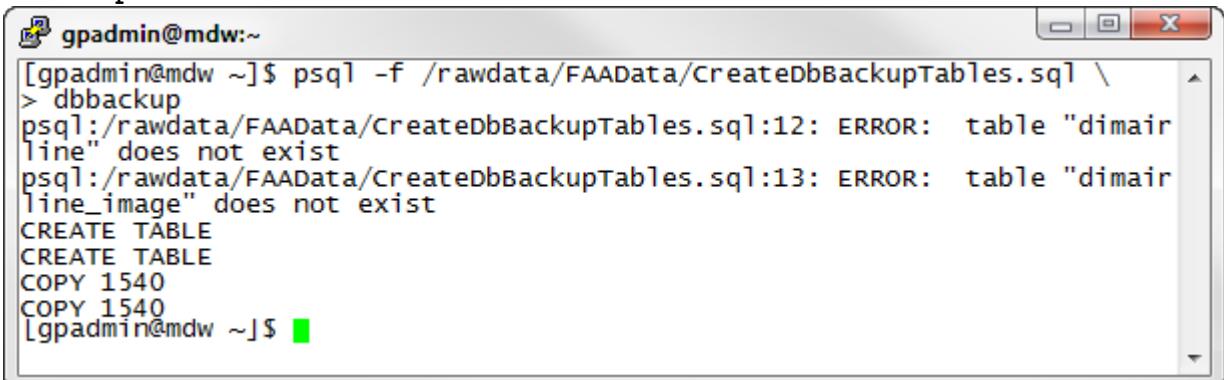
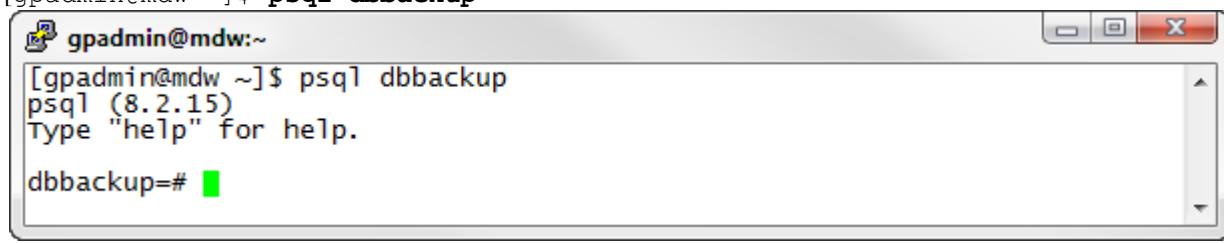
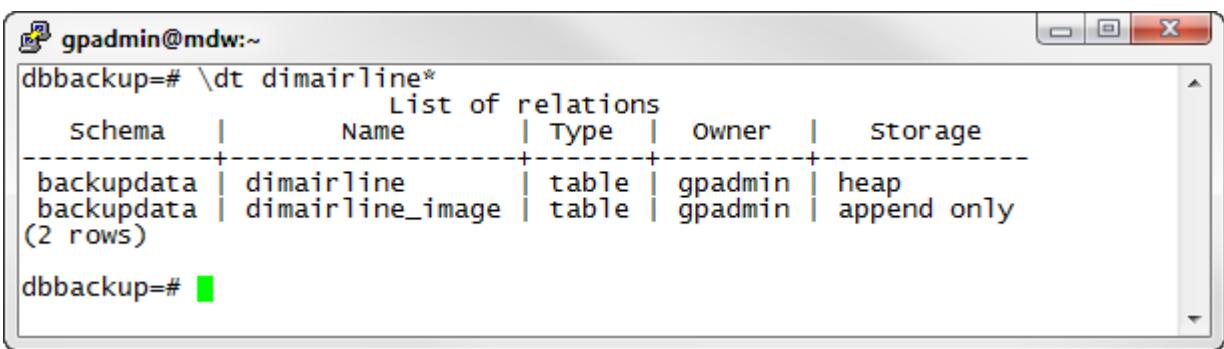
Step	Action
13.	<p>Over the next few steps, you will recover the database from the backup you created. Connect to the gpadmin database as the gpadmin user, if not already connected:</p> <pre>[gpadmin@mdw ~] \$ psql</pre> <p>Rename the faa database:</p> <pre>gpadmin=# ALTER DATABASE faa rename to faa1;</pre> 
14.	<p>Recreate the faa database. You will use this database to recover the data stored in the backup you created.</p> <pre>gpadmin=# CREATE DATABASE faa;</pre>  <p>Connect to the faa database you created and list the tables in the database:</p> <pre>gpadmin=# \c faa faa=# \dt</pre> 
15.	<p>Exit your PSQL session.</p>

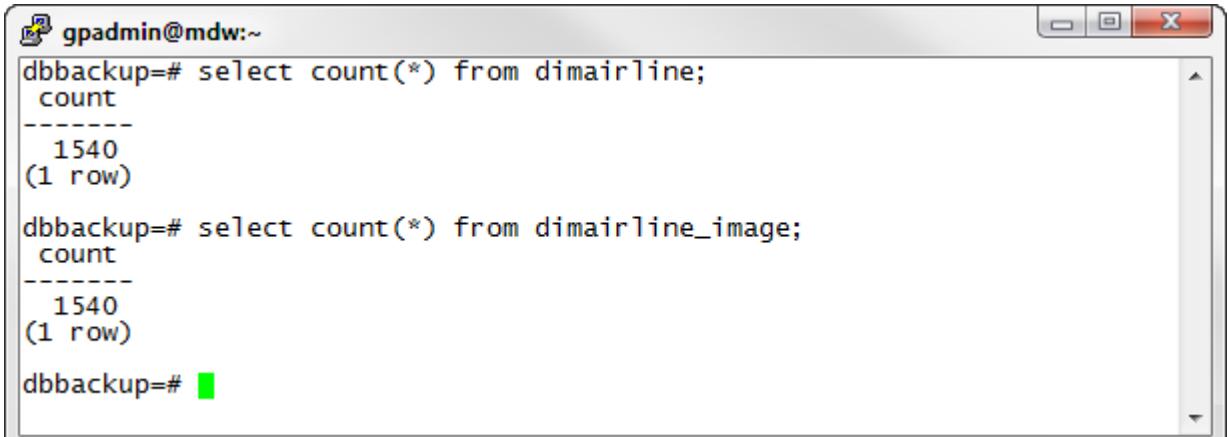
Step	Action
16.	<p>Restore the <code>faa</code> database from the last backup taken. By not including the key identifier for the backupset, <code>gpdbrestore</code> will use the last available backup created to perform the restore operation.</p> <p>Execute the <code>gpdbrestore</code> command to restore the <code>faa</code> database:</p> <pre>[gpadmin@mdw ~]\$ gpdbrestore -s faa</pre>  <p>When prompted, type <code>y</code> and press Enter.</p> <p>The procedure may take a few minutes to complete.</p>

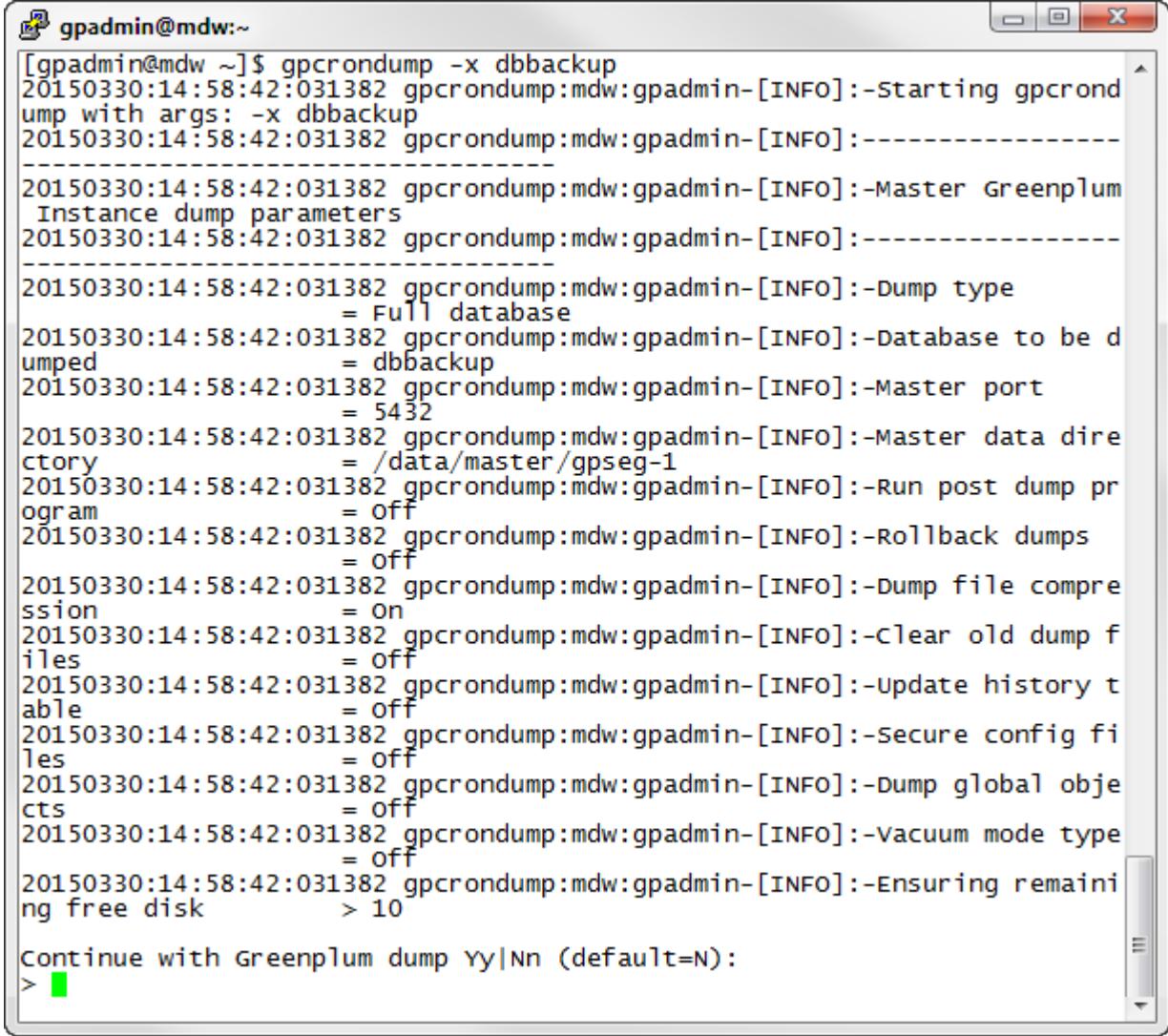
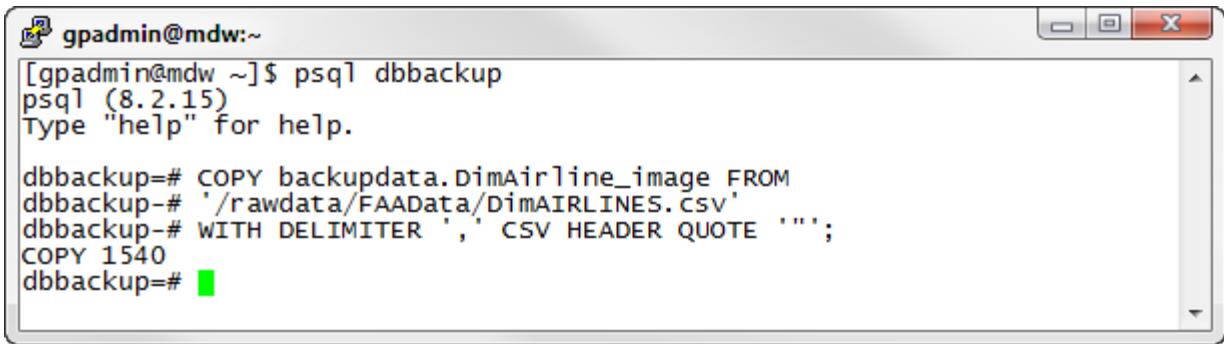
Step	Action																																																																																										
	 <pre>&gt; y 20150330:14:21:02:024705 gpdbrestore:mdw:gpadmin-[INFO]:-gp_restore comm andline: gp_restore -i -h mdw -p 5432 -U gpadmin --gp-i --gp-k=201503301 35701 --gp-l=p --gp-d=db_dumps/20150330 --gp-c -d faa: 20150330:14:26:47:024705 gpdbrestore:mdw:gpadmin-[INFO]:-Commencing anal yze of faa database, please wait 20150330:14:28:13:024705 gpdbrestore:mdw:gpadmin-[INFO]:-Analyze of faa completed without error [gpadmin@mdw ~]\$</pre> <p><b>Note:</b> <code>gpdbrestore -s database_name</code> option looks for the latest set of dump files for the given database name in the segment data directories <code>db_dumps</code> directory on the Greenplum Database array of hosts.</p>																																																																																										
17.	<p>Verify the database has been restored by listing tables from the <code>faa</code> database.</p> <pre>[gpadmin@mdw ~]\$ psql faa faa=# ALTER DATABASE faa SET search_path TO faadata, public, pg_catalog; faa=# \c faa faa=# \dt</pre>  <pre>[gpadmin@mdw ~]\$ psql faa psql (8.2.15) Type "help" for help.  faa=# ALTER DATABASE faa SET search_path TO faadata, public, pg_catalog; ALTER DATABASE faa=# \c faa  You are now connected to database "faa" as user "gpadmin". faa=# \dt</pre>  <table border="1"> <caption>List of relations</caption> <thead> <tr> <th>Schema</th> <th>Name</th> <th>Type</th> <th>Owner</th> <th>Storage</th> </tr> </thead> <tbody> <tr><td>faadata</td><td>dimairline</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>dimairport</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>dimcancellation</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>dimcarrier_history</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>dimday</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>dimdepart_arrive_block</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>dimdistance_group</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>dimdiversions</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>dimmonth</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>dimontime_delay_groups</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>dimquarter</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>dimstate_abr</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>dimstate_fips</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>dimunique_carriers</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>dimworld_area_code</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>dimyesno_resp</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td colspan="5">--More--</td></tr> </tbody> </table>	Schema	Name	Type	Owner	Storage	faadata	dimairline	table	gpadmin	heap	faadata	dimairport	table	gpadmin	heap	faadata	dimcancellation	table	gpadmin	heap	faadata	dimcarrier_history	table	gpadmin	heap	faadata	dimday	table	gpadmin	heap	faadata	dimdepart_arrive_block	table	gpadmin	heap	faadata	dimdistance_group	table	gpadmin	heap	faadata	dimdiversions	table	gpadmin	heap	faadata	dimmonth	table	gpadmin	heap	faadata	dimontime_delay_groups	table	gpadmin	heap	faadata	dimquarter	table	gpadmin	heap	faadata	dimstate_abr	table	gpadmin	heap	faadata	dimstate_fips	table	gpadmin	heap	faadata	dimunique_carriers	table	gpadmin	heap	faadata	dimworld_area_code	table	gpadmin	heap	faadata	dimyesno_resp	table	gpadmin	heap	--More--				
Schema	Name	Type	Owner	Storage																																																																																							
faadata	dimairline	table	gpadmin	heap																																																																																							
faadata	dimairport	table	gpadmin	heap																																																																																							
faadata	dimcancellation	table	gpadmin	heap																																																																																							
faadata	dimcarrier_history	table	gpadmin	heap																																																																																							
faadata	dimday	table	gpadmin	heap																																																																																							
faadata	dimdepart_arrive_block	table	gpadmin	heap																																																																																							
faadata	dimdistance_group	table	gpadmin	heap																																																																																							
faadata	dimdiversions	table	gpadmin	heap																																																																																							
faadata	dimmonth	table	gpadmin	heap																																																																																							
faadata	dimontime_delay_groups	table	gpadmin	heap																																																																																							
faadata	dimquarter	table	gpadmin	heap																																																																																							
faadata	dimstate_abr	table	gpadmin	heap																																																																																							
faadata	dimstate_fips	table	gpadmin	heap																																																																																							
faadata	dimunique_carriers	table	gpadmin	heap																																																																																							
faadata	dimworld_area_code	table	gpadmin	heap																																																																																							
faadata	dimyesno_resp	table	gpadmin	heap																																																																																							
--More--																																																																																											

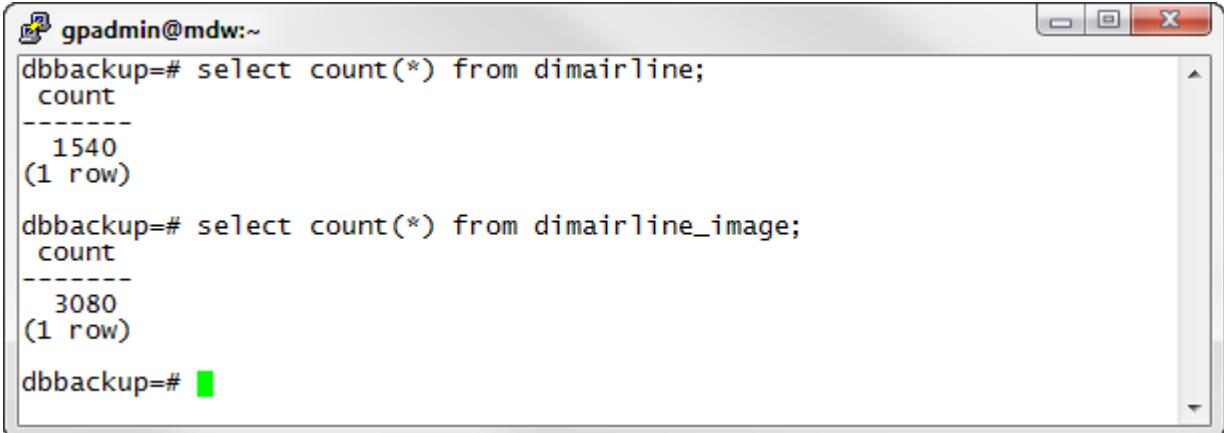
Step	Action
18.	Exit the database before proceeding.

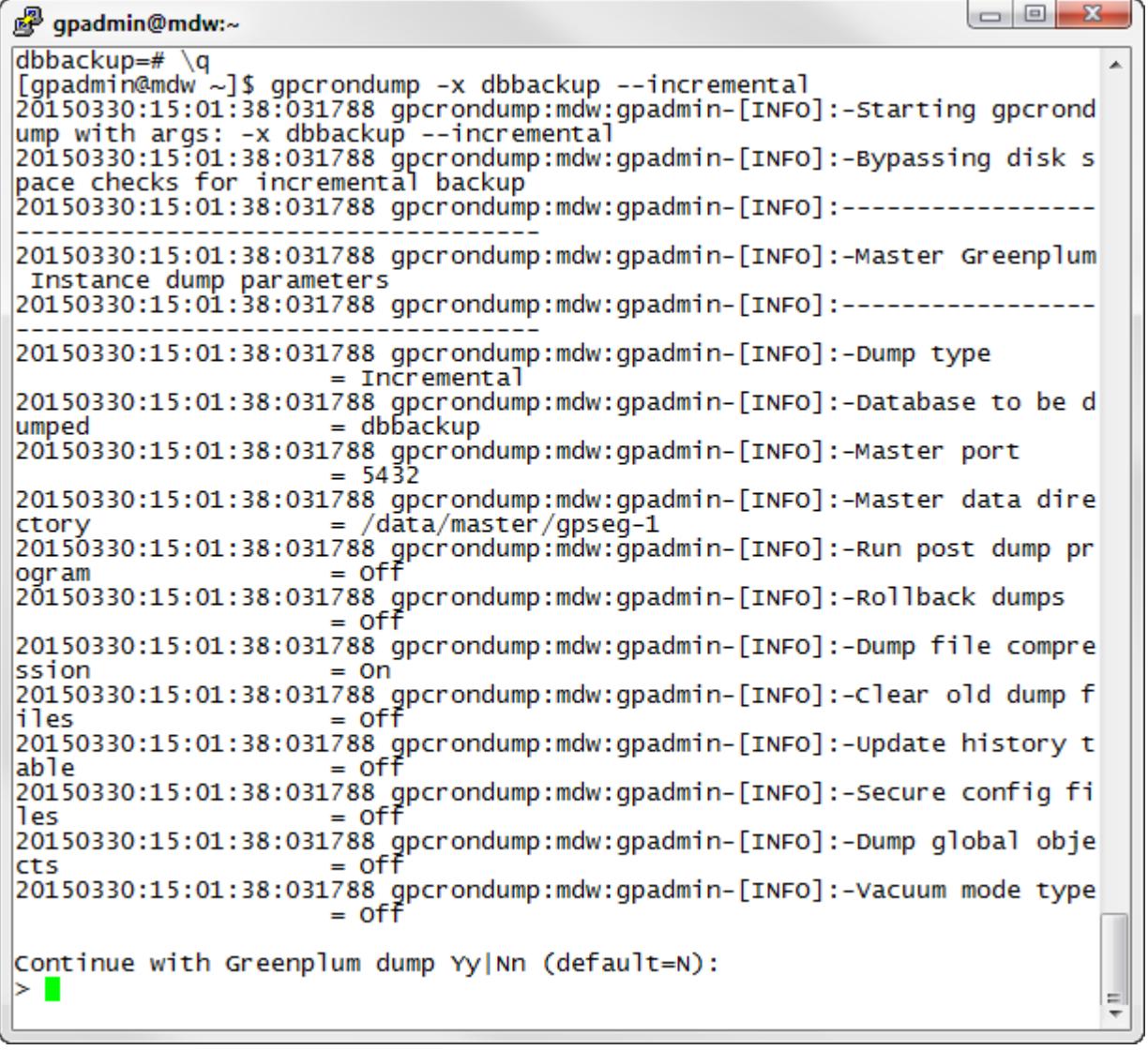
## Lab 15: Task 2 – Incremental Backup and Restore

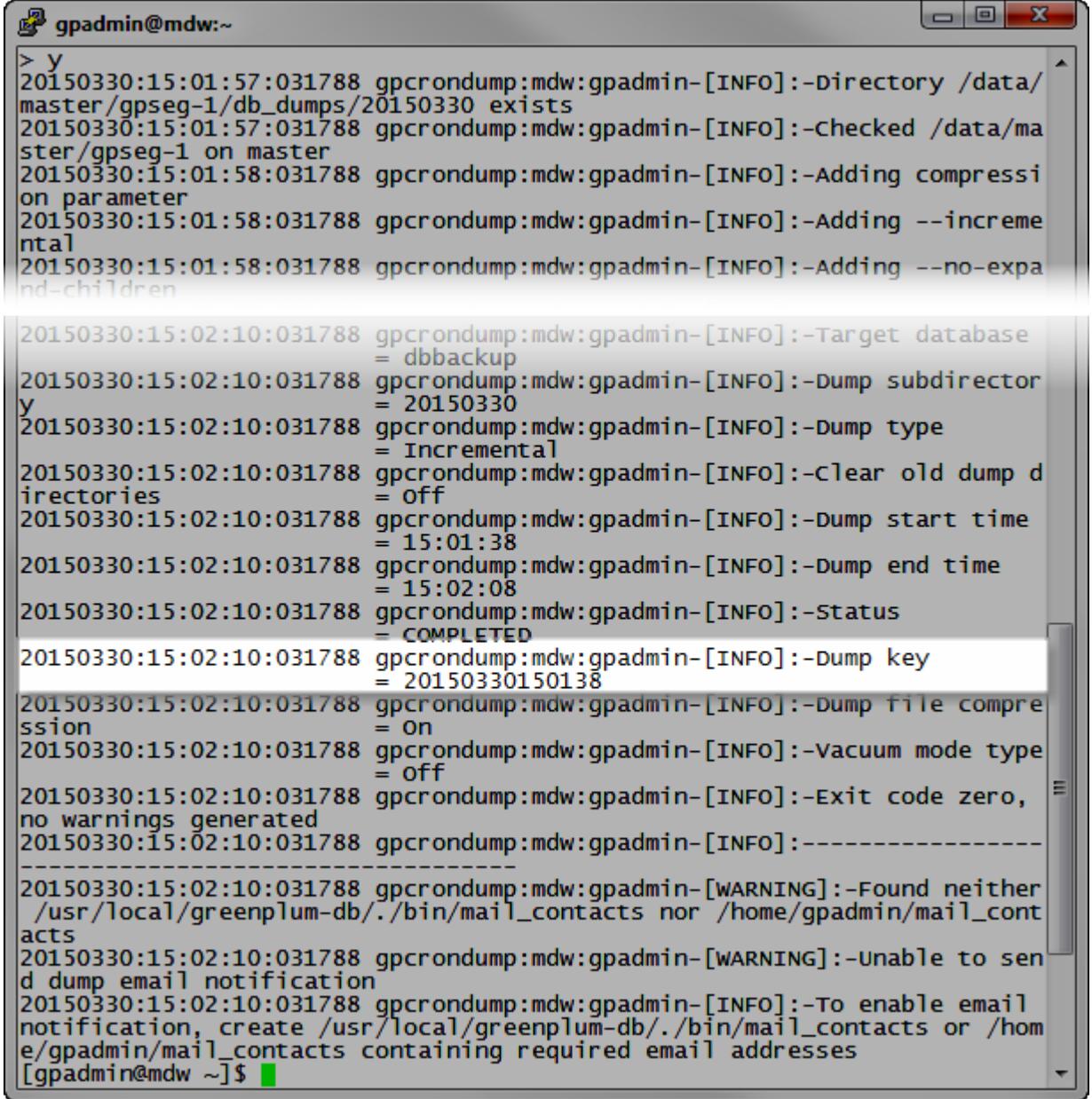
Step	Action
1.	<p>Execute the command below to create two tables in the dbbackup database. The first will be a regular table called dimairline, while the second is an append-only table called dimairline_image.</p> <pre>[gpadmin@mdw ~]\$ psql -f /rawdata/FAAData/CreateDbBackupTables.sql \dbbackup</pre>  <p>The error is displayed only if the table did not previously exist. The table will be created thereafter.</p>
2.	<p>Start a psql session by connecting to the dbbackup database.</p> <pre>[gpadmin@mdw ~]\$ psql dbbackup</pre> 
3.	<p>List the available tables on this database by using the command \dt as shown:</p> <pre>dbbackup=# \dt dimairline*</pre> 

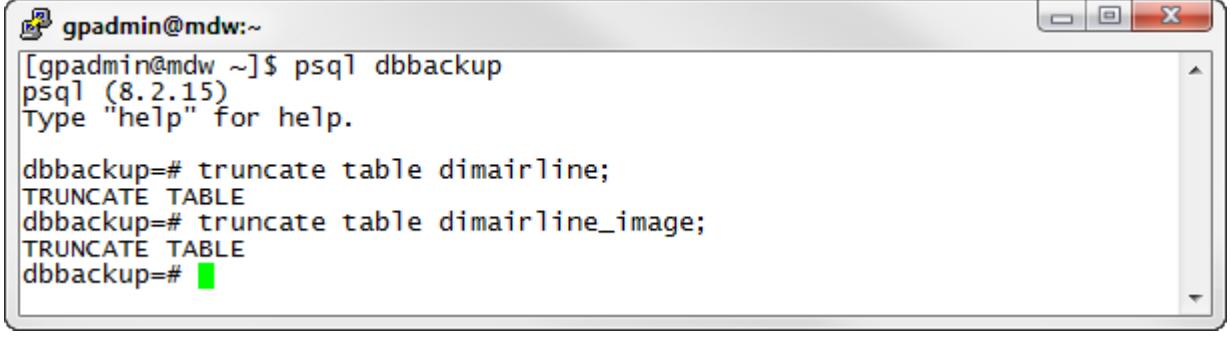
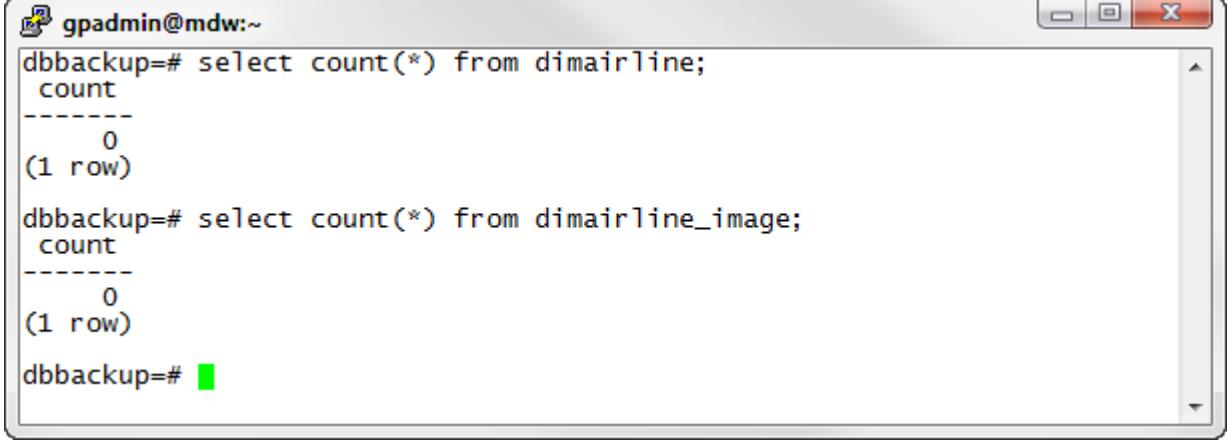
Step	Action
4.	<p>Verify how many records the tables have by running the commands below.</p> <pre>dbbackup=# select count(*) from dimairline; dbbackup=# select count(*) from dimairline_image;</pre>  <p><b>Note:</b> Both tables contain 1,540 records.</p>
5.	Exit the database before proceeding.

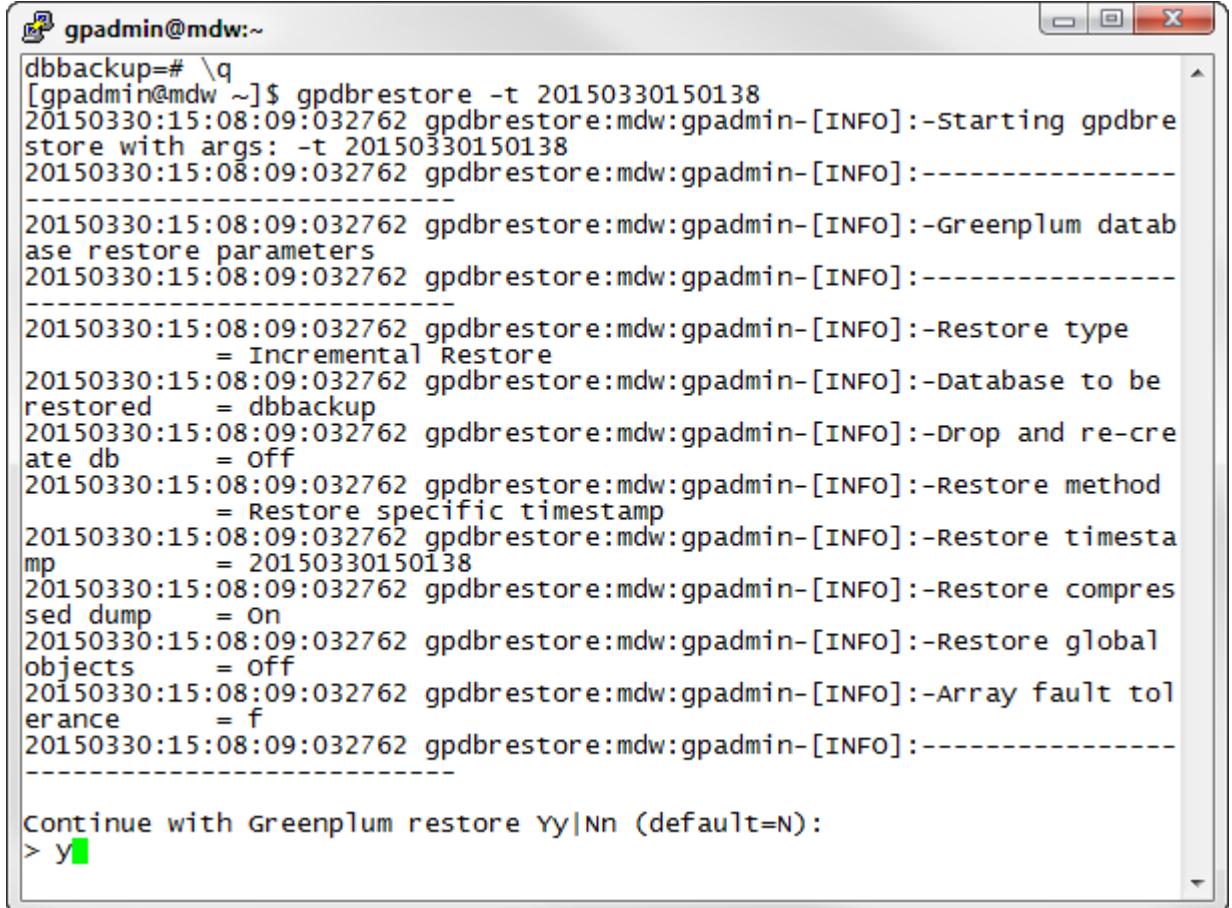
Step	Action
6.	<p>Run a full backup using the following command:</p> <pre>[gpadmin@mdw ~]\$ gpcrondump -x dbbackup</pre>  <p>Respond with <b>y</b> when prompted to continue.</p>
7.	<p>Connect to the dbbackup database as gpadmin and populate the append-only dimairline_image table, as shown:</p> <pre>[gpadmin@mdw ~]\$ psql dbbackup dbbackup=# COPY backupdata.DimAirline_image FROM '/rawdata/FAAData/DimAIRLINES.csv' WITH DELIMITER ',' CSV HEADER QUOTE ''';</pre> 

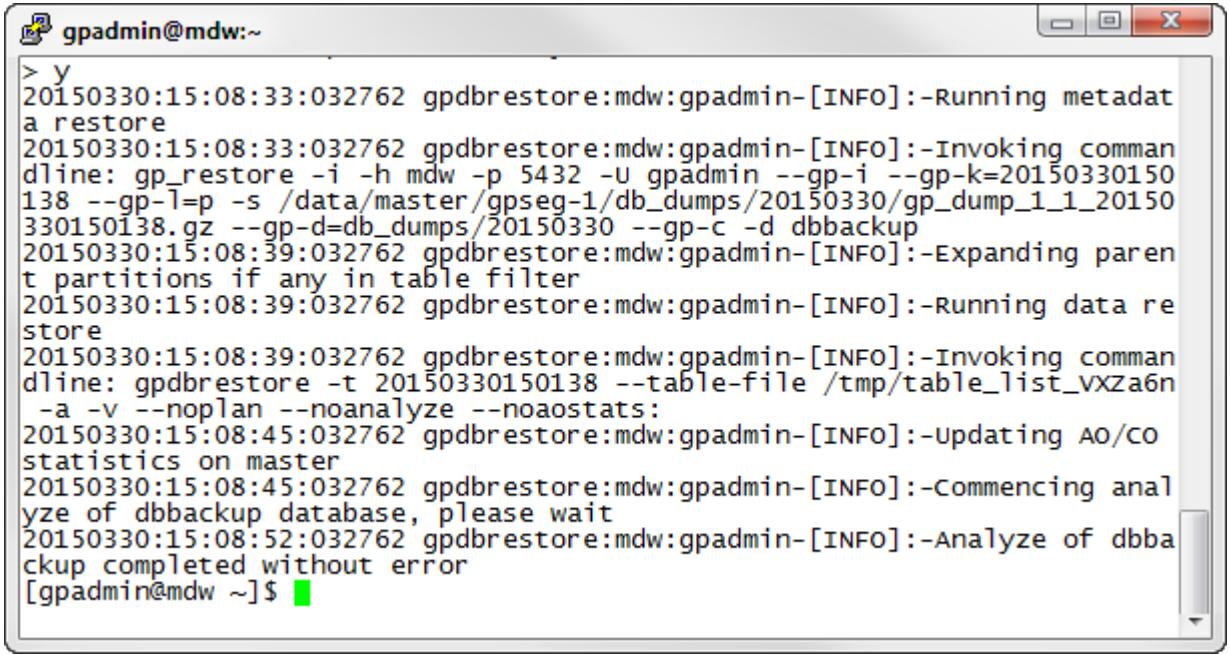
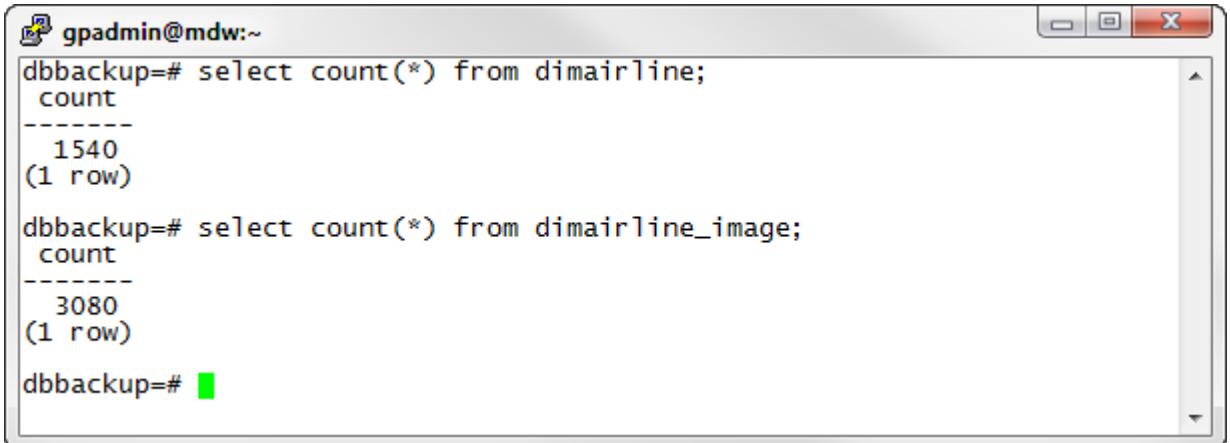
Step	Action
8.	<p>Verify the number of records both the dimairline and dimairline_image tables have by executing the following commands:</p> <pre>dbbackup=# select count(*) from dimairline; dbbackup=# select count(*) from dimairline_image;</pre>  <pre>gpadmin@mdw:~ dbbackup=# select count(*) from dimairline; count -----  1540 (1 row)  dbbackup=# select count(*) from dimairline_image; count -----  3080 (1 row)  dbbackup=# </pre> <p><b>Note:</b> The table dimairline_image now contains 3,080 records.</p>

Step	Action
9.	<p>Exit the database and start an incremental backup for the dbbackup database.</p> <pre>dbbackup=# \q [gpadmin@mdw ~]\$ gpcrondump -x dbbackup --incremental</pre>  <p>Respond with <b>y</b> when prompted to continue.</p>

Step	Action
10.	 <pre> gpadmin@gpadmin@mdw:~\$ &gt; y 20150330:15:01:57:031788 gpcrondump:mdw:gpadmin-[INFO]:-directory /data/master/gpseg-1/db_dumps/20150330 exists 20150330:15:01:57:031788 gpcrondump:mdw:gpadmin-[INFO]:-checked /data/master/gpseg-1 on master 20150330:15:01:58:031788 gpcrondump:mdw:gpadmin-[INFO]:-Adding compression parameter 20150330:15:01:58:031788 gpcrondump:mdw:gpadmin-[INFO]:-Adding --incremental 20150330:15:01:58:031788 gpcrondump:mdw:gpadmin-[INFO]:-Adding --no-expand-children  20150330:15:02:10:031788 gpcrondump:mdw:gpadmin-[INFO]:-Target database = dbbackup 20150330:15:02:10:031788 gpcrondump:mdw:gpadmin-[INFO]:-Dump subdirectory = 20150330 20150330:15:02:10:031788 gpcrondump:mdw:gpadmin-[INFO]:-Dump type = Incremental 20150330:15:02:10:031788 gpcrondump:mdw:gpadmin-[INFO]:-Clear old dump directories = off 20150330:15:02:10:031788 gpcrondump:mdw:gpadmin-[INFO]:-Dump start time = 15:01:38 20150330:15:02:10:031788 gpcrondump:mdw:gpadmin-[INFO]:-Dump end time = 15:02:08 20150330:15:02:10:031788 gpcrondump:mdw:gpadmin-[INFO]:-Status = COMPLETED 20150330:15:02:10:031788 gpcrondump:mdw:gpadmin-[INFO]:-Dump key = 20150330150138 20150330:15:02:10:031788 gpcrondump:mdw:gpadmin-[INFO]:-Dump file compression = On 20150330:15:02:10:031788 gpcrondump:mdw:gpadmin-[INFO]:-vacuum mode type = off 20150330:15:02:10:031788 gpcrondump:mdw:gpadmin-[INFO]:-Exit code zero, no warnings generated 20150330:15:02:10:031788 gpcrondump:mdw:gpadmin-[INFO]:----- -----20150330:15:02:10:031788 gpcrondump:mdw:gpadmin-[WARNING]:-Found neither /usr/local/greenplum-db/.bin/mail_contacts nor /home/gpadmin/mail_contacts 20150330:15:02:10:031788 gpcrondump:mdw:gpadmin-[WARNING]:-unable to send dump email notification 20150330:15:02:10:031788 gpcrondump:mdw:gpadmin-[INFO]:-To enable email notification, create /usr/local/greenplum-db/.bin/mail_contacts or /home/gpadmin/mail_contacts containing required email addresses [gpadmin@mdw ~]\$ </pre> <p><b>Note:</b> The incremental backup provides a Dump key as shown that should be used to restore the incremental backup.</p> <p>Record the dump key here:</p> <hr/>

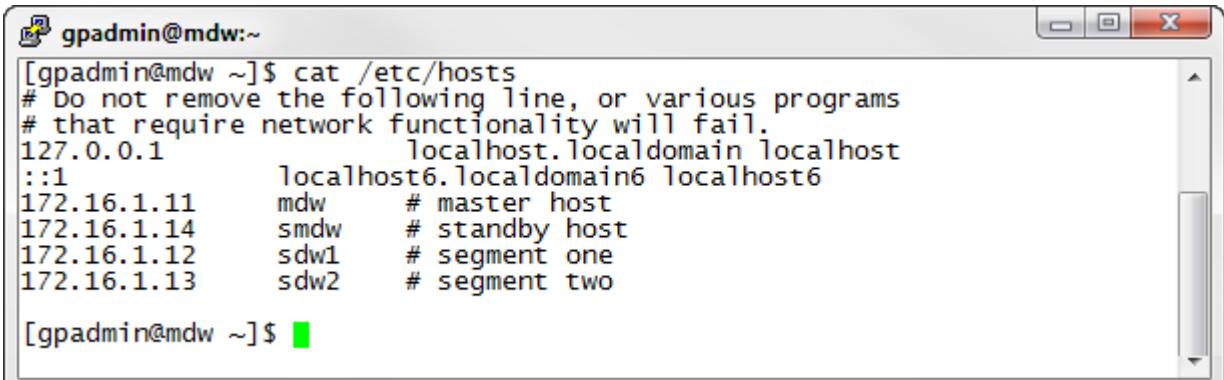
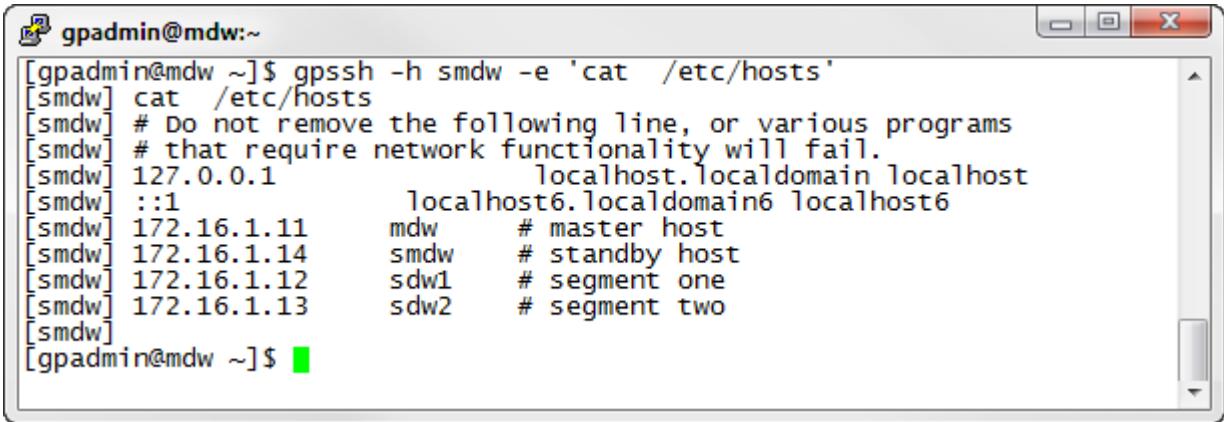
Step	Action
11.	<p>Connect to the dbbackup database and truncate the dimairline and dimairline_image tables.</p> <pre>[gpadmin@mdw ~]\$ psql dbbackup dbbackup=# truncate table dimairline; dbbackup=# truncate table dimairline_image;</pre> 
12.	<p>Verify the number of records in the tables:</p> <pre>dbbackup=# select count(*) from dimairline; dbbackup=# select count(*) from dimairline_image;</pre> 

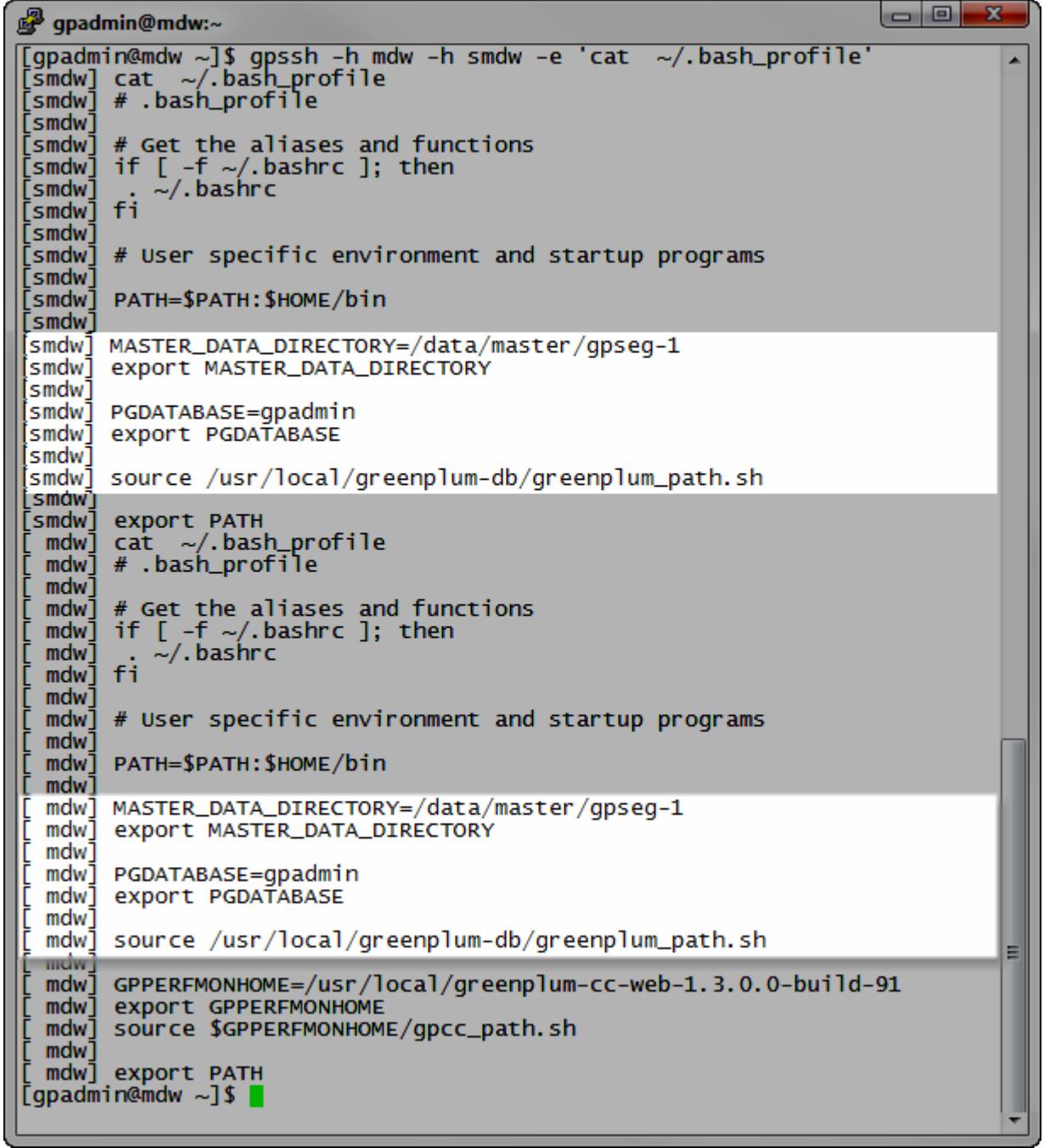
Step	Action
13.	<p>Exit the database and execute the <code>gpdbrestore</code> command to recover the tables.</p> <pre data-bbox="298 228 1122 297">dbbackup=# \q [gpadmin@mdw ~]\$ gpdbrestore -t 20150330150138</pre> <p>Replace the dump key shown here with the dump key you recorded earlier.</p>  <p>Respond with <b>y</b> when prompted to continue.</p>

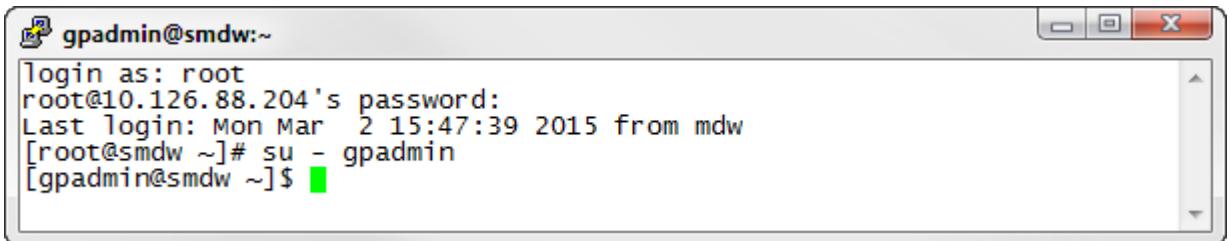
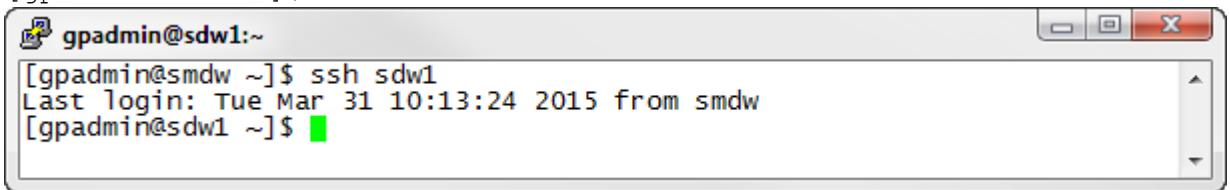
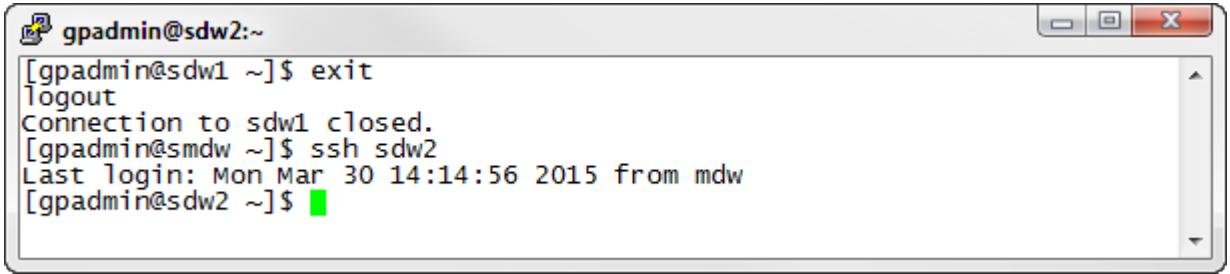
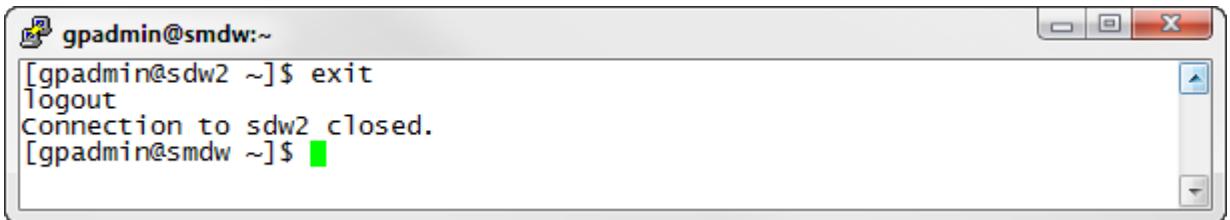
Step	Action
	 <pre> gpadmin@mdw:~  &gt; y 20150330:15:08:33:032762 gpdbrestore:mdw:gpadmin-[INFO]:-Running metadata restore 20150330:15:08:33:032762 gpdbrestore:mdw:gpadmin-[INFO]:-Invoking commandline: gp_restore -i -h mdw -p 5432 -U gpadmin --gp-i --gp-k=20150330150138 --gp-l=p -s /data/master/gpseg-1/db_dumps/20150330/gp_dump_1_1_20150330150138.gz --gp-d=db_dumps/20150330 --gp-c -d dbbackup 20150330:15:08:39:032762 gpdbrestore:mdw:gpadmin-[INFO]:-Expanding parent partitions if any in table filter 20150330:15:08:39:032762 gpdbrestore:mdw:gpadmin-[INFO]:-Running data restore 20150330:15:08:39:032762 gpdbrestore:mdw:gpadmin-[INFO]:-Invoking commandline: gpdbrestore -t 20150330150138 --table-file /tmp/table_list_vXza6n -a -v --noplan --noanalyze --noaoostats: 20150330:15:08:45:032762 gpdbrestore:mdw:gpadmin-[INFO]:-Updating AO/CO statistics on master 20150330:15:08:45:032762 gpdbrestore:mdw:gpadmin-[INFO]:-Commencing analyze of dbbackup database, please wait 20150330:15:08:52:032762 gpdbrestore:mdw:gpadmin-[INFO]:-Analyze of dbbackup completed without error [gpadmin@mdw ~]\$  </pre>
14.	<p>Access the dbbackup database and verify the number of records the tables contain by executing the following commands:</p> <pre> [gpadmin@mdw ~]\$ psql dbbackup dbbackup=# select count(*) from dimairline; dbbackup=# select count(*) from dimairline_image; </pre>  <pre> gpadmin@mdw:~  dbbackup=# select count(*) from dimairline; count -----  1540 (1 row)  dbbackup=# select count(*) from dimairline_image; count -----  3080 (1 row)  dbbackup=# </pre>
15.	<p>Exit the database before proceeding.</p>

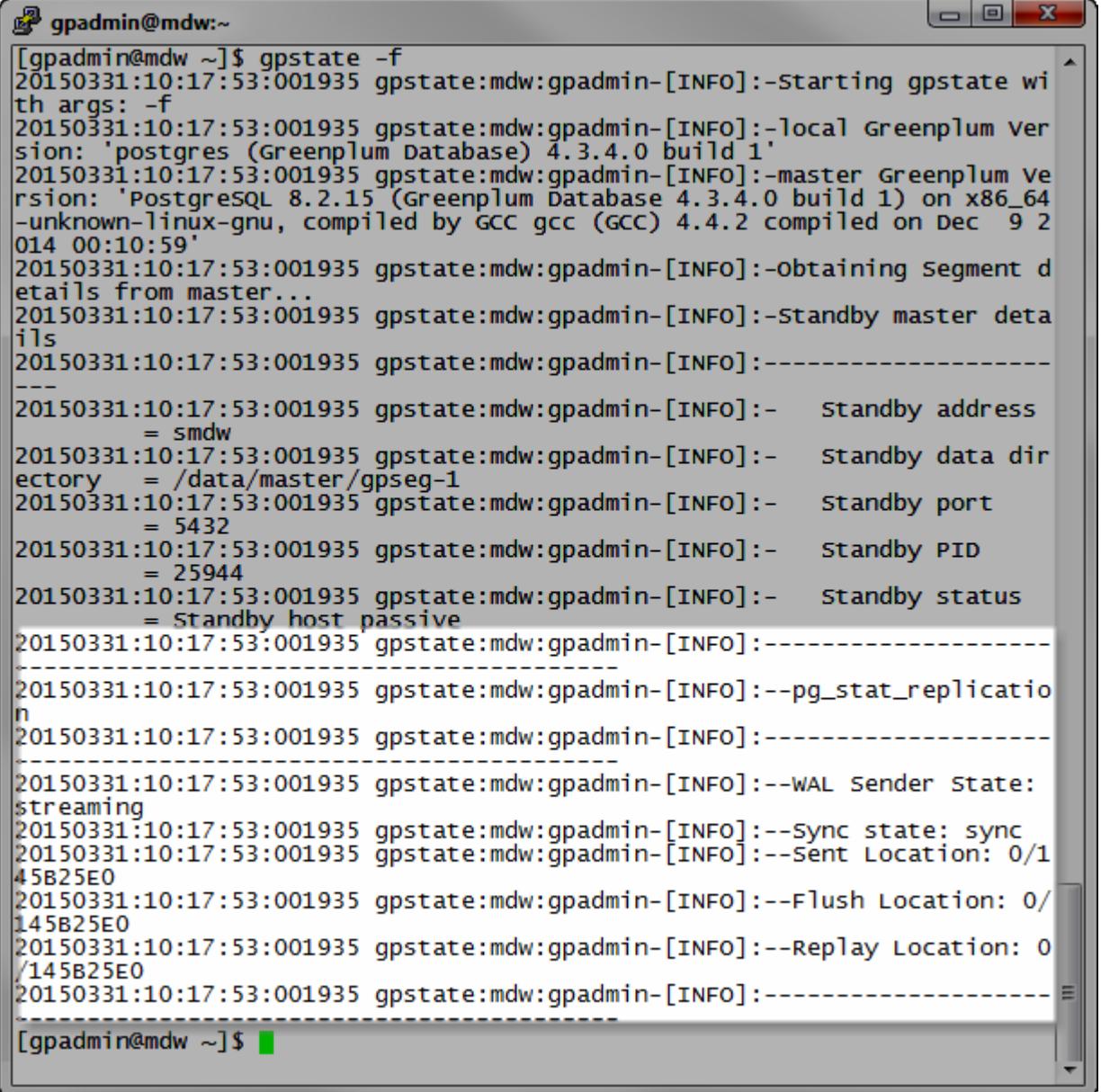
Step	Action
	<p><b>Summary</b></p> <p>Backups are typically automated with <code>gpcrondump</code>, which is a wrapper for <code>gp_dump</code> and <code>pg_dumpall</code>.</p> <p>The <code>gpcrondump</code> utility dumps the contents of a Greenplum Database into SQL utility files, which can then be used to restore the database schema and user data at a later time using <code>gpdbrestore</code>.</p> <p>Keep in mind that a database in the Greenplum Database is actually comprised of several PostgreSQL instances (the master and all active segments), each of which must be dumped individually. The <code>gpcrondump</code> utility takes care of dumping all of the individual instances across the system.</p> <p>Note that the 14 digit timestamp is the number that uniquely identifies the backup job, and is part of the filename for each dump file created by a <code>gp_dump</code> operation. This timestamp must be passed to the <code>gpdbrestore</code> utility when restoring a Greenplum Database.</p> <p>Incremental backups let you backup append-only tables if a change has been made to the table or its contents. The <code>--incremental</code> option to the <code>gpcrondump</code> command lets you take advantage of the space-saving features that come with performing incremental backups on your tables. Restoring from an incremental backup requires that you have all backups from the last full backup.</p>

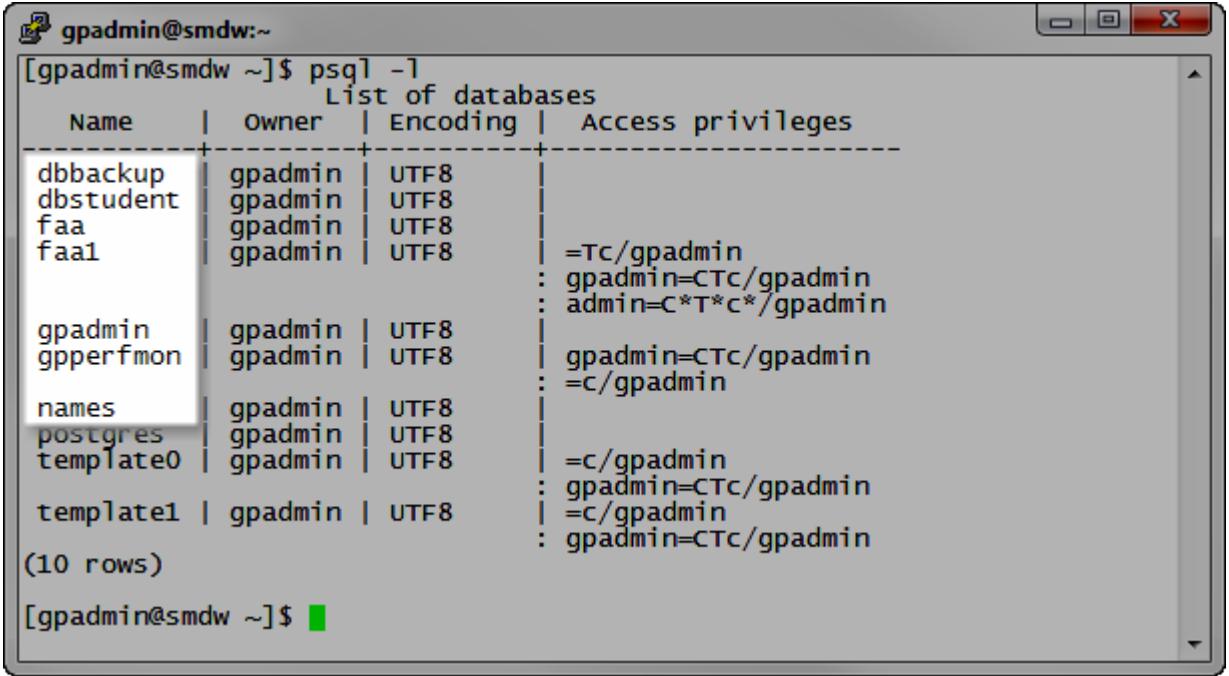
## Lab 15: Task 3 – Recover from a Failed Master (OPTIONAL)

Step	Action
	<p><b>Task Overview</b></p> <p>Your standby server has been installed and configured during the installation of the Greenplum software.</p> <p>You will perform the following steps:</p> <ul style="list-style-type: none"><li>• Failover to the standby server</li><li>• Verify the Greenplum state operating with standby server</li><li>• Fallback to the master server</li></ul>
1.	<p>Before proceeding, verify that the standby server is properly configured.</p> <p>If not already connected, log in as <code>root</code> and switch user to <code>gpadmin</code> on your master server.</p> <p>Verify the file <code>/etc/hosts</code> on the master and standby servers contain the same content.</p> <pre>[gpadmin@mdw ~]\$ cat /etc/hosts</pre>  <pre>gpadmin@mdw:~\$ [gpadmin@mdw ~]\$ cat /etc/hosts # Do not remove the following line, or various programs # that require network functionality will fail. 127.0.0.1      localhost.localdomain localhost ::1            localhost6.localdomain6 localhost6 172.16.1.11    mdw      # master host 172.16.1.14    smdw     # standby host 172.16.1.12    sdw1     # segment one 172.16.1.13    sdw2     # segment two [gpadmin@mdw ~]\$</pre> <p>[gpadmin@mdw ~]\$ <code>gpssh -h smdw -e 'cat /etc/hosts'</code></p>  <pre>gpadmin@mdw:~\$ [gpadmin@mdw ~]\$ gpssh -h smdw -e 'cat /etc/hosts' [smdw] cat /etc/hosts [smdw] # Do not remove the following line, or various programs [smdw] # that require network functionality will fail. [smdw] 127.0.0.1      localhost.localdomain localhost [smdw] ::1            localhost6.localdomain6 localhost6 [smdw] 172.16.1.11    mdw      # master host [smdw] 172.16.1.14    smdw     # standby host [smdw] 172.16.1.12    sdw1     # segment one [smdw] 172.16.1.13    sdw2     # segment two [smdw] [gpadmin@mdw ~]\$</pre>

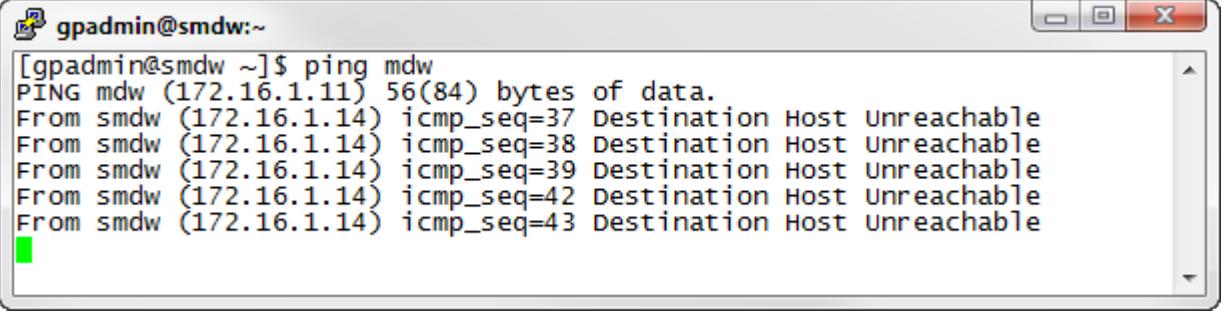
Step	Action
2.	<p>Verify the contents of the <code>.bash_profile</code> file on the standby master server are the same as the <code>.bash_profile</code> file on the master server for the <code>gpadmin</code> user.</p> <pre>[gpadmin@mdw ~]\$ gpssh -h mdw -h smdw -e 'cat ~/.bash_profile'</pre>  <p>As Command Center has not been configured to run on the standby server, you do not have to make changes to the <code>/home/gpadmin/.bash_profile</code> file to include it. All other changes reflecting the Greenplum Database must be the same.</p>

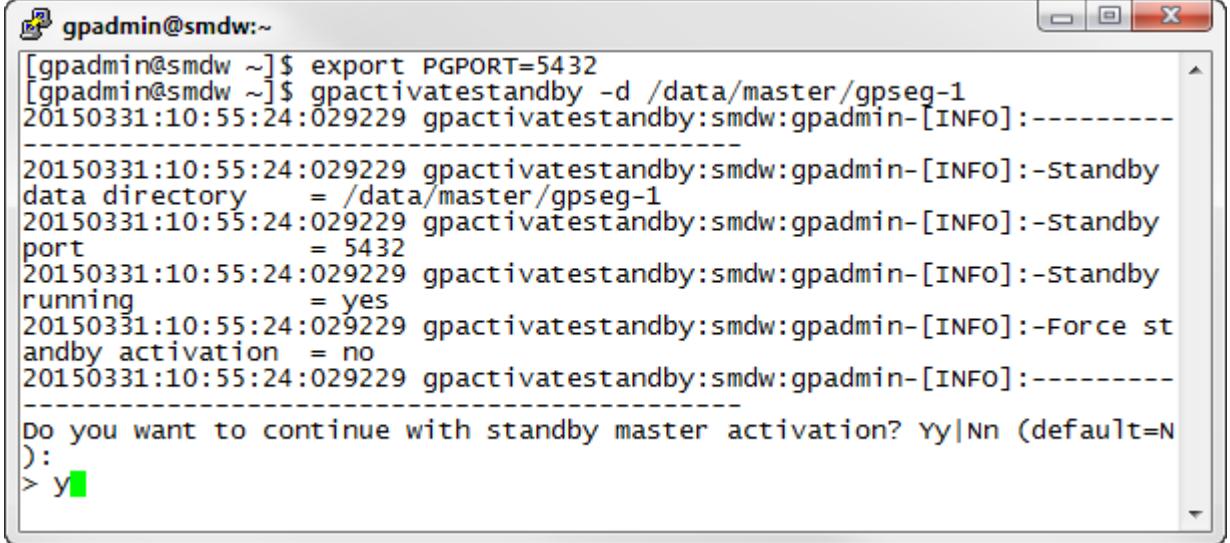
Step	Action
3.	<p>Verify that you can ssh to both segment servers from the standby server.</p> <p>Open a new terminal connection to the standby server, smdw.</p> <p>Connect to the standby server first. Login as root and switch to the gpadmin user account.</p>  <pre>gpadmin@smdw:~\$  Login as: root root@10.126.88.204's password: Last login: Mon Mar  2 15:47:39 2015 from mdw [root@smdw ~]# su - gpadmin [gpadmin@smdw ~]\$ </pre>
4.	<p>From the terminal session where you have connected to the standby server, connect to the first segment server, sdw1.</p> <pre>[gpadmin@smdw ~]\$ ssh sdw1</pre>  <pre>gpadmin@sdw1:~\$  [gpadmin@smdw ~]\$ ssh sdw1 Last login: Tue Mar 31 10:13:24 2015 from smdw [gpadmin@sdw1 ~]\$ </pre> <p>Exit from the first segment server and connect to the second segment server, sdw2.</p> <pre>[gpadmin@sdw1 ~]\$ exit [gpadmin@smdw ~]\$ ssh sdw2</pre>  <pre>gpadmin@sdw2:~\$  [gpadmin@sdw1 ~]\$ exit Logout Connection to sdw1 closed. [gpadmin@smdw ~]\$ ssh sdw2 Last login: Mon Mar 30 14:14:56 2015 from mdw [gpadmin@sdw2 ~]\$ </pre> <p>Exit from the second segment server, sdw2.</p> <pre>[gpadmin@sdw2 ~]\$ exit</pre>  <pre>gpadmin@smdw:~\$  [gpadmin@sdw2 ~]\$ exit Logout Connection to sdw2 closed. [gpadmin@smdw ~]\$ </pre>

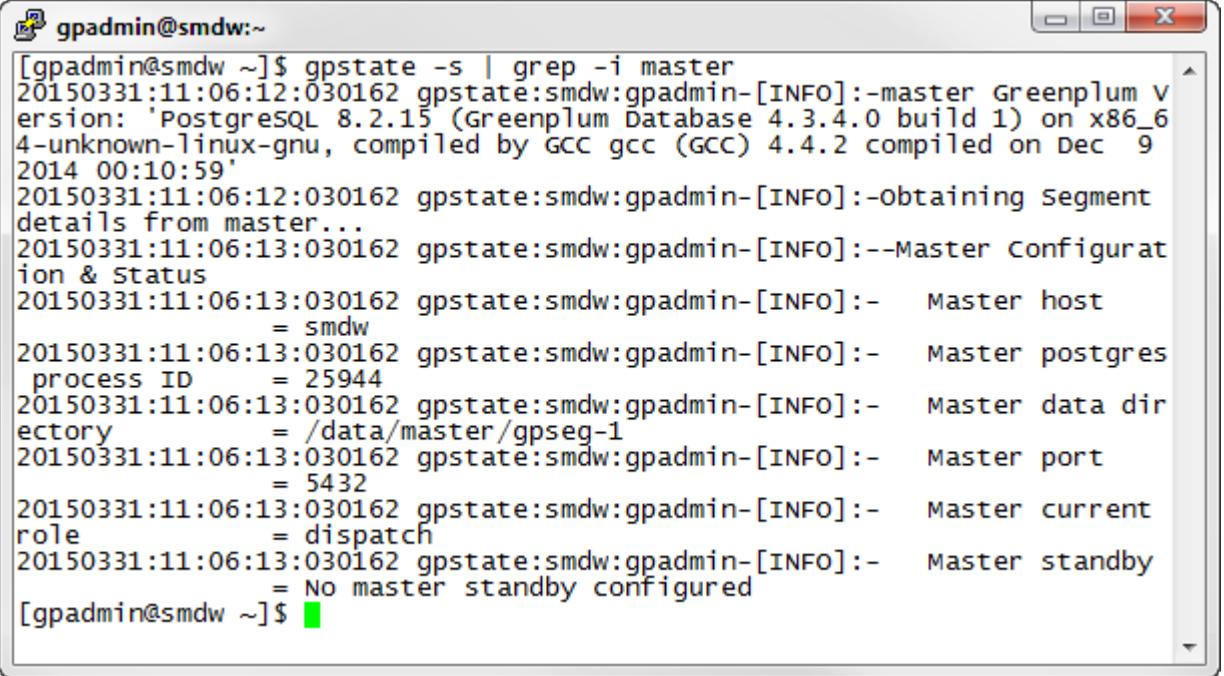
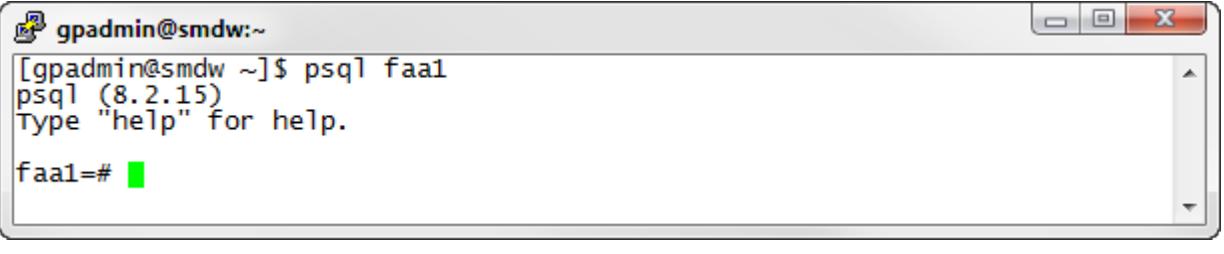
Step	Action
5.	<p>Before initiating a failover, verify the state of the master to standby server to ensure that the database is synchronized. To verify the state, execute the following command on the master server, mdw:</p> <pre data-bbox="306 228 1530 1505">[gpadmin@mdw ~]\$ gpstate -f</pre>  <p>The terminal window displays the following gpstate -f output:</p> <pre>[gpadmin@mdw ~]\$ gpstate -f 20150331:10:17:53:001935 gpstate:mdw:gpadmin-[INFO]:-Starting gpstate with args: -f 20150331:10:17:53:001935 gpstate:mdw:gpadmin-[INFO]:-local Greenplum version: 'postgres (Greenplum Database) 4.3.4.0 build 1' 20150331:10:17:53:001935 gpstate:mdw:gpadmin-[INFO]:-master Greenplum version: 'PostgreSQL 8.2.15 (Greenplum Database 4.3.4.0 build 1) on x86_64-unknown-linux-gnu, compiled by GCC gcc (GCC) 4.4.2 compiled on Dec 9 2014 00:10:59' 20150331:10:17:53:001935 gpstate:mdw:gpadmin-[INFO]:-obtaining segment details from master... 20150331:10:17:53:001935 gpstate:mdw:gpadmin-[INFO]:-Standby master details 20150331:10:17:53:001935 gpstate:mdw:gpadmin-[INFO]:----- ---  20150331:10:17:53:001935 gpstate:mdw:gpadmin-[INFO]:- standby address = smdw 20150331:10:17:53:001935 gpstate:mdw:gpadmin-[INFO]:- standby data directory = /data/master/gpseg-1 20150331:10:17:53:001935 gpstate:mdw:gpadmin-[INFO]:- standby port = 5432 20150331:10:17:53:001935 gpstate:mdw:gpadmin-[INFO]:- standby PID = 25944 20150331:10:17:53:001935 gpstate:mdw:gpadmin-[INFO]:- standby status = Standby host passive 20150331:10:17:53:001935 gpstate:mdw:gpadmin-[INFO]:----- ---  20150331:10:17:53:001935 gpstate:mdw:gpadmin-[INFO]:--pg_stat_replication 20150331:10:17:53:001935 gpstate:mdw:gpadmin-[INFO]:----- ---  20150331:10:17:53:001935 gpstate:mdw:gpadmin-[INFO]:--WAL Sender state: streaming 20150331:10:17:53:001935 gpstate:mdw:gpadmin-[INFO]:--Sync state: sync 20150331:10:17:53:001935 gpstate:mdw:gpadmin-[INFO]:--Sent Location: 0/145B25E0 20150331:10:17:53:001935 gpstate:mdw:gpadmin-[INFO]:--Flush Location: 0/145B25E0 20150331:10:17:53:001935 gpstate:mdw:gpadmin-[INFO]:--Replay Location: 0/145B25E0 20150331:10:17:53:001935 gpstate:mdw:gpadmin-[INFO]:-----</pre> <p>[gpadmin@mdw ~]\$</p>

Step	Action																																																
6.	<p>To safeguard against incidents that may occur in your lab environment, create a backup of all databases in the environment. You will create a backup of all databases except template0, template1, and postgres.</p> <p>First, obtain the list of databases in the environment. The highlighted databases will be backed up.</p> <pre>[gpadmin@smdw ~]\$ psql -l</pre>  <table border="1"> <thead> <tr> <th>Name</th> <th>owner</th> <th>Encoding</th> <th>Access privileges</th> </tr> </thead> <tbody> <tr> <td>dbbackup</td> <td>gpadmin</td> <td>UTF8</td> <td>=Tc/gpadmin</td> </tr> <tr> <td>dbstudent</td> <td>gpadmin</td> <td>UTF8</td> <td>: gpadmin=CTc/gpadmin</td> </tr> <tr> <td>faa</td> <td>gpadmin</td> <td>UTF8</td> <td>: admin=C*T*c*/gpadmin</td> </tr> <tr> <td>faa1</td> <td>gpadmin</td> <td>UTF8</td> <td></td> </tr> <tr> <td>gpadmin</td> <td>gpadmin</td> <td>UTF8</td> <td>gpadmin=CTc/gpadmin</td> </tr> <tr> <td>gpperfmon</td> <td>gpadmin</td> <td>UTF8</td> <td>: =c/gpadmin</td> </tr> <tr> <td>names</td> <td>gpadmin</td> <td>UTF8</td> <td></td> </tr> <tr> <td>postgres</td> <td>gpadmin</td> <td>UTF8</td> <td>=c/gpadmin</td> </tr> <tr> <td>template0</td> <td>gpadmin</td> <td>UTF8</td> <td>: gpadmin=CTc/gpadmin</td> </tr> <tr> <td>template1</td> <td>gpadmin</td> <td>UTF8</td> <td>: =c/gpadmin</td> </tr> <tr> <td>(10 rows)</td> <td></td> <td></td> <td>: gpadmin=CTc/gpadmin</td> </tr> </tbody> </table> <pre>[gpadmin@smdw ~]\$</pre>	Name	owner	Encoding	Access privileges	dbbackup	gpadmin	UTF8	=Tc/gpadmin	dbstudent	gpadmin	UTF8	: gpadmin=CTc/gpadmin	faa	gpadmin	UTF8	: admin=C*T*c*/gpadmin	faa1	gpadmin	UTF8		gpadmin	gpadmin	UTF8	gpadmin=CTc/gpadmin	gpperfmon	gpadmin	UTF8	: =c/gpadmin	names	gpadmin	UTF8		postgres	gpadmin	UTF8	=c/gpadmin	template0	gpadmin	UTF8	: gpadmin=CTc/gpadmin	template1	gpadmin	UTF8	: =c/gpadmin	(10 rows)			: gpadmin=CTc/gpadmin
Name	owner	Encoding	Access privileges																																														
dbbackup	gpadmin	UTF8	=Tc/gpadmin																																														
dbstudent	gpadmin	UTF8	: gpadmin=CTc/gpadmin																																														
faa	gpadmin	UTF8	: admin=C*T*c*/gpadmin																																														
faa1	gpadmin	UTF8																																															
gpadmin	gpadmin	UTF8	gpadmin=CTc/gpadmin																																														
gpperfmon	gpadmin	UTF8	: =c/gpadmin																																														
names	gpadmin	UTF8																																															
postgres	gpadmin	UTF8	=c/gpadmin																																														
template0	gpadmin	UTF8	: gpadmin=CTc/gpadmin																																														
template1	gpadmin	UTF8	: =c/gpadmin																																														
(10 rows)			: gpadmin=CTc/gpadmin																																														

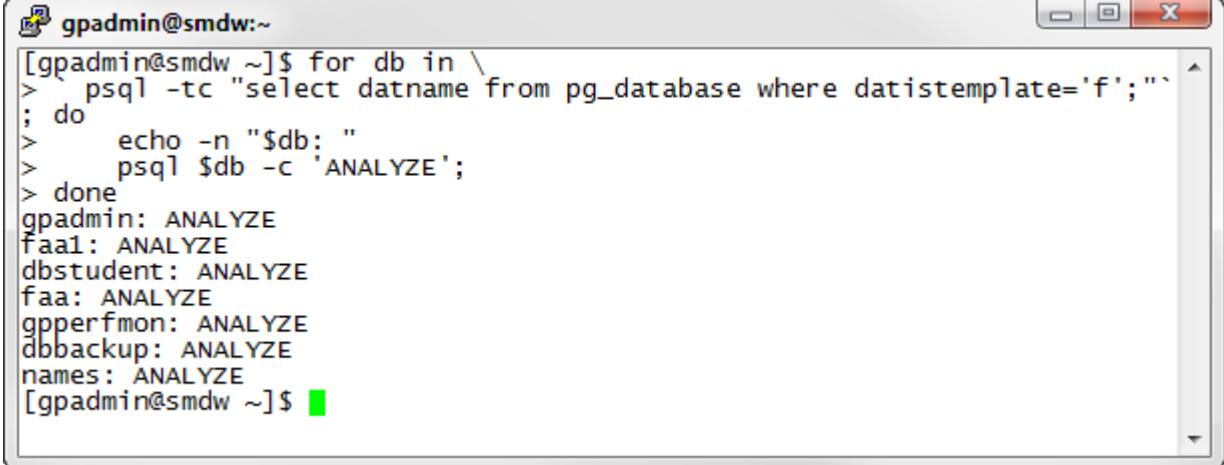
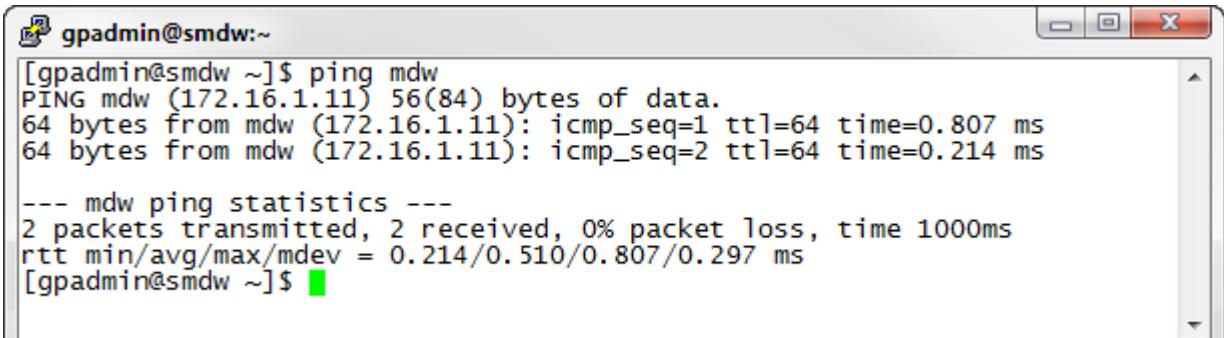
Step	Action
7.	<p>Create a backup of all databases highlighted in the previous step using the <code>gpcrondump</code> command. You can specify multiple databases using the <code>-x</code> option with a comma separated list of databases. You will also copy the <code>pggresql.conf</code> and <code>pg_hba.conf</code> file as part of the backups with the <code>-g</code> option. The <code>-a</code> option will execute the command in non-interactive mode. Backup files will be saved to the <code>/home/gpadmin/db_backup</code> directory on the master and segment servers with the <code>-u</code> option. You will need the configuration files after you complete the failover process.</p> <pre>[gpadmin@mdw ~]\$ gpcrondump -x dbbackup,dbstudent,faa,faa1, \ &gt; gpadmin,gpperfmon,names -u ~/db_backup -g -a</pre>

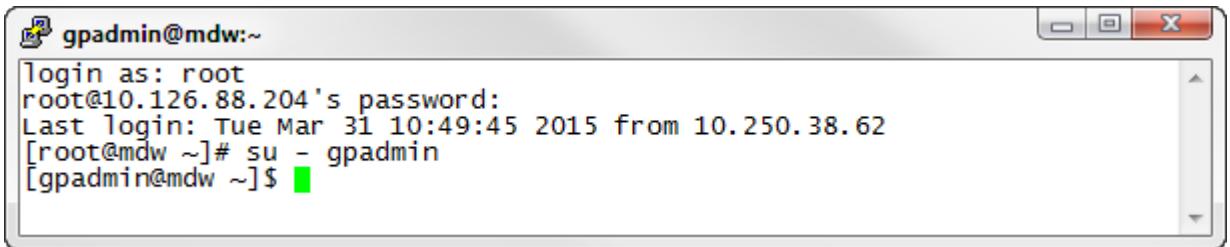
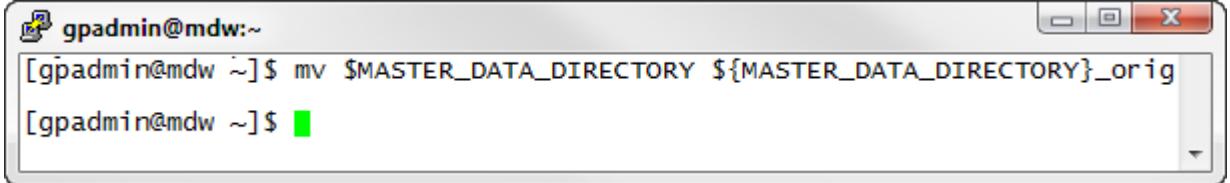
Step	Action
8.	<p>From the master server, switch to the <code>root</code> user and issue the <code>reboot</code> command to reboot the master server. The database will not start automatically as there are no startup scripts in place for the database start up.</p> <pre data-bbox="306 304 675 367">[gpadmin@mdw ~]\$ su - [root@mdw ~]# reboot</pre>  <p>The purpose of this step is to simulate unavailability of the master server. You can then force the standby server to become the new primary master server. Do not proceed until this step has been completed.</p>
9.	<p>Confirm that your master server is down, by pinging it as shown from the standby master server.</p> <pre data-bbox="306 783 768 815">[gpadmin@smdw ~]\$ ping mdw</pre> 

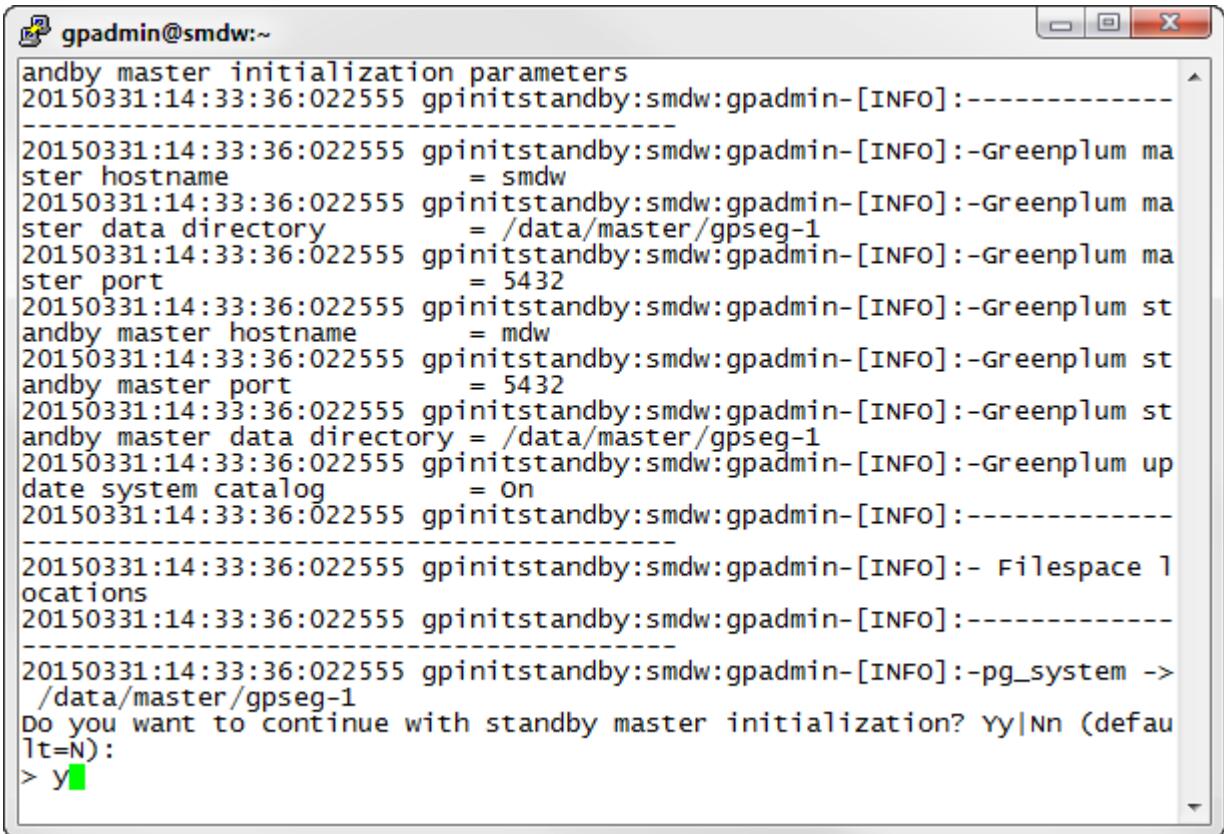
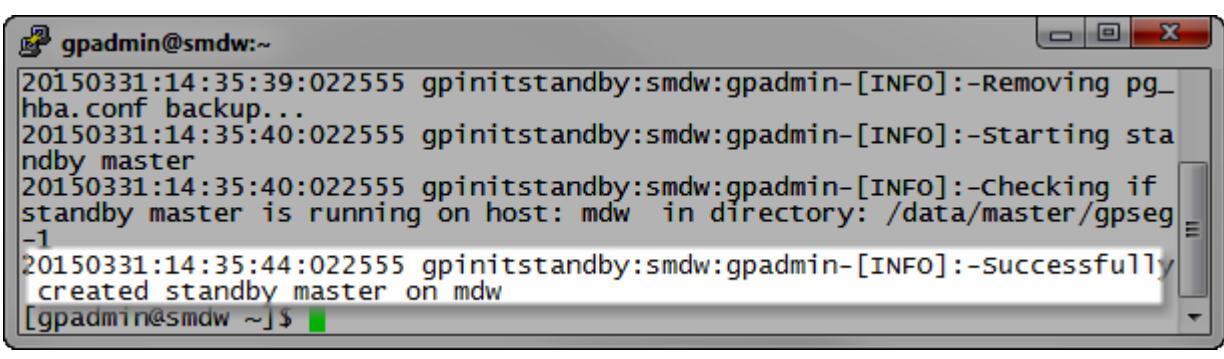
Step	Action
10.	<p>From your standby server, <code>smdw</code>, promote the standby master to be the primary master. You will need to specify the port to use for the database activation. You will continue to use port 5432.</p> <pre>[gpadmin@smdw ~]\$ export PGPORT=5432 [gpadmin@smdw ~]\$ gpactivatestandby -d /data/master/gpseg-1</pre>  <p>Respond with <b>y</b> when asked to continue.</p> <p>It may take a few minutes for the process to complete.</p> <p>Note that the <code>postgresql.conf</code> and <code>pg_hba.conf</code> files are not synchronized as part of the master replication process. Therefore, custom settings preserved on the master are not available here. This therefore required that you set the <code>PGPORT</code> environment variable before promoting the standby server to master.</p>

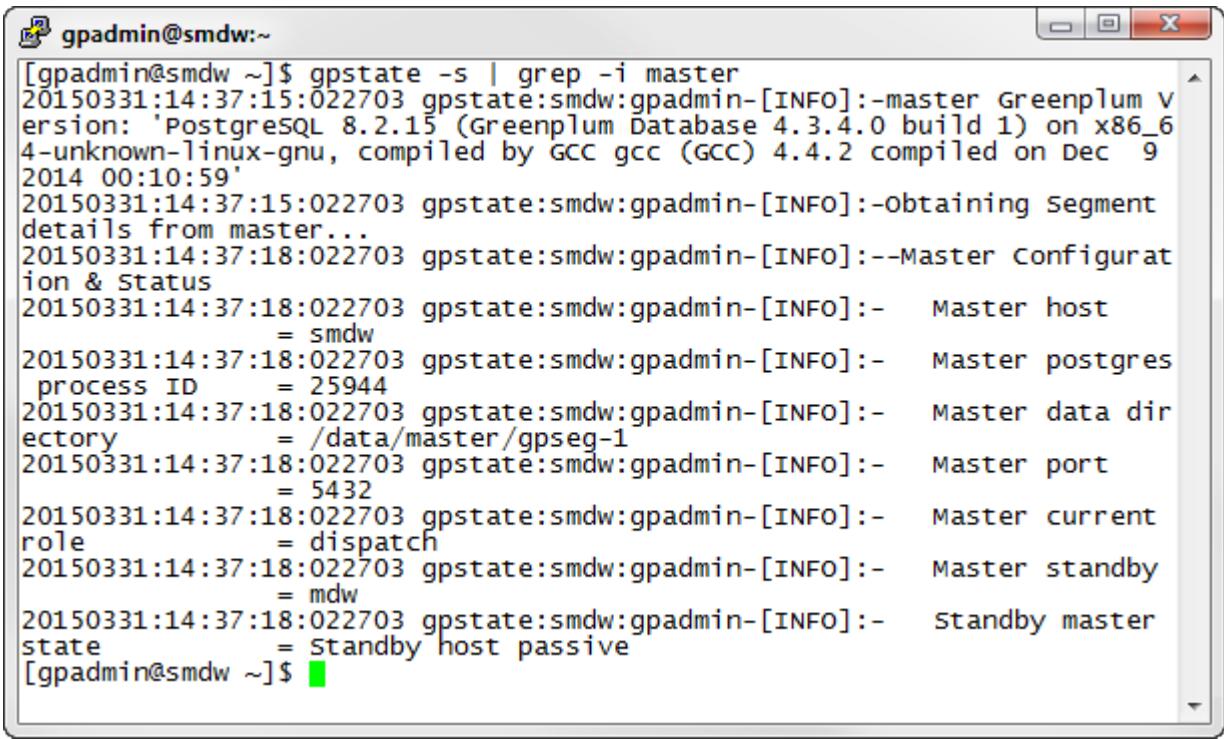
Step	Action
11.	<p>Verify the state of your database by running the gpstate command with the -s option to obtain detailed information. Search the output for strings that contain the word, master.</p> <pre>[gpadmin@smdw ~]\$ gpstate -s   grep -i master</pre>  <p>Note that there is no standby running at this point.</p>
12.	<p>Open a PSQL session to the <b>faa1</b> database to verify the database has been recovered.</p> <pre>[gpadmin@smdw ~]\$ psql faa1</pre> 

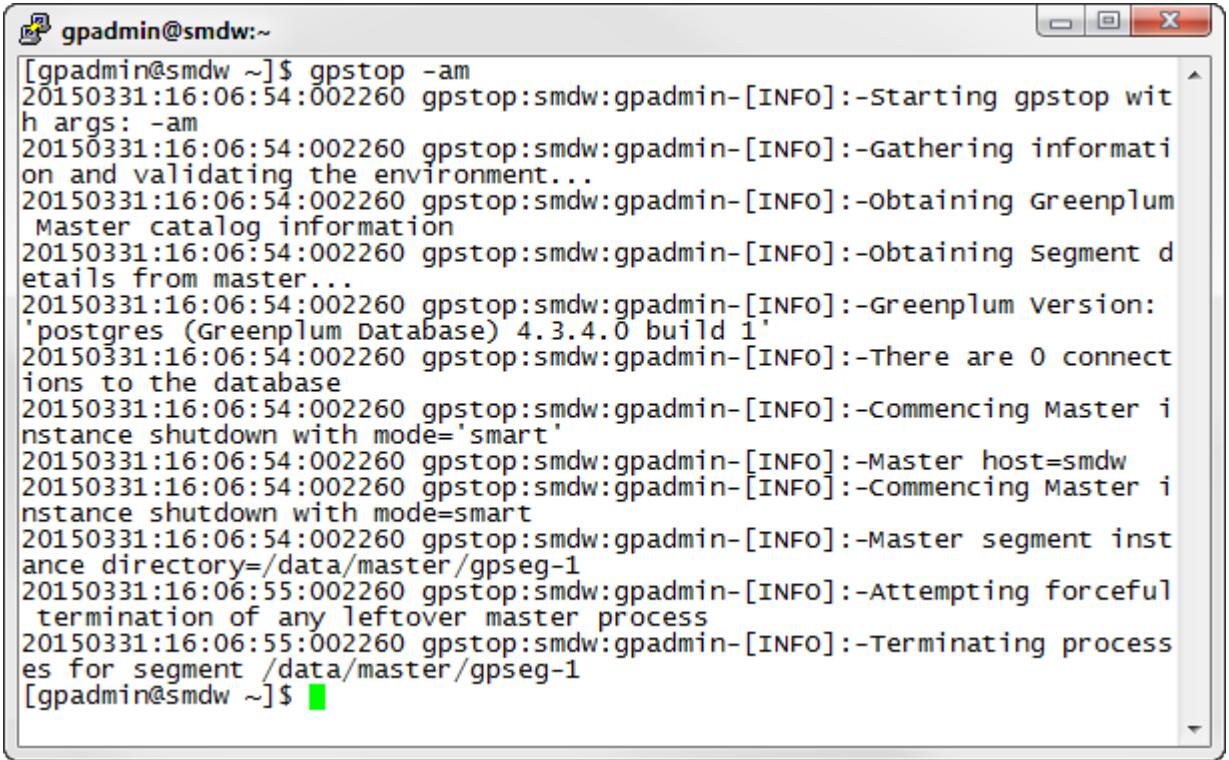
Step	Action																																																																																																																													
13.	<p>Display the user tables for the faa1 database.</p> <pre>faa1=# \dt</pre> <table border="1"> <caption>List of relations</caption> <thead> <tr> <th>Schema</th> <th>Name</th> <th>Type</th> <th>Owner</th> <th>Storage</th> </tr> </thead> <tbody> <tr><td>faadata</td><td>dimairline</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>dimairport</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>dimcancellation</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>dimcarrier_history</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>dimday</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>dimdepart_arrive_block</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>dimdistance_group</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>dimdiversions</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>dimmonth</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>dimontime_delay_groups</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>dimquarter</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>dimstate_abr</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>dimstate_fips</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>dimunique_carriers</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>dimworld_area_code</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>dimyesno_resp</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>fact_temp</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>factortimeperformance</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>factotperf</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>factotperf_1_prt_2</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>factotperf_1_prt_3</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>factotperf_1_prt_y2011</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>orders</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>orders_1_prt_10</td><td>table</td><td>gpadmin</td><td>heap</td></tr> </tbody> </table>	Schema	Name	Type	Owner	Storage	faadata	dimairline	table	gpadmin	heap	faadata	dimairport	table	gpadmin	heap	faadata	dimcancellation	table	gpadmin	heap	faadata	dimcarrier_history	table	gpadmin	heap	faadata	dimday	table	gpadmin	heap	faadata	dimdepart_arrive_block	table	gpadmin	heap	faadata	dimdistance_group	table	gpadmin	heap	faadata	dimdiversions	table	gpadmin	heap	faadata	dimmonth	table	gpadmin	heap	faadata	dimontime_delay_groups	table	gpadmin	heap	faadata	dimquarter	table	gpadmin	heap	faadata	dimstate_abr	table	gpadmin	heap	faadata	dimstate_fips	table	gpadmin	heap	faadata	dimunique_carriers	table	gpadmin	heap	faadata	dimworld_area_code	table	gpadmin	heap	faadata	dimyesno_resp	table	gpadmin	heap	faadata	fact_temp	table	gpadmin	heap	faadata	factortimeperformance	table	gpadmin	heap	faadata	factotperf	table	gpadmin	heap	faadata	factotperf_1_prt_2	table	gpadmin	heap	faadata	factotperf_1_prt_3	table	gpadmin	heap	faadata	factotperf_1_prt_y2011	table	gpadmin	heap	faadata	orders	table	gpadmin	heap	faadata	orders_1_prt_10	table	gpadmin	heap
Schema	Name	Type	Owner	Storage																																																																																																																										
faadata	dimairline	table	gpadmin	heap																																																																																																																										
faadata	dimairport	table	gpadmin	heap																																																																																																																										
faadata	dimcancellation	table	gpadmin	heap																																																																																																																										
faadata	dimcarrier_history	table	gpadmin	heap																																																																																																																										
faadata	dimday	table	gpadmin	heap																																																																																																																										
faadata	dimdepart_arrive_block	table	gpadmin	heap																																																																																																																										
faadata	dimdistance_group	table	gpadmin	heap																																																																																																																										
faadata	dimdiversions	table	gpadmin	heap																																																																																																																										
faadata	dimmonth	table	gpadmin	heap																																																																																																																										
faadata	dimontime_delay_groups	table	gpadmin	heap																																																																																																																										
faadata	dimquarter	table	gpadmin	heap																																																																																																																										
faadata	dimstate_abr	table	gpadmin	heap																																																																																																																										
faadata	dimstate_fips	table	gpadmin	heap																																																																																																																										
faadata	dimunique_carriers	table	gpadmin	heap																																																																																																																										
faadata	dimworld_area_code	table	gpadmin	heap																																																																																																																										
faadata	dimyesno_resp	table	gpadmin	heap																																																																																																																										
faadata	fact_temp	table	gpadmin	heap																																																																																																																										
faadata	factortimeperformance	table	gpadmin	heap																																																																																																																										
faadata	factotperf	table	gpadmin	heap																																																																																																																										
faadata	factotperf_1_prt_2	table	gpadmin	heap																																																																																																																										
faadata	factotperf_1_prt_3	table	gpadmin	heap																																																																																																																										
faadata	factotperf_1_prt_y2011	table	gpadmin	heap																																																																																																																										
faadata	orders	table	gpadmin	heap																																																																																																																										
faadata	orders_1_prt_10	table	gpadmin	heap																																																																																																																										
14.	Exit the database.																																																																																																																													

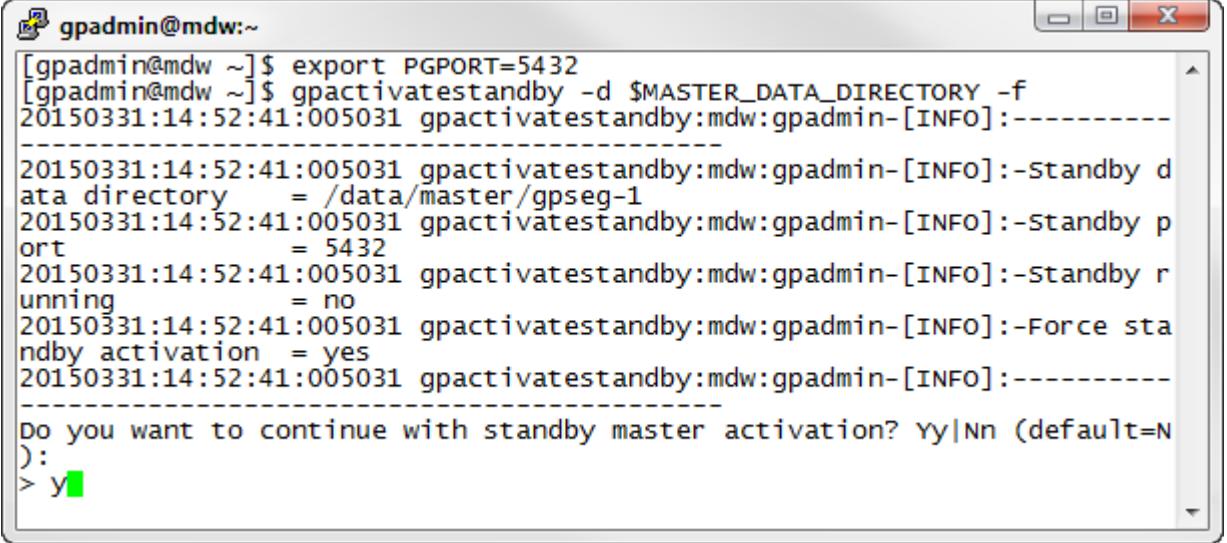
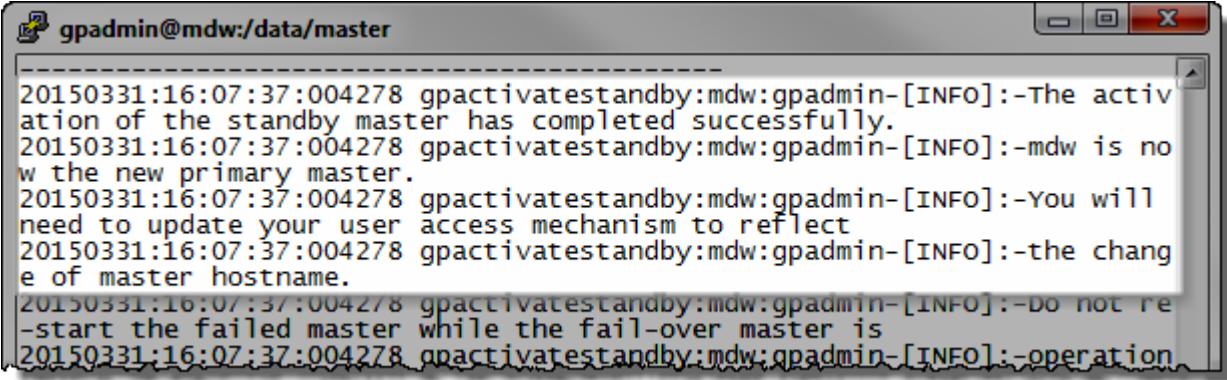
Step	Action
15.	<p>After activating the standby server as the master server, you should update the database query statistics on all databases.</p> <p>For each database, execute the ANALYZE command to update statistics. Use the following script to perform this step.</p> <pre>[gpadmin@smdw ~]\$ for db in \ `psql -tc "select datname from pg_database where datistemplate='f';"`; do   echo -n "\$db: "   psql \$db -c 'ANALYZE'; done</pre> 
16.	<p>After activating a standby master in a recovery scenario and making it your current primary master, you can continue running that instance as your primary master. This assumes that the capabilities and dependability of that host machine are equivalent to the original master host.</p> <p>Before restoring the master and standby instances on original hosts, ensure that the conditions that caused the original failure have been fully fixed.</p> <p>Verify the original master server, mdw, is back online. Ping mdw from smdw.</p> <pre>[gpadmin@smdw ~]\$ ping mdw</pre> 

Step	Action
17.	<p>Reconnect your original terminal session to the original master server, mdw. Login as root and switch to the gpadmin user account.</p>  <pre>gpadmin@mdw:~\$ login as: root root@10.126.88.204's password: Last login: Tue Mar 31 10:49:45 2015 from 10.250.38.62 [root@mdw ~]# su - gpadmin [gpadmin@mdw ~]\$</pre>
18.	<p>On your original master server, mdw, rename the directory /data/master/gpseg-1 to /data/master/gpseg-1_orig. The utility gpinitstandby will recreate the directory and requires that it does not exist.</p> <pre>[gpadmin@mdw ~]\$ mv \$MASTER_DATA_DIRECTORY \ \${MASTER_DATA_DIRECTORY}_orig</pre>  <pre>gpadmin@mdw:~\$ mv \$MASTER_DATA_DIRECTORY \${MASTER_DATA_DIRECTORY}_orig [gpadmin@mdw ~]\$</pre>

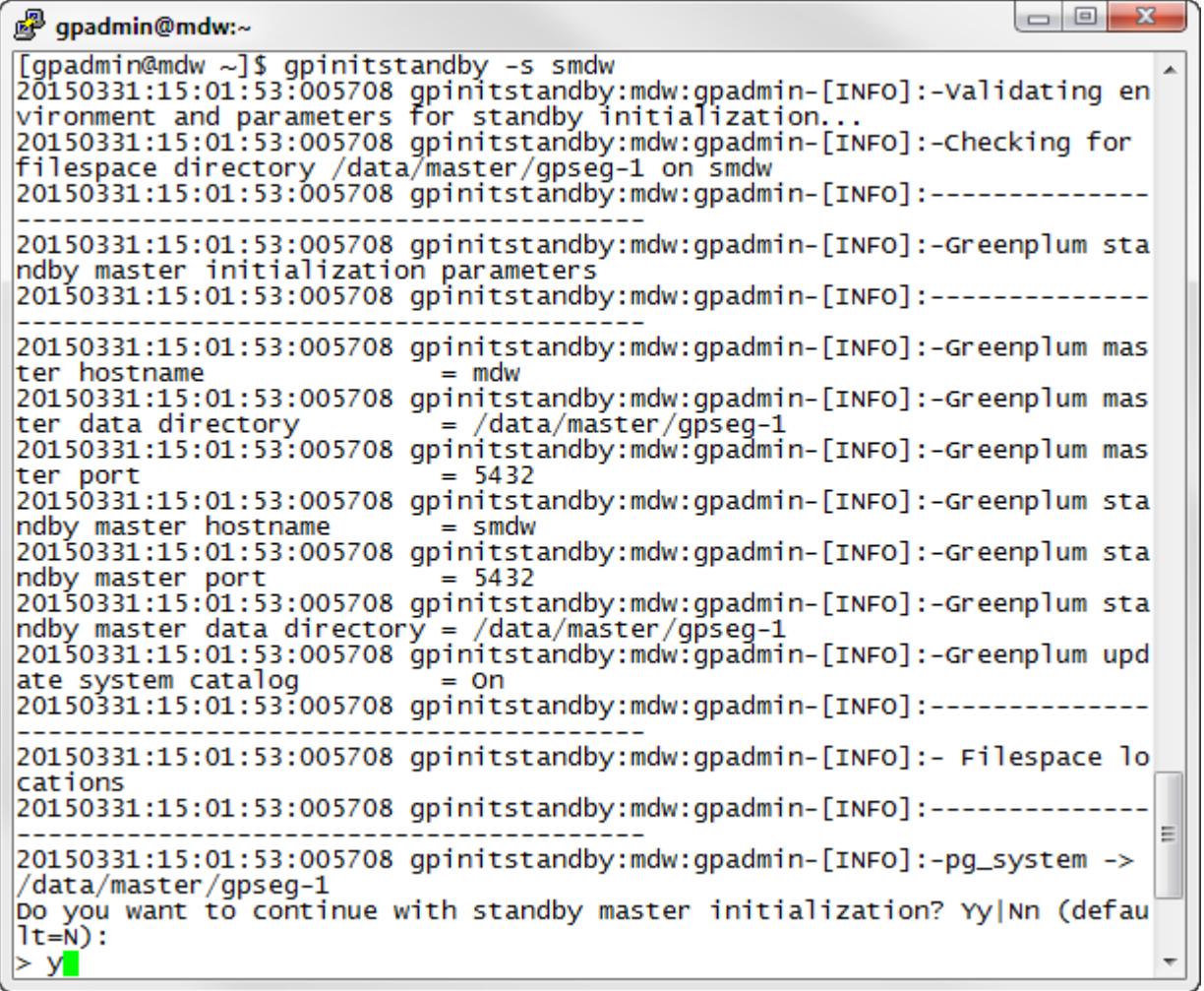
Step	Action
19.	<p>From the standby server, smdw, execute the gpinitstandby command to promote the original master server, mdw, to be the new standby server.</p> <pre>[gpadmin@smdw ~]\$ <b>gpinitstandby -s mdw</b></pre>  <p>You will be prompted to continue. Type <b>y</b> and press Enter to proceed.</p> <p>Once completed, you should see the following output:</p> 

Step	Action
20.	<p>Use the <code>gpstate</code> command to check the status of the standby master. The output of the <code>gpstate</code> command shows that the original standby server, <code>smdw</code>, is now the master server. It also shows that the original master server, <code>mdw</code>, is now the standby server.</p> <pre>[gpadmin@smdw ~]\$ gpstate -s   grep -i master</pre>  <p>You can also obtain the state of the standby and mirrors with the command, <code>gpstate -f</code>.</p>

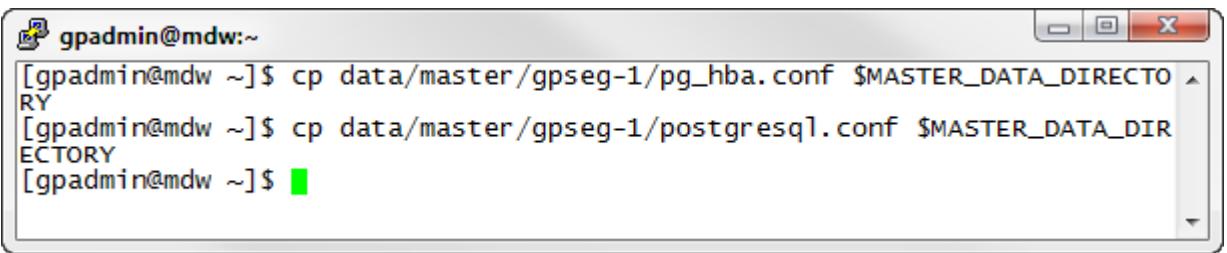
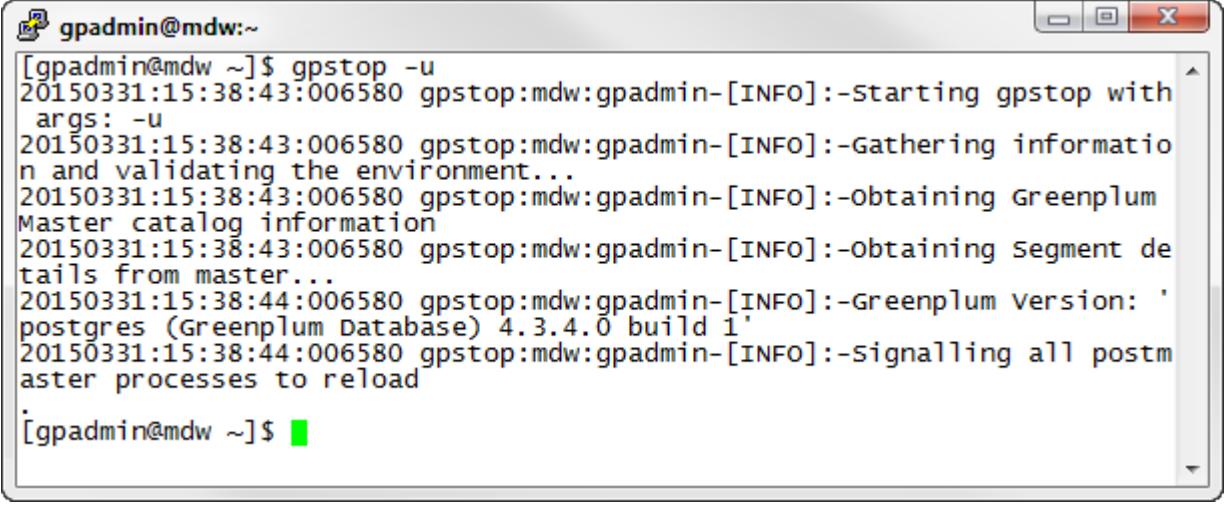
Step	Action
21.	<p>Now that the failover has succeeded, reverse the roles of the master and standby servers so that mdw and smdw are back to their original roles. To perform this task, complete the following steps:</p> <p>On the current master server, smdw, stop the Greenplum database master instance only using the -am option.</p> <pre>[gpadmin@smdw ~]\$ gpstop -am</pre>  <pre>[gpadmin@smdw ~]\$ gpstop -am 20150331:16:06:54:002260 gpstop:smdw:gpadmin-[INFO]:-Starting gpstop with args: -am 20150331:16:06:54:002260 gpstop:smdw:gpadmin-[INFO]:-Gathering information and validating the environment... 20150331:16:06:54:002260 gpstop:smdw:gpadmin-[INFO]:-obtaining Greenplum Master catalog information 20150331:16:06:54:002260 gpstop:smdw:gpadmin-[INFO]:-obtaining Segment details from master... 20150331:16:06:54:002260 gpstop:smdw:gpadmin-[INFO]:-Greenplum version: 'postgres (Greenplum Database) 4.3.4.0 build 1' 20150331:16:06:54:002260 gpstop:smdw:gpadmin-[INFO]:-There are 0 connections to the database 20150331:16:06:54:002260 gpstop:smdw:gpadmin-[INFO]:-Commencing Master instance shutdown with mode='smart' 20150331:16:06:54:002260 gpstop:smdw:gpadmin-[INFO]:-Master host=smdw 20150331:16:06:54:002260 gpstop:smdw:gpadmin-[INFO]:-Commencing Master instance shutdown with mode=smart 20150331:16:06:54:002260 gpstop:smdw:gpadmin-[INFO]:-Master segment instance directory=/data/master/gpseg-1 20150331:16:06:55:002260 gpstop:smdw:gpadmin-[INFO]:-Attempting forceful termination of any leftover master process 20150331:16:06:55:002260 gpstop:smdw:gpadmin-[INFO]:-Terminating processes for segment /data/master/gpseg-1 [gpadmin@smdw ~]\$</pre>

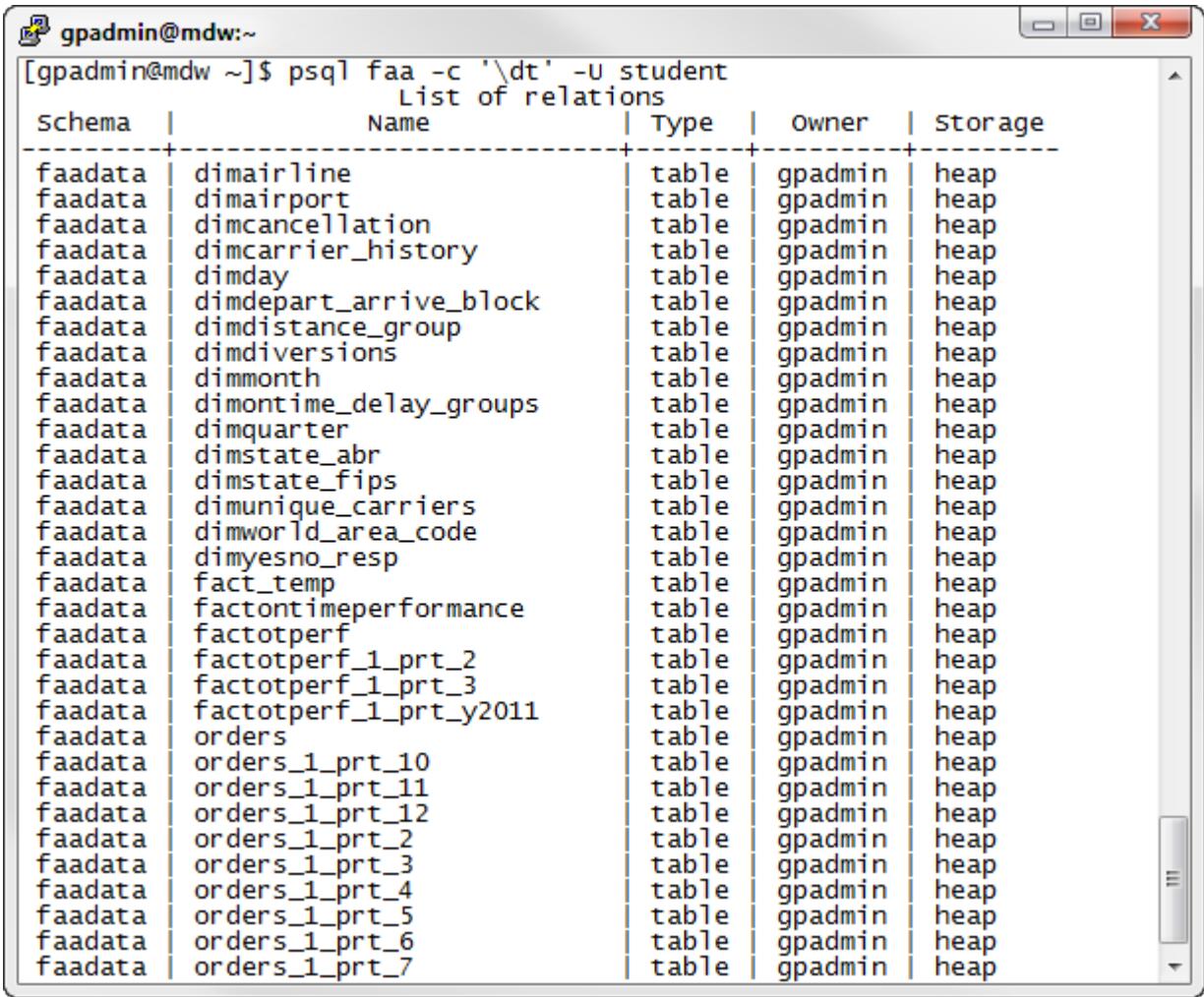
Step	Action
22.	<p>At this point, the database should no longer be running. From the current standby server, mdw, promote mdw to be the active master server. Use the <code>gpactivatestandby</code> utility to perform this task.</p> <pre>[gpadmin@mdw ~]\$ export PGPORT=5432 [gpadmin@mdw ~]\$ gpactivatestandby -d \$MASTER_DATA_DIRECTORY -f</pre>  <p>You will be prompted to continue. Type <b>y</b> and press Enter to proceed.</p> <p>Once completed, you should see the following output:</p> 

Step	Action
23.	<p>On mdw, execute the gpstate utility to determine the state of the active master server. This screen shows that there is no standby server configured and that mdw is back to its original role as the active master server.</p> <pre>[gpadmin@mdw ~]\$ gpstate -s   grep -i master</pre> <p>The terminal window title is "gpadmin@mdw:~". The command run is "[gpadmin@mdw ~]\$ gpstate -s   grep -i master". The output shows Greenplum version information, master host (mdw), master postgreSQL process ID (5280), master data directory (/data/master/gpseg-1), master port (5432), and that no master standby is configured.</p> <pre>[gpadmin@mdw ~]\$ gpstate -s   grep -i master 20150331:14:58:02:005547 gpstate:mdw:gpadmin-[INFO]:-master Greenplum ve rsion: 'PostgreSQL 8.2.15 (Greenplum Database 4.3.4.0 build 1) on x86_64 -unknown-linux-gnu, compiled by GCC gcc (GCC) 4.4.2 compiled on Dec  9 2 014 00:10:59' 20150331:14:58:02:005547 gpstate:mdw:gpadmin-[INFO]:-obtaining segment d etails from master... 20150331:14:58:03:005547 gpstate:mdw:gpadmin-[INFO]:--Master Configurati on &amp; Status 20150331:14:58:03:005547 gpstate:mdw:gpadmin-[INFO]:- Master host = mdw 20150331:14:58:03:005547 gpstate:mdw:gpadmin-[INFO]:- Master postgres process ID = 5280 20150331:14:58:03:005547 gpstate:mdw:gpadmin-[INFO]:- Master data dire ctory = /data/master/gpseg-1 20150331:14:58:03:005547 gpstate:mdw:gpadmin-[INFO]:- Master port = 5432 20150331:14:58:03:005547 gpstate:mdw:gpadmin-[INFO]:- Master current r ole = dispatch 20150331:14:58:03:005547 gpstate:mdw:gpadmin-[INFO]:- Master standby = No master standby configured [gpadmin@mdw ~]\$</pre>
24.	<p>On smdw, start the process for changing this server back to its original role as the standby server. Rename the directory /data/master/gpseg-1 to /data/master/gpseg-1_orig.</p> <pre>[gpadmin@smdw ~]\$ mv \$MASTER_DATA_DIRECTORY \ \${MASTER_DATA_DIRECTORY}_orig</pre> <p>The terminal window title is "gpadmin@smdw:~". The command run is "[gpadmin@smdw ~]\$ mv \$MASTER_DATA_DIRECTORY \ \${MASTER_DATA_DIRECTORY}_orig". The output shows the directory /data/master/gpseg-1 being renamed to /data/master/gpseg-1_orig.</p> <pre>[gpadmin@smdw ~]\$ mv \$MASTER_DATA_DIRECTORY \ &gt; \${MASTER_DATA_DIRECTORY}_orig [gpadmin@smdw ~]\$</pre> <p>This step is necessary as the procedure to initialize a standby server into the Greenplum cluster will create the master data directory.</p>

Step	Action
25.	<p>From mdw, execute the gpinitstandby utility to promote the smdw server to the standby role.</p> <pre data-bbox="306 219 975 255">[gpadmin@mdw ~]\$ gpinitstandby -s smdw</pre>  <p>You will be prompted to continue. Type <b>y</b> and press Enter to proceed.</p>

Step	Action
26.	<p>Run the gpstate utility to re-verify the state of the master and standby servers. The screen below shows that the original roles for mdw (master) and smdw (standby) have been restored.</p> <pre>[gpadmin@mdw ~]\$ gpstate -s   grep -i master</pre> <p>The terminal window shows the command [gpadmin@mdw ~]\$ gpstate -s   grep -i master being run. The output displays various system information and configuration details, including the master host (mdw), master postres process ID (5280), master data directory (/data/master/gpseg-1), master port (5432), master current role (dispatch), master standby (smdw), and standby master state (Standby host passive). The session ends with [gpadmin@mdw ~]\$.</p>
27.	<p>Recover the postgresql.conf and pg_hba.conf file from your backups and push them to the \$MASTER_DATA_DIRECTORY locations. This will overwrite the existing copies but will give you back your customized versions.</p> <p>Search the backup directory you created in this task for files with the name gp_master_config_files_*.tar.</p> <pre>[gpadmin@mdw ~]\$ find ~/db_backup -name 'gp_master_config_files_*.tar'</pre> <p>The terminal window shows the command [gpadmin@mdw ~]\$ find ~/db_backup -name 'gp_master_config_files_*.tar' being run. The output shows a single tar file named /home/gpadmin/db_backup/db_dumps/20150331/gp_master_config_files_201503311103817.tar located in the db_backup directory. The session ends with [gpadmin@mdw ~]\$.</p>

Step	Action
28.	<p>Extract the contents of the tarred file from the previous step.</p> <pre>[gpadmin@mdw ~]\$ tar xvf /home/gpadmin/db_backup/db_dumps/20150331/gp_master_config_files_20150 331103817.tar</pre> 
29.	<p>Copy the pg_hba.conf file and the postgresql.conf file to \$MASTER_DATA_DIRECTORY.</p> <pre>[gpadmin@mdw ~]\$ cp data/master/gpseg-1/pg_hba.conf \$MASTER_DATA_DIRECTORY [gpadmin@mdw ~]\$ cp data/master/gpseg-1/postgresql.conf \$MASTER_DATA_DIRECTORY</pre> 
30.	<p>Re-read the configuration files with the gpstop -u command.</p>  <pre>[gpadmin@mdw ~]\$ gpstop -u 20150331:15:38:43:006580 gpstop:mdw:gpadmin-[INFO]:-starting gpstop with args: -u 20150331:15:38:43:006580 gpstop:mdw:gpadmin-[INFO]:-Gathering informatio n and validating the environment... 20150331:15:38:43:006580 gpstop:mdw:gpadmin-[INFO]:-obtaining Greenplum Master catalog information 20150331:15:38:43:006580 gpstop:mdw:gpadmin-[INFO]:-obtaining Segment de tails from master... 20150331:15:38:44:006580 gpstop:mdw:gpadmin-[INFO]:-Greenplum version: ' postgres (Greenplum Database) 4.3.4.0 build 1' 20150331:15:38:44:006580 gpstop:mdw:gpadmin-[INFO]:-signalling all postm aster processes to reload . [gpadmin@mdw ~]\$</pre>

Step	Action																																																																																																																																																																					
31.	<p>As a test, verify your non-superuser account, in this case student, can list the tables in the faa database.</p> <pre>[gpadmin@mdw ~]\$ psql faa -c '\dt' -U student</pre>  <table border="1"> <thead> <tr> <th>Schema</th> <th>Name</th> <th>Type</th> <th>Owner</th> <th>Storage</th> </tr> </thead> <tbody> <tr><td>faadata</td><td>dimairline</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>dimairport</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>dimcancellation</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>dimcarrier_history</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>dimday</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>dimdepart_arrive_block</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>dimdistance_group</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>dimdiversions</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>dimmonth</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>dimontime_delay_groups</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>dimquarter</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>dimstate_abr</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>dimstate_fips</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>dimunique_carriers</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>dimworld_area_code</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>dimyesno_resp</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>fact_temp</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>factontimeperformance</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>factotperf</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>factotperf_1_prt_2</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>factotperf_1_prt_3</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>factotperf_1_prt_y2011</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>orders</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>orders_1_prt_10</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>orders_1_prt_11</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>orders_1_prt_12</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>orders_1_prt_2</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>orders_1_prt_3</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>orders_1_prt_4</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>orders_1_prt_5</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>orders_1_prt_6</td><td>table</td><td>gpadmin</td><td>heap</td></tr> <tr><td>faadata</td><td>orders_1_prt_7</td><td>table</td><td>gpadmin</td><td>heap</td></tr> </tbody> </table>	Schema	Name	Type	Owner	Storage	faadata	dimairline	table	gpadmin	heap	faadata	dimairport	table	gpadmin	heap	faadata	dimcancellation	table	gpadmin	heap	faadata	dimcarrier_history	table	gpadmin	heap	faadata	dimday	table	gpadmin	heap	faadata	dimdepart_arrive_block	table	gpadmin	heap	faadata	dimdistance_group	table	gpadmin	heap	faadata	dimdiversions	table	gpadmin	heap	faadata	dimmonth	table	gpadmin	heap	faadata	dimontime_delay_groups	table	gpadmin	heap	faadata	dimquarter	table	gpadmin	heap	faadata	dimstate_abr	table	gpadmin	heap	faadata	dimstate_fips	table	gpadmin	heap	faadata	dimunique_carriers	table	gpadmin	heap	faadata	dimworld_area_code	table	gpadmin	heap	faadata	dimyesno_resp	table	gpadmin	heap	faadata	fact_temp	table	gpadmin	heap	faadata	factontimeperformance	table	gpadmin	heap	faadata	factotperf	table	gpadmin	heap	faadata	factotperf_1_prt_2	table	gpadmin	heap	faadata	factotperf_1_prt_3	table	gpadmin	heap	faadata	factotperf_1_prt_y2011	table	gpadmin	heap	faadata	orders	table	gpadmin	heap	faadata	orders_1_prt_10	table	gpadmin	heap	faadata	orders_1_prt_11	table	gpadmin	heap	faadata	orders_1_prt_12	table	gpadmin	heap	faadata	orders_1_prt_2	table	gpadmin	heap	faadata	orders_1_prt_3	table	gpadmin	heap	faadata	orders_1_prt_4	table	gpadmin	heap	faadata	orders_1_prt_5	table	gpadmin	heap	faadata	orders_1_prt_6	table	gpadmin	heap	faadata	orders_1_prt_7	table	gpadmin	heap
Schema	Name	Type	Owner	Storage																																																																																																																																																																		
faadata	dimairline	table	gpadmin	heap																																																																																																																																																																		
faadata	dimairport	table	gpadmin	heap																																																																																																																																																																		
faadata	dimcancellation	table	gpadmin	heap																																																																																																																																																																		
faadata	dimcarrier_history	table	gpadmin	heap																																																																																																																																																																		
faadata	dimday	table	gpadmin	heap																																																																																																																																																																		
faadata	dimdepart_arrive_block	table	gpadmin	heap																																																																																																																																																																		
faadata	dimdistance_group	table	gpadmin	heap																																																																																																																																																																		
faadata	dimdiversions	table	gpadmin	heap																																																																																																																																																																		
faadata	dimmonth	table	gpadmin	heap																																																																																																																																																																		
faadata	dimontime_delay_groups	table	gpadmin	heap																																																																																																																																																																		
faadata	dimquarter	table	gpadmin	heap																																																																																																																																																																		
faadata	dimstate_abr	table	gpadmin	heap																																																																																																																																																																		
faadata	dimstate_fips	table	gpadmin	heap																																																																																																																																																																		
faadata	dimunique_carriers	table	gpadmin	heap																																																																																																																																																																		
faadata	dimworld_area_code	table	gpadmin	heap																																																																																																																																																																		
faadata	dimyesno_resp	table	gpadmin	heap																																																																																																																																																																		
faadata	fact_temp	table	gpadmin	heap																																																																																																																																																																		
faadata	factontimeperformance	table	gpadmin	heap																																																																																																																																																																		
faadata	factotperf	table	gpadmin	heap																																																																																																																																																																		
faadata	factotperf_1_prt_2	table	gpadmin	heap																																																																																																																																																																		
faadata	factotperf_1_prt_3	table	gpadmin	heap																																																																																																																																																																		
faadata	factotperf_1_prt_y2011	table	gpadmin	heap																																																																																																																																																																		
faadata	orders	table	gpadmin	heap																																																																																																																																																																		
faadata	orders_1_prt_10	table	gpadmin	heap																																																																																																																																																																		
faadata	orders_1_prt_11	table	gpadmin	heap																																																																																																																																																																		
faadata	orders_1_prt_12	table	gpadmin	heap																																																																																																																																																																		
faadata	orders_1_prt_2	table	gpadmin	heap																																																																																																																																																																		
faadata	orders_1_prt_3	table	gpadmin	heap																																																																																																																																																																		
faadata	orders_1_prt_4	table	gpadmin	heap																																																																																																																																																																		
faadata	orders_1_prt_5	table	gpadmin	heap																																																																																																																																																																		
faadata	orders_1_prt_6	table	gpadmin	heap																																																																																																																																																																		
faadata	orders_1_prt_7	table	gpadmin	heap																																																																																																																																																																		

Step	Action
	<p><b>Summary</b></p> <p>If the master server fails, the standby server can be used to bring the database back online and accessible to users. If a virtual IP address has been defined for the master and standby server, the virtual IP address can be used by the standby server so that users do not have to use a different IP address or hostname to access the database.</p> <p>Greenplum Database uses log replication to synchronize data between the master server and the backup server. Committed transactions are synchronized from the master server to the standby server. Should the master become unavailable, the standby can be promoted to act as the master until the master becomes available again.</p> <p>The replication process is maintained by a WAL process running on the master server and the standby server. You can verify the state of synchronization by using the <code>gpstate -f</code> command or selecting against the <code>pg_stat_replication</code> view. This view contains the process id (<code>procpid</code> field), the state (<code>state</code> field), and the synchronization state (<code>sync_state</code> field) along with other information on the WAL process.</p> <p>Note that while transactions are synchronized, the <code>postgresql.conf</code> and <code>pg_hba.conf</code> files are not. Maintain a backup copy of these files and be prepared to incorporate the changes to those files on the standby server should you need to perform a failover.</p> <p>Additionally, when promoting the original master server to its original role, retrieve the backup copies of those files and push them to the recovered data directory.</p>

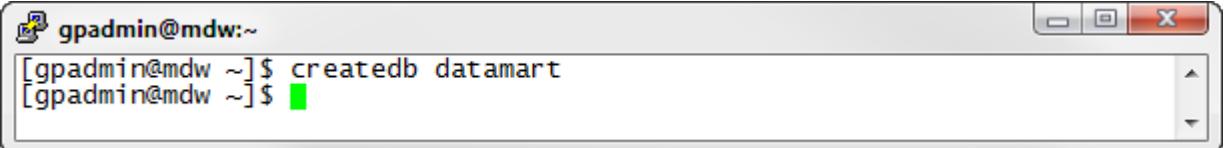
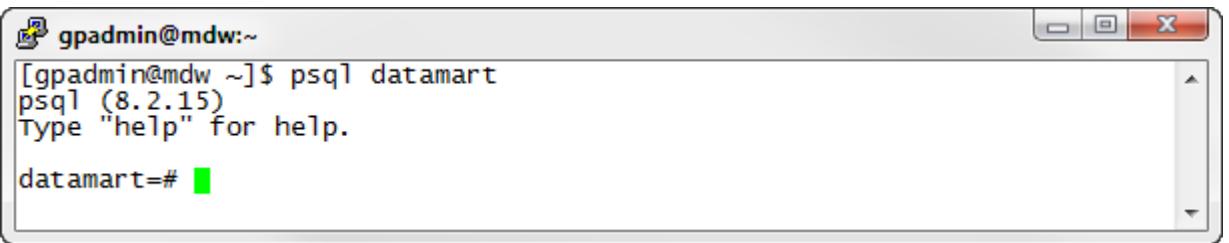
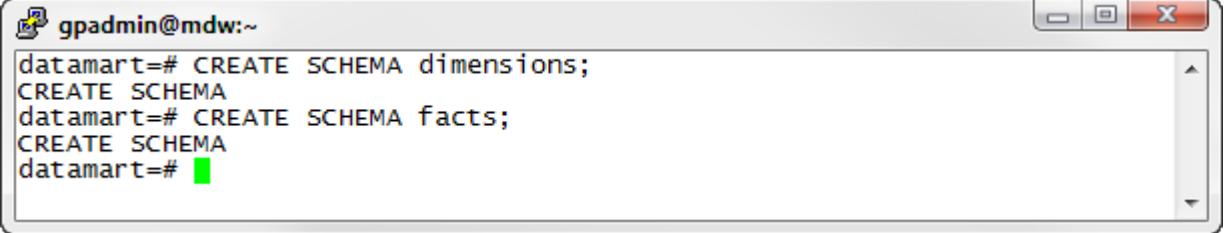
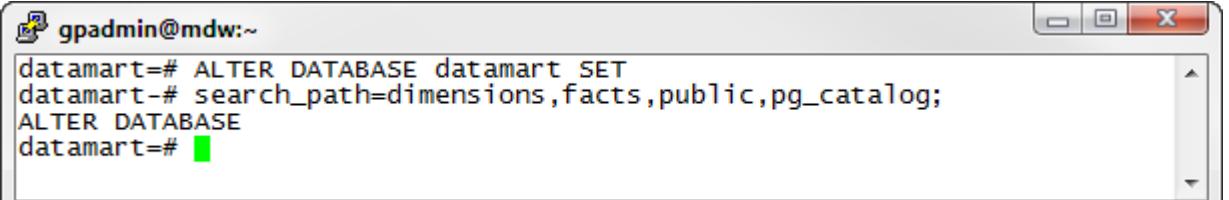
*End of Lab Exercise*

This page intentionally left blank

## Lab 16. Data Modeling

<b>Purpose:</b>	In this lab, you create the database objects that will be used for follow-on labs for demonstrating data modeling and design decisions.
<b>Tasks:</b>	Students perform the following task: Create the <code>datamart</code> database and database objects.
<b>References:</b>	Module 7: Data Modeling and Design <ul style="list-style-type: none"><li>• Lesson: Data Modeling</li></ul>

## Lab 16: Task – Create the datamart Database and Objects

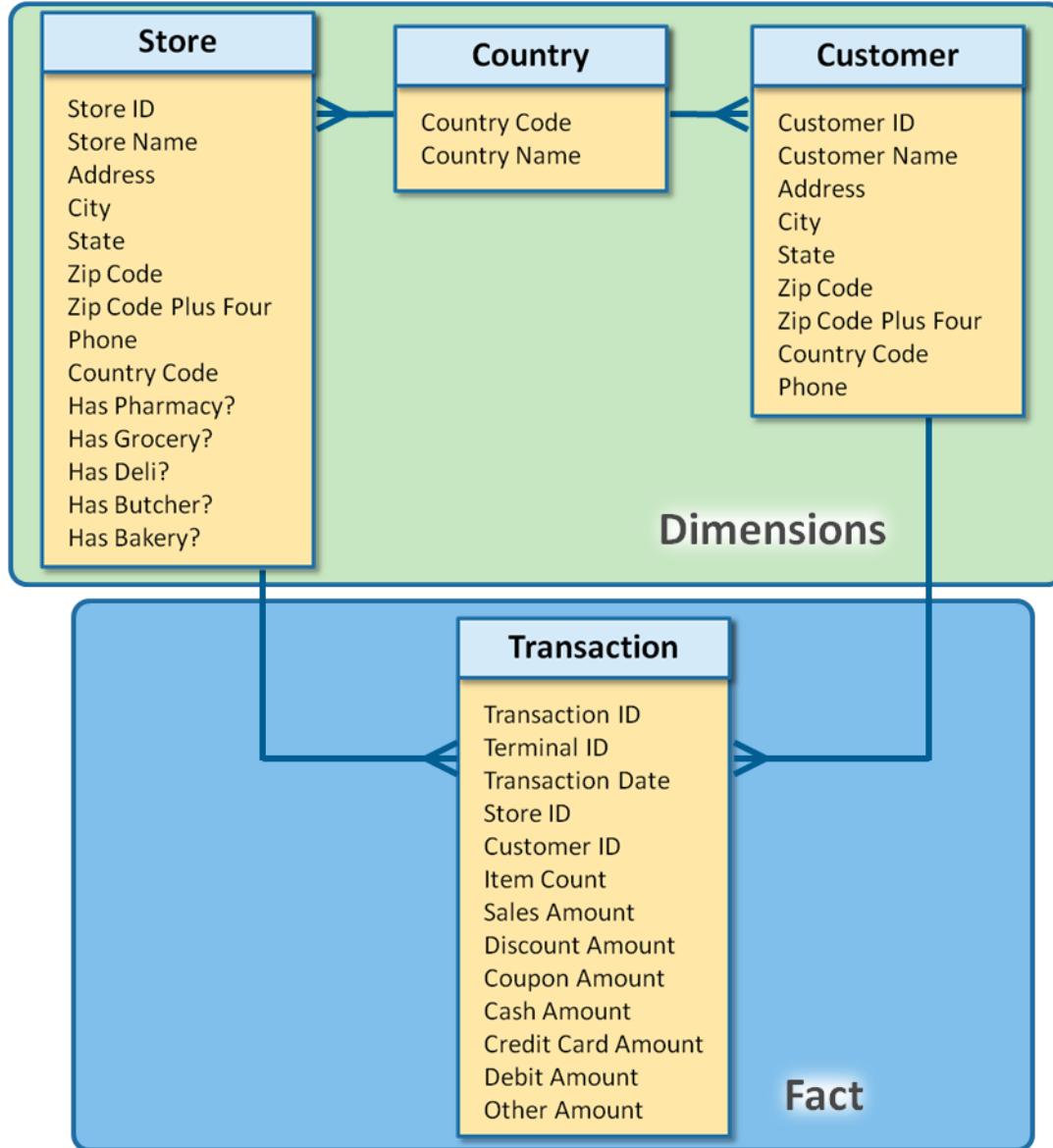
Step	Action
1.	<p>Create the datamart database using either of the methods discussed. From the Linux command prompt:</p> <pre>[gpadmin@mdw ~]\$ createdb datamart</pre> 
2.	<p>Access the datamart database as gpadmin:</p> <pre>[gpadmin@mdw ~]\$ psql datamart</pre> 
3.	<p>Create the facts and dimensions schema for the database:</p> <pre>datamart=# CREATE SCHEMA dimensions; datamart=# CREATE SCHEMA facts;</pre> 
4.	<p>Set the default search path for the datamart database to include the facts and dimensions schemas. You may wish to include public and pg_catalog as well.</p> <pre>datamart=# ALTER DATABASE datamart SET search_path=dimensions,facts,public,pg_catalog;</pre> 

End of Lab Exercise

This page intentionally left blank

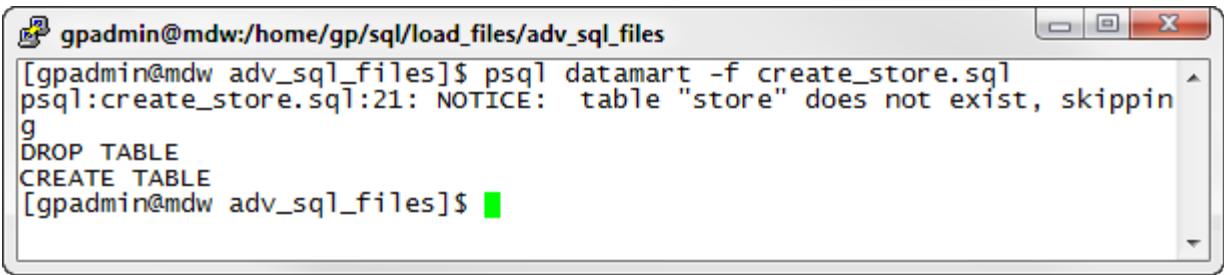
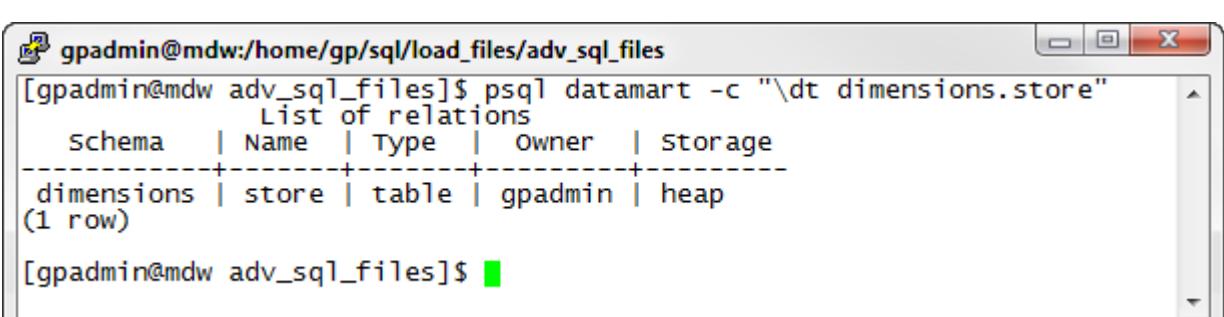
## Lab 17. Physical Design Decisions

<b>Purpose:</b>	In this lab, you create table objects based on business requirements provided in the lab. The choice of data types, constraints, distribution keys, and partitioning is up to you.  The logical data model to support the business model is provided below.
<b>Tasks:</b>	Students perform the following tasks: <ul style="list-style-type: none"><li>• Create the Store dimension</li><li>• Create the Country dimension</li><li>• Create the Customer dimension</li><li>• Create the Transaction fact table</li><li>• Load dimension and fact data</li></ul>
<b>References:</b>	Module 7 – Data Modeling and Design <ul style="list-style-type: none"><li>• Lesson: Physical Design Decisions</li></ul>



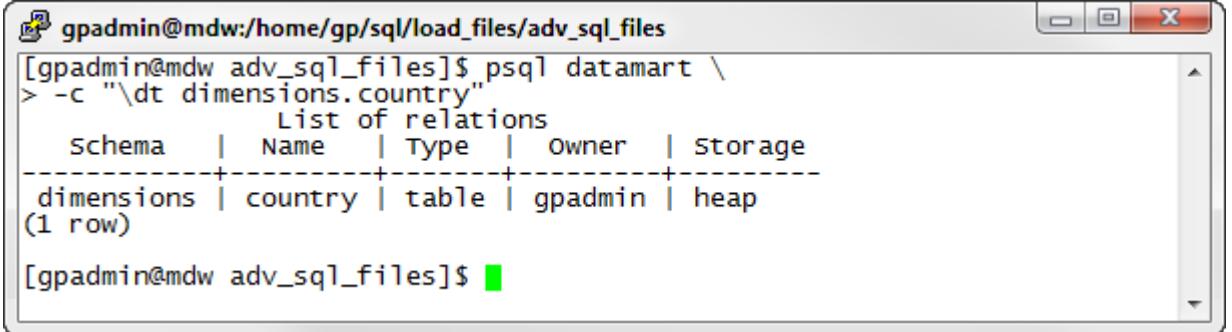
## Lab 17: Task 1 – Create the Store Dimension

Step	Action																																																												
1.	<p>The business requirements for the Store dimension are as follows:</p> <ol style="list-style-type: none"> <li>1. Each store is given a numeric identifier. This identifier is created in the source system (OLTP) and passed to the data warehouse. The Store id is unique for each store and may not be null.</li> <li>2. All address attributes are required.</li> <li>3. The phone number is stored as (XXX)XXX-XXXX formatted character string.</li> <li>4. The store name must allow for upper and lower case data and may contain single or double quotes in the name. The store name may not be null.</li> <li>5. The “Has” columns are passed as either 0 (false) or 1 (true) to the data warehouse. The default value for these columns in the OLTP system is 0.</li> <li>6. The country code is three characters (e.g., USA, CAN). Currently the business only operates in the United States and Canada.</li> <li>7. There are currently around thirty five stores in the US and Canada.</li> </ol> <p>Based on these requirements, implement the Store dimension.</p>																																																												
2.	Determine the table name for the Store entity? _____																																																												
3.	<p>Determine the column name, data type, and any constraints for each attribute of the Store entity.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr style="background-color: #cccccc;"> <th style="text-align: left; padding: 2px;">Attribute Name</th> <th style="text-align: left; padding: 2px;">Column Name</th> <th style="text-align: left; padding: 2px;">Data Type</th> <th style="text-align: left; padding: 2px;">Constraint</th> </tr> </thead> <tbody> <tr> <td style="padding: 2px;">Store ID</td><td style="padding: 2px;"></td><td style="padding: 2px;"></td><td style="padding: 2px;"></td></tr> <tr> <td style="padding: 2px;">Store Name</td><td style="padding: 2px;"></td><td style="padding: 2px;"></td><td style="padding: 2px;"></td></tr> <tr> <td style="padding: 2px;">Address</td><td style="padding: 2px;"></td><td style="padding: 2px;"></td><td style="padding: 2px;"></td></tr> <tr> <td style="padding: 2px;">City</td><td style="padding: 2px;"></td><td style="padding: 2px;"></td><td style="padding: 2px;"></td></tr> <tr> <td style="padding: 2px;">State</td><td style="padding: 2px;"></td><td style="padding: 2px;"></td><td style="padding: 2px;"></td></tr> <tr> <td style="padding: 2px;">Zip Code</td><td style="padding: 2px;"></td><td style="padding: 2px;"></td><td style="padding: 2px;"></td></tr> <tr> <td style="padding: 2px;">Zip Code Plus Four</td><td style="padding: 2px;"></td><td style="padding: 2px;"></td><td style="padding: 2px;"></td></tr> <tr> <td style="padding: 2px;">Phone</td><td style="padding: 2px;"></td><td style="padding: 2px;"></td><td style="padding: 2px;"></td></tr> <tr> <td style="padding: 2px;">Country Code</td><td style="padding: 2px;"></td><td style="padding: 2px;"></td><td style="padding: 2px;"></td></tr> <tr> <td style="padding: 2px;">Has Pharmacy</td><td style="padding: 2px;"></td><td style="padding: 2px;"></td><td style="padding: 2px;"></td></tr> <tr> <td style="padding: 2px;">Has Grocery</td><td style="padding: 2px;"></td><td style="padding: 2px;"></td><td style="padding: 2px;"></td></tr> <tr> <td style="padding: 2px;">Has Deli</td><td style="padding: 2px;"></td><td style="padding: 2px;"></td><td style="padding: 2px;"></td></tr> <tr> <td style="padding: 2px;">Has Butcher</td><td style="padding: 2px;"></td><td style="padding: 2px;"></td><td style="padding: 2px;"></td></tr> <tr> <td style="padding: 2px;">Has Bakery</td><td style="padding: 2px;"></td><td style="padding: 2px;"></td><td style="padding: 2px;"></td></tr> </tbody> </table>	Attribute Name	Column Name	Data Type	Constraint	Store ID				Store Name				Address				City				State				Zip Code				Zip Code Plus Four				Phone				Country Code				Has Pharmacy				Has Grocery				Has Deli				Has Butcher				Has Bakery			
Attribute Name	Column Name	Data Type	Constraint																																																										
Store ID																																																													
Store Name																																																													
Address																																																													
City																																																													
State																																																													
Zip Code																																																													
Zip Code Plus Four																																																													
Phone																																																													
Country Code																																																													
Has Pharmacy																																																													
Has Grocery																																																													
Has Deli																																																													
Has Butcher																																																													
Has Bakery																																																													
4.	What will be the DISTRIBUTION KEY? Is it UNIQUE? _____																																																												

Step	Action
5.	<p>Create an SQL file with the <code>CREATE</code> statement. Be sure to include any column and table constraints. There is an example (non-optimized) file called <code>create_store.sql</code> in the <code>/home/gp/sql/load_files/adv_sql_files</code> that you can use as a guide.</p> <pre>[gpadmin@mdw ~]\$ cd /home/gp/sql/load_files/adv_sql_files [gpadmin@mdw adv_sql_files]\$ vi create_store.sql</pre> <p>You will be accessing the directory <code>/home/gp/sql/load_files</code> directory throughout this lab. To save some time, create an environment variable that points to that directory: <code>export LF=/home/gp/sql/load_files</code></p>
6.	<p>Execute the script as gpadmin by starting a <code>psql</code> session and passing in the file name with the <code>-f</code> option. Include the datamart database as part of the <code>psql</code> command:</p> <pre>[gpadmin@mdw adv_sql_files]\$ psql datamart -f create_store.sql</pre>  <p>You should see the words <code>CREATE TABLE</code> immediately under the Linux prompt if there were no errors.</p>
7.	<p>Verify the table has been created:</p> <pre>[gpadmin@mdw adv_sql_files]\$ psql datamart -c "\dt dimensions.store"</pre>  <p>One row with the table name should be displayed.</p>

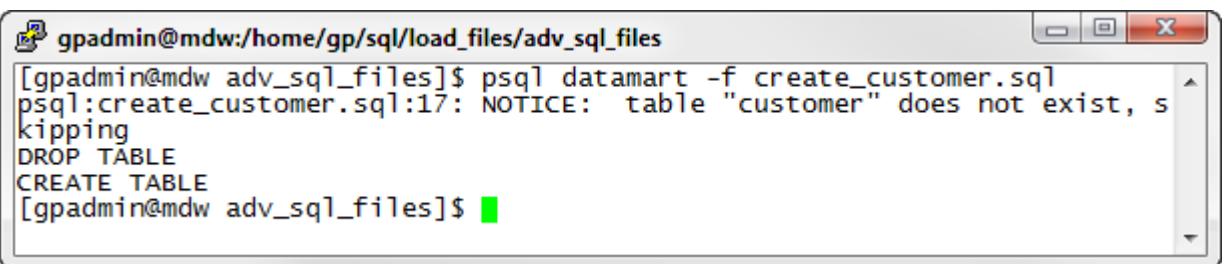
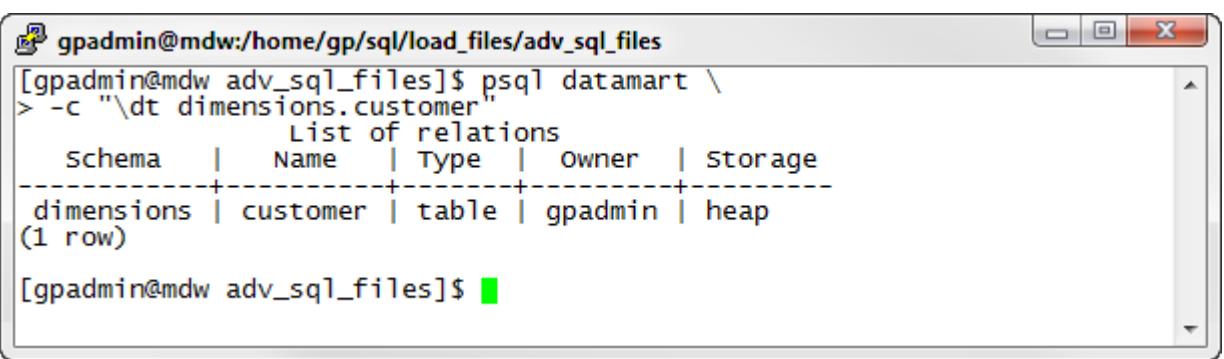
## Lab 17: Task 2 – Create the Country Dimension

Step	Action												
1.	<p>The business requirements for the Country dimension are as follows:</p> <ol style="list-style-type: none"> <li>1. The country code is three characters, such as USA or CAN. Currently, the business only operates in the United States and Canada. This column cannot be NULL.</li> <li>2. The country name is a text column.</li> </ol> <p>Based on these requirements, implement the Country dimension.</p>												
2.	Determine the table name for the Country entity from the LDM. <hr/>												
3.	<p>Determine the column name, data type and any constraints for each attribute of the Store entity.</p> <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr style="background-color: #4f81bd; color: white;"> <th style="padding: 2px;">Attribute Name</th> <th style="padding: 2px;">Column Name</th> <th style="padding: 2px;">Data Type</th> <th style="padding: 2px;">Constraint</th> </tr> </thead> <tbody> <tr> <td style="padding: 2px;">Country Code</td> <td style="padding: 2px;"></td> <td style="padding: 2px;"></td> <td style="padding: 2px;"></td> </tr> <tr> <td style="padding: 2px;">Country Name</td> <td style="padding: 2px;"></td> <td style="padding: 2px;"></td> <td style="padding: 2px;"></td> </tr> </tbody> </table>	Attribute Name	Column Name	Data Type	Constraint	Country Code				Country Name			
Attribute Name	Column Name	Data Type	Constraint										
Country Code													
Country Name													
4.	<p>What will be the DISTRIBUTION KEY? Is it UNIQUE?</p> <hr/>												
5.	<p>Create an SQL file with the CREATE TABLE statement.</p> <p>Be sure to include any column and table constraints.</p> <p>There is an example (non-optimized) file called <code>create_country.sql</code> in the <code>/home/gp/sql/load_files/adv_sql_files</code> that you can use as a guide.</p> <pre>[gpadmin@mdw adv_sql_files]\$ vi create_country.sql</pre>												
6.	<p>Execute the script as gpadmin by starting a psql session and passing in the file name with the <code>-f</code> option. Include the datamart database as part of the psql command:</p> <pre>[gpadmin@mdw adv_sql_files]\$ psql datamart -f create_country.sql</pre> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;">  gpadmin@mdw:/home/gp/sql/load_files/adv_sql_files <pre>[gpadmin@mdw adv_sql_files]\$ psql datamart -f create_country.sql psql:create_country.sql:11: NOTICE:  table "country" does not exist, skipping DROP TABLE CREATE TABLE [gpadmin@mdw adv_sql_files]\$ █</pre> </div> <p>You should see the words CREATE TABLE immediately under the UNIX prompt if there were no errors.</p>												

Step	Action
7.	<p>Verify the table has been created:</p> <pre>[gpadmin@mdw adv_sql_files]\$ psql datamart \ -c "\dt dimensions.country"</pre>  <p>One row with the table name should be displayed.</p>

## Lab 17: Task 3 – Create the Customer Dimension

Step	Action																																								
1.	<p>The business requirements for the Customer dimension are as follows:</p> <ol style="list-style-type: none"> <li>1. Each customer is tracked by their phone number as part of their membership rewards program.</li> <li>2. There is a default customer with a (999)999-9999 customer id that is used when a transaction occurs for a customer that is not a member of the rewards program.</li> <li>3. The customer name may not be NULL.</li> <li>4. All address attributes are required.</li> <li>5. The phone number is stored as (XXX)XXX-XXXX formatted character string.</li> <li>6. The country code is three characters (e.g., USA, CAN). Currently the business only operates in the United States and Canada.</li> <li>7. There are approximately 10,000 customers signed up for the rewards program.</li> <li>8. The customer id will have to be generated during the ETL process as it is not a part of the OLTP system.</li> </ol> <p>Based on these requirements, implement the Customer dimension.</p>																																								
2.	Determine the table name for the Customer entity from the LDM. <hr/>																																								
3.	<p>Determine the column name, data type and any constraints for each attribute of the Store entity.</p> <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr style="background-color: #4f81bd; color: white;"> <th style="padding: 5px;">Attribute Name</th> <th style="padding: 5px;">Column Name</th> <th style="padding: 5px;">Data Type</th> <th style="padding: 5px;">Constraint</th> </tr> </thead> <tbody> <tr> <td style="padding: 5px;">Customer ID</td><td style="padding: 5px;"></td><td style="padding: 5px;"></td><td style="padding: 5px;"></td></tr> <tr> <td style="padding: 5px;">Customer Name</td><td style="padding: 5px;"></td><td style="padding: 5px;"></td><td style="padding: 5px;"></td></tr> <tr> <td style="padding: 5px;">Address</td><td style="padding: 5px;"></td><td style="padding: 5px;"></td><td style="padding: 5px;"></td></tr> <tr> <td style="padding: 5px;">City</td><td style="padding: 5px;"></td><td style="padding: 5px;"></td><td style="padding: 5px;"></td></tr> <tr> <td style="padding: 5px;">State</td><td style="padding: 5px;"></td><td style="padding: 5px;"></td><td style="padding: 5px;"></td></tr> <tr> <td style="padding: 5px;">Zip Code</td><td style="padding: 5px;"></td><td style="padding: 5px;"></td><td style="padding: 5px;"></td></tr> <tr> <td style="padding: 5px;">Zip Code Plus Four</td><td style="padding: 5px;"></td><td style="padding: 5px;"></td><td style="padding: 5px;"></td></tr> <tr> <td style="padding: 5px;">Phone</td><td style="padding: 5px;"></td><td style="padding: 5px;"></td><td style="padding: 5px;"></td></tr> <tr> <td style="padding: 5px;">Country Code</td><td style="padding: 5px;"></td><td style="padding: 5px;"></td><td style="padding: 5px;"></td></tr> </tbody> </table>	Attribute Name	Column Name	Data Type	Constraint	Customer ID				Customer Name				Address				City				State				Zip Code				Zip Code Plus Four				Phone				Country Code			
Attribute Name	Column Name	Data Type	Constraint																																						
Customer ID																																									
Customer Name																																									
Address																																									
City																																									
State																																									
Zip Code																																									
Zip Code Plus Four																																									
Phone																																									
Country Code																																									
4.	What will be the DISTRIBUTION KEY? Is it UNIQUE? <hr/>																																								

Step	Action															
5.	<p>Create an SQL file with the CREATE TABLE statement. Be sure to include any column and table constraints. There is an example (non-optimized) file called <code>create_customer.sql</code> in the <code>/home/gp/sql/load_files /adv_sql_files</code> that you can use as a guide.</p> <pre data-bbox="331 407 1228 439">[gpadmin@mdw adv_sql_files]\$ vi create_customer.sql</pre>															
6.	<p>Execute the script as gpadmin by starting a psql session and passing in the file name with the <code>-f</code> option. Include the datamart database as part of the psql command:</p> <pre data-bbox="314 608 1457 639">[gpadmin@mdw adv_sql_files]\$ psql datamart -f create_customer.sql</pre>  <p>You should see the words CREATE TABLE immediately under the UNIX prompt if there were no errors.</p>															
7.	<p>Verify that the table has been created:</p> <pre data-bbox="314 1146 1090 1210">[gpadmin@mdw adv_sql_files]\$ psql datamart \ -c "\dt dimensions.customer"</pre>  <table border="1"> <thead> <tr> <th data-bbox="372 1358 470 1379">Schema</th> <th data-bbox="510 1358 592 1379">Name</th> <th data-bbox="633 1358 714 1379">Type</th> <th data-bbox="755 1358 837 1379">Owner</th> <th data-bbox="878 1358 1122 1379">Storage</th> </tr> </thead> <tbody> <tr> <td data-bbox="331 1421 445 1442">dimensions</td> <td data-bbox="486 1421 600 1442">customer</td> <td data-bbox="641 1421 722 1442">table</td> <td data-bbox="763 1421 845 1442">gpadmin</td> <td data-bbox="886 1421 967 1442">heap</td> </tr> <tr> <td data-bbox="331 1453 445 1474">(1 row)</td> <td data-bbox="486 1453 600 1474"></td> <td data-bbox="641 1453 722 1474"></td> <td data-bbox="763 1453 845 1474"></td> <td data-bbox="886 1453 967 1474"></td> </tr> </tbody> </table>	Schema	Name	Type	Owner	Storage	dimensions	customer	table	gpadmin	heap	(1 row)				
Schema	Name	Type	Owner	Storage												
dimensions	customer	table	gpadmin	heap												
(1 row)																

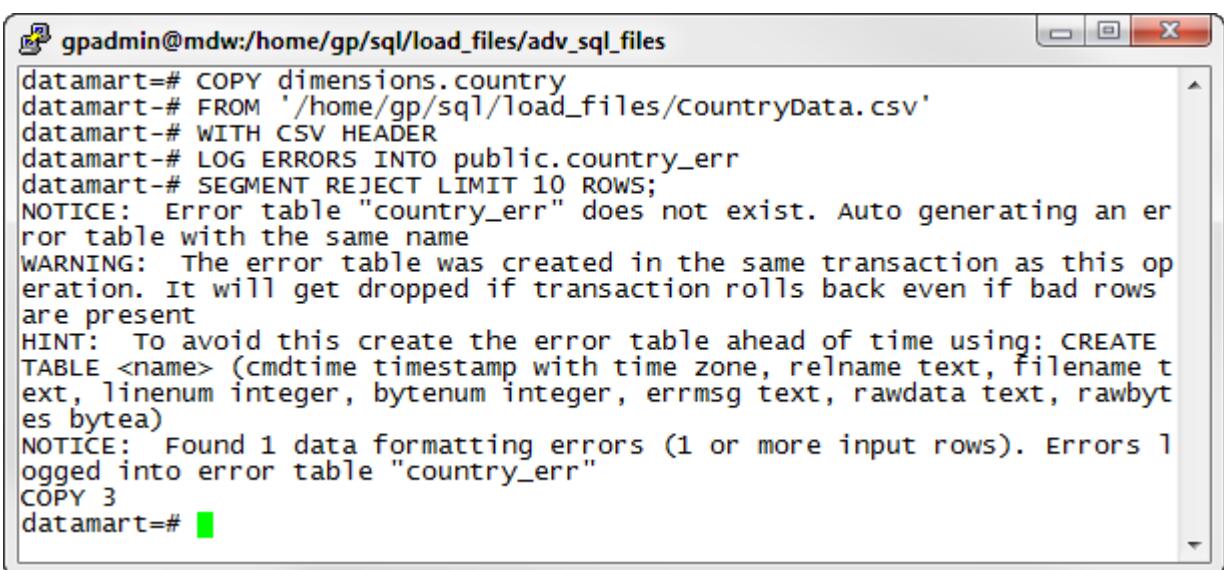
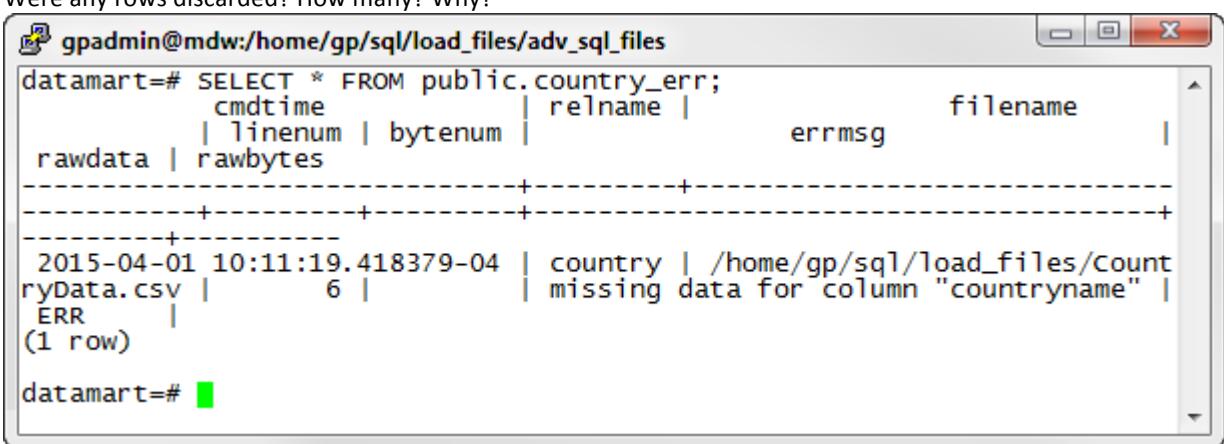
## Lab 17: Task 4 – Create the Transaction Fact Table

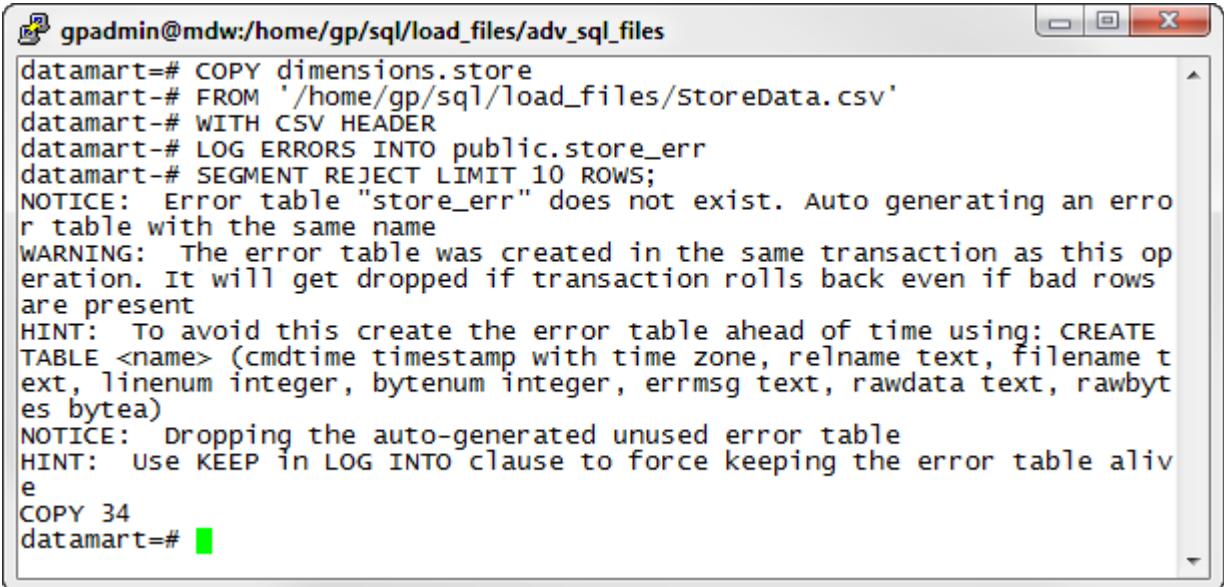
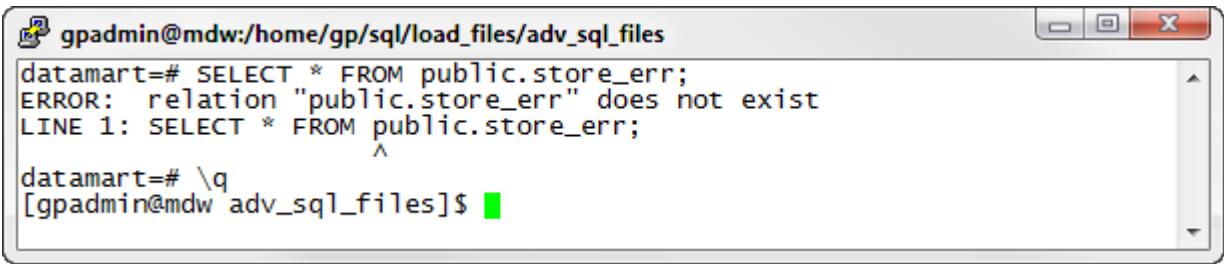
Step	Action
1.	<p>The business requirements for the Transaction fact table are as follows:</p> <ol style="list-style-type: none"> <li>1. The transaction represents all transactions that occur by a customer, at a checkout stand, at a given store by day. (In data warehousing this is called the “shopping basket” type of transaction.) It is, in effect, a summary of the all items and includes tax, payment methods, etc.</li> <li>2. Each transaction is given a numeric identifier. This identifier is created in the source system (OLTP) and passed to the data warehouse. The transaction id is unique for each customer transaction and may not be null.</li> <li>3. If the customer is not a rewards member, the default phone number is (999)999-9999. Otherwise rewards customers are tracked by their phone numbers.</li> <li>4. The terminal id may not be null. All terminals are numbered from 1-N in each store. There are no more than 24 terminals in any given store at implementation time.</li> <li>5. Item count may not be zero or negative.</li> <li>6. Sales Amount may not be negative.</li> <li>7. Taxes are represented in the “other amount” attribute.</li> <li>8. Cash Amount + Debit Amount + Credit Card Amount may not exceed Sales Amount plus Other Amount.</li> <li>9. There is an average of 100,000 transactions per day across the entire business.</li> <li>10. Ninety percent of all transactions involve a rewards member customer.</li> <li>11. The transaction date is a timestamp containing the date with the time. The business only wishes to store the date portion of the transaction date.</li> <li>12. The historical data that will be loaded only goes back to the beginning of 2008.</li> </ol>
2.	Determine the table name for Transaction entity. _____

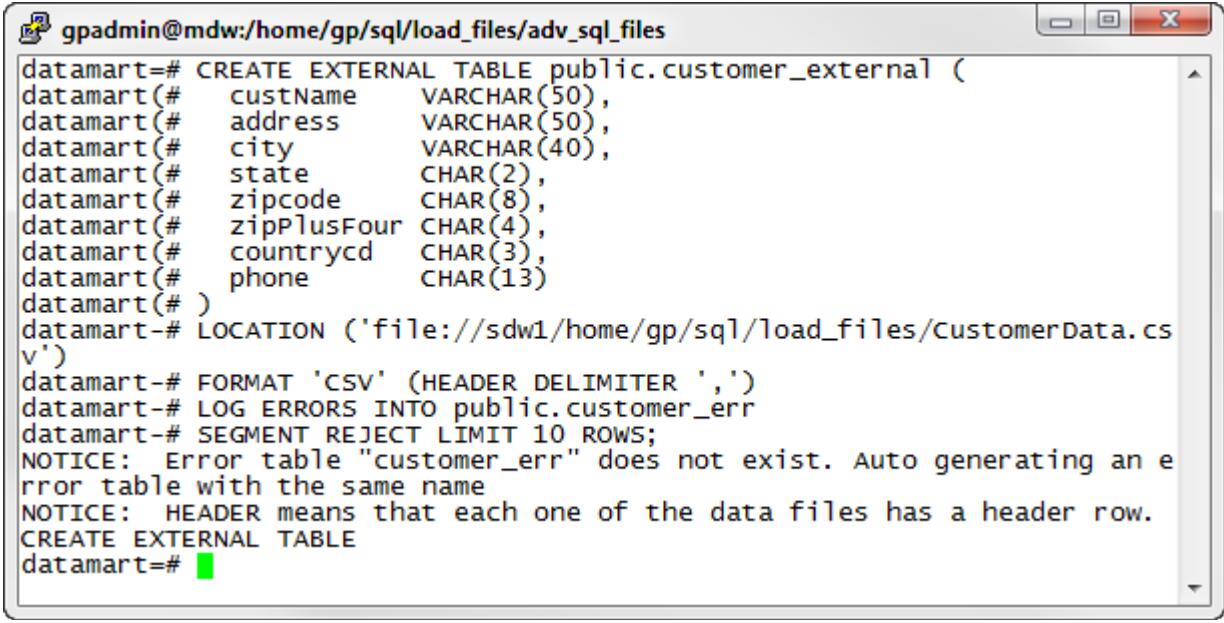
Step	Action																																																								
3.	<p>Determine the column name, data type, and any constraints for each attribute of the Transaction entity.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr style="background-color: #4f81bd; color: white;"> <th style="text-align: left;">Attribute Name</th><th style="text-align: left;">Column Name</th><th style="text-align: left;">Data Type</th><th style="text-align: left;">Constraint</th></tr> </thead> <tbody> <tr><td>Transaction ID</td><td></td><td></td><td></td></tr> <tr><td>terminal ID</td><td></td><td></td><td></td></tr> <tr><td>Transaction Date</td><td></td><td></td><td></td></tr> <tr><td>Store ID</td><td></td><td></td><td></td></tr> <tr><td>Customer ID</td><td></td><td></td><td></td></tr> <tr><td>Item Count</td><td></td><td></td><td></td></tr> <tr><td>Sales Amount</td><td></td><td></td><td></td></tr> <tr><td>Discount Amount</td><td></td><td></td><td></td></tr> <tr><td>Coupon Amount</td><td></td><td></td><td></td></tr> <tr><td>Cash Amount</td><td></td><td></td><td></td></tr> <tr><td>Credit Card Amount</td><td></td><td></td><td></td></tr> <tr><td>Debit Amount</td><td></td><td></td><td></td></tr> <tr><td>Other Amount</td><td></td><td></td><td></td></tr> </tbody> </table>	Attribute Name	Column Name	Data Type	Constraint	Transaction ID				terminal ID				Transaction Date				Store ID				Customer ID				Item Count				Sales Amount				Discount Amount				Coupon Amount				Cash Amount				Credit Card Amount				Debit Amount				Other Amount			
Attribute Name	Column Name	Data Type	Constraint																																																						
Transaction ID																																																									
terminal ID																																																									
Transaction Date																																																									
Store ID																																																									
Customer ID																																																									
Item Count																																																									
Sales Amount																																																									
Discount Amount																																																									
Coupon Amount																																																									
Cash Amount																																																									
Credit Card Amount																																																									
Debit Amount																																																									
Other Amount																																																									
4.	<p>What will be the DISTRIBUTION KEY? Is it UNIQUE?</p> <hr/>																																																								
5.	<p>Create an SQL file with the CREATE TABLE statement. Be sure to include any column and table constraints.</p> <p>There is an example (non-optimized) file called <code>create_transaction.sql</code> in the <code>/home/gp/sql/load_files /adv_sql_files</code> that you can use as a guide.</p> <pre>[gpadmin@mdw adv_sql_files]\$ vi create_transaction.sql</pre>																																																								
6.	<p>Execute the script as <code>gpadmin</code> by starting a <code>psql</code> session and passing in the file name with the <code>-f</code> option. Include the datamart database as part of the <code>psql</code> command:</p> <pre>[gpadmin@mdw adv_sql_files]\$ psql datamart -f create_transaction.sql</pre> <div style="border: 1px solid black; padding: 5px; width: fit-content;">  gpadmin@mdw:/home/gp/sql/load_files/adv_sql_files <pre>[gpadmin@mdw adv_sql_files]\$ psql datamart -f create_transaction.sql psql:create_transaction.sql:22: NOTICE:  table "transaction" does not exist, skipping DROP TABLE CREATE TABLE [gpadmin@mdw adv_sql_files]\$ █</pre> </div>																																																								

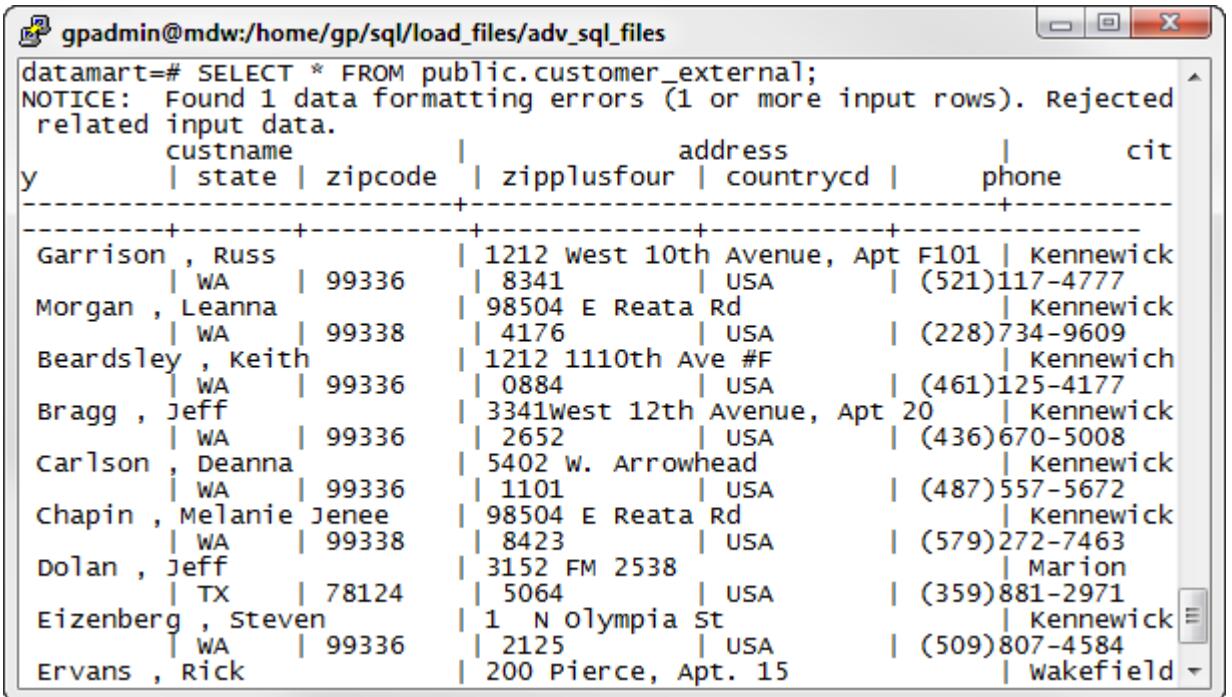
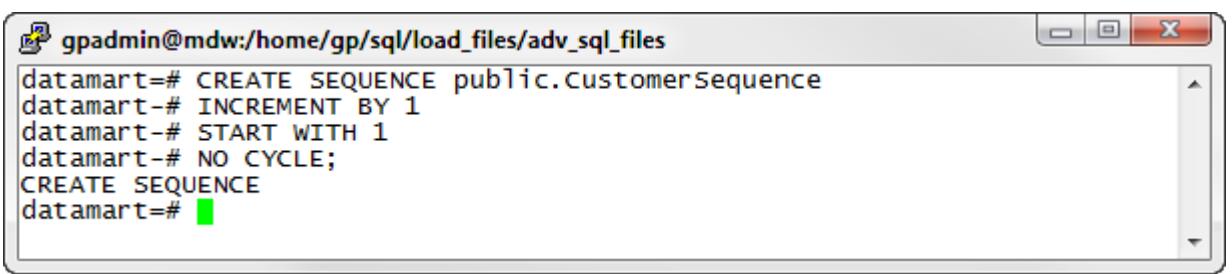
Step	Action
7.	<p>Verify the table has been created:</p> <pre>[gpadmin@mdw adv_sql_files]\$ psql datamart -c "\d+ facts.transaction"</pre> <pre>[gpadmin@mdw adv_sql_files]\$ psql datamart -c "\d+ facts.transaction" Table "facts.transaction"  Column   Type        Modifiers   Storage   Description -----+-----+-----+-----+-----+  transid   bigint                  plain      terminalid   integer                plain      transdate   date                   plain      storeid   smallint               plain      customerid   integer               plain      itemcnt   integer               plain      salesamt   numeric(9,2)               main      taxamt   numeric(9,2)               main      discountamt   numeric(9,2)   default 0.0   main      couponamt   numeric(9,2)   default 0.0   main      cashamt   numeric(9,2)   default 0.0   main      checkamt   numeric(9,2)   default 0.0   main      ccamt   numeric(9,2)   default 0.0   main      debitamt   numeric(9,2)   default 0.0   main      otheramt   numeric(9,2)   default 0.0   main      Has OIDs: no Distributed by: (transid)  [gpadmin@mdw adv_sql_files]\$</pre>

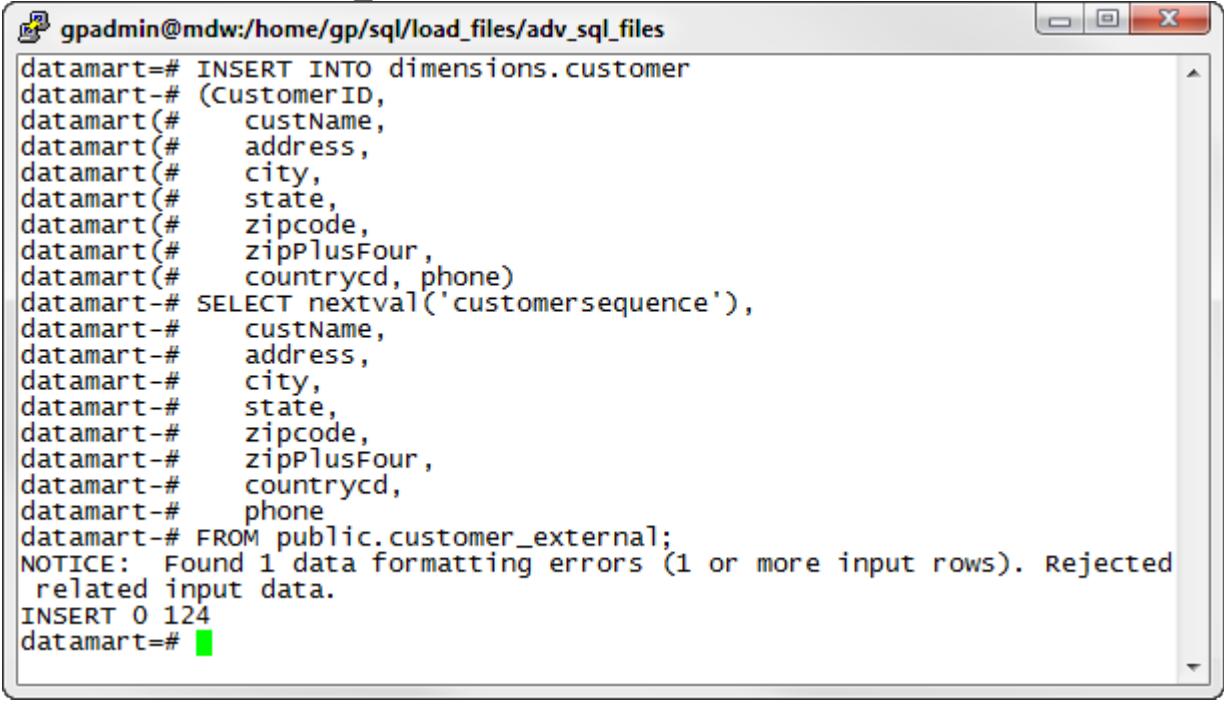
## Lab 17: Task 5 – Load Dimension and Fact Data

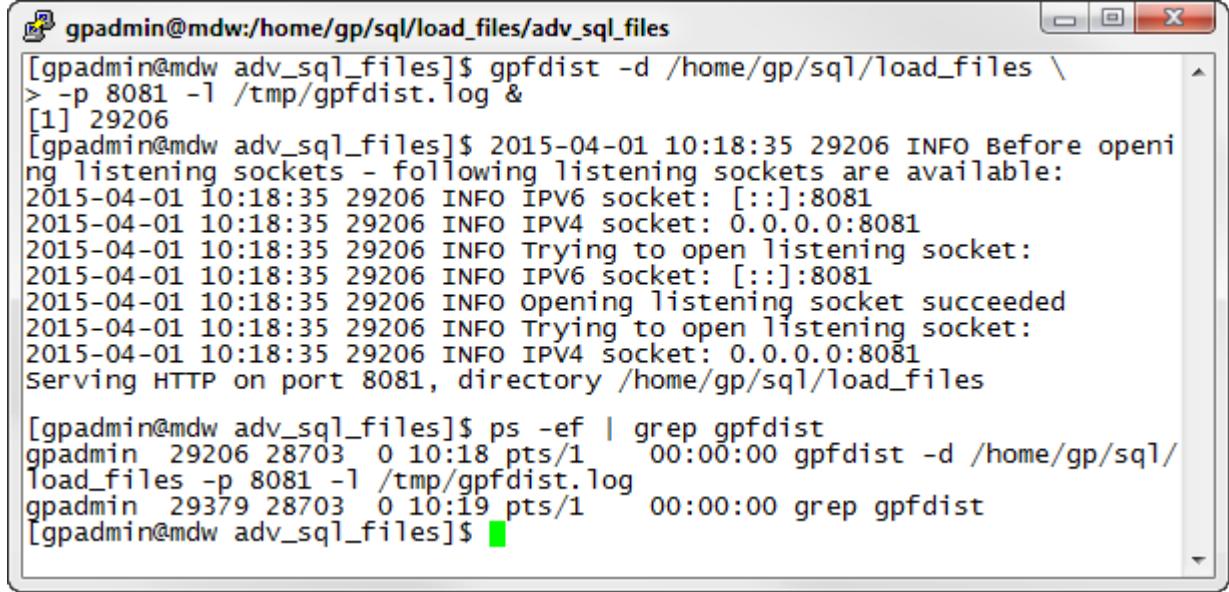
Step	Action																		
1.	<p>Connect to the datamart database as gpadmin.</p> <pre>[gpadmin@mdw adv_sql_files]\$ psql datamart</pre>																		
2.	<p>Load the dimensions.country table using the COPY command. Copy the data from /home/gp/sql/load_files/CountryData.csv into the dimensions.country table.</p> <pre>datamart=# COPY dimensions.country FROM '/home/gp/sql/load_files/CountryData.csv' WITH CSV HEADER LOG ERRORS INTO public.country_err SEGMENT REJECT LIMIT 10 ROWS;</pre> 																		
3.	<p>Check the error table, public.country_err for any errors:</p> <pre>datamart=# SELECT * FROM public.country_err;</pre> <p>Were any rows discarded? How many? Why?</p>  <table border="1"> <thead> <tr> <th>cmdtime</th> <th>relname</th> <th>filename</th> </tr> <tr> <th>linenum</th> <th>bytenum</th> <th>errmsg</th> </tr> <tr> <th>rawdata</th> <th>rawbytes</th> <th></th> </tr> </thead> <tbody> <tr> <td>2015-04-01 10:11:19.418379-04</td> <td>country</td> <td>/home/gp/sql/load_files/CountryData.csv</td> </tr> <tr> <td>ERR</td> <td>6</td> <td>missing data for column "countryname"</td> </tr> <tr> <td>(1 row)</td> <td></td> <td></td> </tr> </tbody> </table>	cmdtime	relname	filename	linenum	bytenum	errmsg	rawdata	rawbytes		2015-04-01 10:11:19.418379-04	country	/home/gp/sql/load_files/CountryData.csv	ERR	6	missing data for column "countryname"	(1 row)		
cmdtime	relname	filename																	
linenum	bytenum	errmsg																	
rawdata	rawbytes																		
2015-04-01 10:11:19.418379-04	country	/home/gp/sql/load_files/CountryData.csv																	
ERR	6	missing data for column "countryname"																	
(1 row)																			

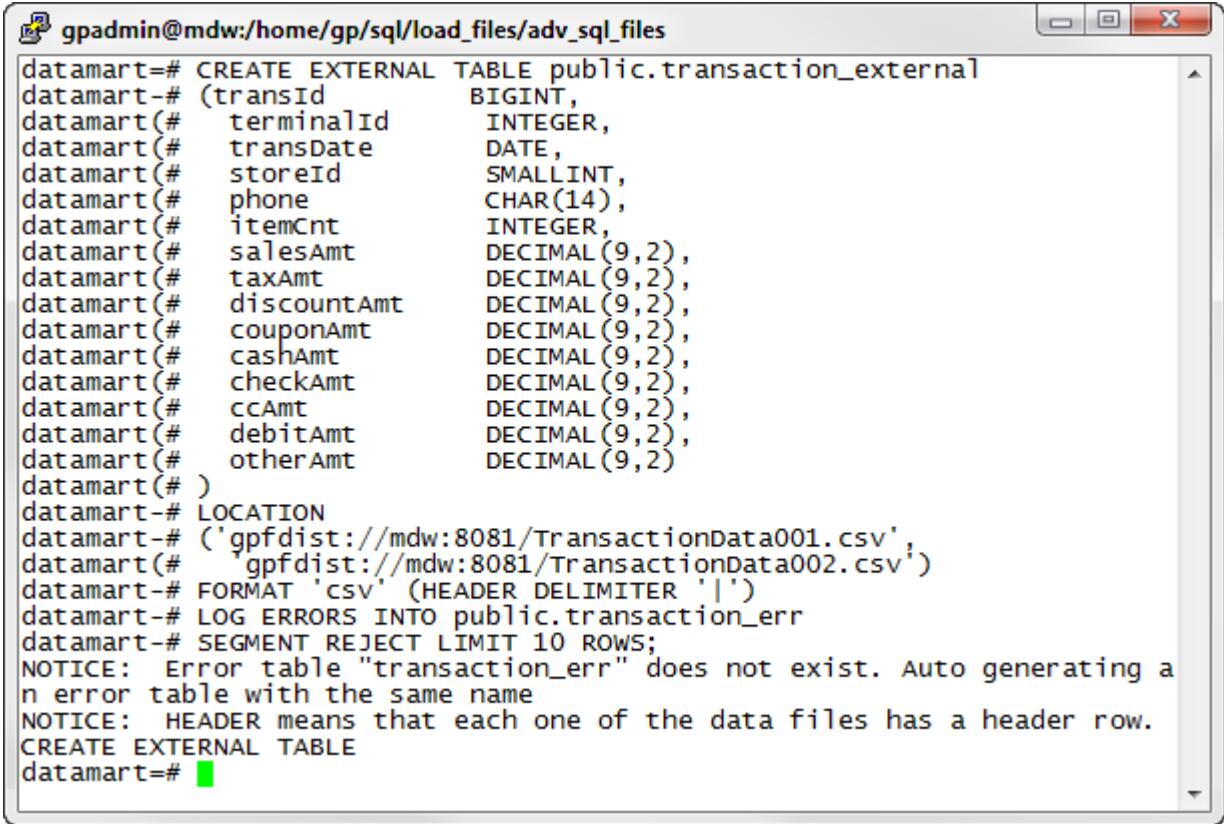
Step	Action
4.	<p>Load the dimensions.store table using the COPY command. Copy the data from /home/gp/sql/load_files/StoreData.csv into the dimensions.store table.</p> <pre>datamart=# COPY dimensions.store FROM '/home/gp/sql/load_files/StoreData.csv' WITH CSV HEADER LOG ERRORS INTO public.store_err SEGMENT REJECT LIMIT 10 ROWS;</pre> 
5.	<p>Check the error table, <code>public.store_err</code> for any errors:</p> <pre>datamart=# SELECT * FROM public.store_err;</pre>  <p>Were any rows discarded? How many? Why?</p> <hr/> <hr/>
6.	<p>Connect to the <code>datamart</code> database as <code>gpadmin</code>.</p> <pre>[gpadmin@mdw adv_sql_files]\$ psql datamart</pre>

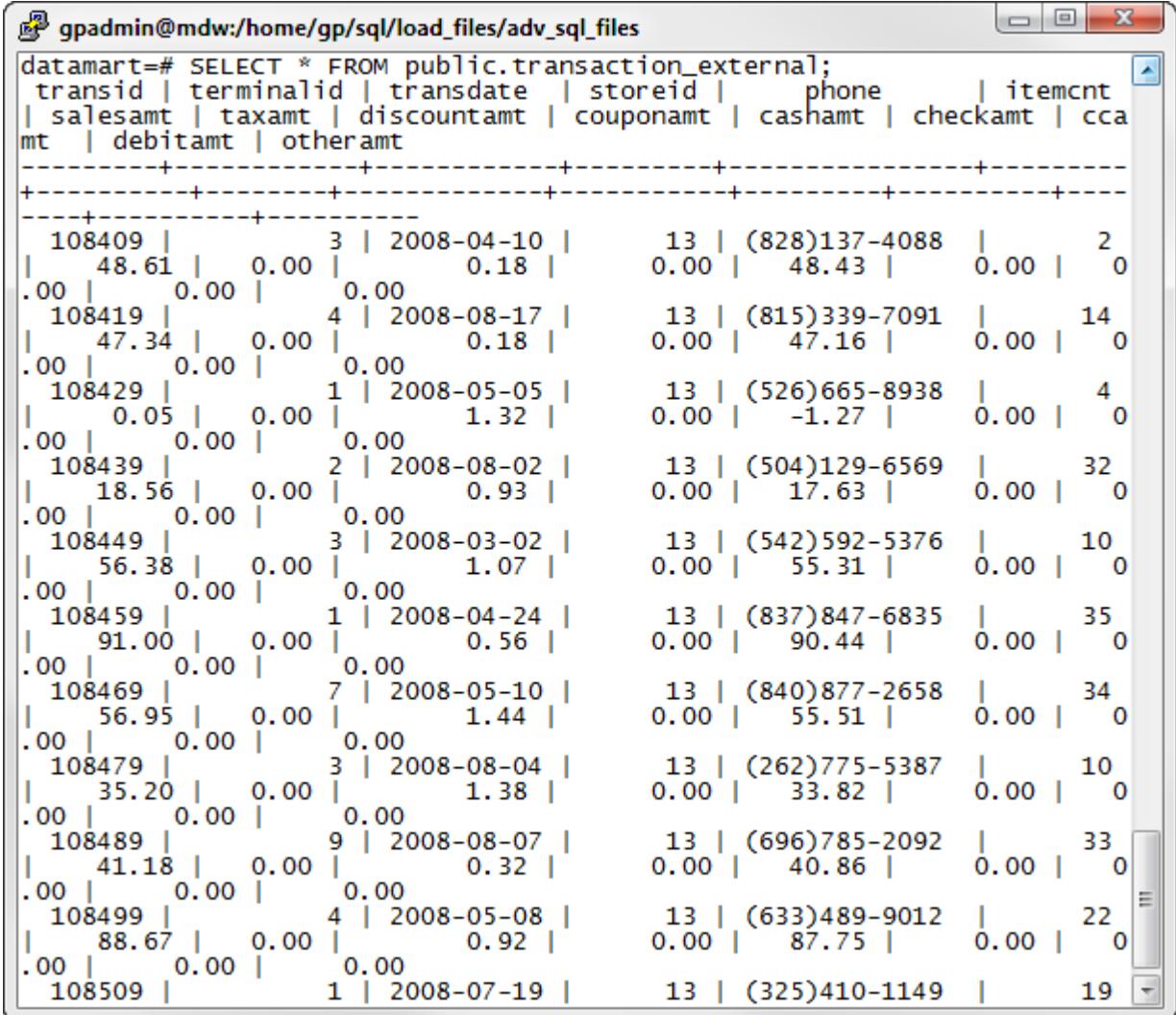
Step	Action
7.	<p>Load the dimensions.customer table using an external table. Create the external table using the following syntax:</p> <pre>datamart=# CREATE EXTERNAL TABLE public.customer_external (     custName      VARCHAR(50),     address       VARCHAR(50),     city          VARCHAR(40),     state         CHAR(2),     zipcode       CHAR(8),     zipPlusFour   CHAR(4),     countrycd    CHAR(3),     phone         CHAR(13) ) LOCATION ('file:///sdw1/home/gp/sql/load_files/CustomerData.csv') FORMAT 'CSV' (HEADER DELIMITER ',') LOG ERRORS INTO public.customer_err SEGMENT REJECT LIMIT 10 ROWS;</pre> 

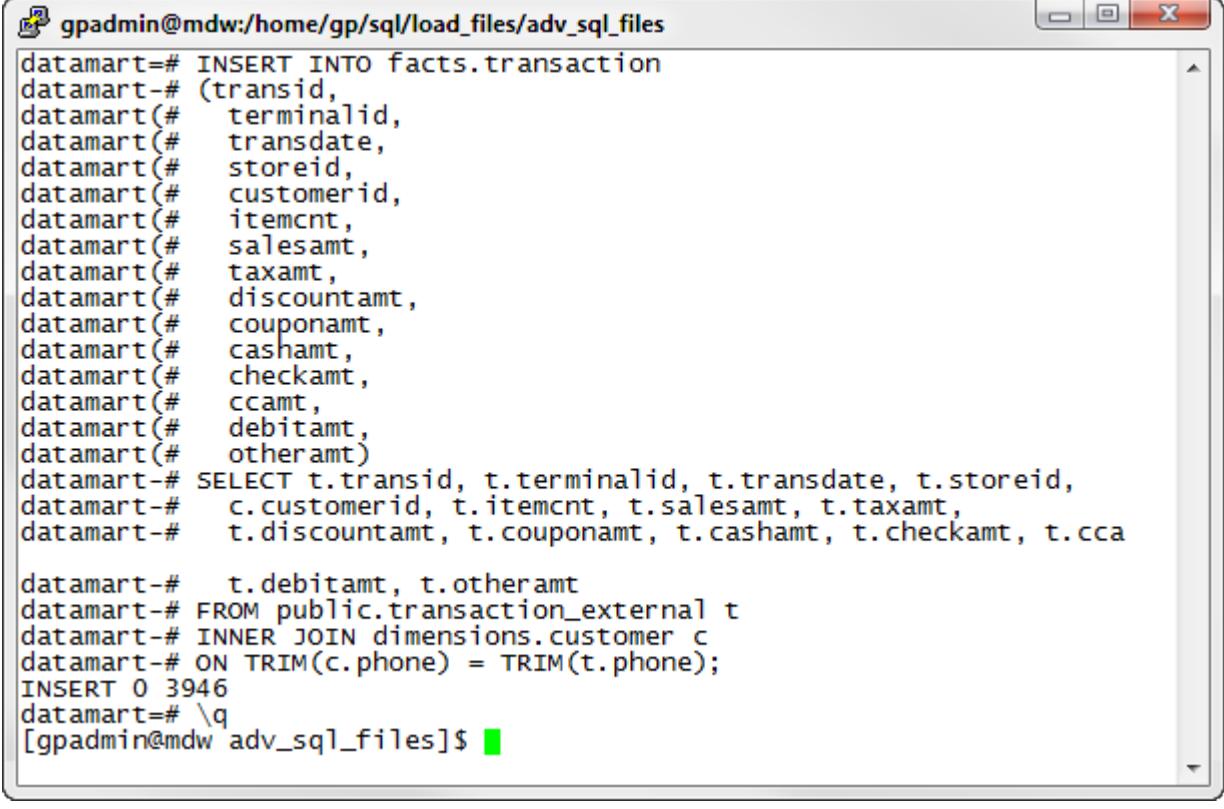
Step	Action
8.	<p>Validate that the table was created correctly:</p> <pre>datamart=# SELECT * FROM public.customer_external;</pre> 
9.	<p>The customers for this business are tracked by their phone numbers. A customer ID must be created for each of the customers. You will use a sequence number to create a unique customer ID for each customer. Create a sequence to support the data load of the customer dimension:</p> <pre>datamart=# CREATE SEQUENCE public.CustomerSequence INCREMENT BY 1 START WITH 1 NO CYCLE;</pre> 

Step	Action
10.	<p>Insert the rows from the external table into the target table. Note that you are performing a transform of the ELT process, so you will need to specify all of the columns from the source table. Note that the sequence name is all lower case.</p> <pre data-bbox="306 304 1057 931"> datamart=# INSERT INTO dimensions.customer (CustomerID,  custName,  address,  city,  state,  zipcode,  zipPlusFour,  countrycd, phone) SELECT nextval('customersequence'),  custName,  address,  city,  state,  zipcode,  zipPlusFour,  countrycd,  phone FROM public.customer_external; </pre> 
11.	<p>Were any rows discarded? How many? Why?</p> <hr/> <hr/> <hr/>

Step	Action
12.	Exit the PSQL session.
13.	<p>Load the Transaction data using gpfdist and an external table. You will perform a single transformation during the <code>INSERT... SELECT</code> and that is a join to the Customer dimension using the phone column to get the <code>customerid</code>.</p> <p>Start a <code>gpfdist</code> session and verify that it is executing:</p> <pre>[gpadmin@mdw adv_sql_files]\$ gpfdist -d /home/gp/sql/load_files \ -p 8081 -l /tmp/gpfdist.log &amp;</pre> <p>Verify that <code>gpfdist</code> is running.</p> <pre>[gpadmin@mdw adv_sql_files]\$ ps -ef   grep gpfdist</pre>  <p><b>Note:</b> If you receive an internal error that <code>gpfdist</code> cannot create a socket, verify that another <code>gpfdist</code> process is not running on the port number you specified.</p>
14.	<p>Connect to the datamart database as <code>gpadmin</code>.</p> <pre>[gpadmin@mdw adv_sql_files]\$ psql datamart</pre>

Step	Action
15.	<p>Create a gpfdist based external table to load the Transaction fact table. Connect to the datamart database and use the following DDL to create the external table:</p> <pre>datamart=# CREATE EXTERNAL TABLE public.transaction_external (transId          BIGINT,  terminalId      INTEGER,  transDate       DATE,  storeId         SMALLINT,  phone           CHAR(14),  itemCnt         INTEGER,  salesAmt        DECIMAL(9,2),  taxAmt          DECIMAL(9,2),  discountAmt     DECIMAL(9,2),  couponAmt       DECIMAL(9,2),  cashAmt         DECIMAL(9,2),  checkAmt        DECIMAL(9,2),  ccAmt           DECIMAL(9,2),  debitAmt        DECIMAL(9,2),  otherAmt        DECIMAL(9,2) ) LOCATION ('gpfdist://mdw:8081/TransactionData001.csv',  'gpfdist://mdw:8081/TransactionData002.csv') FORMAT 'csv' (HEADER DELIMITER ' ') LOG ERRORS INTO public.transaction_err SEGMENT REJECT LIMIT 10 ROWS;</pre> 

Step	Action
16.	<p>Validate that the table was created correctly:</p> <pre>datamart=# SELECT * FROM public.transaction_external;</pre>  <pre>datamart=# SELECT * FROM public.transaction_external; transid   terminalid   transdate   storeid   phone   itemcnt   salesamt   taxamt   discountamt   couponamt   cashamt   checkamt   cca mt   debitamt   otheramt -----+-----+-----+-----+-----+-----+-----+ -----+-----+-----+-----+-----+-----+-----+ 108409   3   2008-04-10   13   (828)137-4088   2   48.61   0.00   0.18   0.00   48.43   0.00   0 .00   0.00   0.00   108419   4   2008-08-17   13   (815)339-7091   14   47.34   0.00   0.18   0.00   47.16   0.00   0 .00   0.00   0.00   108429   1   2008-05-05   13   (526)665-8938   4   0.05   0.00   1.32   0.00   -1.27   0.00   0 .00   0.00   0.00   108439   2   2008-08-02   13   (504)129-6569   32   18.56   0.00   0.93   0.00   17.63   0.00   0 .00   0.00   0.00   108449   3   2008-03-02   13   (542)592-5376   10   56.38   0.00   1.07   0.00   55.31   0.00   0 .00   0.00   0.00   108459   1   2008-04-24   13   (837)847-6835   35   91.00   0.00   0.56   0.00   90.44   0.00   0 .00   0.00   0.00   108469   7   2008-05-10   13   (840)877-2658   34   56.95   0.00   1.44   0.00   55.51   0.00   0 .00   0.00   0.00   108479   3   2008-08-04   13   (262)775-5387   10   35.20   0.00   1.38   0.00   33.82   0.00   0 .00   0.00   0.00   108489   9   2008-08-07   13   (696)785-2092   33   41.18   0.00   0.32   0.00   40.86   0.00   0 .00   0.00   0.00   108499   4   2008-05-08   13   (633)489-9012   22   88.67   0.00   0.92   0.00   87.75   0.00   0 .00   0.00   0.00   108509   1   2008-07-19   13   (325)410-1149   19</pre>

Step	Action
17.	<p>Insert the rows from the external table into the target Transaction fact table in psql. Join the external transaction table to the customer dimension using the phone number column to get the customer id. If there were hundreds of thousands or millions of rows in the source external files, you should consider loading a specific child table or using gload.</p> <pre data-bbox="306 297 1416 1072"> datamart=# INSERT INTO facts.transaction (transid,  terminalid,  transdate,  storeid,  customerid,  itemcnt,  salesamt,  taxamt,  discountamt,  couponamt,  cashamt,  checkamt,  ccamt,  debitamt,  otheramt) SELECT t.transid, t.terminalid, t.transdate, t.storeid, c.customerid, t.itemcnt, t.salesamt, t.taxamt, t.discountamt, t.couponamt, t.cashamt, t.checkamt, t.ccamt, t.debitamt, t.otheramt FROM public.transaction_external t INNER JOIN dimensions.customer c ON TRIM(c.phone) = TRIM(t.phone); </pre> 

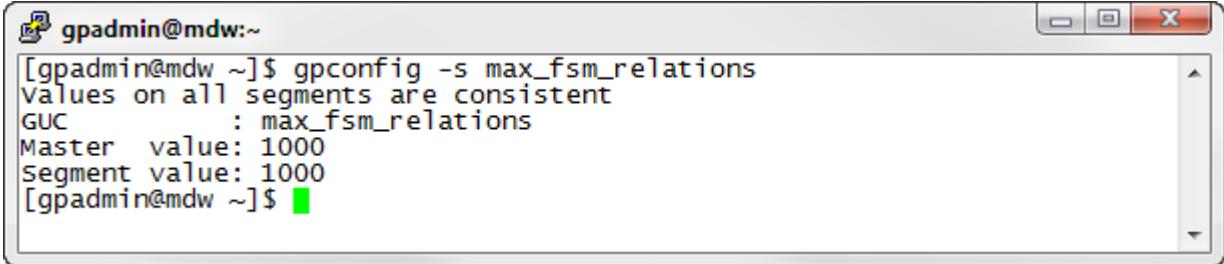
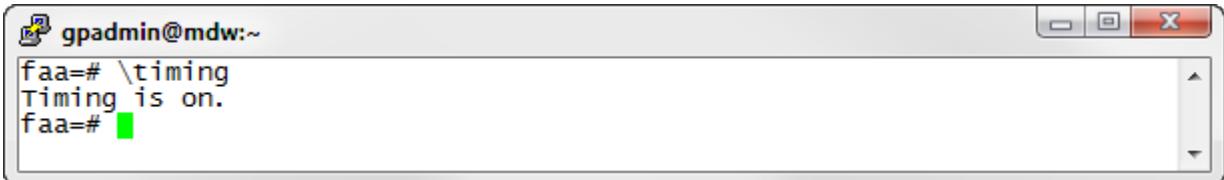
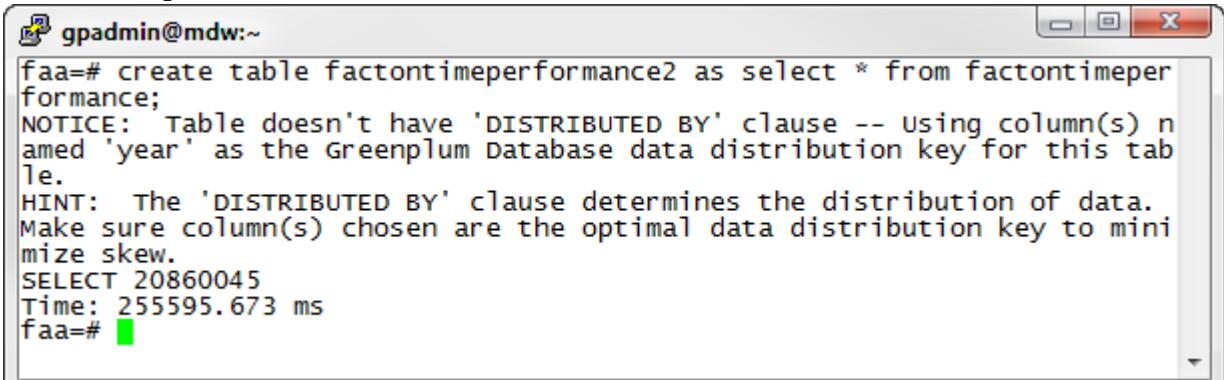
Step	Action
18.	Change back to the home directory. [gpadmin@mdw adv_sql_files]\$ <b>cd</b>
19.	Were any rows discarded? How many? Why? <hr/> <hr/> <hr/>
20.	Why did you need the TRIM function? What are the implications for loading large rows where we need to apply a function to the join conditions? <hr/> <hr/> <hr/>

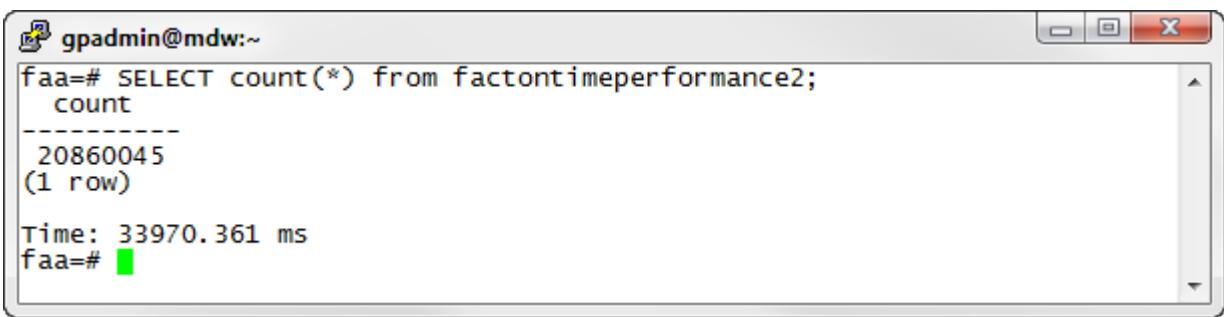
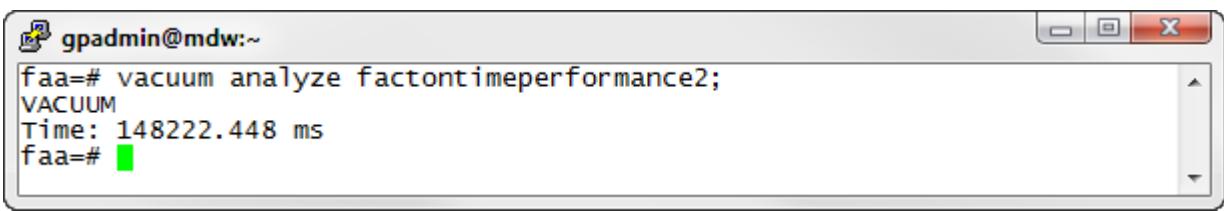
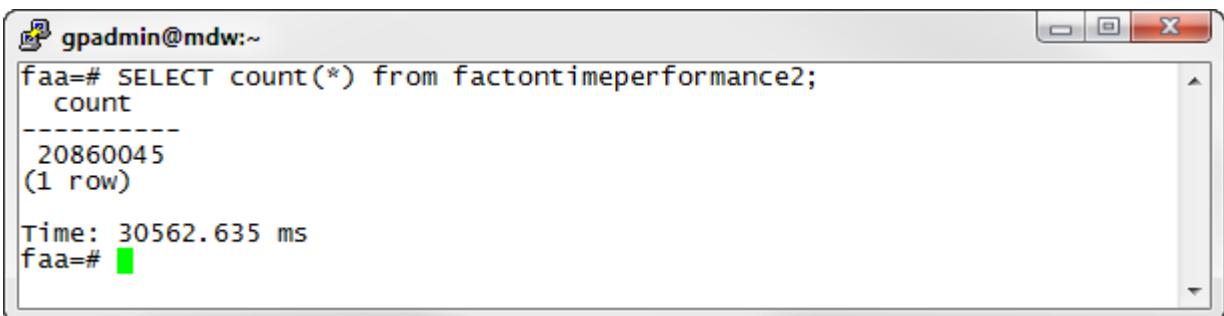
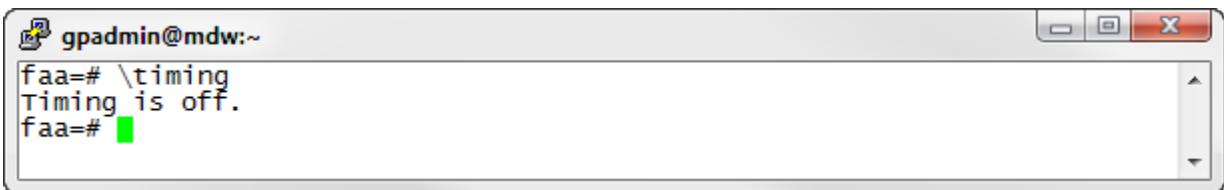
*End of Lab Exercise*

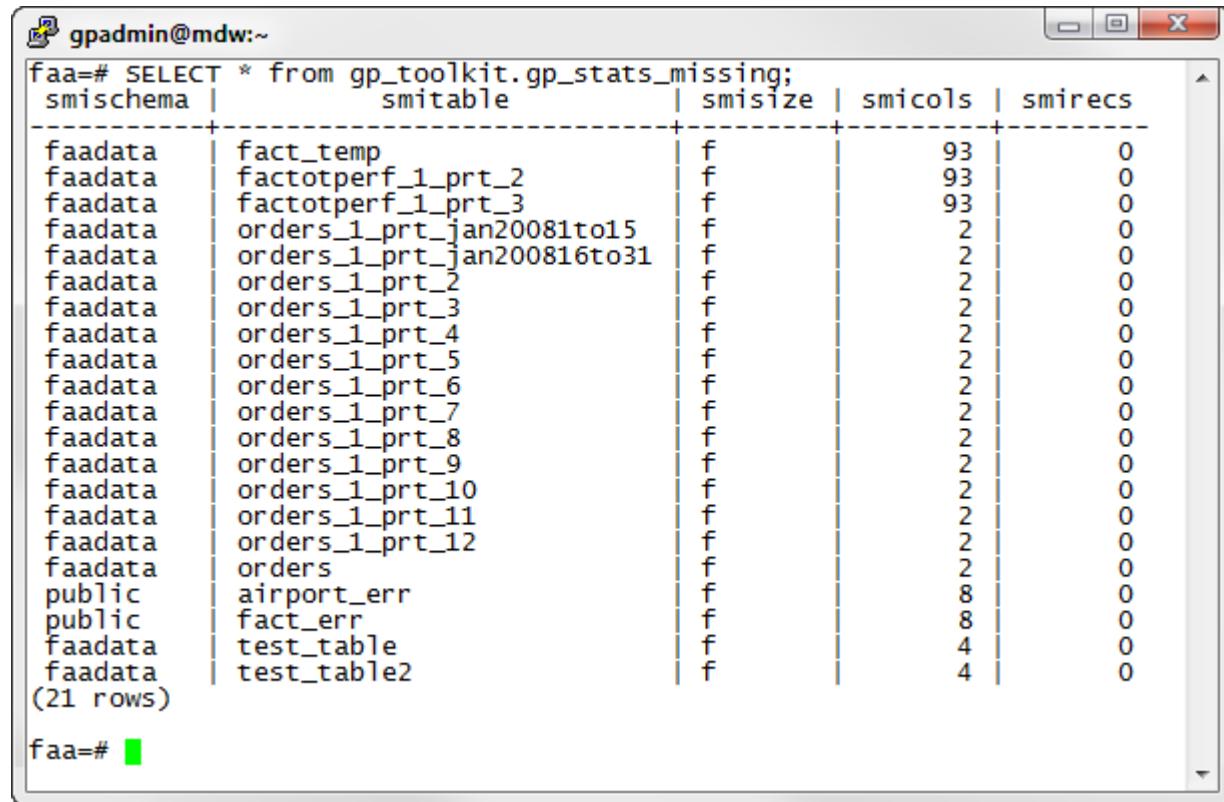
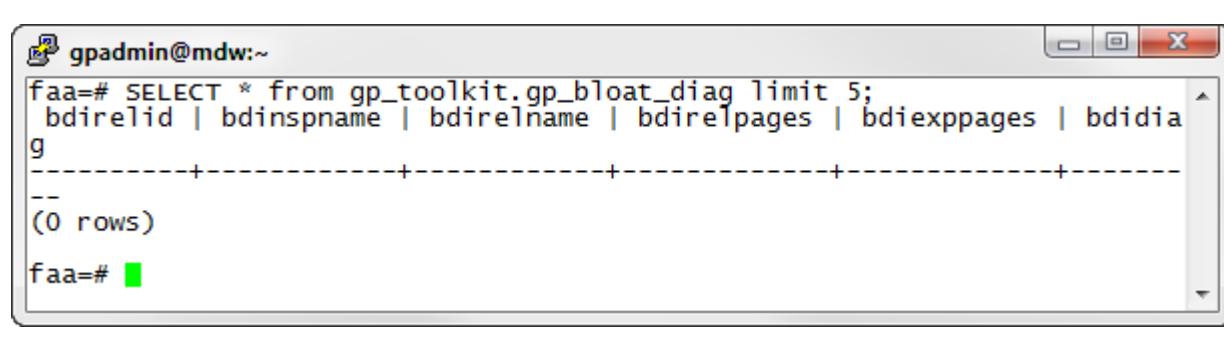
## Lab 18. Database Tuning

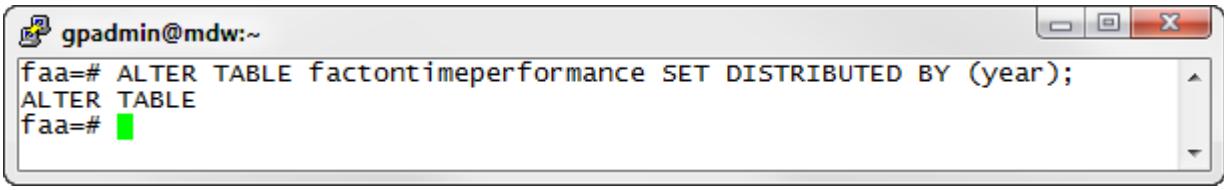
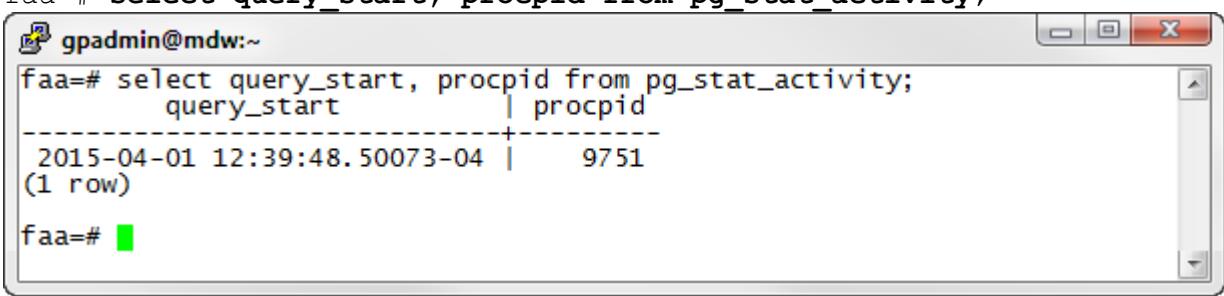
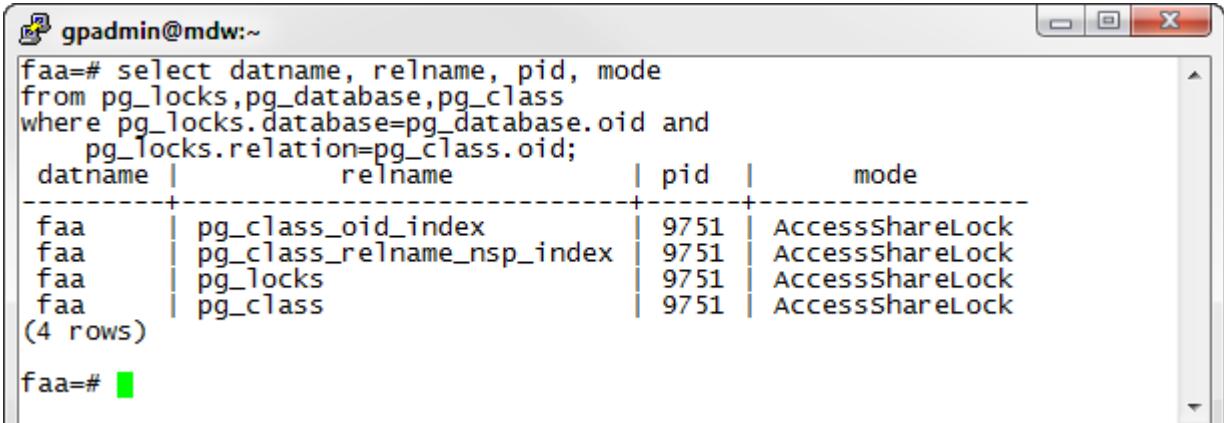
<b>Purpose:</b>	In this lab, you use a combination of tools, commands, and schemas to analyze the overall performance of your environment. You will examine how the system behaves when retrieving data or updating tables.
<b>Tasks:</b>	Students perform the following task: Tune the database and queries.
<b>References:</b>	<p>Module 8 – Performance Analysis and Tuning</p> <ul style="list-style-type: none"><li>• Lesson: Join Tables – Types and Methods</li><li>• Lesson: Database Tuning</li><li>• Lesson: Query Profiling</li></ul>

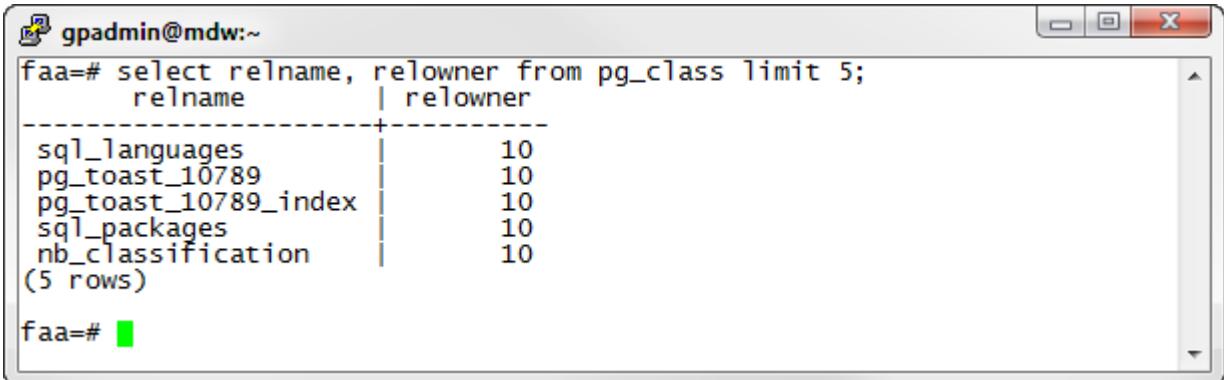
## Lab 18: Task – Tune the Database and Queries

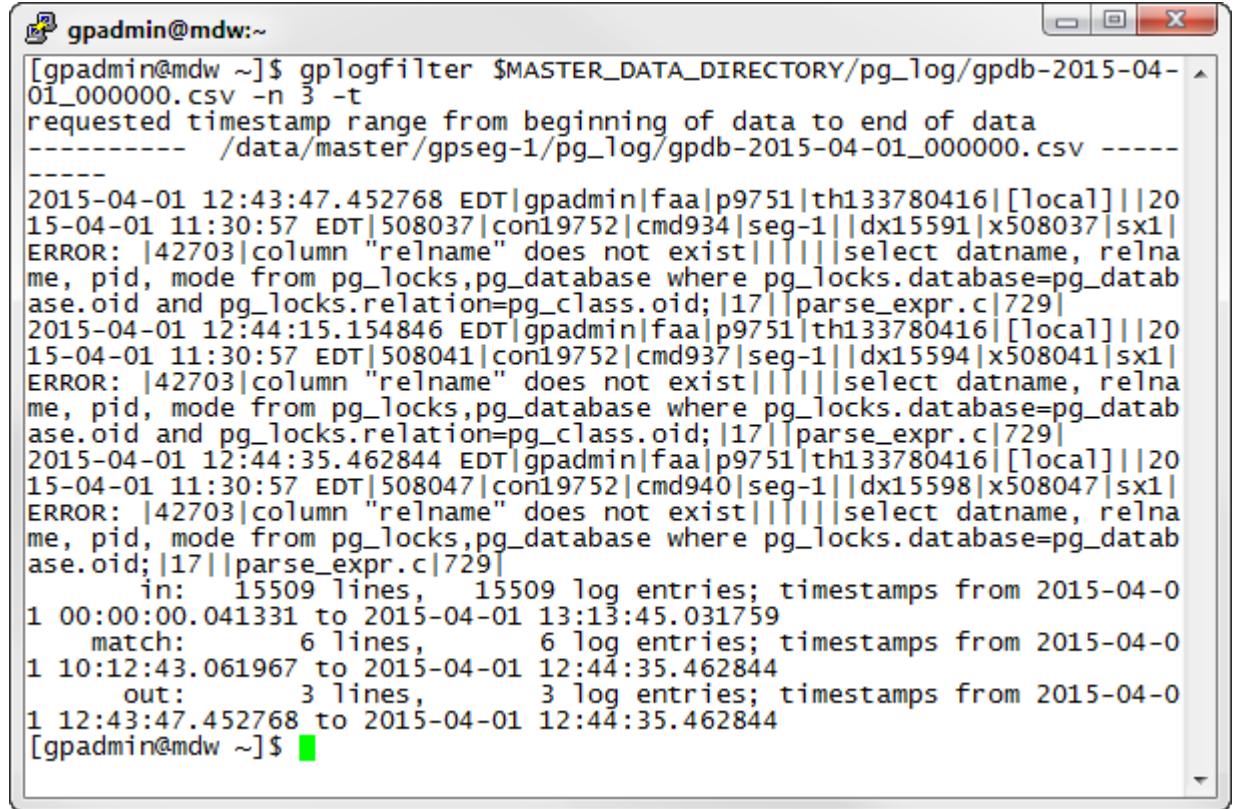
Step	Action
1.	<p>While logged in as gpadmin on your master server, verify your parameters are as shown below.</p> <pre>[gpadmin@mdw ~]\$ gpconfig -s parameter</pre> <p>Verify the following parameters:</p> <pre>max_fsm_relations = 1000 max_fsm_pages = 200000 work_mem = 32MB maintenance_work_mem = 64MB</pre> <p>For example, to search for max_fsm_relations, type the following:</p> <pre>[gpadmin@mdw ~]\$ gpconfig -s max_fsm_relations</pre> 
2.	<p>Load data in a table and check performance. Connect to the faa database as gpadmin.</p> <pre>[gpadmin@mdw ~]\$ psql faa</pre>
3.	<p>The PSQL timing parameter shows how long it takes to complete a command. The command will be used to compare the time it takes to execute SELECT statements on specific tables.</p> <p>Turn timing on:</p> <pre>faa=# \timing</pre> 
4.	<p>Create a new table and load as shown. This step will create a new table and load it with more than 20 million records thus it might take a couple of minutes.</p> <pre>faa=# create table factontimeperformance2 as select * from factontimeperformance;</pre> 

Step	Action
5.	Verify the data loaded: <pre>faa=# SELECT count(*) from factontimeperformance2;</pre>  <pre>gpadmin@mdw:~\$ faa=# SELECT count(*) from factontimeperformance2; count ----- 20860045 (1 row)  Time: 33970.361 ms faa=# </pre>
6.	Analyze and vacuum the table: <pre>faa=# VACUUM ANALYZE factontimeperformance2;</pre>  <pre>gpadmin@mdw:~\$ faa=# VACUUM ANALYZE factontimeperformance2; VACUUM Time: 148222.448 ms faa=# </pre>
7.	Re-check performance: <pre>faa=# SELECT count(*) from factontimeperformance2;</pre>  <pre>gpadmin@mdw:~\$ faa=# SELECT count(*) from factontimeperformance2; count ----- 20860045 (1 row)  Time: 30562.635 ms faa=# </pre> <p>Note that the time it took to perform this query should be less than the first time the query was executed. The VACUUM ANALYZE step you performed provided more up to date statistical information to the query planner so that it could determine the most optimal plan for the query.</p>
8.	Disable PSQL timing. Use the \timing command to toggle the setting to off. <pre>faa=# \timing</pre>  <pre>gpadmin@mdw:~\$ faa=# \timing Timing is off. faa=# </pre>

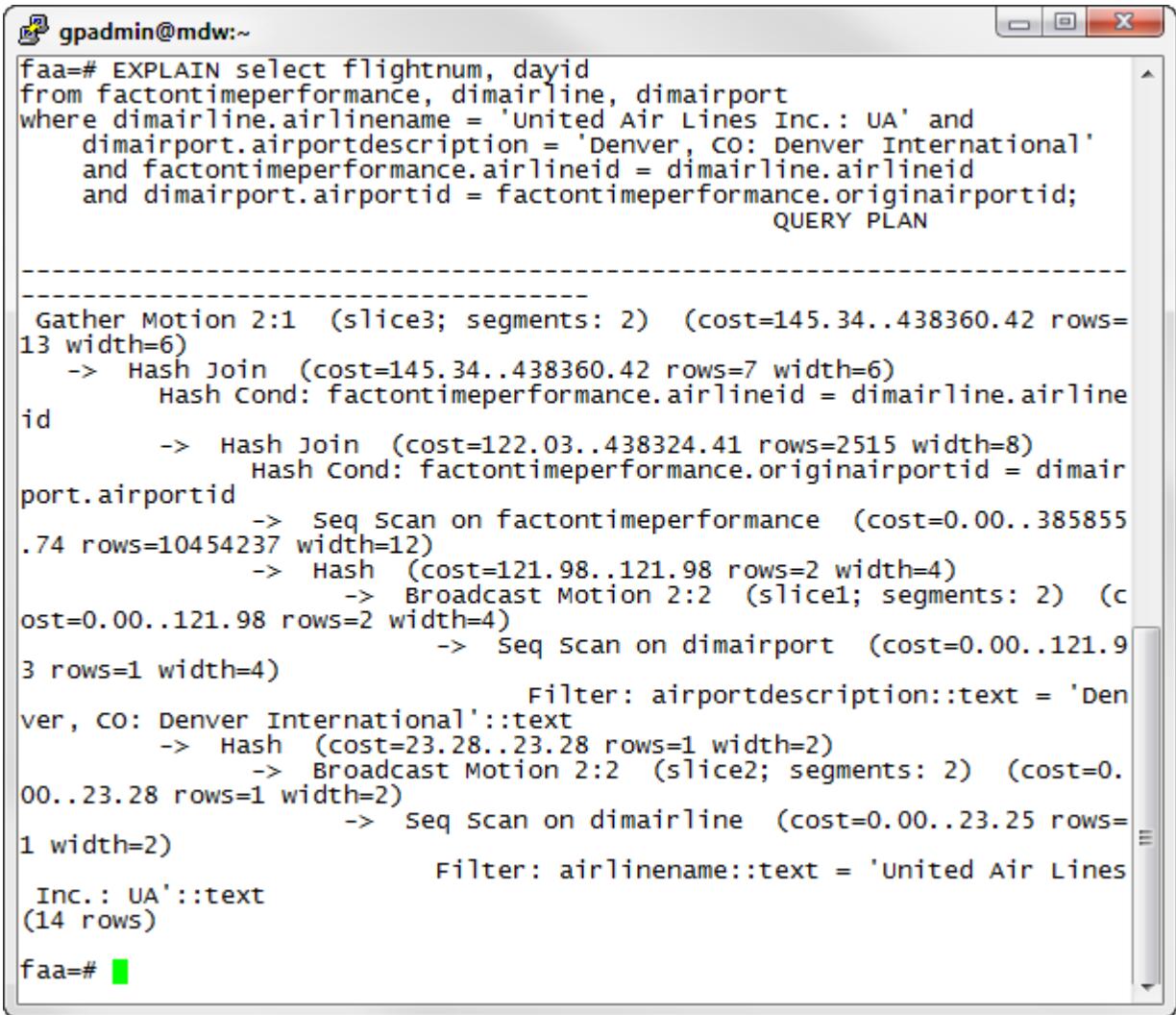
Step	Action																																																																																																														
9.	<p>Greenplum provides an administrative schema called <code>gp_toolkit</code> that you can use to query the system catalogs, log files, and operating environment for system status information. The <code>gp_toolkit</code> schema contains a number of views that you can access using SQL commands. The <code>gp_toolkit</code> schema is accessible to all database users, although some objects may require superuser permissions.</p> <p>Look for tables that do not have statistics:</p> <pre>faa=# SELECT * from gp_toolkit.gp_stats_missing;</pre>  <table border="1"> <thead> <tr> <th>smischema</th> <th>smitable</th> <th>smisize</th> <th>smicols</th> <th>smirecs</th> </tr> </thead> <tbody> <tr><td>faadata</td><td>fact_temp</td><td>f</td><td>93</td><td>0</td></tr> <tr><td>faadata</td><td>factotperf_1_prt_2</td><td>f</td><td>93</td><td>0</td></tr> <tr><td>faadata</td><td>factotperf_1_prt_3</td><td>f</td><td>93</td><td>0</td></tr> <tr><td>faadata</td><td>orders_1_prt_jan20081to15</td><td>f</td><td>2</td><td>0</td></tr> <tr><td>faadata</td><td>orders_1_prt_jan200816to31</td><td>f</td><td>2</td><td>0</td></tr> <tr><td>faadata</td><td>orders_1_prt_2</td><td>f</td><td>2</td><td>0</td></tr> <tr><td>faadata</td><td>orders_1_prt_3</td><td>f</td><td>2</td><td>0</td></tr> <tr><td>faadata</td><td>orders_1_prt_4</td><td>f</td><td>2</td><td>0</td></tr> <tr><td>faadata</td><td>orders_1_prt_5</td><td>f</td><td>2</td><td>0</td></tr> <tr><td>faadata</td><td>orders_1_prt_6</td><td>f</td><td>2</td><td>0</td></tr> <tr><td>faadata</td><td>orders_1_prt_7</td><td>f</td><td>2</td><td>0</td></tr> <tr><td>faadata</td><td>orders_1_prt_8</td><td>f</td><td>2</td><td>0</td></tr> <tr><td>faadata</td><td>orders_1_prt_9</td><td>f</td><td>2</td><td>0</td></tr> <tr><td>faadata</td><td>orders_1_prt_10</td><td>f</td><td>2</td><td>0</td></tr> <tr><td>faadata</td><td>orders_1_prt_11</td><td>f</td><td>2</td><td>0</td></tr> <tr><td>faadata</td><td>orders_1_prt_12</td><td>f</td><td>2</td><td>0</td></tr> <tr><td>faadata</td><td>orders</td><td>f</td><td>2</td><td>0</td></tr> <tr><td>public</td><td>airport_err</td><td>f</td><td>8</td><td>0</td></tr> <tr><td>public</td><td>fact_err</td><td>f</td><td>8</td><td>0</td></tr> <tr><td>faadata</td><td>test_table</td><td>f</td><td>4</td><td>0</td></tr> <tr><td>faadata</td><td>test_table2</td><td>f</td><td>4</td><td>0</td></tr> </tbody> </table> <pre>(21 rows)</pre> <pre>faa=# █</pre>	smischema	smitable	smisize	smicols	smirecs	faadata	fact_temp	f	93	0	faadata	factotperf_1_prt_2	f	93	0	faadata	factotperf_1_prt_3	f	93	0	faadata	orders_1_prt_jan20081to15	f	2	0	faadata	orders_1_prt_jan200816to31	f	2	0	faadata	orders_1_prt_2	f	2	0	faadata	orders_1_prt_3	f	2	0	faadata	orders_1_prt_4	f	2	0	faadata	orders_1_prt_5	f	2	0	faadata	orders_1_prt_6	f	2	0	faadata	orders_1_prt_7	f	2	0	faadata	orders_1_prt_8	f	2	0	faadata	orders_1_prt_9	f	2	0	faadata	orders_1_prt_10	f	2	0	faadata	orders_1_prt_11	f	2	0	faadata	orders_1_prt_12	f	2	0	faadata	orders	f	2	0	public	airport_err	f	8	0	public	fact_err	f	8	0	faadata	test_table	f	4	0	faadata	test_table2	f	4	0
smischema	smitable	smisize	smicols	smirecs																																																																																																											
faadata	fact_temp	f	93	0																																																																																																											
faadata	factotperf_1_prt_2	f	93	0																																																																																																											
faadata	factotperf_1_prt_3	f	93	0																																																																																																											
faadata	orders_1_prt_jan20081to15	f	2	0																																																																																																											
faadata	orders_1_prt_jan200816to31	f	2	0																																																																																																											
faadata	orders_1_prt_2	f	2	0																																																																																																											
faadata	orders_1_prt_3	f	2	0																																																																																																											
faadata	orders_1_prt_4	f	2	0																																																																																																											
faadata	orders_1_prt_5	f	2	0																																																																																																											
faadata	orders_1_prt_6	f	2	0																																																																																																											
faadata	orders_1_prt_7	f	2	0																																																																																																											
faadata	orders_1_prt_8	f	2	0																																																																																																											
faadata	orders_1_prt_9	f	2	0																																																																																																											
faadata	orders_1_prt_10	f	2	0																																																																																																											
faadata	orders_1_prt_11	f	2	0																																																																																																											
faadata	orders_1_prt_12	f	2	0																																																																																																											
faadata	orders	f	2	0																																																																																																											
public	airport_err	f	8	0																																																																																																											
public	fact_err	f	8	0																																																																																																											
faadata	test_table	f	4	0																																																																																																											
faadata	test_table2	f	4	0																																																																																																											
10.	<p>Look for table bloat from lack of vacuuming:</p> <pre>faa=# SELECT * from gp_toolkit.gp_bloat_diag limit 5;</pre>  <table border="1"> <thead> <tr> <th>bdirelid</th> <th>bdinspname</th> <th>bdirelname</th> <th>bdirelpages</th> <th>bdiexppages</th> <th>bdidia</th> </tr> </thead> <tbody> <tr><td></td><td></td><td></td><td></td><td></td><td></td></tr> </tbody> </table> <pre>-- (0 rows)</pre> <pre>faa=# █</pre>	bdirelid	bdinspname	bdirelname	bdirelpages	bdiexppages	bdidia																																																																																																								
bdirelid	bdinspname	bdirelname	bdirelpages	bdiexppages	bdidia																																																																																																										

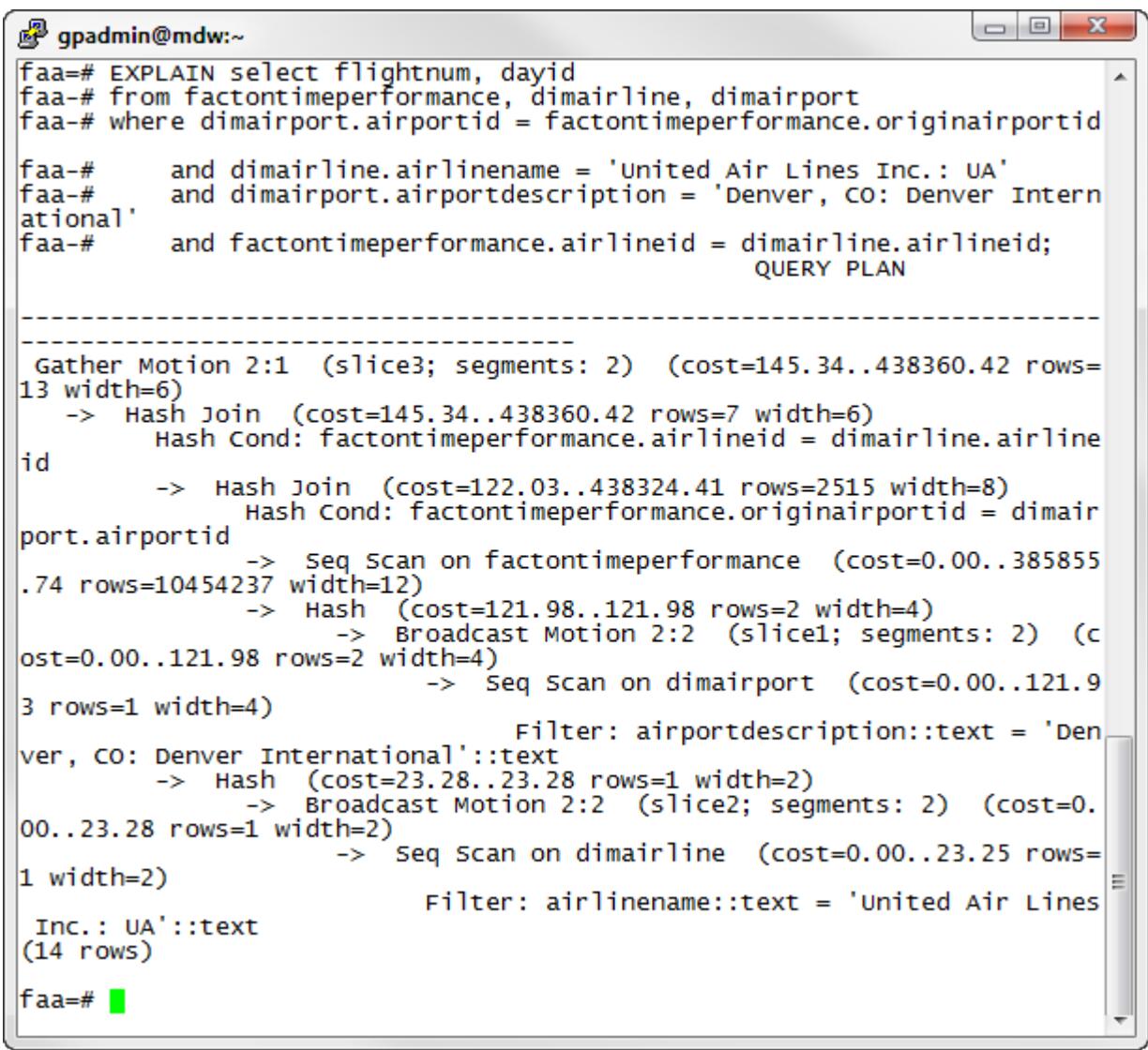
Step	Action
11.	<p>If there is performance issue based on how a table is distributed, the distribution can be changed with an ALTER TABLE command:</p> <pre>faa=# ALTER TABLE factontimeperformance SET DISTRIBUTED BY (year);</pre>  <p><b>Note:</b> This command may take several minutes to process.</p>
12.	<p>Examine other views such as the pg_stat_activity, pg_locks and pg_class views.</p> <p>pg_stat_activity has information about current running queries.</p> <p>Look for the start time of all current queries in the database:</p> <pre>faa=# select query_start, procpid from pg_stat_activity;</pre> 
13.	<p>pg_locks has data about locks in the database. This syntax shows the locks that are on the tables being accessed.</p> <pre>faa=# select datname, relname, pid, mode from pg_locks,pg_database,pg_class where pg_locks.database=pg_database.oid and pg_locks.relation=pg_class.oid;</pre>  <p>In this case, the tables being accessed are the ones referenced in the query just executed. There are no other accesses on the database at this time. Look for locks in the database:</p>

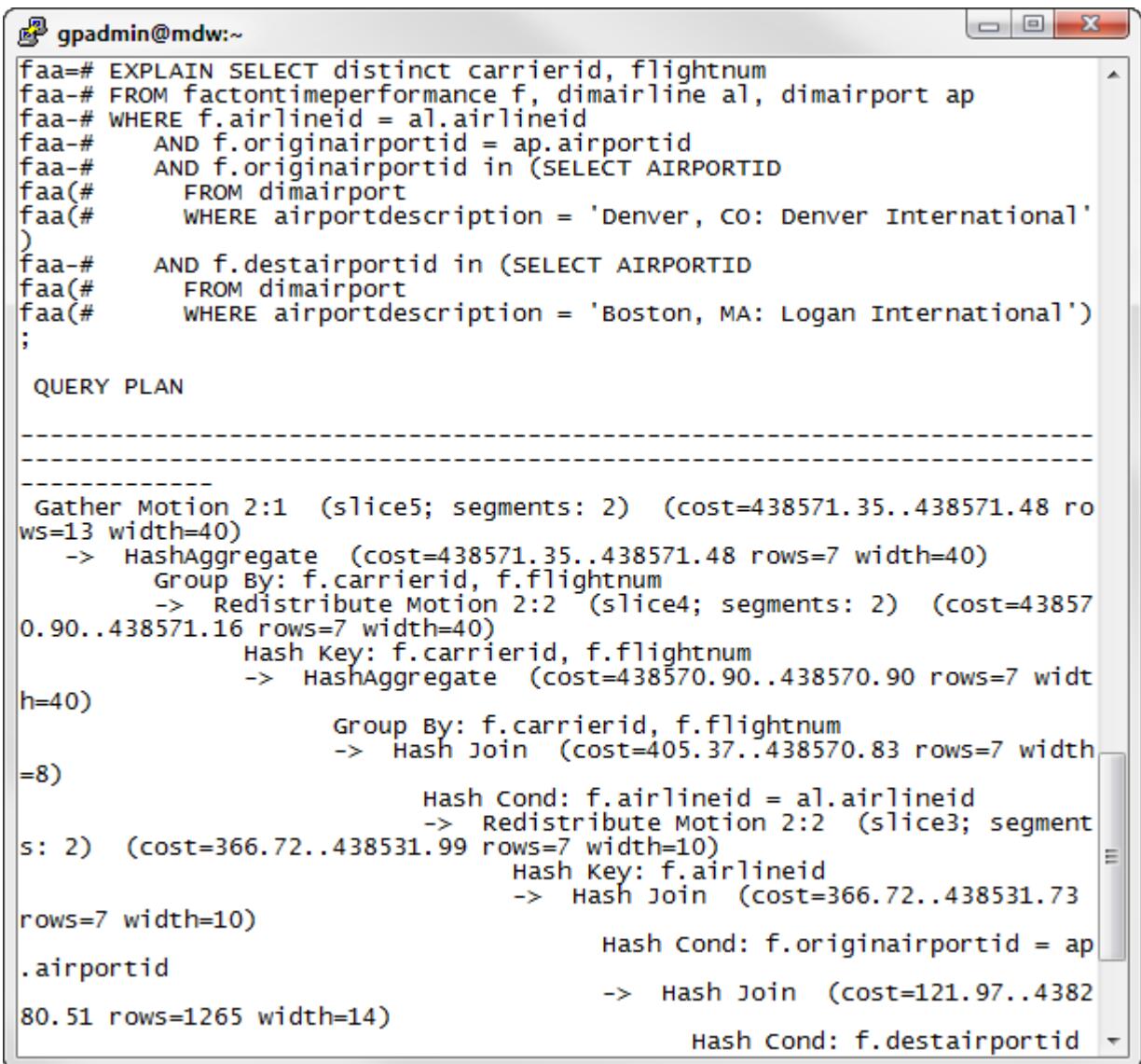
Step	Action
14.	<p>Look for owner and object type in the database:</p> <pre>faa=# select relname, relowner from pg_class limit 5;</pre>  <pre>gpadmin@mdw:~ faa=# select relname, relowner from pg_class limit 5;       relname         relowner -----+-----  sql_languages           10  pg_toast_10789          10  pg_toast_10789_index        10  sql_packages            10  nb_classification        10 (5 rows)  faa=# </pre>
15.	Exit your PSQL session.

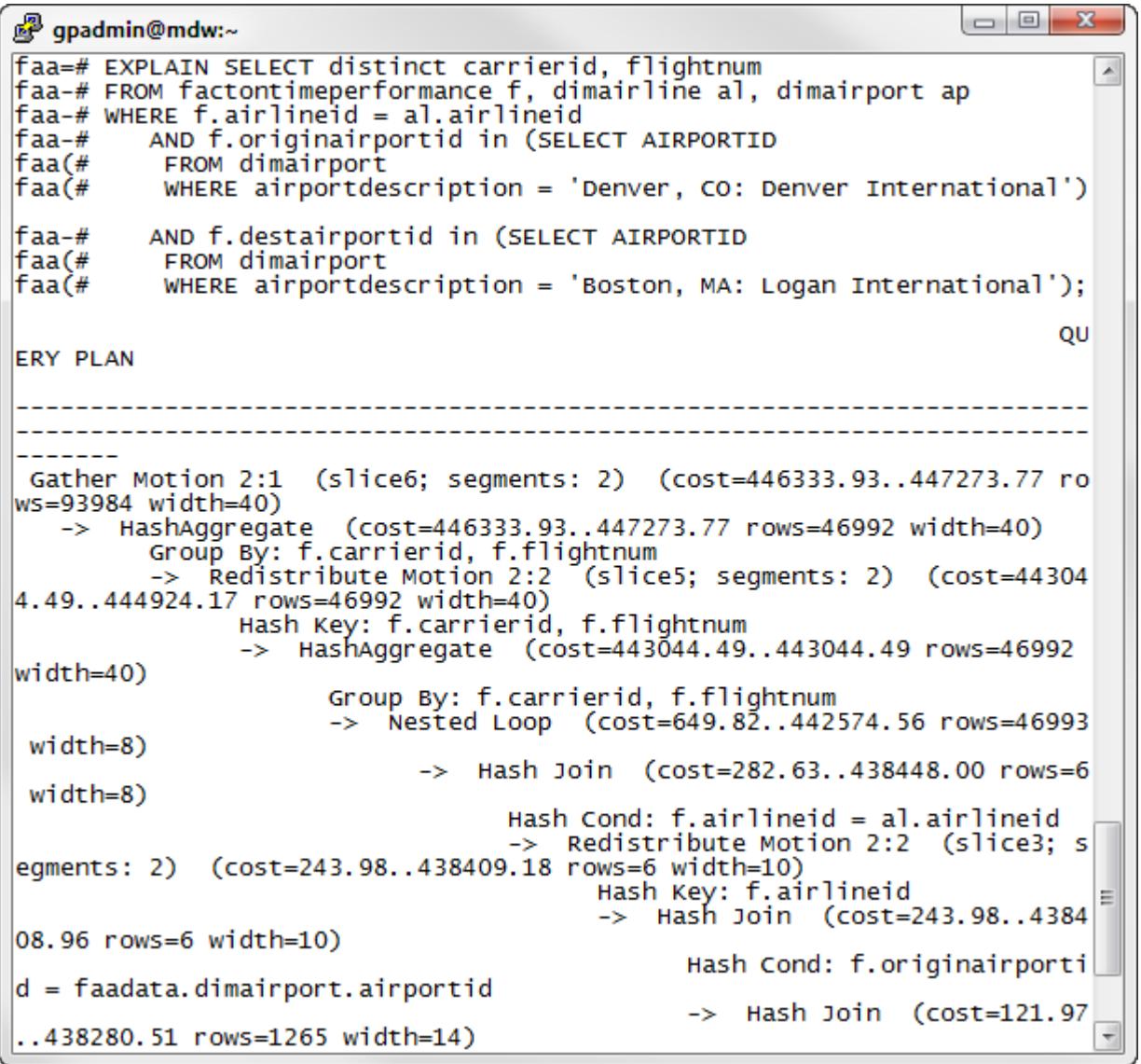
Step	Action
16.	<p>Database logs are another place to look for errors or tuning issues. In UNIX as the <code>gpadmin</code> user</p> <p>The following are the database log files that reside on the system:</p> <ul style="list-style-type: none"> <li>• <code>pg_xlog</code> contains Greenplum Write Ahead Logs (WAL, Greenplum implementation of transaction logging) files (normally 16MB in size, each).</li> <li>• <code>pg_clog</code> contains the commit log files which contain transaction commit status of a transaction. One main purpose is to perform a database recovery in case of a crash by replaying these logs.</li> <li>• <code>pg_log</code> contains the database instance logs by date. This is where you will find instance errors like missing files or filled up disks. It also holds the connection information.</li> </ul> <p>Identify the logs that exist and use <code>gplogfilter</code> to access the content. The <code>gplogfilter</code> syntax here is looking at one file instead of all log files (the default behavior) and looks for the last three error messages generated in the file you specify.</p> <pre>[gpadmin@mdw ~]\$ ls \$MASTER_DATA_DIRECTORY/pg_log gpdb-2015-03-31_160631.csv  gpdb-2015-04-01_000000.csv  startup.log [gpadmin@mdw ~]\$ gplogfilter \$MASTER_DATA_DIRECTORY/pg_log/gpdb-2015-04-01_000000.csv -n 3 -t</pre>  <p><b>Note:</b> Replace the file name shown with a file that exists in your <code>pg_log</code> environment.</p>

Step	Action
17.	<p>Use UNIX operating system commands to see performance from the OS point of view:</p> <ul style="list-style-type: none"> <li>• <code>top</code> is used to get information on CPU and memory performance.</li> <li>• <code>df</code> provides information on the capacity of the file systems.</li> <li>• <code>ps -ef   grep postgres</code> lets you look for postgres processes currently running.</li> <li>• <code>vmstat</code> is a utility that provides information on virtual memory usage on the system.</li> <li>• <code>netstat</code> is a network status utility.</li> <li>• <code>gpstate</code> is a Greenplum utility that provides information on the state of the cluster.</li> </ul>

Step	Action
18.	<p>Getting an explain plan out of a query in Greenplum is easy. Add the word EXPLAIN to the front of your query to see how the query optimizer will execute your query. You can then change the query to see if you can affect the cost to make it lower.</p> <p>Open another PSQL session and run the two queries below. Verify the differences in the EXPLAIN plan:</p> <pre>faa=# EXPLAIN select flightnum, dayid from factontimeperformance, dimairline, dimairport where dimairline.airlinename = 'United Air Lines Inc.: UA' and       dimairport.airportdescription = 'Denver, CO: Denver International'       and factontimeperformance.airlineid = dimairline.airlineid       and dimairport.airportid = factontimeperformance.originairportid;</pre> 

Step	Action
19.	<pre>faa=# EXPLAIN select flightnum, dayid   from factontimeperformance, dimairline, dimairport   where dimairport.airportid = factontimeperformance.originairportid     and dimairline.airlinename = 'United Air Lines Inc.: UA'     and dimairport.airportdescription = 'Denver, CO: Denver International'     and factontimeperformance.airlineid = dimairline.airlineid;</pre>  <p><b>Note:</b> Both of the queries result in the same execution plan. Based on the statistics Greenplum will optimize the query the same way regardless of the ordering of the WHERE clause.</p>

Step	Action
20.	<p>Generate execution plans for the following two queries and compare the execution plans:</p> <pre> faa=# EXPLAIN SELECT distinct carrierid, flightnum   FROM factontimeperformance f, dimairline al, dimairport ap  WHERE f.airlineid = al.airlineid       AND f.originairportid = ap.airportid       AND f.originairportid in (SELECT AIRPORTID         FROM dimairport         WHERE airportdescription = 'Denver, CO: Denver International')   AND f.destairportid in (SELECT AIRPORTID     FROM dimairport     WHERE airportdescription = 'Boston, MA: Logan International'); </pre> 

Step	Action
21.	<pre> faa=# EXPLAIN SELECT distinct carrierid, flightnum FROM factontimeperformance f, dimairline al, dimairport ap WHERE f.airlineid = al.airlineid       AND f.originairportid in (SELECT AIRPORTID                                  FROM dimairport                                  WHERE airportdescription= 'Denver, CO: Denver International')       AND f.destairportid in (SELECT AIRPORTID                                  FROM dimairport                                  WHERE airportdescription = 'Boston, MA: Logan International'); </pre>  <p><b>Note:</b> While both of the queries display the same results, they have very different execution plans. The second query omits one of the join relations and creates a partial Cartesian product which has much slower performance. This can be identified in the execution plan from the additional Materialize step that is required.</p>
22.	Exit your PSQL session.

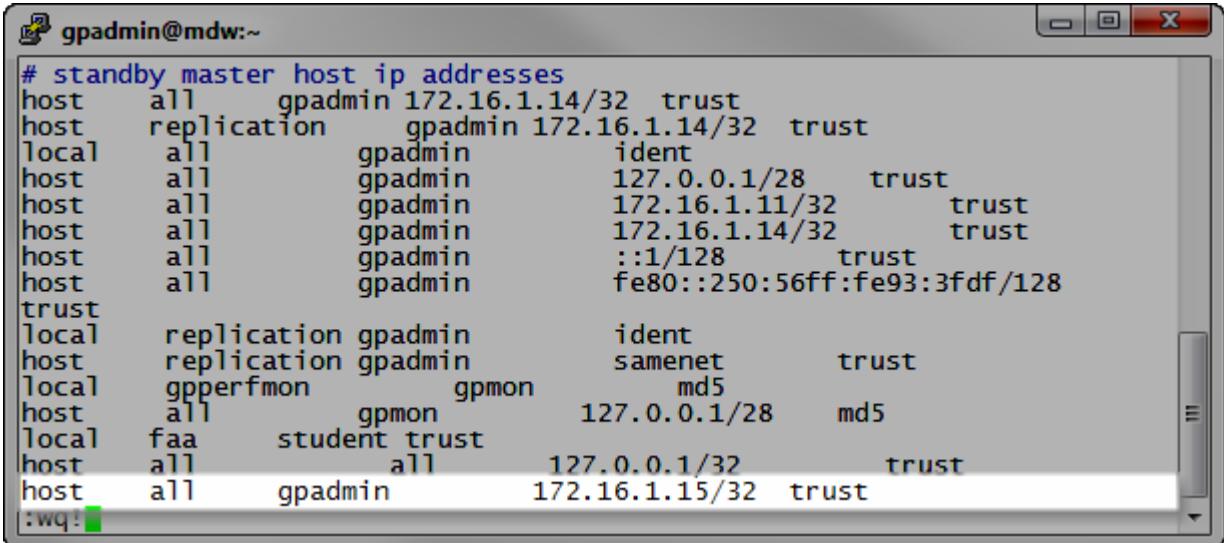
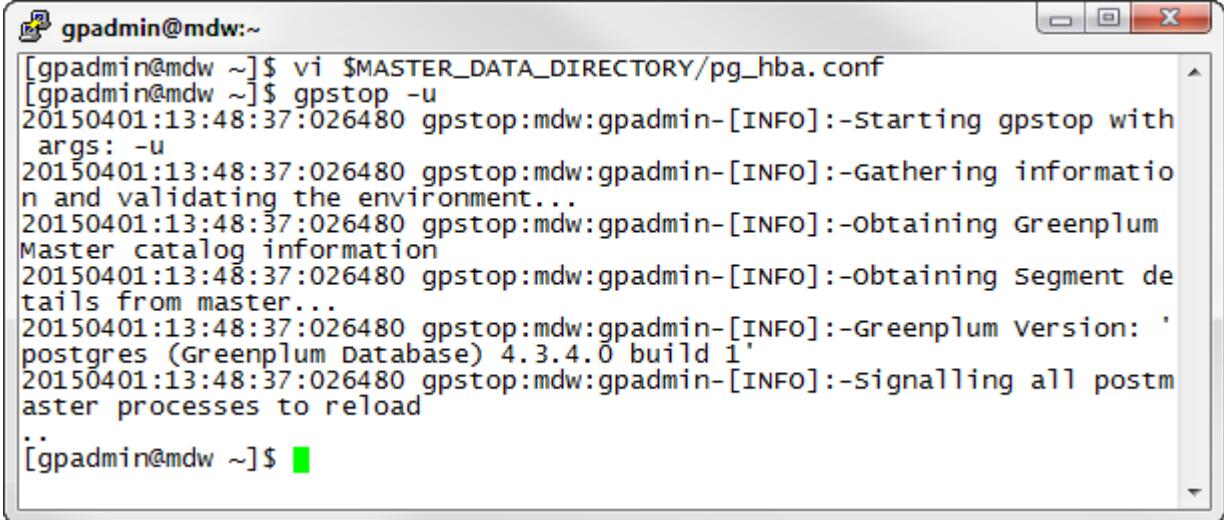
Step	Action
23.	<p><b>Summary</b></p> <p>The EXPLAIN command allows you to view the query plan for a query. EXPLAIN ANALYZE will actually run the query and show you the plan that was executed but does not return results.</p> <p>Query plans are read from bottom to top and show a tree plan of nodes. A node represents a database operation, such as a table scan, a join, or a sort. Greenplum Database plans will also show motion nodes, which are operations that move tuples between the segment instances or from the segment instances to the master. Examining query plans helps uncover areas where performance can be improved.</p> <p>In addition to query plans, using UNIX-based commands and tools, Greenplum clients can provide detailed information on how the system is behaving. The hardware and network has a very strong impact on performance, so those should always remain a part of your performance tuning goals.</p>

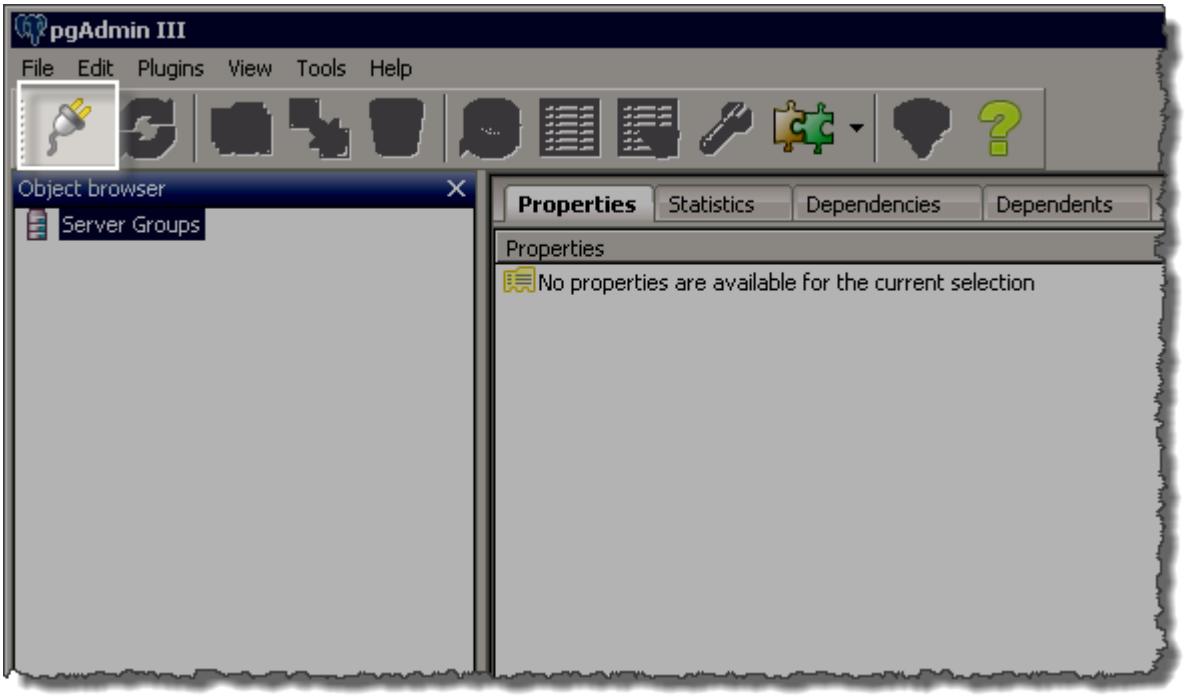
*End of Lab Exercise*

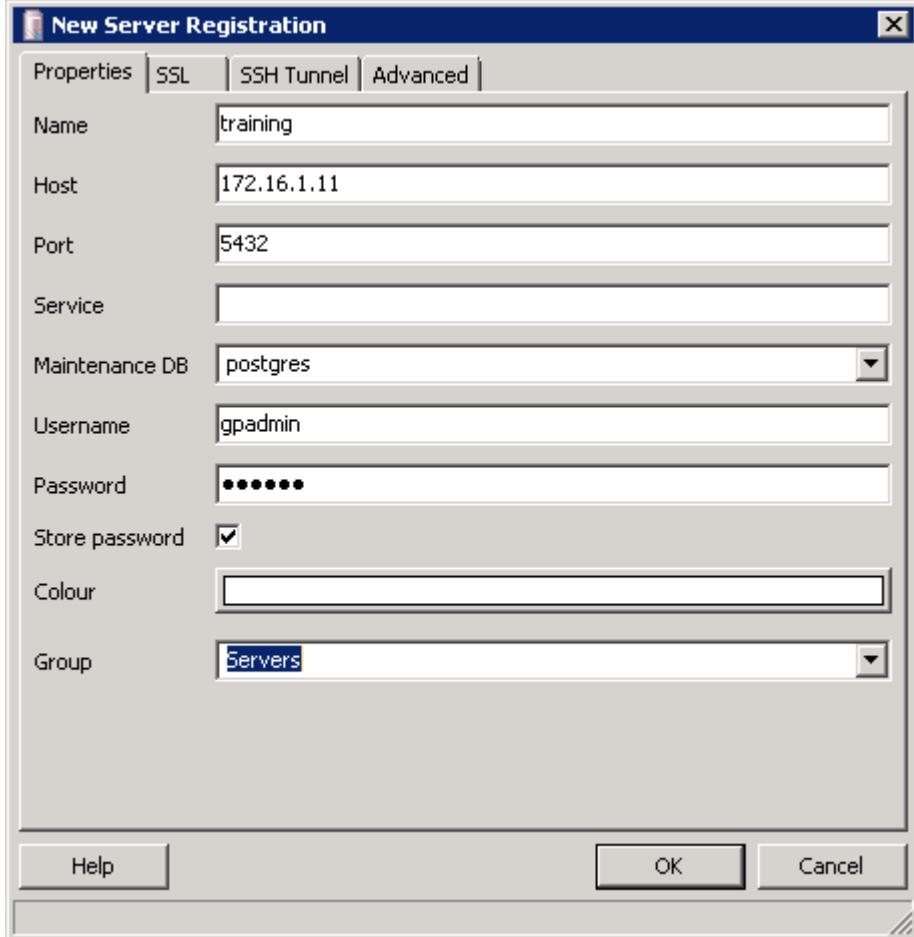
## Lab 19. Explain the EXPLAIN Plan – Analyzing Queries

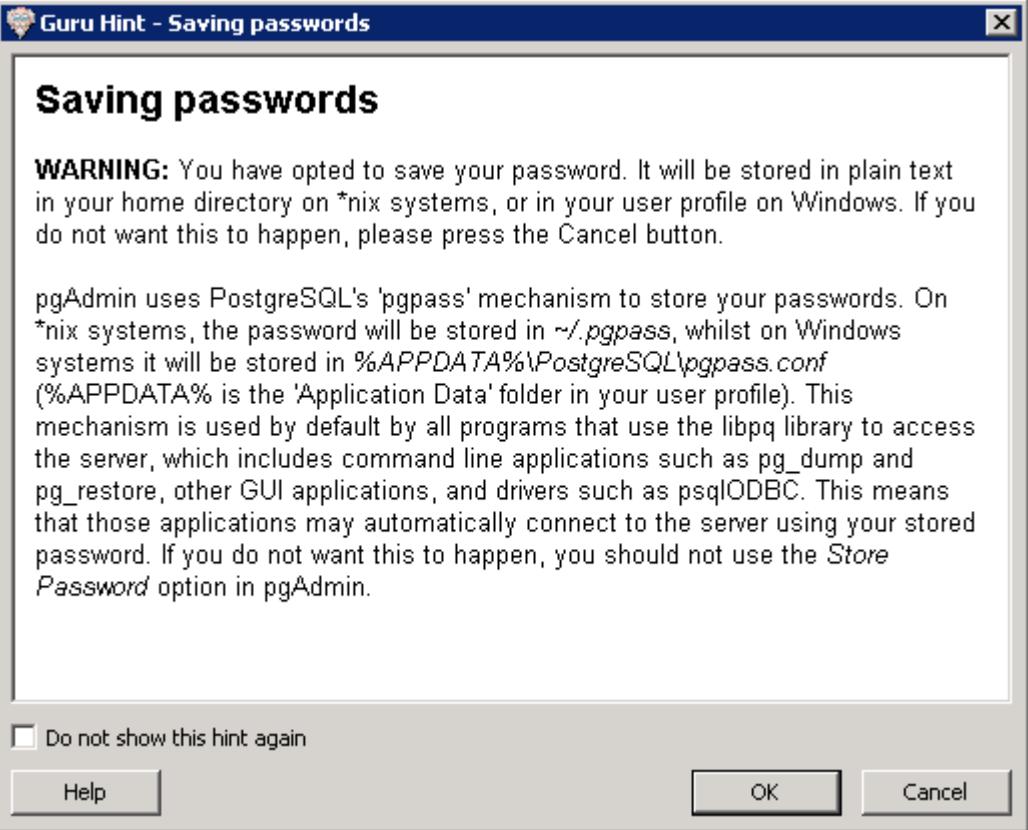
<b>Purpose:</b>	In this lab, you will be given a set of queries to analyze. The data that you have previously inserted deliberately does not have statistics collected for it. Do not use the EXPLAIN or ANALYZE commands until you are instructed to do so. You will want to compare the before and after effects of statistics.
<b>Tasks:</b>	Students perform the following task: Analyze queries.
<b>References:</b>	Module 8 – Performance Analysis and Tuning <ul style="list-style-type: none"><li>• Explain the Explain Plan – Analyzing Queries</li></ul>

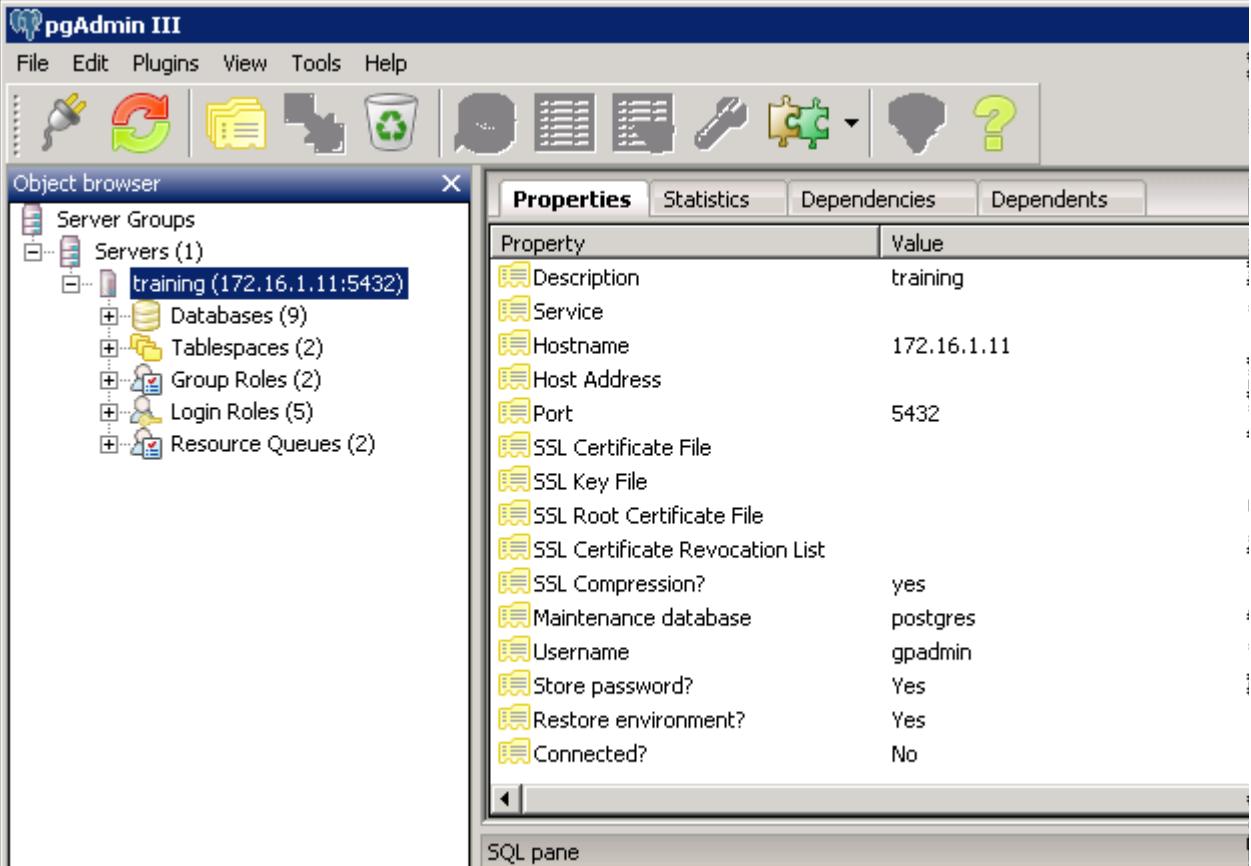
## Lab 19: Task 1– Configure Databases in pgAdmin III

Step	Action
	<p>pgAdmin III offers a graphical interface to interact with the Greenplum Database. The tool can be used to view database objects, execute queries, and perform some management tasks in the database. You will configure pgAdmin III to make it easier to execute and the labs in this exercise. Feel free to use it for other exercises. You must first configure pgAdmin III by completing two main tasks:</p> <ul style="list-style-type: none"> <li>• Updating the <code>pg_hba.conf</code> file to provide access to connections from your Windows environment where you will be running pgAdmin III</li> <li>• Creating a connection to the database within pgAdmin III</li> </ul>
1.	<p>From your terminal connection the master server, <code>mdw</code>, modify the <code>\$MASTER_DATA_DIRECTORY/pg_hba.conf</code> file and add the following entry to the bottom of the file:</p> <pre>host all gpadmin 172.16.1.15/32 trust</pre>  <p>Save and exit the file.</p>
2.	<p>Re-read the configuration file with the <code>gpstop</code> command.</p> 
3.	<p>Double-click the pgAdmin III icon on your desktop to start the application.</p>

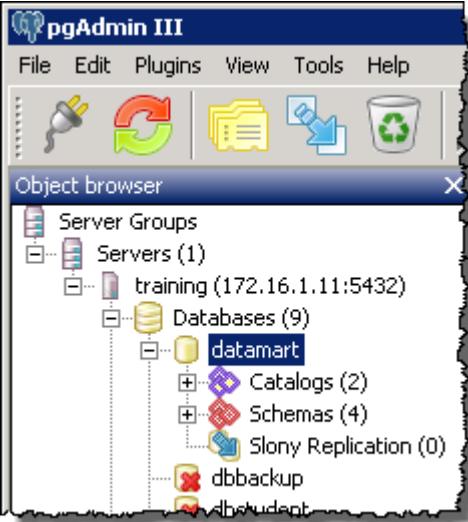
Step	Action
4.	<p>Click the icon to add a connection to the environment.</p>  <p>The screenshot shows the pgAdmin III interface. The title bar reads "pgAdmin III". The menu bar includes File, Edit, Plugins, View, Tools, and Help. The toolbar contains various icons: a power plug, a gear, a folder, a database, a table, a chart, a wrench, a puzzle piece, a brain, and a question mark. The left pane is titled "Object browser" and shows a "Server Groups" node. The right pane has tabs for Properties, Statistics, Dependencies, and Dependents. The "Properties" tab is selected, displaying the message "No properties are available for the current selection".</p>

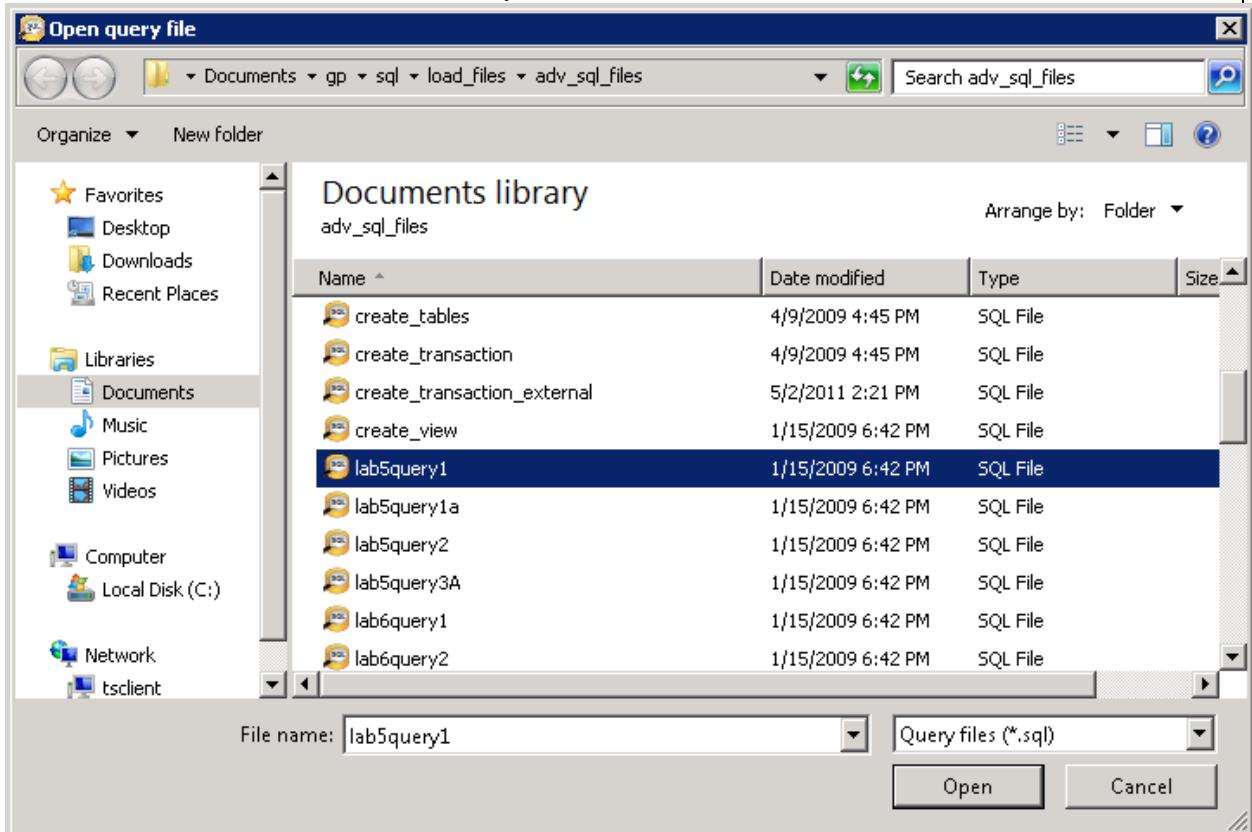
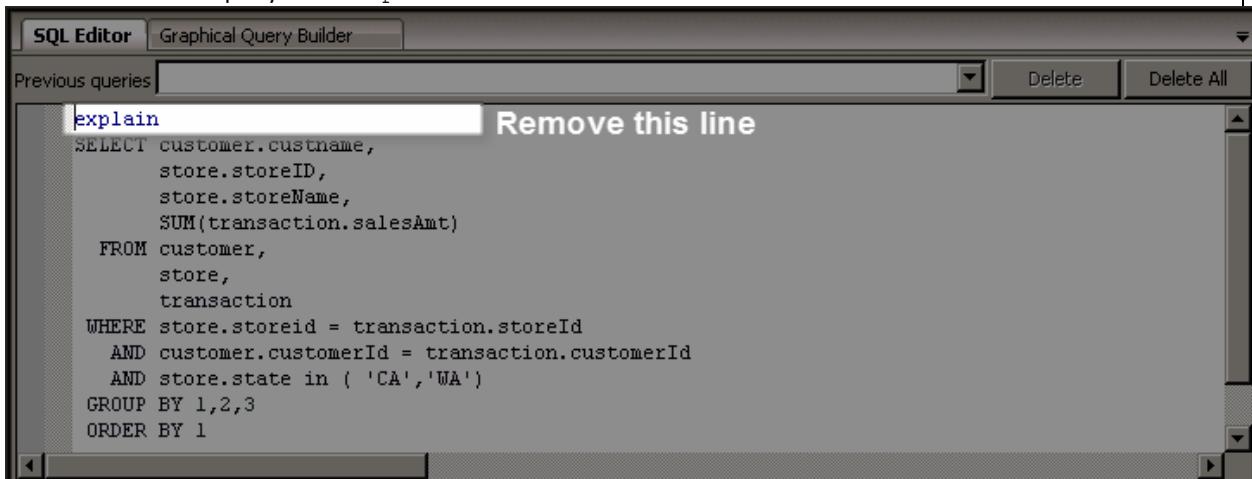
Step	Action
5.	<p>Provide the following information to register your server connection and click <b>OK</b> to complete the process:</p> <ul style="list-style-type: none"> <li>• Name: training</li> <li>• Host: 172.16.1.11</li> <li>• Port: 5432</li> <li>• Password: gpadmin</li> </ul> 

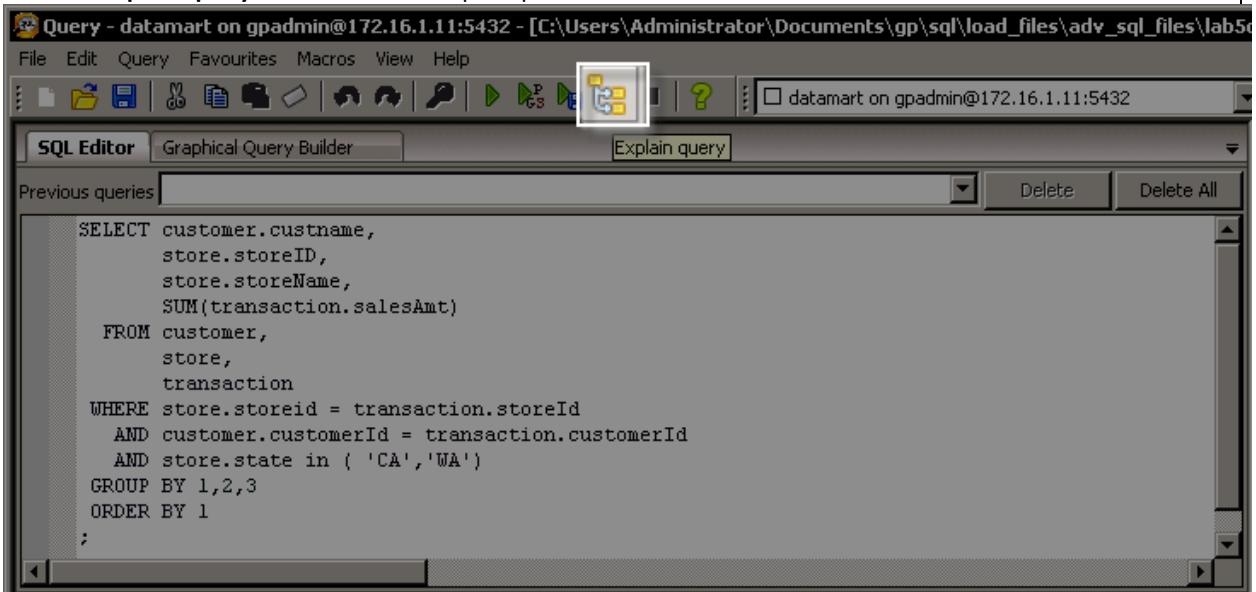
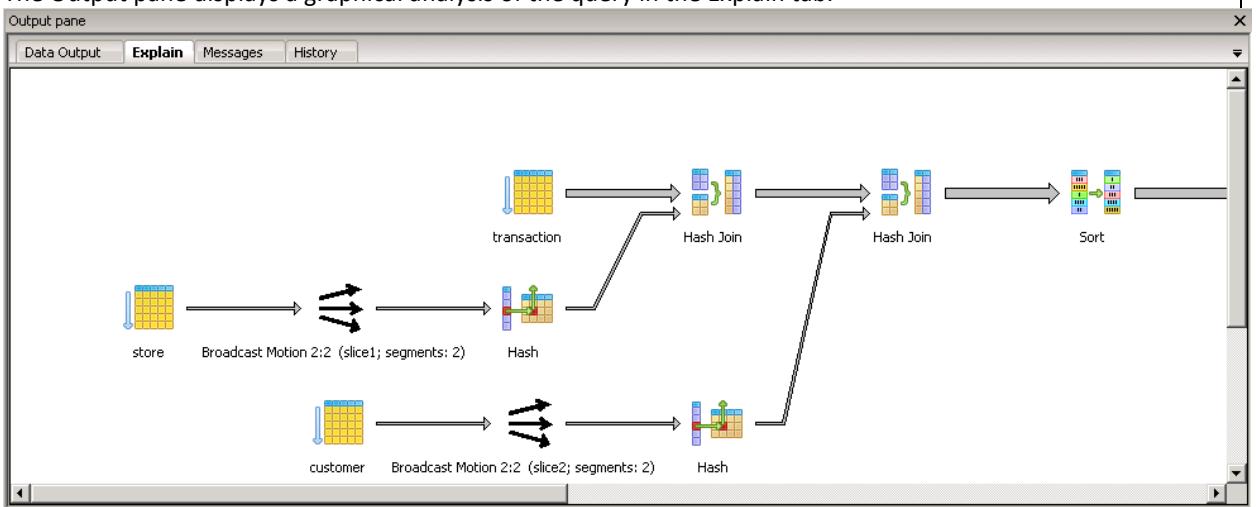
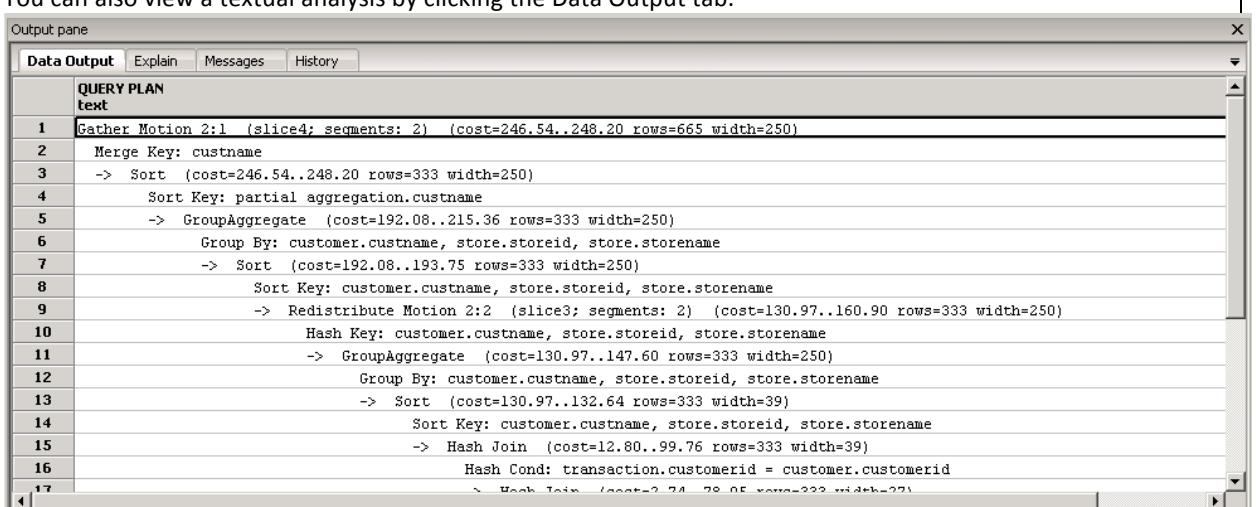
Step	Action
6.	<p>A warning will be displayed emphasizing the issues with storing your password. In your production environment, you would not do this. However, in this lab environment, you will proceed as it simplifies and eases your access. Click <b>OK</b> to proceed.</p>  <p><b>Saving passwords</b></p> <p><b>WARNING:</b> You have opted to save your password. It will be stored in plain text in your home directory on *nix systems, or in your user profile on Windows. If you do not want this to happen, please press the Cancel button.</p> <p>pgAdmin uses PostgreSQL's 'pgpass' mechanism to store your passwords. On *nix systems, the password will be stored in <code>~/.pgpass</code>, whilst on Windows systems it will be stored in <code>%APPDATA%\PostgreSQL\pgpass.conf</code> (<code>%APPDATA%</code> is the 'Application Data' folder in your user profile). This mechanism is used by default by all programs that use the libpq library to access the server, which includes command line applications such as <code>pg_dump</code> and <code>pg_restore</code>, other GUI applications, and drivers such as psqlODBC. This means that those applications may automatically connect to the server using your stored password. If you do not want this to happen, you should not use the <i>Store Password</i> option in pgAdmin.</p> <p><input type="checkbox"/> Do not show this hint again</p> <p>Help      OK      Cancel</p>

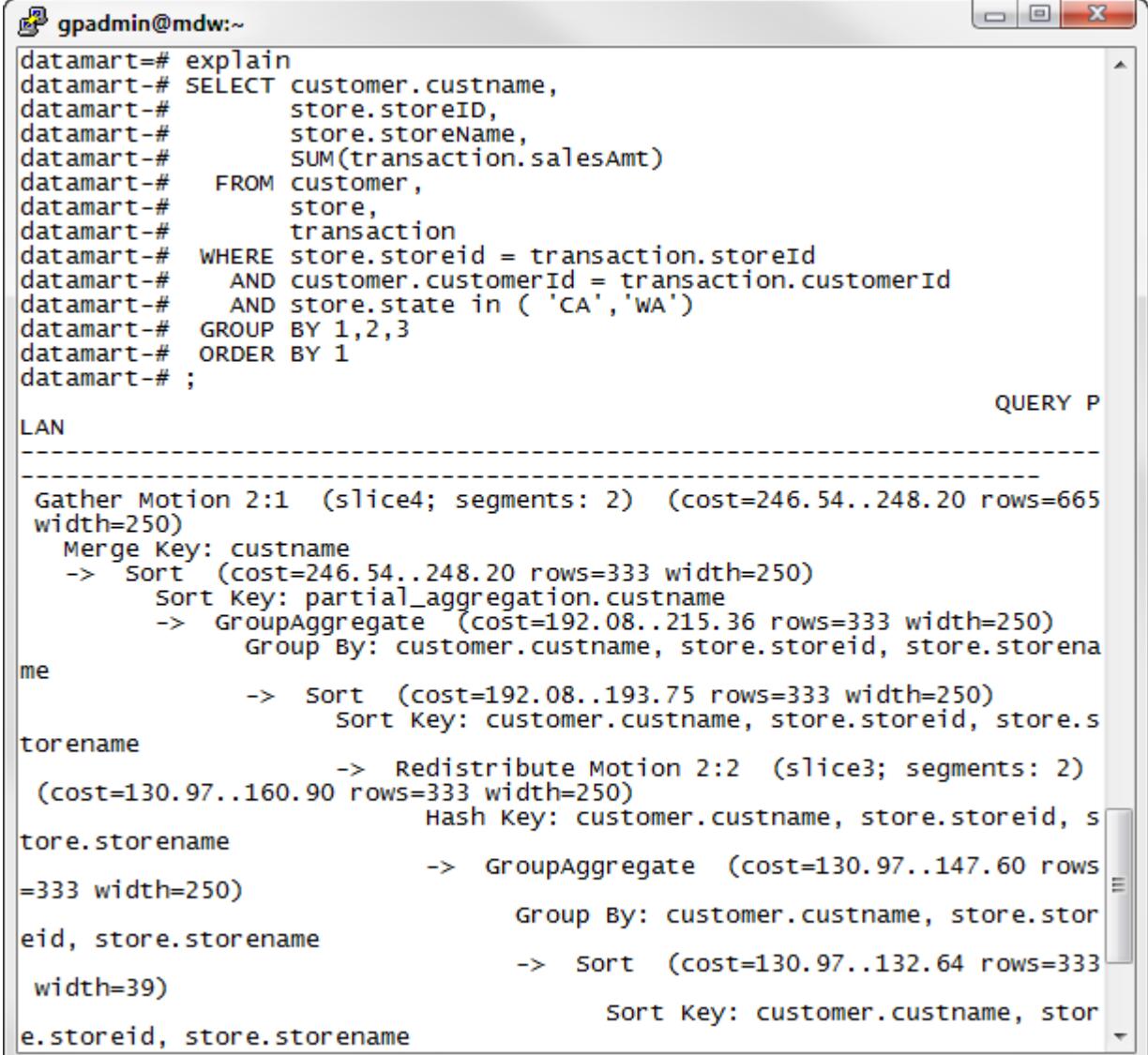
Step	Action																																
7.	<p>The session will now be saved. Double-click the session under the Servers category to initiate a connection to the database.</p>  <p>The screenshot shows the pgAdmin III interface. The Object browser on the left lists 'Server Groups' and 'Servers (1)'. Under 'Servers', there is one entry: 'training (172.16.1.11:5432)' which has children: 'Databases (9)', 'Tablespaces (2)', 'Group Roles (2)', 'Login Roles (5)', and 'Resource Queues (2)'. To the right of the Object browser is the 'Properties' tab of a properties pane. The properties listed are:</p> <table border="1"> <thead> <tr> <th>Property</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Description</td> <td>training</td> </tr> <tr> <td>Service</td> <td></td> </tr> <tr> <td>Hostname</td> <td>172.16.1.11</td> </tr> <tr> <td>Host Address</td> <td></td> </tr> <tr> <td>Port</td> <td>5432</td> </tr> <tr> <td>SSL Certificate File</td> <td></td> </tr> <tr> <td>SSL Key File</td> <td></td> </tr> <tr> <td>SSL Root Certificate File</td> <td></td> </tr> <tr> <td>SSL Certificate Revocation List</td> <td></td> </tr> <tr> <td>SSL Compression?</td> <td>yes</td> </tr> <tr> <td>Maintenance database</td> <td>postgres</td> </tr> <tr> <td>Username</td> <td>gpadmin</td> </tr> <tr> <td>Store password?</td> <td>Yes</td> </tr> <tr> <td>Restore environment?</td> <td>Yes</td> </tr> <tr> <td>Connected?</td> <td>No</td> </tr> </tbody> </table> <p>Below the properties pane is a 'SQL pane'.</p> <p>You now have access to the database through pgAdmin III.</p>	Property	Value	Description	training	Service		Hostname	172.16.1.11	Host Address		Port	5432	SSL Certificate File		SSL Key File		SSL Root Certificate File		SSL Certificate Revocation List		SSL Compression?	yes	Maintenance database	postgres	Username	gpadmin	Store password?	Yes	Restore environment?	Yes	Connected?	No
Property	Value																																
Description	training																																
Service																																	
Hostname	172.16.1.11																																
Host Address																																	
Port	5432																																
SSL Certificate File																																	
SSL Key File																																	
SSL Root Certificate File																																	
SSL Certificate Revocation List																																	
SSL Compression?	yes																																
Maintenance database	postgres																																
Username	gpadmin																																
Store password?	Yes																																
Restore environment?	Yes																																
Connected?	No																																

## Lab 19: Task 2 – Analyze Queries

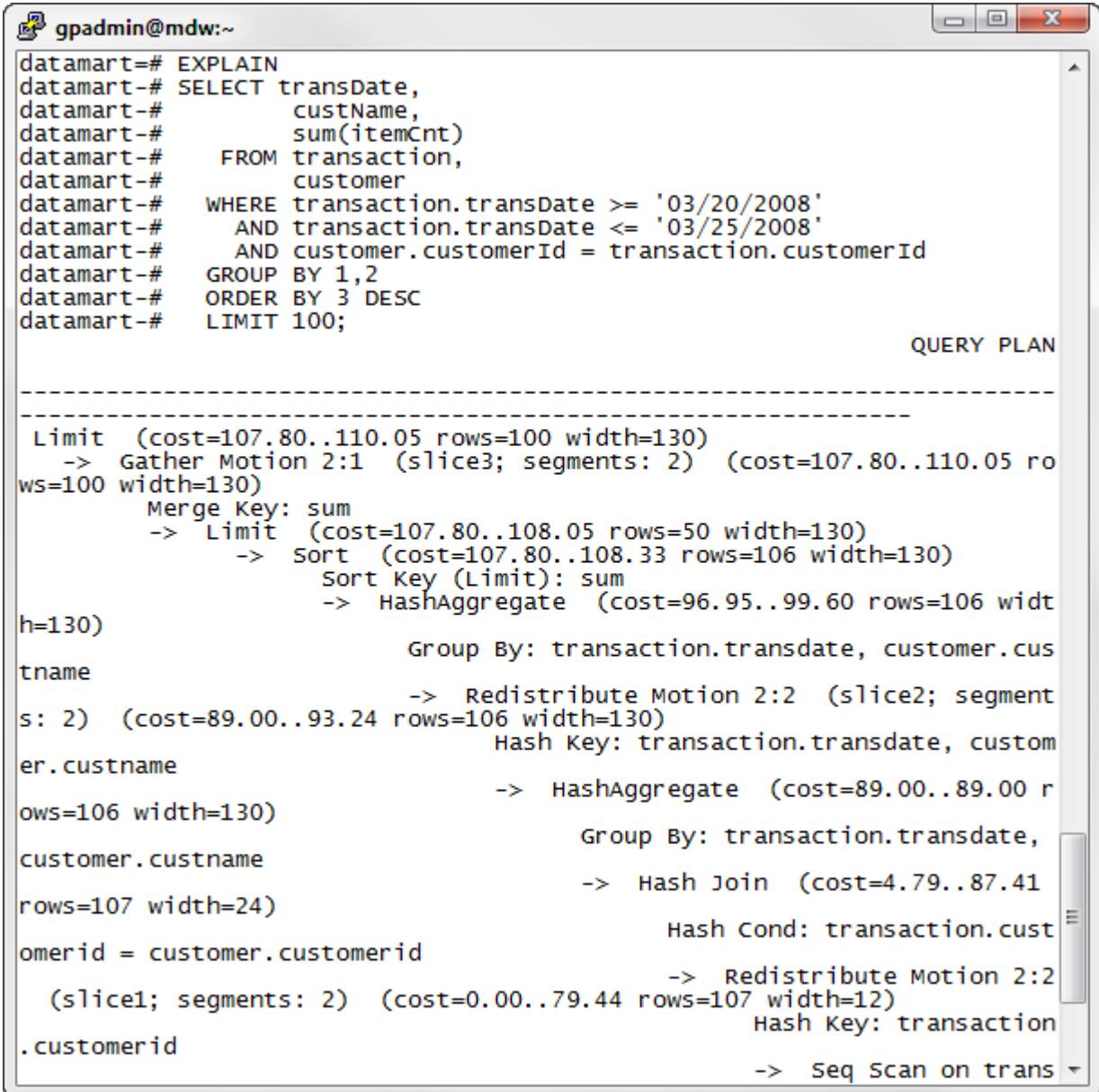
Step	Action
	<p>You can choose to use pgAdmin III to view the explain plans for the queries you will examine in this task. Instructions will be provided for both for the first exercise. Remaining exercises will use PSQL. However, feel free to use pgAdmin III to view the query plans or execute queries.</p>
1.	<p>If you wish to view the query plans in PSQL, proceed to step 8. In pgAdmin III, click on the <b>datamart</b> database. This will be the database used for query executions.</p> 
2.	<p>Click the Execute Arbitrary SQL queries icon to proceed.</p> 
3.	<p>Click the <b>Open file</b> icon.</p> 

Step	Action
4.	<p>From the Documents folder, select the file, gp&gt;sql&gt;load_files&gt;adv_sql_files&gt;lab5query1 and either double-click the filename or click <b>Open</b>.</p> 
5.	<p>The first line of the query is the explain command. Remove this line.</p>  <p>While you can leave this line in the file and use the Execute query button to view the text of the explain plan, we will instead use the pgAdmin III explain plan method to view a graphical breakdown of the plan.</p>

Step	Action
6.	<p>Click the <b>Explain query</b> icon to view the explain plan.</p>  <pre data-bbox="388 397 1008 734"> SELECT customer.custname,        store.storeID,        store.storeName,        SUM(transaction.salesAmt)   FROM customer,        store,        transaction  WHERE store.storeid = transaction.storeId    AND customer.customerId = transaction.customerId    AND store.state in ( 'CA','WA')  GROUP BY 1,2,3  ORDER BY 1 ;</pre>
7.	<p>The Output pane displays a graphical analysis of the query in the Explain tab.</p>  <p>You can also view a textual analysis by clicking the Data Output tab.</p>  <pre data-bbox="329 1410 1374 1854"> 1  QUERY PLAN text 1  1 Gather Motion 2:1 (slice4; segments: 2) (cost=246.54..248.20 rows=665 width=250) 2  2   Merge Key: custname 3  3     -&gt; Sort (cost=246.54..248.20 rows=333 width=250) 4  4       Sort Key: partial aggregation.custname 5  5       -&gt; GroupAggregate (cost=192.08..215.36 rows=333 width=250) 6  6         Group By: customer.custname, store.storeid, store.storename 7  7         -&gt; Sort (cost=192.08..193.75 rows=333 width=250) 8  8           Sort Key: customer.custname, store.storeid, store.storename 9  9           -&gt; Redistribute Motion 2:2 (slice3; segments: 2) (cost=130.97..160.90 rows=333 width=250) 10 10             Hash Key: customer.custname, store.storeid, store.storename 11 11             -&gt; GroupAggregate (cost=130.97..147.60 rows=333 width=250) 12 12               Group By: customer.custname, store.storeid, store.storename 13 13               -&gt; Sort (cost=130.97..132.64 rows=333 width=39) 14 14                 Sort Key: customer.custname, store.storeid, store.storename 15 15                 -&gt; Hash Join (cost=12.80..99.76 rows=333 width=39) 16 16                   Hash Cond: transaction.customerid = customer.customerid 17 17                     -&gt; Hash Join (cost=2.24..72.05 rows=333 width=250) </pre> <p>Proceed to step 10.</p>

Step	Action
8.	<p>Complete the following steps to view the explain plan in PSQL.  Access the datamart database as gpadmin.</p> <pre>[gpadmin@mdw ~]\$ psql datamart</pre>
9.	<p>Execute an explain plan for a simple query without analyzing the data. This query can be found in the file /home/gp/sql/load_files/adv_sql_files/lab5query1.sql.</p> <pre>datamart=# \i /home/gp/sql/load_files/adv_sql_files/lab5query1.sql</pre>  <p>The screenshot shows a terminal window titled "gpadmin@mdw:~". The command entered is "explain \i /home/gp/sql/load_files/adv_sql_files/lab5query1.sql". The output displays the detailed execution plan for the query, which involves gathering motion, merging, sorting, and redistributing data across multiple segments. The plan includes information about cost, rows, width, and specific keys used for each operation.</p>
10.	<p>Which tables are being scanned?</p> <hr/> <hr/> <hr/>

Step	Action
11.	<p>Is there any data motion or Broadcasts? Which table(s)?</p> <hr/> <hr/> <hr/>
12.	<p>How could you reduce the number of tables being scanned?</p> <hr/> <hr/> <hr/>
13.	<p>If you add a WHERE clause to get transactions for the month of May 2008 only, how does the explain plan change? This SQL can be found in the file lab5query1a.sql.</p> <p><b>AND transaction.transdate BETWEEN '2008-05-01' AND '2008-05-31'</b></p> <hr/> <hr/> <hr/>

Step	Action
14.	<p>Execute another explain plan for a simple query without analyzing the data. This query is found in the file /home/gp/sql/load_files/adv_sql_files/lab5query2.sql:</p> <pre data-bbox="313 276 1468 308">datamart=# \i /home/gp/sql/load_files/adv_sql_files/lab5query2.sql</pre>  <pre data-bbox="313 382 1494 1522"> datamart=# EXPLAIN datamart-# SELECT transDate, datamart-#           custName, datamart-#           sum(itemCnt) datamart-#      FROM transaction, datamart-#           customer datamart-#     WHERE transaction.transDate &gt;= '03/20/2008' datamart-#       AND transaction.transDate &lt;= '03/25/2008' datamart-#       AND customer.customerId = transaction.customerId datamart-#   GROUP BY 1,2 datamart-#   ORDER BY 3 DESC datamart-#   LIMIT 100;  QUERY PLAN ----- Limit  (cost=107.80..110.05 rows=100 width=130)   -&gt; Gather Motion 2:1  (slice3; segments: 2)  (cost=107.80..110.05 rows=100 width=130)       Merge Key: sum       -&gt; Limit  (cost=107.80..108.05 rows=50 width=130)           -&gt; Sort  (cost=107.80..108.33 rows=106 width=130)               Sort Key (Limit): sum               -&gt; HashAggregate  (cost=96.95..99.60 rows=106 width=130)                   Group By: transaction.transdate, customer.cus                   tname                   -&gt; Redistribute Motion 2:2  (slice2; segment                       s: 2)  (cost=89.00..93.24 rows=106 width=130)                       Hash Key: transaction.transdate, custom                       er.custname                       -&gt; HashAggregate  (cost=89.00..89.00 r                           ows=106 width=130)                           Group By: transaction.transdate,                           customer.custname                           -&gt; Hash Join  (cost=4.79..87.41                             rows=107 width=24)                             Hash Cond: transaction.cust                             omerid = customer.customerid                             -&gt; Redistribute Motion 2:2                                 (slice1; segments: 2)  (cost=0.00..79.44 rows=107 width=12)                                 Hash Key: transaction.cust                                 .customerid                                 -&gt; Seq Scan on trans </pre>
15.	<p>Which tables are being scanned?</p> <hr/> <hr/> <hr/>

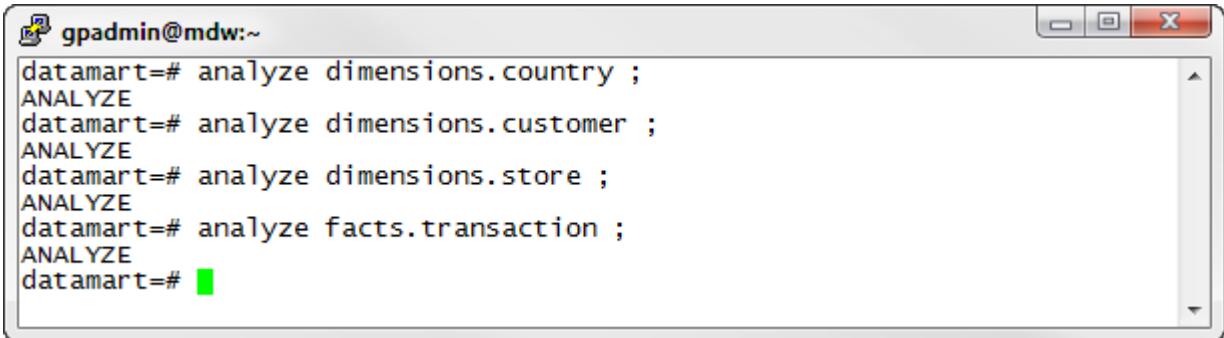
Step	Action
16.	Is there any data motion or Broadcasts? Which table(s)? _____ _____ _____
17.	If you partitioned your tables, how many “transaction” tables are being scanned? Why? _____ _____ _____
	<p><b>Summary</b></p> <p>You should execute an EXPLAIN statement with any new SQL that accesses large tables, any SQL that is running slowly, or against data objects that are new to the system. This helps you to determine, in advance of going into production, the likely path that the optimizer will take when performing the query. The EXPLAIN utility can produce a plan for any SQL statement, except EXPLAIN itself, so you are encouraged to take advantage of this functionality as often as possible.</p>

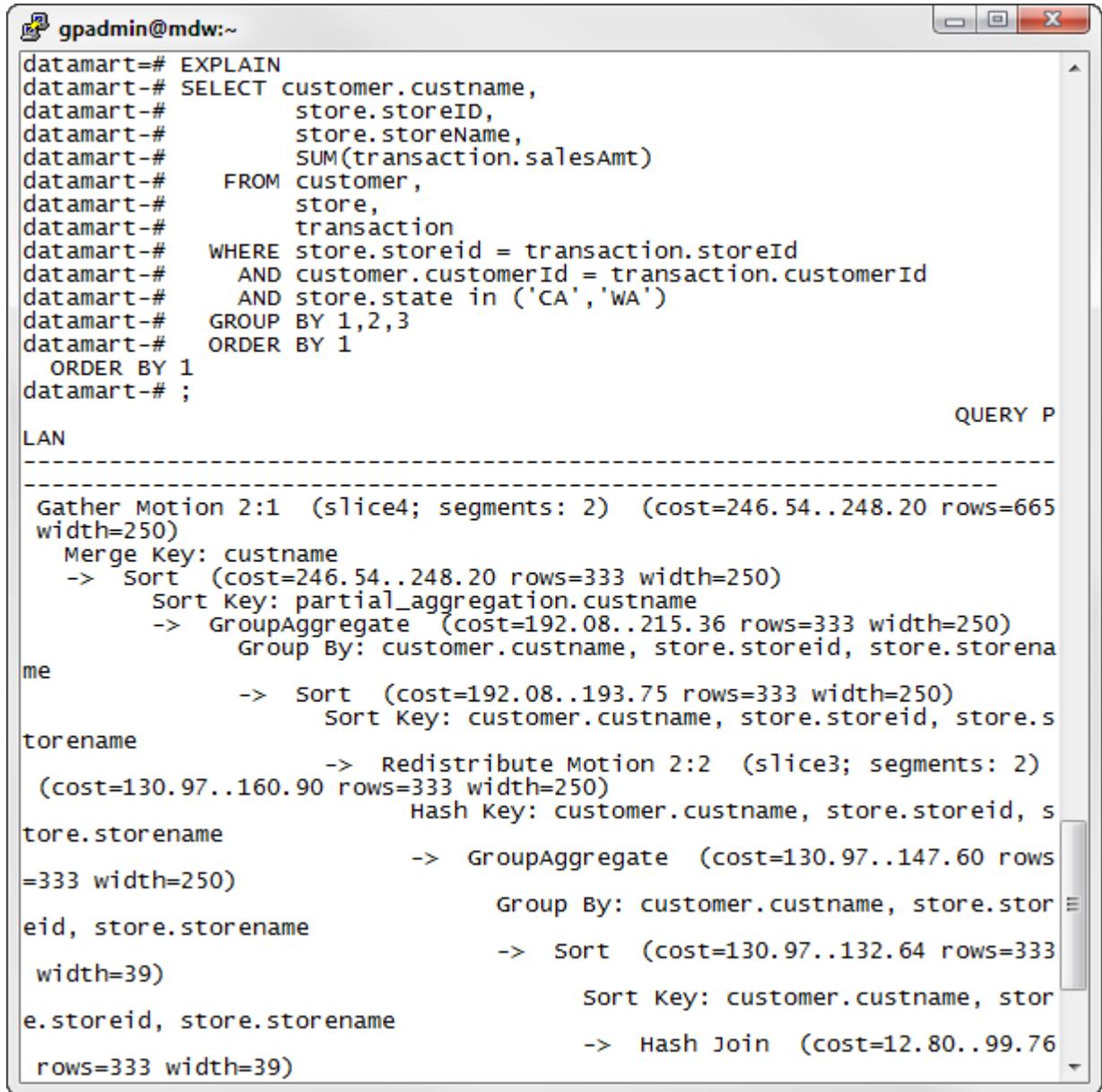
*End of Lab Exercise*

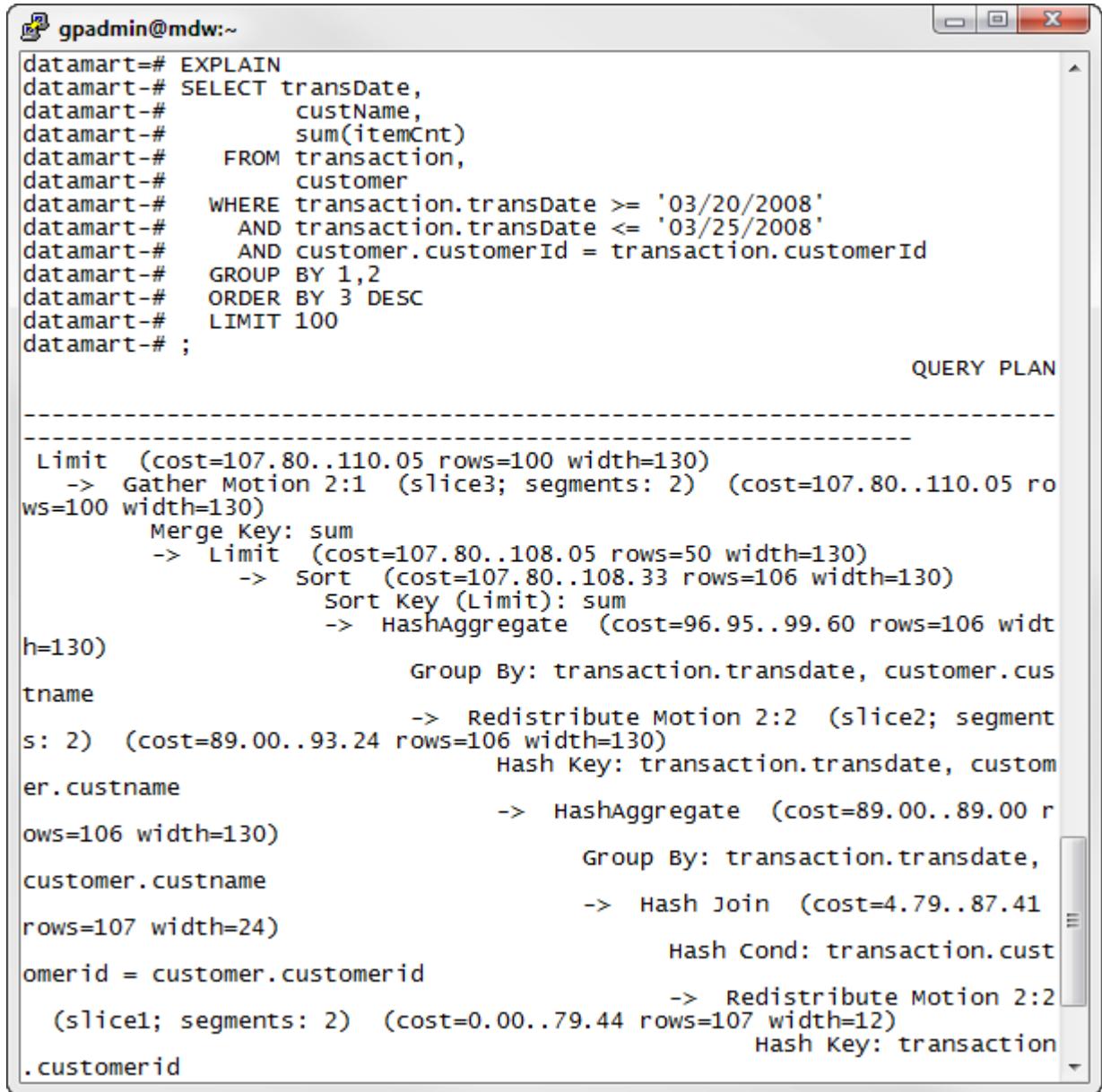
## Lab 20. Improve Performance with Statistics

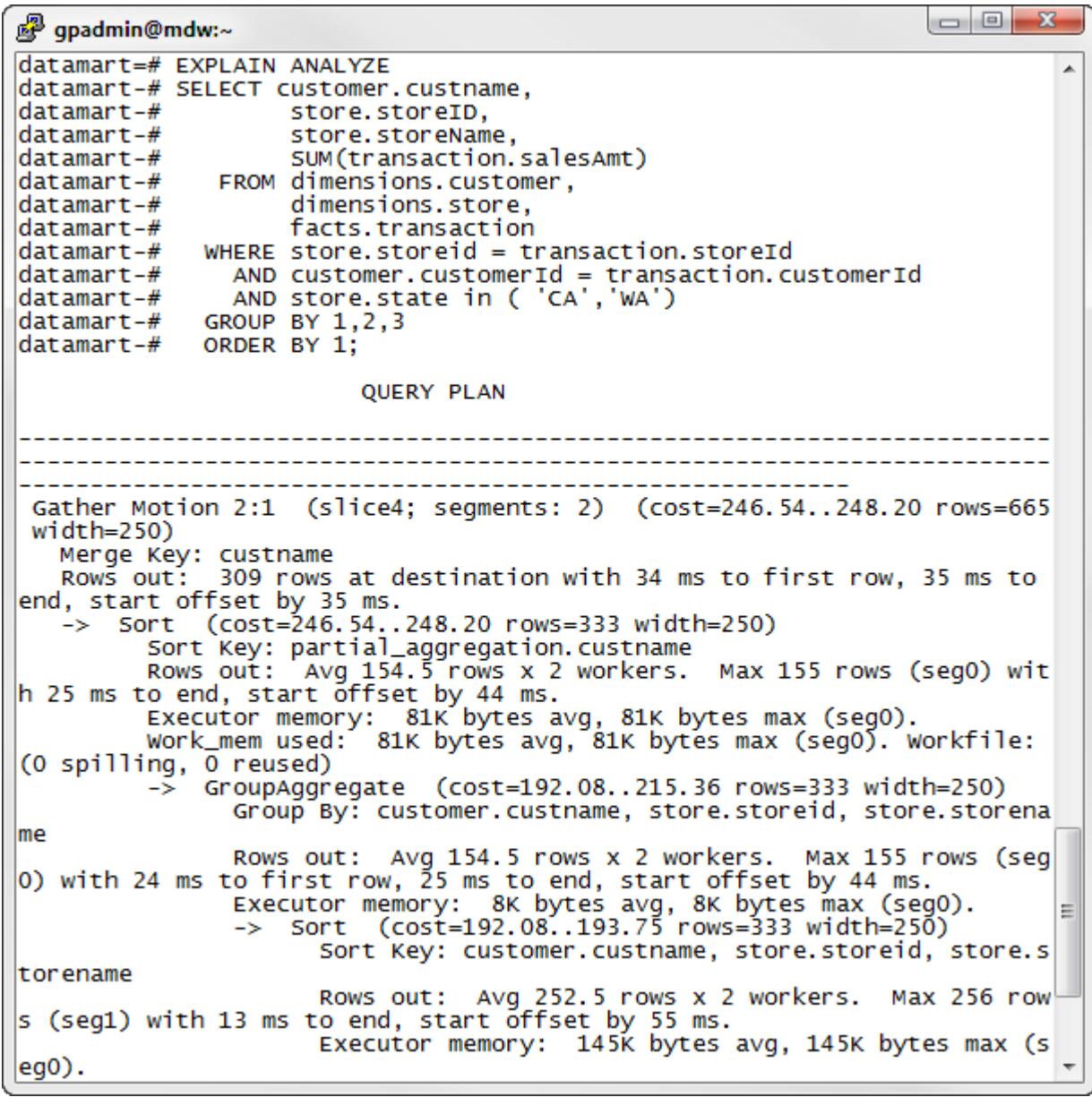
<b>Purpose:</b>	In this lab you will gather statistics on all of the dimension and fact tables in your data warehouse. You will then do an EXPLAIN ANALYZE step for the two queries from the previous lab.
<b>Tasks:</b>	Students perform the following task: Gather statistics and analyze queries.
<b>References:</b>	Module 8 – Performance Analysis and Tuning <ul style="list-style-type: none"><li>• Lesson: Improve Performance with Statistics</li></ul>

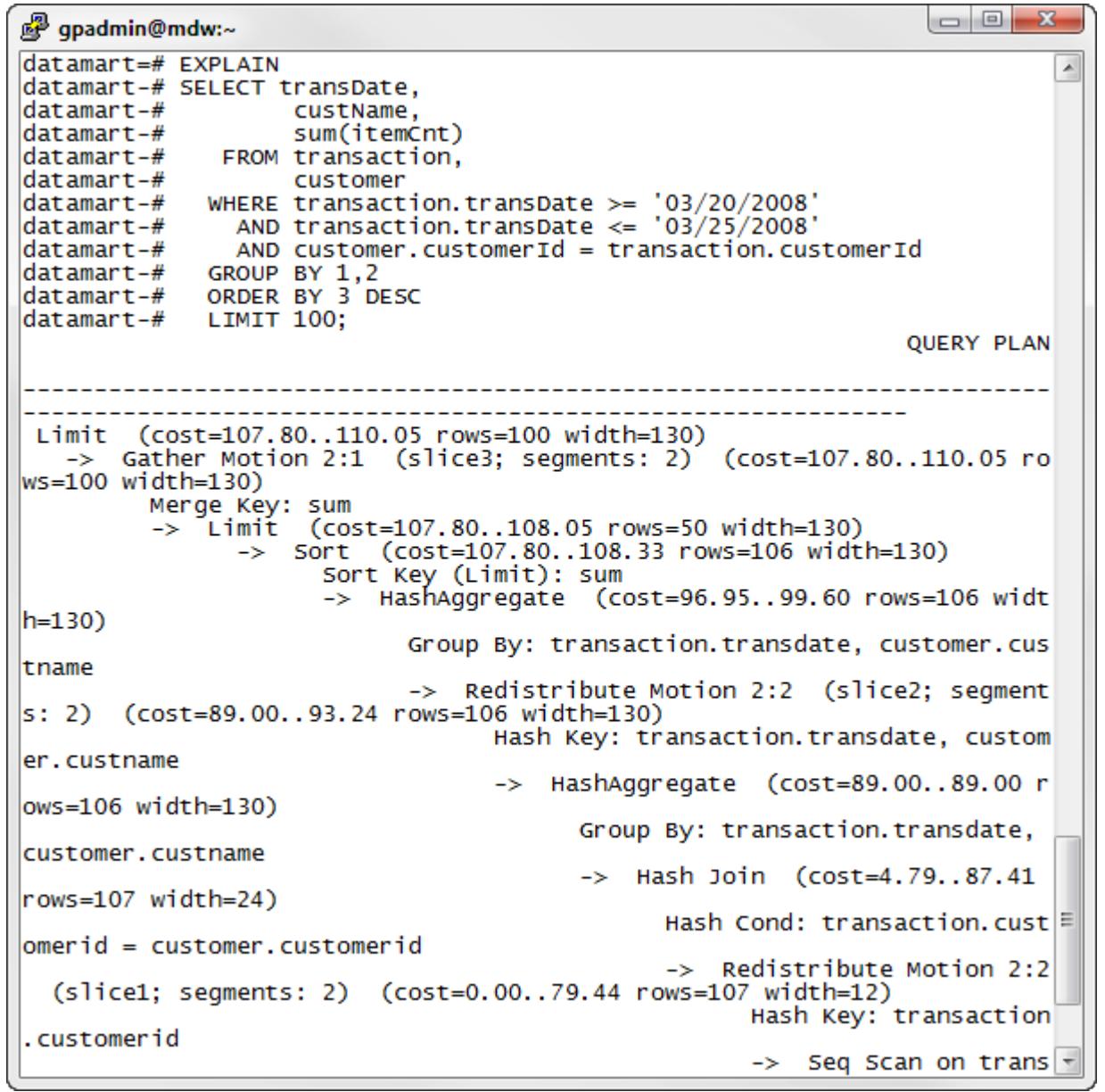
## Lab 20: Task – Gather Statistics and Analyze Queries

Step	Action
1.	Connect to the datamart database as gpadmin.
2.	Analyze each of dimensions and facts tables using the ANALYZE command: 1. Analyze the dimensions.country table: <code>datamart=# analyze dimensions.country;</code> 2. Analyze the dimensions.customer table: <code>datamart=# analyze dimensions.customer;</code> 3. Analyze the dimensions.store table: <code>datamart=# analyze dimensions.store;</code> 4. Analyze the facts.transaction table <code>datamart=# analyze facts.transaction;</code>   <pre>gpadmin@mdw:~\$ datamart=# analyze dimensions.country ; ANALYZE datamart=# analyze dimensions.customer ; ANALYZE datamart=# analyze dimensions.store ; ANALYZE datamart=# analyze facts.transaction ; ANALYZE datamart=#</pre>

Step	Action
3.	<p>Execute the first query from the previous plan with the EXPLAIN command. This query is found in the /home/gp/sql/load_files/adv_sql_files/lab5query1.sql file:</p> <pre data-bbox="306 276 1468 308">datamart=# \i /home/gp/sql/load_files/adv_sql_files/lab5query1.sql</pre>  <p>Are there any differences between the query plans with statistics?</p> <hr/> <hr/> <hr/>

Step	Action
4.	<p>Execute the second explain query plan from the previous lab using the EXPLAIN command. This query is found in the /home/gp/sql/load_files/adv_sql_files/lab5query2.sql file:</p> <pre data-bbox="306 276 1468 308">datamart=# \i /home/gp/sql/load_files/adv_sql_files/lab5query2.sql</pre>  <pre data-bbox="306 340 1535 1558"> gpadmin@mdw:~ datamart=# EXPLAIN datamart# SELECT transDate, datamart#           custName, datamart#           sum(itemCnt) datamart#     FROM transaction, datamart#           customer datamart#   WHERE transaction.transDate &gt;= '03/20/2008' datamart#       AND transaction.transDate &lt;= '03/25/2008' datamart#       AND customer.customerId = transaction.customerId datamart#   GROUP BY 1,2 datamart#   ORDER BY 3 DESC datamart#   LIMIT 100 datamart# ; </pre> <p style="text-align: right;">QUERY PLAN</p> <hr/> <pre data-bbox="306 825 1535 1558"> ----- Limit (cost=107.80..110.05 rows=100 width=130)   -&gt; Gather Motion 2:1 (slice3; segments: 2) (cost=107.80..110.05 rows=100 width=130)       Merge Key: sum       -&gt; Limit (cost=107.80..108.05 rows=50 width=130)           -&gt; Sort (cost=107.80..108.33 rows=106 width=130)               Sort Key (Limit): sum               -&gt; HashAggregate (cost=96.95..99.60 rows=106 width=130)                   Group By: transaction.transdate, customer.custname                   -&gt; Redistribute Motion 2:2 (slice2; segments: 2) (cost=89.00..93.24 rows=106 width=130)                       Hash Key: transaction.transdate, customer.custname                       -&gt; HashAggregate (cost=89.00..89.00 rows=106 width=130)                           Group By: transaction.transdate, customer.custname                           -&gt; Hash Join (cost=4.79..87.41 rows=107 width=24)                               Hash Cond: transaction.customerId = customer.customerId                               -&gt; Redistribute Motion 2:2 (slice1; segments: 2) (cost=0.00..79.44 rows=107 width=12)                                   Hash Key: transaction.customerId </pre> <p>Do you note any differences between this output and the output from the previous lab?</p> <hr/> <hr/> <hr/>

Step	Action
5.	<p>Execute these same queries with an EXPLAIN ANALYZE statement to compare actual execution demographics with the projected execution demographics. This query is found in the /home/gp/sql/load_files/adv_sql_files/lab6query1.sql file:</p> <pre data-bbox="306 312 1481 342">datamart=# \i /home/gp/sql/load_files/adv_sql_files/lab6query1.sql</pre>
	 <pre data-bbox="323 439 1498 1600"> datamart=# EXPLAIN ANALYZE datamart-# SELECT customer.custname, datamart-#           store.storeID, datamart-#           store.storeName, datamart-#           SUM(transaction.salesAmt) datamart-#     FROM dimensions.customer, datamart-#           dimensions.store, datamart-#           facts.transaction datamart-#   WHERE store.storeid = transaction.storeId datamart-#     AND customer.customerId = transaction.customerId datamart-#     AND store.state in ( 'CA', 'WA') datamart-# GROUP BY 1,2,3 datamart-# ORDER BY 1;  QUERY PLAN  ----- Gather Motion 2:1  (slice4; segments: 2)  (cost=246.54..248.20 rows=665 width=250)   Merge Key: custname   Rows out: 309 rows at destination with 34 ms to first row, 35 ms to end, start offset by 35 ms.     -&gt; Sort (cost=246.54..248.20 rows=333 width=250)       Sort Key: partial_aggregation.custname       Rows out: Avg 154.5 rows x 2 workers. Max 155 rows (seg0) with 25 ms to end, start offset by 44 ms.         Executor memory: 81K bytes avg, 81K bytes max (seg0).         work_mem used: 81K bytes avg, 81K bytes max (seg0). workfile: (0 spilling, 0 reused)           -&gt; GroupAggregate (cost=192.08..215.36 rows=333 width=250)             Group By: customer.custname, store.storeid, store.storename             Rows out: Avg 154.5 rows x 2 workers. Max 155 rows (seg0) with 24 ms to first row, 25 ms to end, start offset by 44 ms.               Executor memory: 8K bytes avg, 8K bytes max (seg0).               -&gt; Sort (cost=192.08..193.75 rows=333 width=250)                 Sort Key: customer.custname, store.storeid, store.storename                 Rows out: Avg 252.5 rows x 2 workers. Max 256 rows (seg1) with 13 ms to end, start offset by 55 ms.                   Executor memory: 145K bytes avg, 145K bytes max (seg0). </pre>
6.	<p>What do you note about this output when compared to an EXPLAIN?</p> <hr/> <hr/> <hr/>

Step	Action
7.	<p>Execute the second query with EXPLAIN ANALYZE as well. This query is found in the /home/gp/sql/load_files/adv_sql_files/lab6query2.sql file:</p> <pre data-bbox="306 276 1468 308">datamart=# \i /home/gp/sql/load_files/adv_sql_files/lab6query2.sql</pre>  <pre data-bbox="323 397 1498 1522"> gpadmin@mdw:~ datamart=# EXPLAIN datamart-# SELECT transDate, datamart-#           custName, datamart-#           sum(itemCnt) datamart-#     FROM transaction, datamart-#           customer datamart-#    WHERE transaction.transDate &gt;= '03/20/2008' datamart-#        AND transaction.transDate &lt;= '03/25/2008' datamart-#        AND customer.customerId = transaction.customerId datamart-#   GROUP BY 1,2 datamart-# ORDER BY 3 DESC datamart-# LIMIT 100;  QUERY PLAN  ----- Limit  (cost=107.80..110.05 rows=100 width=130)  -&gt; Gather Motion 2:1  (slice3; segments: 2)  (cost=107.80..110.05 rows=100 width=130)      Merge Key: sum      -&gt; Limit  (cost=107.80..108.05 rows=50 width=130)          -&gt; Sort  (cost=107.80..108.33 rows=106 width=130)              Sort Key (Limit): sum              -&gt; HashAggregate  (cost=96.95..99.60 rows=106 width=130)                  Group By: transaction.transdate, customer.cus                  tname                  -&gt; Redistribute Motion 2:2  (slice2; segment                      s: 2)  (cost=89.00..93.24 rows=106 width=130)                      Hash Key: transaction.transdate, custom                      er.custname                      -&gt; HashAggregate  (cost=89.00..89.00 r                          ows=106 width=130)                          Group By: transaction.transdate,                          customer.custname                          -&gt; Hash Join  (cost=4.79..87.41                              rows=107 width=24)                              Hash Cond: transaction.cust                              omerid = customer.customerid                              -&gt; Redistribute Motion 2:2                                  (slice1; segments: 2)  (cost=0.00..79.44 rows=107 width=12)                                  Hash Key: transaction                                  .customerid                                  -&gt; Seq Scan on trans </pre>
8.	<p>What do you note about this output when compared to an EXPLAIN?</p> <hr/> <hr/> <hr/>

Step	Action
	<p><b>Summary</b></p> <p>ANALYZE is a SQL command that updates the database statistics used by the query planner. In this use case, a query is run both before and after doing ANALYZE. After statistics are collected, the query planner chooses a HashAggregate operation over a much slower GroupAggregate operation. Without any statistics, the query planner could not estimate how many records might be returned, and therefore could not determine if there was sufficient work memory to do the aggregations in memory. The planner always takes the safe route and does aggregations by reading/writing from disk, which is significantly slower.</p>

*End of Lab Exercise*

This page intentionally left blank

## Lab 21. Indexing Strategies

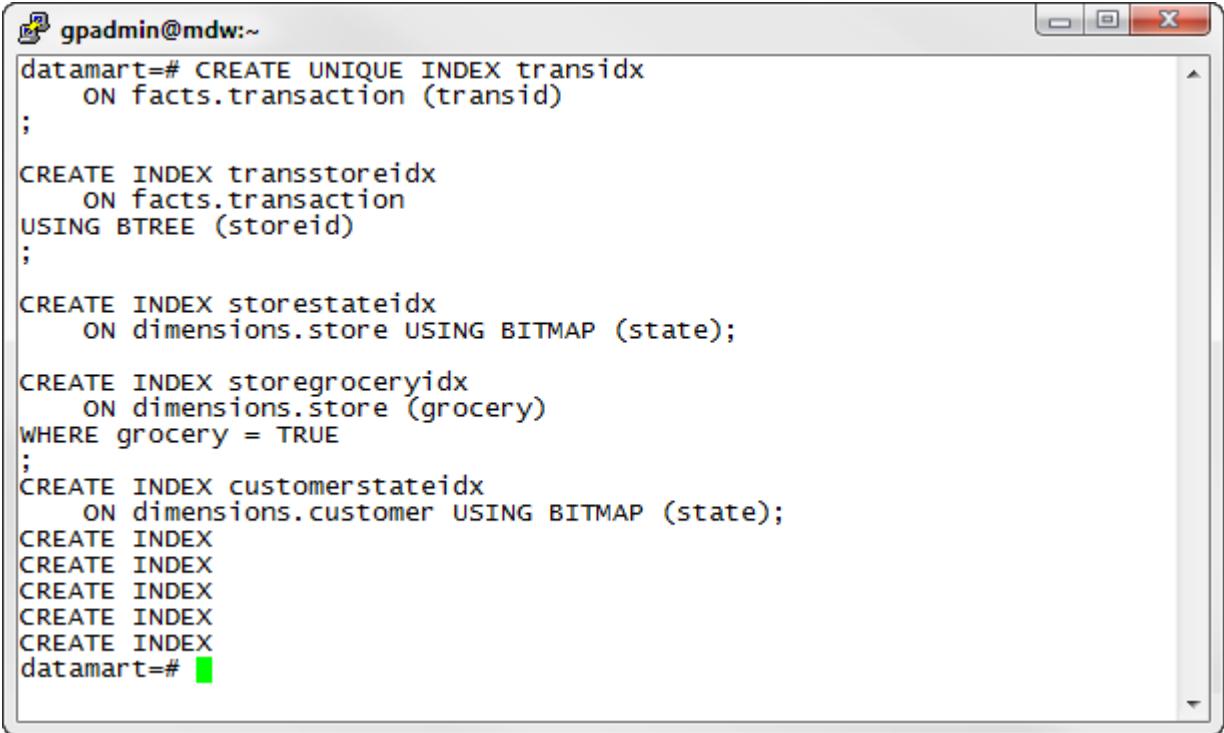
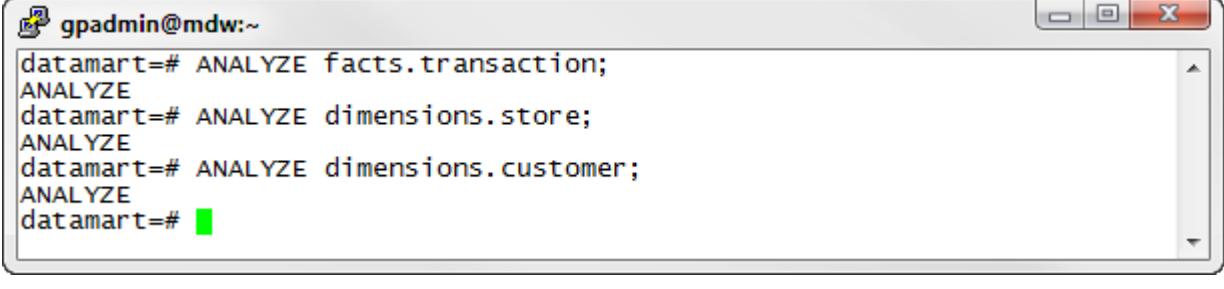
<b>Purpose:</b>	In this lab, you determine requirements for secondary indexes on the fact table TRANSACTION. You will then create the secondary indexes.
-----------------	--

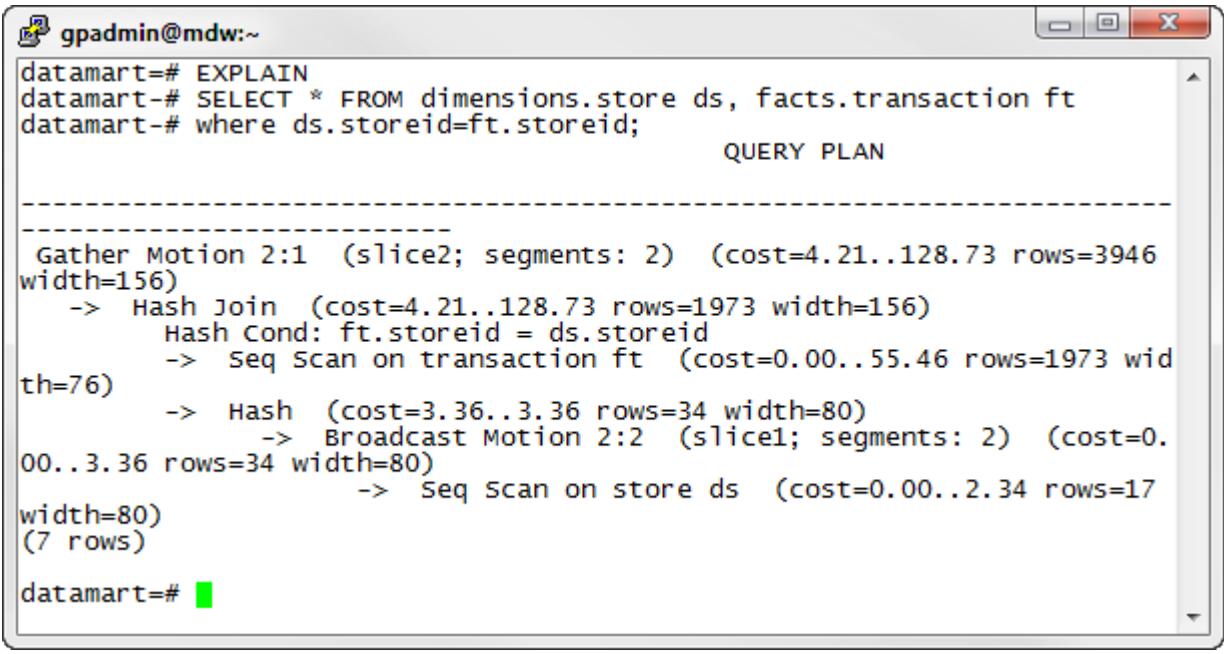
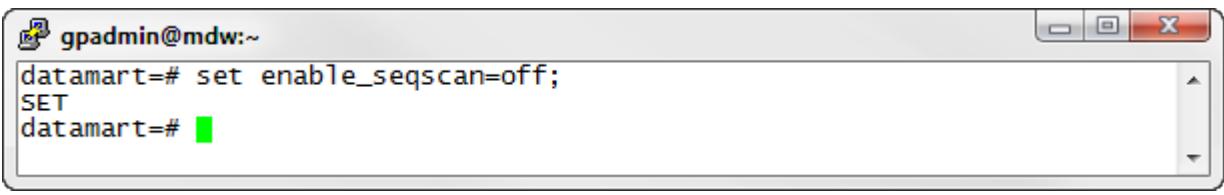
<b>Tasks:</b>	Students perform the following task: Create indexes to support queries
---------------	--

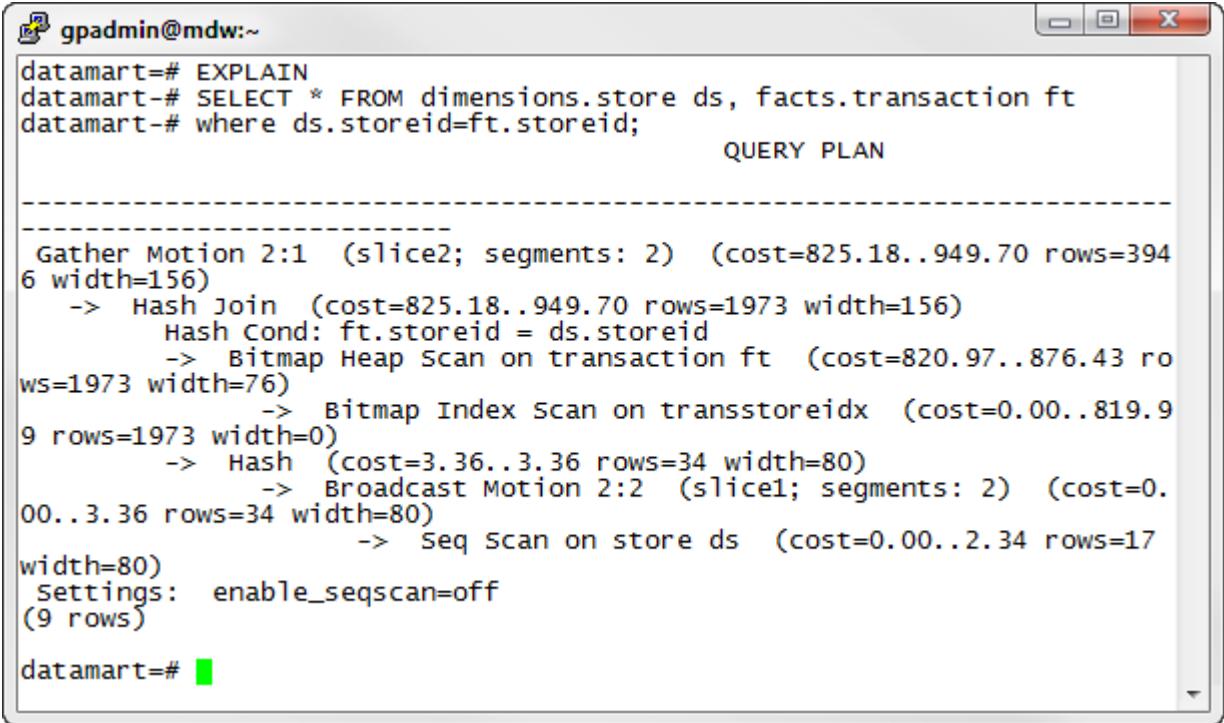
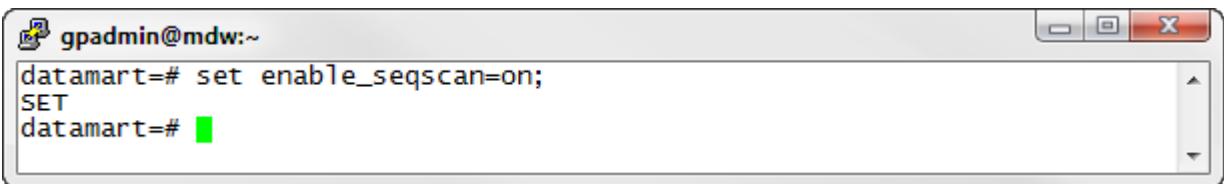
<b>References:</b>	Module 8 – Performance Analysis and Tuning <ul style="list-style-type: none"><li>• Lesson: Indexing Strategies</li></ul>
--------------------	--

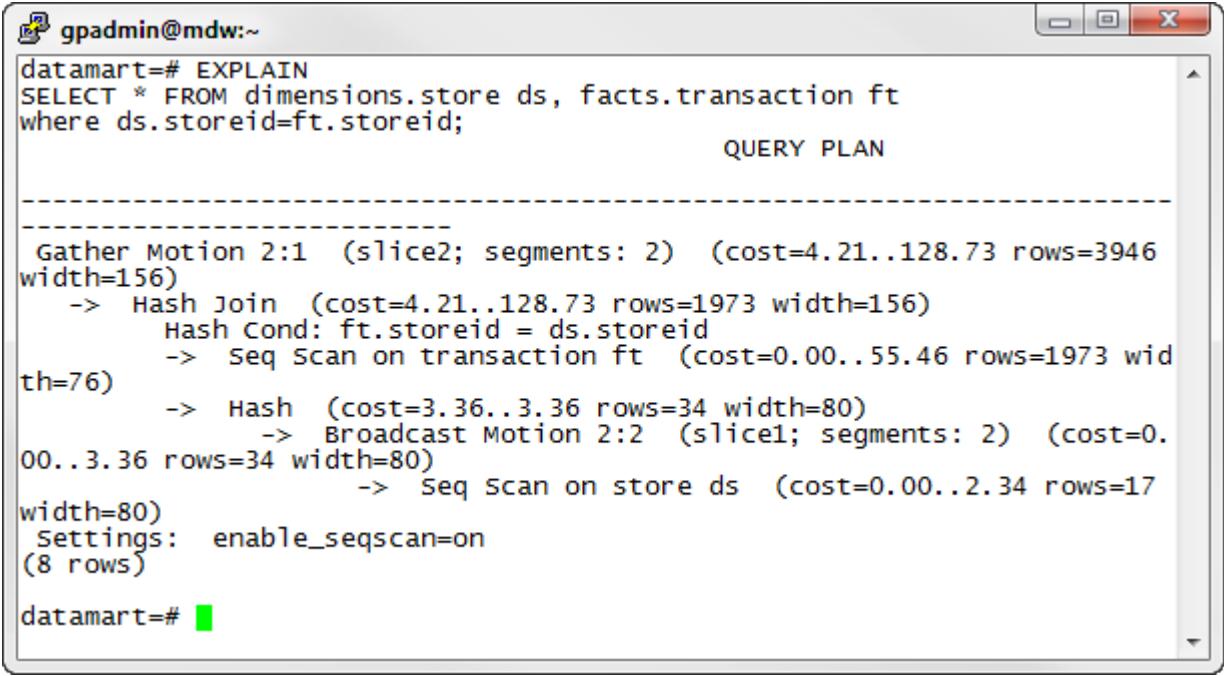
## Lab 21: Task – Create Indexes to Support Queries

Step	Action																		
1.	<p>Here are some facts that you can use to develop your index strategies:</p> <ul style="list-style-type: none"><li>• Store managers often query the transaction table for their store(s) using the storeid.</li><li>• Business Intelligence (BI) tools execute a lot of queries for stores and customers by state for one or more states.</li><li>• One of the executive dashboards queries transactions at stores based on the store feature grocery.</li><li>• Auditing directly accesses transactions by the transaction id.</li></ul> <p>Use the following matrix to determine which indexes you need to build to support the above analysis. A list of the tables that require indexes has been provided:</p> <table border="1"><thead><tr><th>Table Name</th><th>Column(s)</th><th>Index Type</th></tr></thead><tbody><tr><td>transaction</td><td></td><td></td></tr><tr><td>store</td><td></td><td></td></tr><tr><td>customer</td><td></td><td></td></tr><tr><td>store</td><td></td><td></td></tr><tr><td>transaction</td><td></td><td></td></tr></tbody></table>	Table Name	Column(s)	Index Type	transaction			store			customer			store			transaction		
Table Name	Column(s)	Index Type																	
transaction																			
store																			
customer																			
store																			
transaction																			
2.	Connect to the datamart database as gpadmin.																		

Step	Action
3.	<p>Using the CREATE INDEX statement syntax, create the above indexes for your database. Note that to index the transactions table, you will need to index each of the child tables.</p> <p>The sample script, /home/gp/sql/load_files/adv_sql_files/lab7createidx.sql, can be used for guidance.</p> <p>Execute the query file:</p> <pre>datamart=# \i /home/gp/sql/load_files/adv_sql_files/lab7createidx.sql</pre>  <pre>gpadmin@mdw:~ datamart=# CREATE UNIQUE INDEX transidx     ON facts.transaction (transid) ;  CREATE INDEX transstoreidx     ON facts.transaction USING BTREE (storeid) ;  CREATE INDEX storestateidx     ON dimensions.store USING BITMAP (state);  CREATE INDEX storegroceryidx     ON dimensions.store (grocery) WHERE grocery = TRUE ; CREATE INDEX customerstateidx     ON dimensions.customer USING BITMAP (state); CREATE INDEX CREATE INDEX CREATE INDEX CREATE INDEX CREATE INDEX CREATE INDEX datamart=# </pre>
4.	<p>ANALYZE your database tables with new indexes:</p> <ol style="list-style-type: none"> <li>1. Analyze the facts.transaction table:           <pre>datamart=# ANALYZE facts.transaction;</pre> </li> <li>2. Analyze the dimensions.store table:           <pre>datamart=# ANALYZE dimensions.store;</pre> </li> <li>3. Analyze the dimensions.customer table:           <pre>datamart=# ANALYZE dimensions.customer;</pre> </li> </ol>  <pre>gpadmin@mdw:~ datamart=# ANALYZE facts.transaction; ANALYZE datamart=# ANALYZE dimensions.store; ANALYZE datamart=# ANALYZE dimensions.customer; ANALYZE datamart=# </pre>

Step	Action
5.	<p>Determine if your indexes are being used by the optimizer by running the EXPLAIN utility with some simple queries that should use the index.</p> <p>If the indexes are not being used, it is likely because of the small size of the data set. Try changing the enable_seqscan configuration parameter to off if that is the case. Re-execute the EXPLAIN queries and then execute the queries. Be sure to set enable_seqscan back to on before exiting psql. Execute an EXPLAIN against a query:</p> <pre data-bbox="290 460 1263 551">datamart=# EXPLAIN SELECT * FROM dimensions.store ds, facts.transaction ft where ds.storeid=ft.storeid;</pre> 
6.	<p>Disable sequential scans:</p> <pre data-bbox="290 1305 887 1339">datamart=# set enable_seqscan=off;</pre> 

Step	Action
7.	<p>Execute EXPLAIN against the query again:</p> <pre>datamart=# EXPLAIN SELECT * FROM dimensions.store ds, facts.transaction ft where ds.storeid=ft.storeid;</pre>  <pre>gpadmin@mdw:~\$ datamart=# EXPLAIN datamart-# SELECT * FROM dimensions.store ds, facts.transaction ft datamart-# where ds.storeid=ft.storeid;                QUERY PLAN ----- Gather Motion 2:1  (slice2; segments: 2)  (cost=825.18..949.70 rows=394 6 width=156)     -&gt;  Hash Join  (cost=825.18..949.70 rows=1973 width=156)         Hash Cond: ft.storeid = ds.storeid         -&gt;  Bitmap Heap Scan on transaction ft  (cost=820.97..876.43 ro ws=1973 width=76)             -&gt;  Bitmap Index Scan on transstoreidx  (cost=0.00..819.9 9 rows=1973 width=0)             -&gt;  Hash  (cost=3.36..3.36 rows=34 width=80)                 -&gt;  Broadcast Motion 2:2  (slice1; segments: 2)  (cost=0. 00..3.36 rows=34 width=80)                     -&gt;  Seq Scan on store ds  (cost=0.00..2.34 rows=17 width=80)     Settings: enable_seqscan=off (9 rows)  datamart=# </pre>
8.	<p>Set enable_seqscan to on:</p> <pre>datamart=# set enable_seqscan=on;</pre>  <pre>gpadmin@mdw:~\$ datamart=# set enable_seqscan=on; SET datamart=# </pre>

Step	Action
9.	<p>Execute EXPLAIN against the query again:</p> <pre>datamart=# EXPLAIN SELECT * FROM dimensions.store ds, facts.transaction ft where ds.storeid=ft.storeid;</pre>  <pre>gpadmin@mdw:~\$ datamart=# EXPLAIN SELECT * FROM dimensions.store ds, facts.transaction ft where ds.storeid=ft.storeid;  QUERY PLAN ----- Gather Motion 2:1  (slice2; segments: 2)  (cost=4.21..128.73 rows=3946 width=156) -&gt;  Hash Join  (cost=4.21..128.73 rows=1973 width=156)     Hash Cond: ft.storeid = ds.storeid         -&gt;  Seq Scan on transaction ft  (cost=0.00..55.46 rows=1973 width=76)             -&gt;  Hash  (cost=3.36..3.36 rows=34 width=80)                 -&gt;  Broadcast Motion 2:2  (slice1; segments: 2)  (cost=0.00..3.36 rows=34 width=80)                     -&gt;  Seq Scan on store ds  (cost=0.00..2.34 rows=17 width=80) Settings: enable_seqscan=on (8 rows)  datamart=#</pre>
	<p><b>Summary</b></p> <p>In most traditional databases, indexes can greatly improve data access times. However, in a distributed database such as Greenplum, indexes should be used more sparingly. The Greenplum Database is very fast at sequential scanning. Indexes use a random seek pattern to locate records on disk. Also, unlike a traditional database, the data is distributed across the segments. This means each segment scans a smaller portion of the overall data in order to get the result. If using table partitioning, the total data to scan may be even a fraction of that.</p> <p>Greenplum recommends that you first try your query workload without adding any additional indexes. Indexes are more likely to improve performance for OLTP type workloads, where the query is returning a single record or a very small data set. Typically, a business intelligence (BI) query workload returns very large data sets, and thus does not make efficient use of indexes.</p> <p>Note that the Greenplum Database will automatically create PRIMARY KEY indexes for tables with primary keys. If you are experiencing unsatisfactory performance, you may try adding indexes to see if performance improves.</p>

End of Lab Exercise

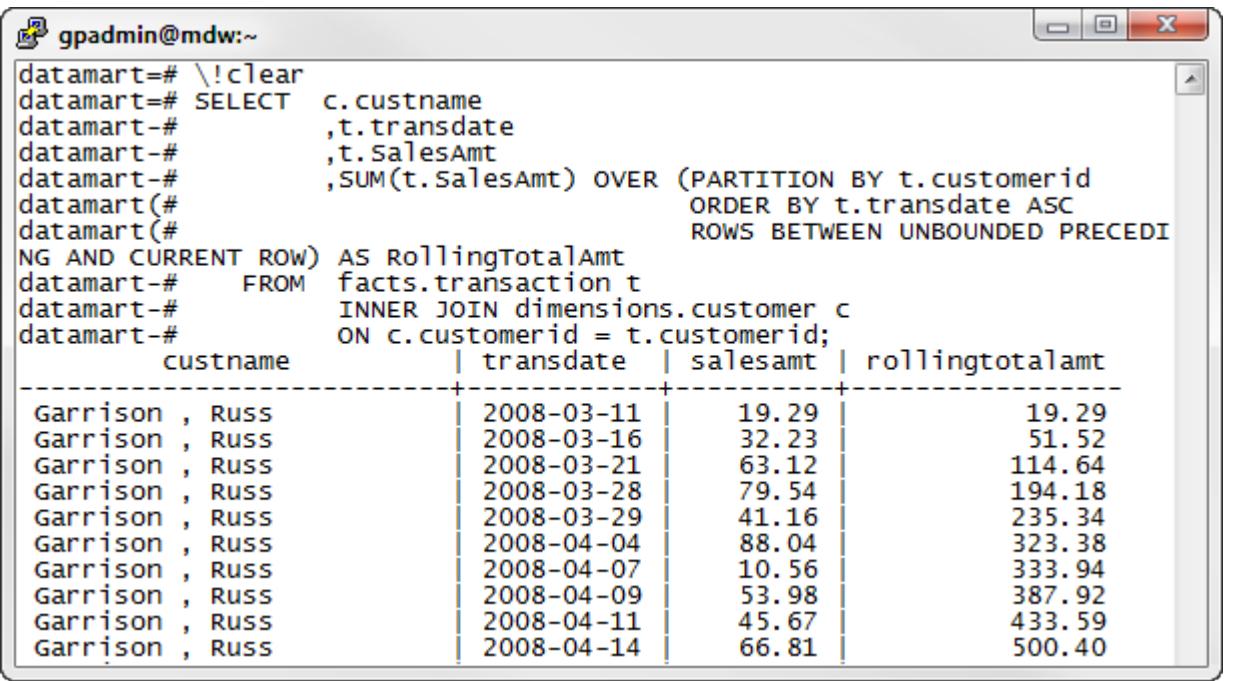
This page intentionally left blank

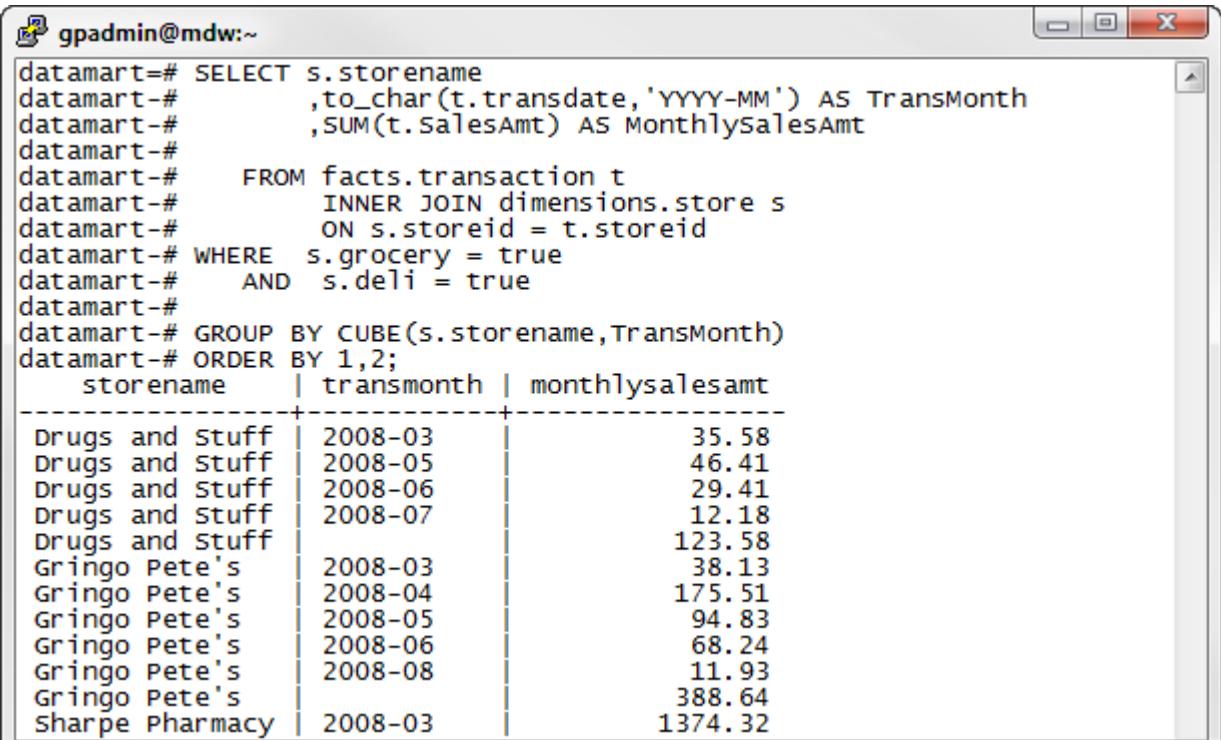
## Lab 22. Advanced Reporting Using OLAP

<b>Purpose:</b>	In this lab, you will be given a set of reporting requirements. You will write the SQL using the various OLAP functions discussed in the module. You may use the examples in the module as the basis for your queries.
<b>Tasks:</b>	Students perform the following task: Create reporting queries using OLAP functions
<b>References:</b>	Module 9 – Developing Reports Using Advanced SQL <ul style="list-style-type: none"><li>• Lesson: Advanced Reporting Using OLAP</li></ul>

## Lab 22: Task – Advanced Reporting Using OLAP

Step	Action
1.	<p>The reporting requirements for which you will develop queries using OLAP functions are as follows:</p> <ul style="list-style-type: none"><li>• Create a query that gives a rolling total of the Sales Amount by customer. Give the customer name.</li><li>• Create a query that gives a monthly total of the Sales Amount for all stores with a grocery and a deli by month, by state summarizing at each control break (state and month). (Hint: try one of the grouping sets like rollup or cube.)</li><li>• Create a query that lists the running total of each amount column by month, by store. Display the store name with the measures. (Hint: See the example in the module that uses a “derived table” or “in-line view”.)</li><li>• Create a query that lists the top customer by Sales Amount, by month to support a customer rewards marketing campaign. (Hint: You will likely need 2 “derived tables” or <i>in-line views</i> to make this query work. It is admittedly tricky!)</li><li>• Create a query that lists the stores, by month ranked by the most items sold.</li></ul> <p><b>Note:</b> When working with dates, the easiest method to convert a date into month/year is with the following syntax: <code>TO_CHAR(date, 'yyyy-mm')</code>.</p>
2.	Connect to the <code>datamart</code> database as <code>gpadmin</code> .

Step	Action																																												
3.	<p>The query for the first requirement makes use of the SUM moving window function.</p> <p>See the lab example in the, /home/gp/sql/load_files/adv_sql_files/lab8query1.sql file if you are having problems writing this SQL. The content is as follows:</p> <pre>datamart=# SELECT c.custname       ,t.transdate       ,t.SalesAmt       ,SUM(t.SalesAmt) OVER (PARTITION BY t.customerid                                 ORDER BY t.transdate ASC                                 ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) AS RollingTotalAmt    FROM facts.transaction t INNER JOIN dimensions.customer c   ON c.customerid = t.customerid;</pre>  <table border="1"> <thead> <tr> <th>custname</th> <th>transdate</th> <th>salesamt</th> <th>rollingtotalamt</th> </tr> </thead> <tbody> <tr><td>Garrison , Russ</td><td>2008-03-11</td><td>19.29</td><td>19.29</td></tr> <tr><td>Garrison , Russ</td><td>2008-03-16</td><td>32.23</td><td>51.52</td></tr> <tr><td>Garrison , Russ</td><td>2008-03-21</td><td>63.12</td><td>114.64</td></tr> <tr><td>Garrison , Russ</td><td>2008-03-28</td><td>79.54</td><td>194.18</td></tr> <tr><td>Garrison , Russ</td><td>2008-03-29</td><td>41.16</td><td>235.34</td></tr> <tr><td>Garrison , Russ</td><td>2008-04-04</td><td>88.04</td><td>323.38</td></tr> <tr><td>Garrison , Russ</td><td>2008-04-07</td><td>10.56</td><td>333.94</td></tr> <tr><td>Garrison , Russ</td><td>2008-04-09</td><td>53.98</td><td>387.92</td></tr> <tr><td>Garrison , Russ</td><td>2008-04-11</td><td>45.67</td><td>433.59</td></tr> <tr><td>Garrison , Russ</td><td>2008-04-14</td><td>66.81</td><td>500.40</td></tr> </tbody> </table>	custname	transdate	salesamt	rollingtotalamt	Garrison , Russ	2008-03-11	19.29	19.29	Garrison , Russ	2008-03-16	32.23	51.52	Garrison , Russ	2008-03-21	63.12	114.64	Garrison , Russ	2008-03-28	79.54	194.18	Garrison , Russ	2008-03-29	41.16	235.34	Garrison , Russ	2008-04-04	88.04	323.38	Garrison , Russ	2008-04-07	10.56	333.94	Garrison , Russ	2008-04-09	53.98	387.92	Garrison , Russ	2008-04-11	45.67	433.59	Garrison , Russ	2008-04-14	66.81	500.40
custname	transdate	salesamt	rollingtotalamt																																										
Garrison , Russ	2008-03-11	19.29	19.29																																										
Garrison , Russ	2008-03-16	32.23	51.52																																										
Garrison , Russ	2008-03-21	63.12	114.64																																										
Garrison , Russ	2008-03-28	79.54	194.18																																										
Garrison , Russ	2008-03-29	41.16	235.34																																										
Garrison , Russ	2008-04-04	88.04	323.38																																										
Garrison , Russ	2008-04-07	10.56	333.94																																										
Garrison , Russ	2008-04-09	53.98	387.92																																										
Garrison , Russ	2008-04-11	45.67	433.59																																										
Garrison , Russ	2008-04-14	66.81	500.40																																										

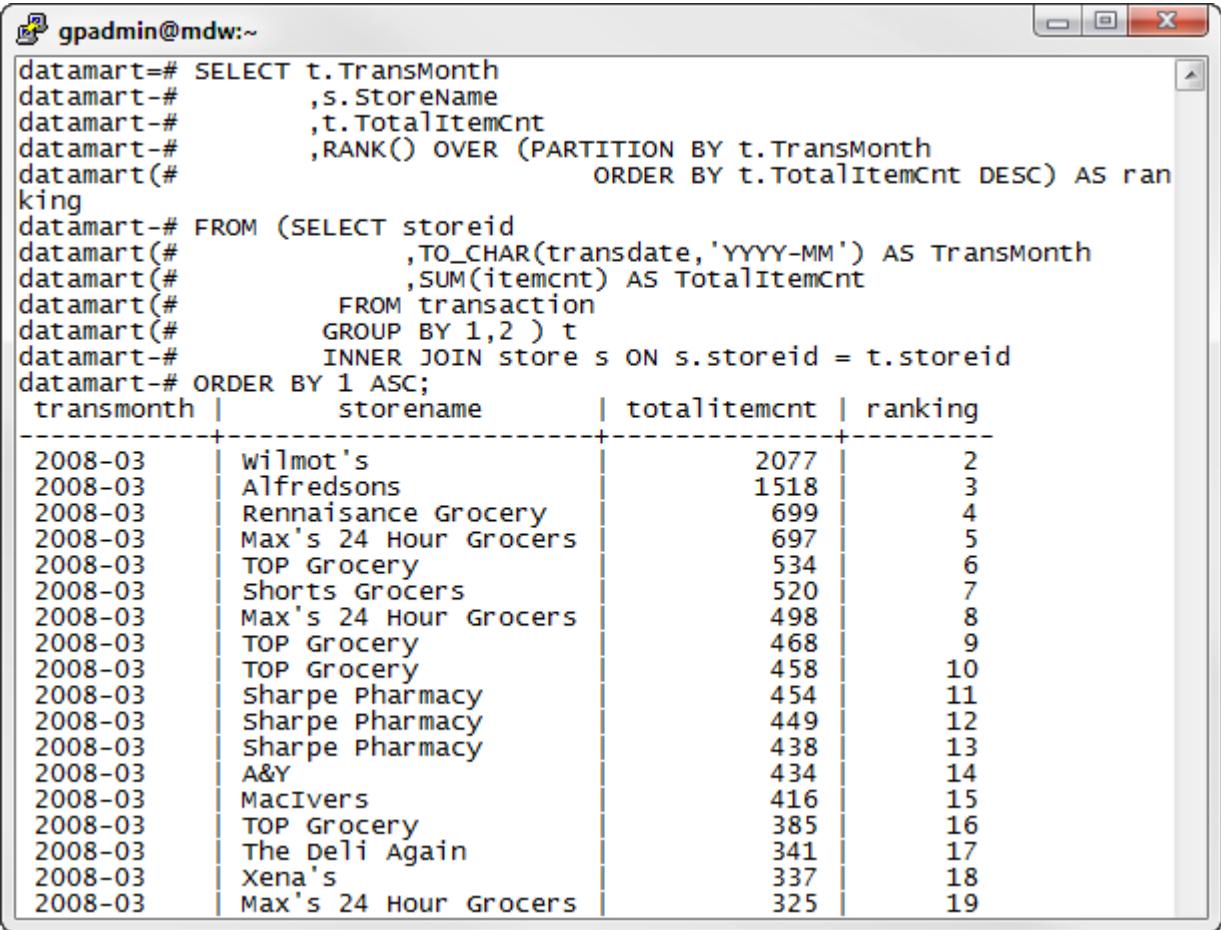
Step	Action
4.	<p>The query for the second requirement works really well with the CUBE group function. You can use the ORDER BY clause of the SQL statement to display the results in a more readable format.</p> <p>See the lab example in the /home/gp/sql/load_files/adv_sql_files/lab8query2.sql file if you need assistance.</p> <pre>datamart=# SELECT s.storename            ,to_char(t.transdate,'YYYY-MM') AS TransMonth            ,SUM(t.SalesAmt) AS MonthlySalesAmt        FROM facts.transaction t       INNER JOIN dimensions.store s       ON s.storeid = t.storeid  WHERE s.grocery = true    AND s.deli = true  GROUP BY CUBE(s.storename,TransMonth) ORDER BY 1,2;</pre>  <pre>gpadmin@mdw:~\$ datamart=# SELECT s.storename datamart-#           ,to_char(t.transdate,'YYYY-MM') AS TransMonth datamart-#           ,SUM(t.SalesAmt) AS MonthlySalesAmt datamart-# datamart-#           FROM facts.transaction t datamart-#           INNER JOIN dimensions.store s datamart-#           ON s.storeid = t.storeid datamart-# WHERE s.grocery = true datamart-#   AND s.deli = true datamart-# datamart-# GROUP BY CUBE(s.storename,TransMonth) datamart-# ORDER BY 1,2;           storename   transmonth   monthlysalesamt -----+-----+----- Drugs and Stuff   2008-03           35.58 Drugs and Stuff   2008-05           46.41 Drugs and Stuff   2008-06           29.41 Drugs and Stuff   2008-07           12.18 Drugs and Stuff              123.58 Gringo Pete's     2008-03           38.13 Gringo Pete's     2008-04          175.51 Gringo Pete's     2008-05           94.83 Gringo Pete's     2008-06           68.24 Gringo Pete's     2008-08           11.93 Gringo Pete's                388.64 Sharpe Pharmacy   2008-03      1374.32</pre>

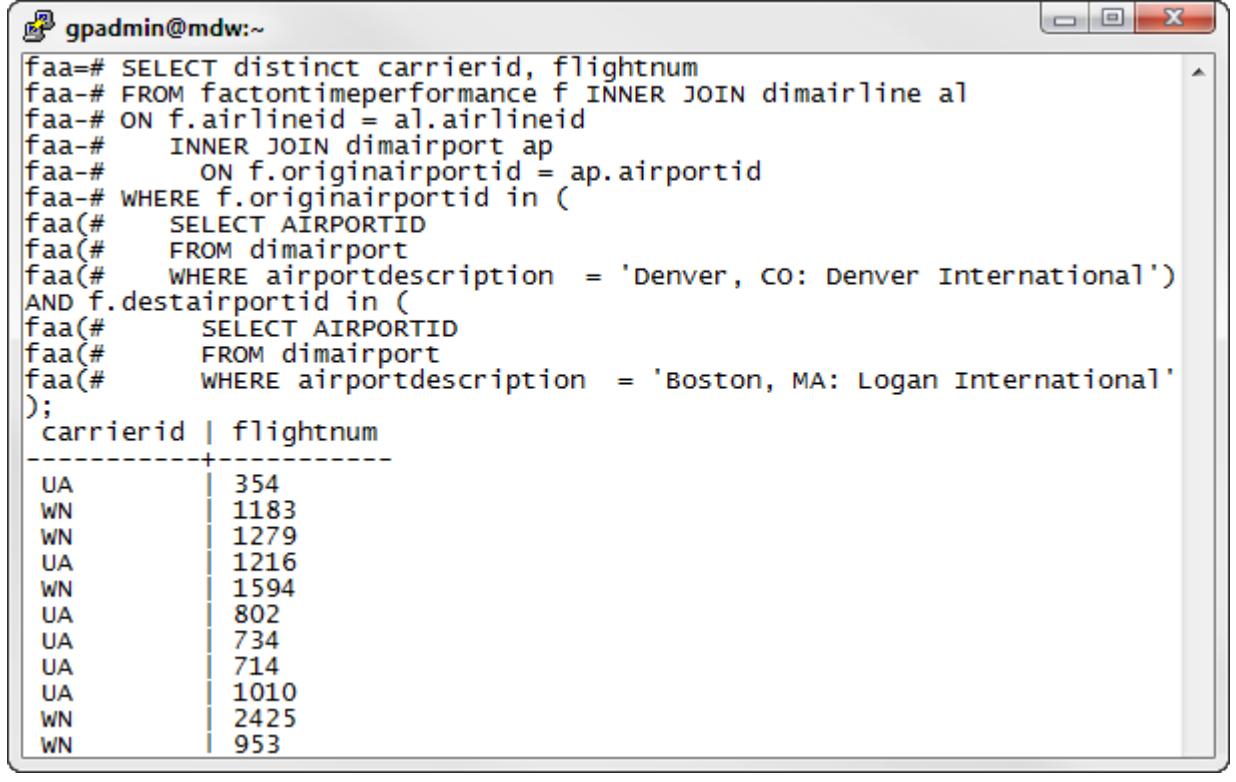
Step	Action
5.	<p>The query for the third requirement is tricky.</p> <p>Consider using a derived table or an in-line view to do a summary by month and store of each of the amount columns. Then apply the SUM window function to each of those columns in your table.</p> <p>See the lab example in the  <code>/home/gp/sql/load_files/adv_sql_files/lab8query3.sql</code> file if you need a hint.</p> <pre> datamart=# SELECT s.storename       ,t.TransMonth       ,SUM(t.salesamt) OVER (PARTITION BY s.storename ORDER BY t.TransMonth) AS TotalSalesAmt       ,SUM(t.taxamt) OVER (PARTITION BY s.storename ORDER BY t.TransMonth) AS TotalTaxAmt       ,SUM(t.discountamt) OVER (PARTITION BY s.storename ORDER BY t.TransMonth) AS TotalDiscountAmt       ,SUM(t.couponamt) OVER (PARTITION BY s.storename ORDER BY t.TransMonth) AS TotalCouponAmt       ,SUM(t.cashamt) OVER (PARTITION BY s.storename ORDER BY t.TransMonth) AS TotalCashAmt       ,SUM(t.checkamt) OVER (PARTITION BY s.storename ORDER BY t.TransMonth) AS TotalCheckAmt       ,SUM(t.ccamt) OVER (PARTITION BY s.storename ORDER BY t.TransMonth) AS TotalCCAmt       ,SUM(t.debitamt) OVER (PARTITION BY s.storename ORDER BY t.TransMonth) AS TotalDebitAmt       ,SUM(t.otheramt) OVER (PARTITION BY s.storename ORDER BY t.TransMonth) AS TotalOtherAmt  FROM (SELECT storeid       ,TO_CHAR(transdate,'YYYY-MM') AS TransMonth       ,SUM(salesamt) AS SalesAmt       ,SUM(taxamt) AS TaxAmt       ,SUM(discountamt) AS DiscountAmt       ,SUM(couponamt) AS CouponAmt       ,SUM(cashamt) AS CashAmt       ,SUM(checkamt) AS CheckAmt       ,SUM(ccamt) AS CCAmt       ,SUM(debitamt) AS DebitAmt       ,SUM(otheramt) AS OtherAmt       FROM facts.transaction       GROUP BY 1,2) t       INNER JOIN dimensions.store s       ON s.storeid = t.storeid; </pre>

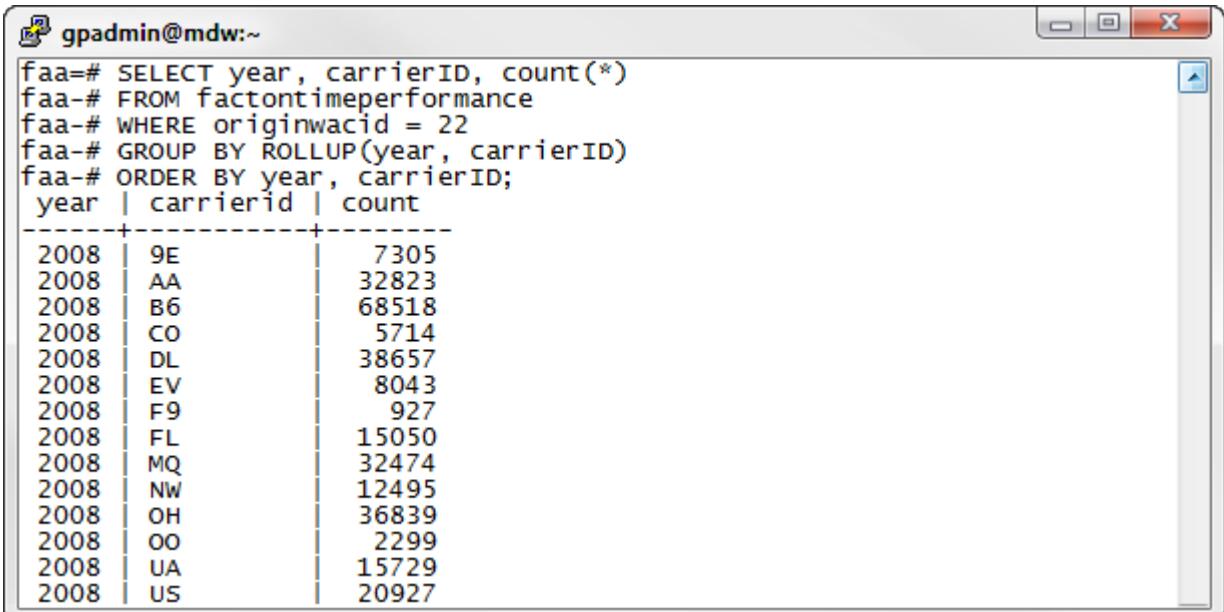
Step	Action
	<pre>gpadmin@mdw:~\$ datamart=# SELECT s.storename datamart-# ,t.TransMonth datamart-# , SUM(t.salesamt) OVER (PARTITION BY s.storename ORDER BY t.TransMonth) AS TotalSalesAmt datamart-# , SUM(t.taxamt) OVER (PARTITION BY s.storename ORDER B Y t.TransMonth) AS TotalTaxAmt datamart-# , SUM(t.discountamt) OVER (PARTITION BY s.storename OR DER BY t.TransMonth) AS TotalDiscountAmt datamart-# , SUM(t.couponamt) OVER (PARTITION BY s.storename ORDE R BY t.TransMonth) AS TotalCouponAmt datamart-# , SUM(t.cashamt) OVER (PARTITION BY s.storename ORDER BY t.TransMonth) AS TotalCashAmt datamart-# , SUM(t.checkamt) OVER (PARTITION BY s.storename ORDER BY t.TransMonth) AS TotalCheckAmt datamart-# , SUM(t.ccamt) OVER (PARTITION BY s.storename ORDER BY t.TransMonth) AS TotalCCAmt datamart-# , SUM(t.debitamt) OVER (PARTITION BY s.storename ORDER BY t.TransMonth) AS TotalDebitAmt datamart-# , SUM(t.otheramt) OVER (PARTITION BY s.storename ORDER BY t.TransMonth) AS TotalOtherAmt datamart-# datamart-# FROM (SELECT storeid datamart(# ,TO_CHAR(transdate,'YYYY-MM') AS TransMonth datamart(# , SUM(salesamt) AS SalesAmt datamart(# , SUM(taxamt) AS TaxAmt datamart(# , SUM(discountamt) AS DiscountAmt datamart(# , SUM(couponamt) AS CouponAmt datamart(# , SUM(cashamt) AS CashAmt datamart(# , SUM(checkamt) AS CheckAmt datamart(# , SUM(ccamt) AS CCAmt datamart(# , SUM(debitamt) AS DebitAmt datamart(# , SUM(otheramt) AS OtherAmt datamart(#     FROM facts.transaction datamart(#     GROUP BY 1,2) t datamart-#     INNER JOIN dimensions.store s datamart-#     ON s.storeid = t.storeid; storename   transmonth   totalsalesamt   totaltaxamt   tota ldiscountamt   totalcouponamt   totalcashamt   totalcheckamt   totalccam t   totaldebitamt   totalotheramt -----+-----+-----+-----+-----+-----+-----+ Alfredsons   2008-03   3795.71   299.80   124.02   632.5   12.38   17.95   1365.86  </pre>

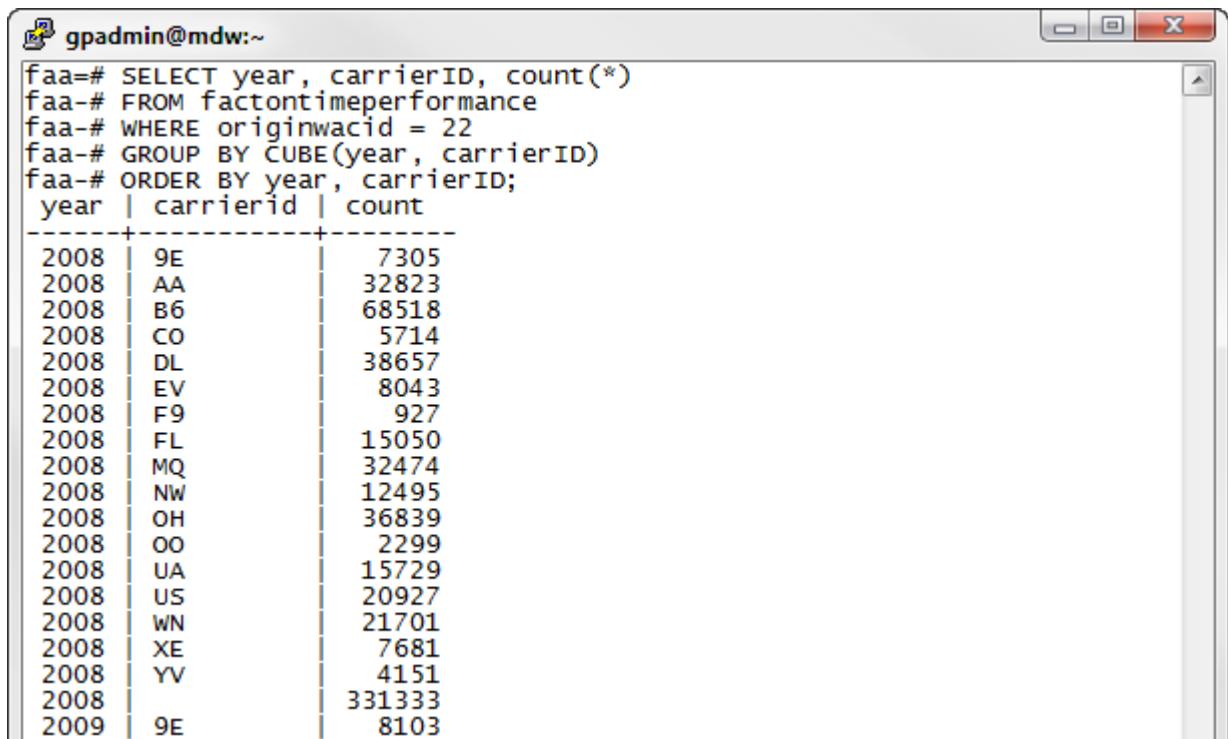
Step	Action
6.	<p>The query for the fourth requirement is harder. You should consider a derived table or in-line view. However, you will need two:</p> <ul style="list-style-type: none"> <li>• The first one does the summarization at your control breaks.</li> <li>• The second gets the ranking for you and then executes the final statement against the last derived table to get the results.</li> </ul> <p>See the lab example in the /home/gp/sql/load_files/adv_sql_files/lab8query4.sql file for details and hints. The contents of the file are displayed below:</p> <pre> datamart=# SELECT Y.TransMonth     ,Y.custname AS CustomerName     ,Y.TotalSalesAmt FROM (SELECT X.TransMonth     ,c.custname     ,X.TotalSalesAmt     ,RANK() OVER (PARTITION BY X.TransMonth                   ORDER BY X.TotalSalesAmt DESC) AS CustomerRanking      FROM (SELECT TO_CHAR(t.transdate,'YYYY-MM') AS TransMonth         ,t.customerid         ,SUM(t.salesamt) AS TotalSalesAmt         FROM facts.transaction t         GROUP BY 1,2     ) X     INNER JOIN dimensions.customer c     ON c.customerid = X.customerid ) Y WHERE Y.CustomerRanking = 1 ORDER BY 1 ASC; </pre>

Step	Action
	<pre> gpadmin@mdw:~ datamart=# SELECT Y.TransMonth datamart-#           ,Y.custname AS CustomerName datamart-#           ,Y.TotalsalesAmt datamart-# FROM (SELECT X.TransMonth datamart(#           ,c.custname datamart(#           ,X.TotalsalesAmt datamart(#           ,RANK() OVER (PARTITION BY X.TransMonth datamart(#                           ORDER BY X.TotalsalesAmt DESC) AS CustomerRanking datamart(#  datamart(#     FROM (SELECT TO_CHAR(t.transdate,'YYYY-MM') AS Trans Month datamart(#           ,t.customerid datamart(#           ,SUM(t.salesamt) AS TotalsalesAmt datamart(#           FROM facts.transaction t datamart(#           GROUP BY 1,2 datamart(#           ) X datamart(#           INNER JOIN dimensions.customer c datamart(#           ON c.customerid = X.customerid datamart(#           ) Y datamart-# WHERE Y.CustomerRanking = 1 datamart-# ORDER BY 1 ASC; transmonth   customername   totalsalesamt -----+-----+----- 2008-03   Beardsley , Keith   586.73 2008-04   Jones , Matthew   621.78 2008-05   McGrath , Martin   678.06 2008-06   Begin , Robin   768.86 2008-07   Fuller , Cathey   700.77 2008-08   Bender , John   612.43 (6 rows)  datamart=# </pre>

Step	Action																																																																												
7.	<p>The query for the last requirement is similar to the query example in the lesson. Consider using a window partition to solve this query.</p> <p>See the lab example in the /home/gp/sql/load_files/adv_sql_files/lab8query5.sql file for hints. The content is as follows:</p> <pre>datamart=# SELECT t.TransMonth     ,s.StoreName     ,t.TotalItemCnt     ,RANK() OVER (PARTITION BY t.TransMonth                   ORDER BY t.TotalItemCnt DESC) AS ranking FROM (SELECT storeid         ,TO_CHAR(transdate,'YYYY-MM') AS TransMonth         ,SUM(itemcnt) AS TotalItemCnt       FROM transaction      GROUP BY 1,2 ) t   INNER JOIN store s ON s.storeid = t.storeid ORDER BY 1 ASC;</pre>  <p>The screenshot shows a terminal window titled "gpadmin@mdw:~". The command entered is the same as the one above, followed by a copy-paste of the query results. The results are displayed in a table with four columns: transmonth, storename, totalitemcnt, and ranking. The data shows various stores across different months, ranked by their total item count.</p> <table border="1"> <thead> <tr> <th>transmonth</th> <th>storename</th> <th>totalitemcnt</th> <th>ranking</th> </tr> </thead> <tbody> <tr><td>2008-03</td><td>wilmot's</td><td>2077</td><td>2</td></tr> <tr><td>2008-03</td><td>Alfredsons</td><td>1518</td><td>3</td></tr> <tr><td>2008-03</td><td>Rennaisance Grocery</td><td>699</td><td>4</td></tr> <tr><td>2008-03</td><td>Max's 24 Hour Grocers</td><td>697</td><td>5</td></tr> <tr><td>2008-03</td><td>TOP Grocery</td><td>534</td><td>6</td></tr> <tr><td>2008-03</td><td>Shorts Grocers</td><td>520</td><td>7</td></tr> <tr><td>2008-03</td><td>Max's 24 Hour Grocers</td><td>498</td><td>8</td></tr> <tr><td>2008-03</td><td>TOP Grocery</td><td>468</td><td>9</td></tr> <tr><td>2008-03</td><td>TOP Grocery</td><td>458</td><td>10</td></tr> <tr><td>2008-03</td><td>Sharpe Pharmacy</td><td>454</td><td>11</td></tr> <tr><td>2008-03</td><td>Sharpe Pharmacy</td><td>449</td><td>12</td></tr> <tr><td>2008-03</td><td>Sharpe Pharmacy</td><td>438</td><td>13</td></tr> <tr><td>2008-03</td><td>A&amp;Y</td><td>434</td><td>14</td></tr> <tr><td>2008-03</td><td>MacIvers</td><td>416</td><td>15</td></tr> <tr><td>2008-03</td><td>TOP Grocery</td><td>385</td><td>16</td></tr> <tr><td>2008-03</td><td>The Deli Again</td><td>341</td><td>17</td></tr> <tr><td>2008-03</td><td>Xena's</td><td>337</td><td>18</td></tr> <tr><td>2008-03</td><td>Max's 24 Hour Grocers</td><td>325</td><td>19</td></tr> </tbody> </table>	transmonth	storename	totalitemcnt	ranking	2008-03	wilmot's	2077	2	2008-03	Alfredsons	1518	3	2008-03	Rennaisance Grocery	699	4	2008-03	Max's 24 Hour Grocers	697	5	2008-03	TOP Grocery	534	6	2008-03	Shorts Grocers	520	7	2008-03	Max's 24 Hour Grocers	498	8	2008-03	TOP Grocery	468	9	2008-03	TOP Grocery	458	10	2008-03	Sharpe Pharmacy	454	11	2008-03	Sharpe Pharmacy	449	12	2008-03	Sharpe Pharmacy	438	13	2008-03	A&Y	434	14	2008-03	MacIvers	416	15	2008-03	TOP Grocery	385	16	2008-03	The Deli Again	341	17	2008-03	Xena's	337	18	2008-03	Max's 24 Hour Grocers	325	19
transmonth	storename	totalitemcnt	ranking																																																																										
2008-03	wilmot's	2077	2																																																																										
2008-03	Alfredsons	1518	3																																																																										
2008-03	Rennaisance Grocery	699	4																																																																										
2008-03	Max's 24 Hour Grocers	697	5																																																																										
2008-03	TOP Grocery	534	6																																																																										
2008-03	Shorts Grocers	520	7																																																																										
2008-03	Max's 24 Hour Grocers	498	8																																																																										
2008-03	TOP Grocery	468	9																																																																										
2008-03	TOP Grocery	458	10																																																																										
2008-03	Sharpe Pharmacy	454	11																																																																										
2008-03	Sharpe Pharmacy	449	12																																																																										
2008-03	Sharpe Pharmacy	438	13																																																																										
2008-03	A&Y	434	14																																																																										
2008-03	MacIvers	416	15																																																																										
2008-03	TOP Grocery	385	16																																																																										
2008-03	The Deli Again	341	17																																																																										
2008-03	Xena's	337	18																																																																										
2008-03	Max's 24 Hour Grocers	325	19																																																																										
8.	<p>Connect to the faa database to execute queries against larger data sets.</p> <pre>datamart=# \c faa</pre>																																																																												

Step	Action
9.	<p>The following query lists flights from Denver International airport to Logan International (Boston) airport:</p> <pre> faa=# SELECT distinct carrierid, flightnum FROM factontimeperformance f INNER JOIN dimairline al ON f.airlineid = al.airlineid INNER JOIN dimairport ap     ON f.originairportid = ap.airportid WHERE f.originairportid in (     SELECT AIRPORTID     FROM dimairport     WHERE airportdescription = 'Denver, CO: Denver International')AND f.destairportid in (     SELECT AIRPORTID     FROM dimairport     WHERE airportdescription = 'Boston, MA: Logan International'); </pre>  <pre> gpadmin@mdw:~ faa=# SELECT distinct carrierid, flightnum faa-# FROM factontimeperformance f INNER JOIN dimairline al faa-# ON f.airlineid = al.airlineid faa-# INNER JOIN dimairport ap faa-#     ON f.originairportid = ap.airportid faa-# WHERE f.originairportid in ( faa(#     SELECT AIRPORTID faa(#     FROM dimairport faa(#     WHERE airportdescription = 'Denver, CO: Denver International') AND f.destairportid in ( faa(#         SELECT AIRPORTID faa(#         FROM dimairport faa(#         WHERE airportdescription = 'Boston, MA: Logan International' ); carrierid   flightnum -----+----- UA          354 WN          1183 WN          1279 UA          1216 WN          1594 UA          802 UA          734 UA          714 UA          1010 WN          2425 WN          953 </pre>

Step	Action																																													
10.	<p>List the number of flights for each airline carrier leaving New York (originwacid=22). Provide the information for each carrier per year. Display the grand total number of flights in the year.</p> <p>Use the ROLLUP operator to display all the results at once:</p> <pre>faa=# SELECT year, carrierID, count(*) FROM factontimeperformance WHERE originwacid = 22 GROUP BY ROLLUP(year, carrierID) ORDER BY year, carrierID;</pre>  <table border="1"> <thead> <tr> <th>year</th> <th>carrierid</th> <th>count</th> </tr> </thead> <tbody> <tr><td>2008</td><td>9E</td><td>7305</td></tr> <tr><td>2008</td><td>AA</td><td>32823</td></tr> <tr><td>2008</td><td>B6</td><td>68518</td></tr> <tr><td>2008</td><td>CO</td><td>5714</td></tr> <tr><td>2008</td><td>DL</td><td>38657</td></tr> <tr><td>2008</td><td>EV</td><td>8043</td></tr> <tr><td>2008</td><td>F9</td><td>927</td></tr> <tr><td>2008</td><td>FL</td><td>15050</td></tr> <tr><td>2008</td><td>MQ</td><td>32474</td></tr> <tr><td>2008</td><td>NW</td><td>12495</td></tr> <tr><td>2008</td><td>OH</td><td>36839</td></tr> <tr><td>2008</td><td>OO</td><td>2299</td></tr> <tr><td>2008</td><td>UA</td><td>15729</td></tr> <tr><td>2008</td><td>US</td><td>20927</td></tr> </tbody> </table>	year	carrierid	count	2008	9E	7305	2008	AA	32823	2008	B6	68518	2008	CO	5714	2008	DL	38657	2008	EV	8043	2008	F9	927	2008	FL	15050	2008	MQ	32474	2008	NW	12495	2008	OH	36839	2008	OO	2299	2008	UA	15729	2008	US	20927
year	carrierid	count																																												
2008	9E	7305																																												
2008	AA	32823																																												
2008	B6	68518																																												
2008	CO	5714																																												
2008	DL	38657																																												
2008	EV	8043																																												
2008	F9	927																																												
2008	FL	15050																																												
2008	MQ	32474																																												
2008	NW	12495																																												
2008	OH	36839																																												
2008	OO	2299																																												
2008	UA	15729																																												
2008	US	20927																																												

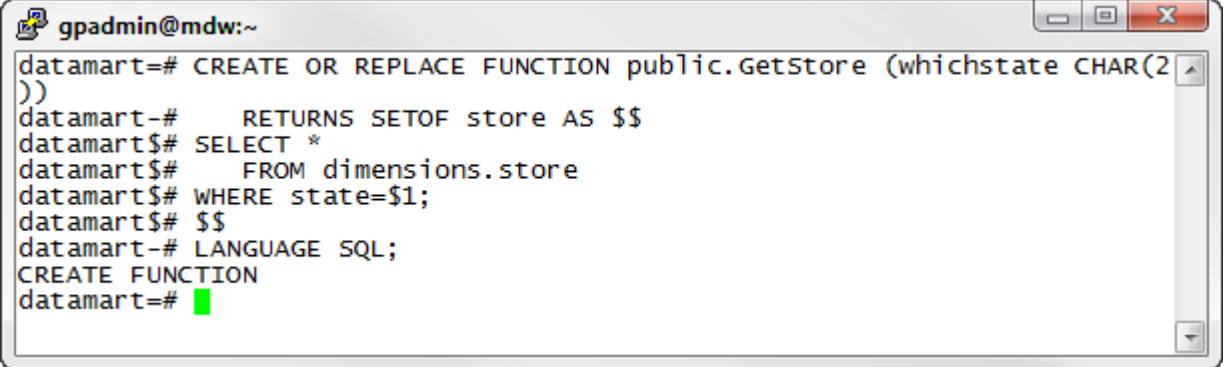
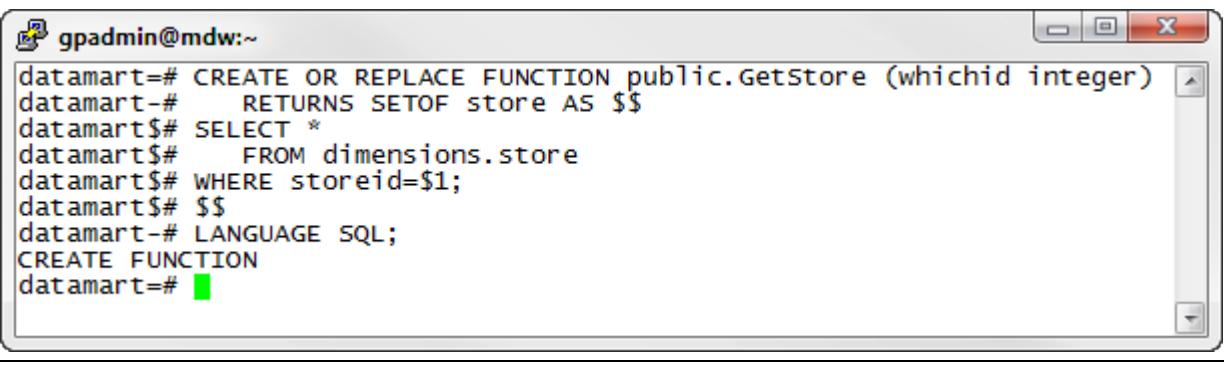
Step	Action																																																												
11.	<p>List the number of flights leaving New York on a carrier and year basis. Display the total number of flights in the year and the total number of flights by a carrier. Use the cube operator to display all the results at once:</p> <pre>faa=# SELECT year, carrierID, count(*) FROM factontimeperformance WHERE originwacid = 22 GROUP BY CUBE(year, carrierID) ORDER BY year, carrierID;</pre>  <table border="1"> <thead> <tr> <th>year</th> <th>carrierid</th> <th>count</th> </tr> </thead> <tbody> <tr><td>2008</td><td>9E</td><td>7305</td></tr> <tr><td>2008</td><td>AA</td><td>32823</td></tr> <tr><td>2008</td><td>B6</td><td>68518</td></tr> <tr><td>2008</td><td>CO</td><td>5714</td></tr> <tr><td>2008</td><td>DL</td><td>38657</td></tr> <tr><td>2008</td><td>EV</td><td>8043</td></tr> <tr><td>2008</td><td>F9</td><td>927</td></tr> <tr><td>2008</td><td>FL</td><td>15050</td></tr> <tr><td>2008</td><td>MQ</td><td>32474</td></tr> <tr><td>2008</td><td>NW</td><td>12495</td></tr> <tr><td>2008</td><td>OH</td><td>36839</td></tr> <tr><td>2008</td><td>OO</td><td>2299</td></tr> <tr><td>2008</td><td>UA</td><td>15729</td></tr> <tr><td>2008</td><td>US</td><td>20927</td></tr> <tr><td>2008</td><td>WN</td><td>21701</td></tr> <tr><td>2008</td><td>XE</td><td>7681</td></tr> <tr><td>2008</td><td>YV</td><td>4151</td></tr> <tr><td>2008</td><td></td><td>331333</td></tr> <tr><td>2009</td><td>9E</td><td>8103</td></tr> </tbody> </table>	year	carrierid	count	2008	9E	7305	2008	AA	32823	2008	B6	68518	2008	CO	5714	2008	DL	38657	2008	EV	8043	2008	F9	927	2008	FL	15050	2008	MQ	32474	2008	NW	12495	2008	OH	36839	2008	OO	2299	2008	UA	15729	2008	US	20927	2008	WN	21701	2008	XE	7681	2008	YV	4151	2008		331333	2009	9E	8103
year	carrierid	count																																																											
2008	9E	7305																																																											
2008	AA	32823																																																											
2008	B6	68518																																																											
2008	CO	5714																																																											
2008	DL	38657																																																											
2008	EV	8043																																																											
2008	F9	927																																																											
2008	FL	15050																																																											
2008	MQ	32474																																																											
2008	NW	12495																																																											
2008	OH	36839																																																											
2008	OO	2299																																																											
2008	UA	15729																																																											
2008	US	20927																																																											
2008	WN	21701																																																											
2008	XE	7681																																																											
2008	YV	4151																																																											
2008		331333																																																											
2009	9E	8103																																																											
	<p><b>Summary</b></p> <p>Use the OLAP grouping functionality whenever you need to display totals and sub-totals based on your group predicates.</p> <p>Use the OLAP window expressions whenever you want to utilize the values in prior rows for comparison or further aggregation. This precludes having to make multiple scans of the same table in order to accomplish these comparisons.</p>																																																												

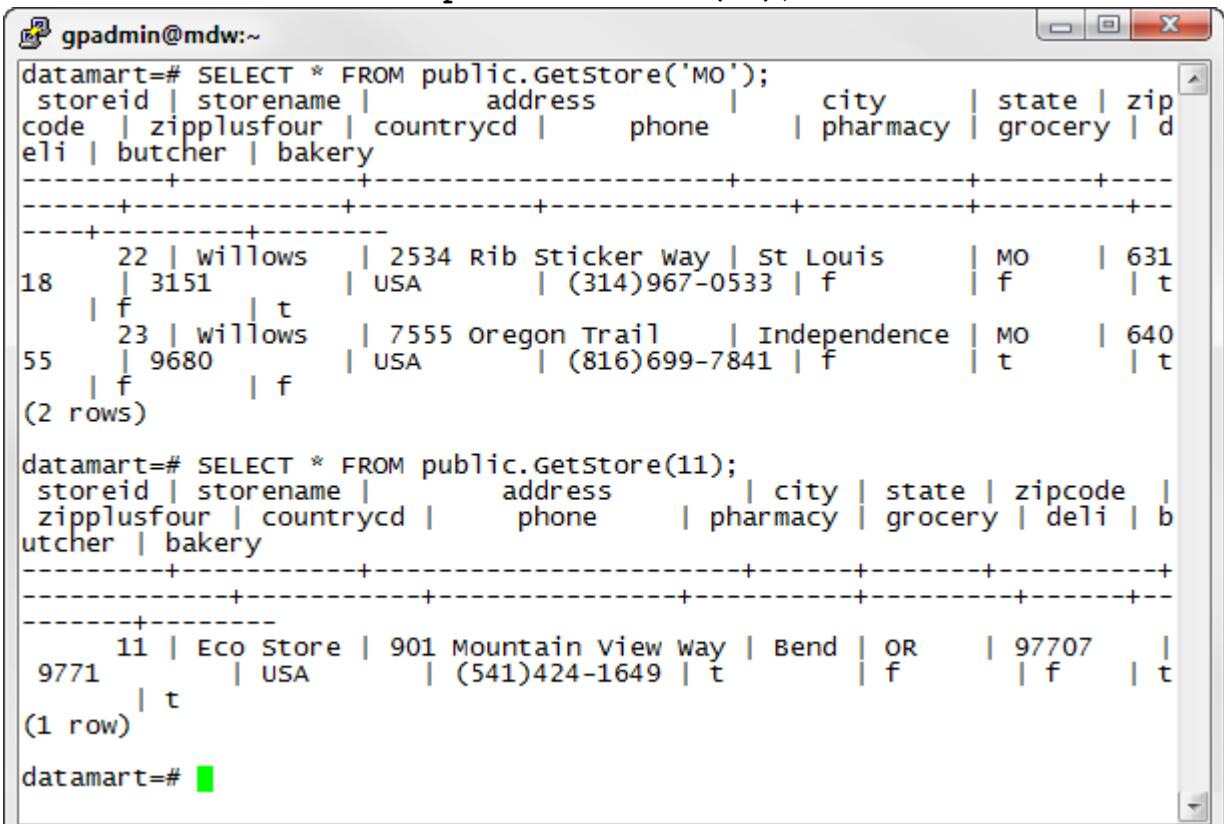
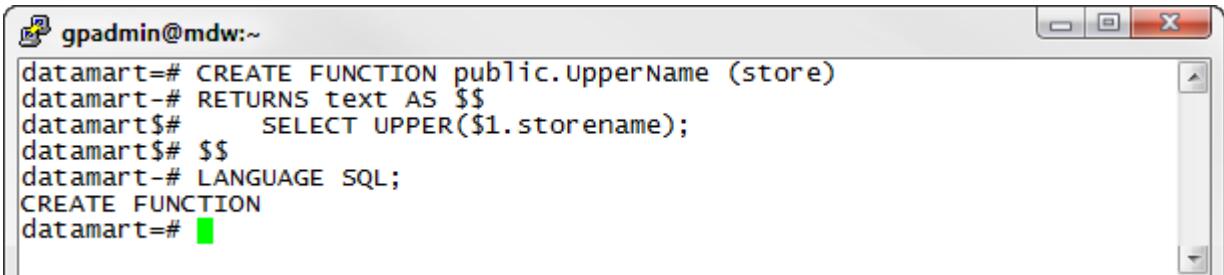
End of Lab Exercise

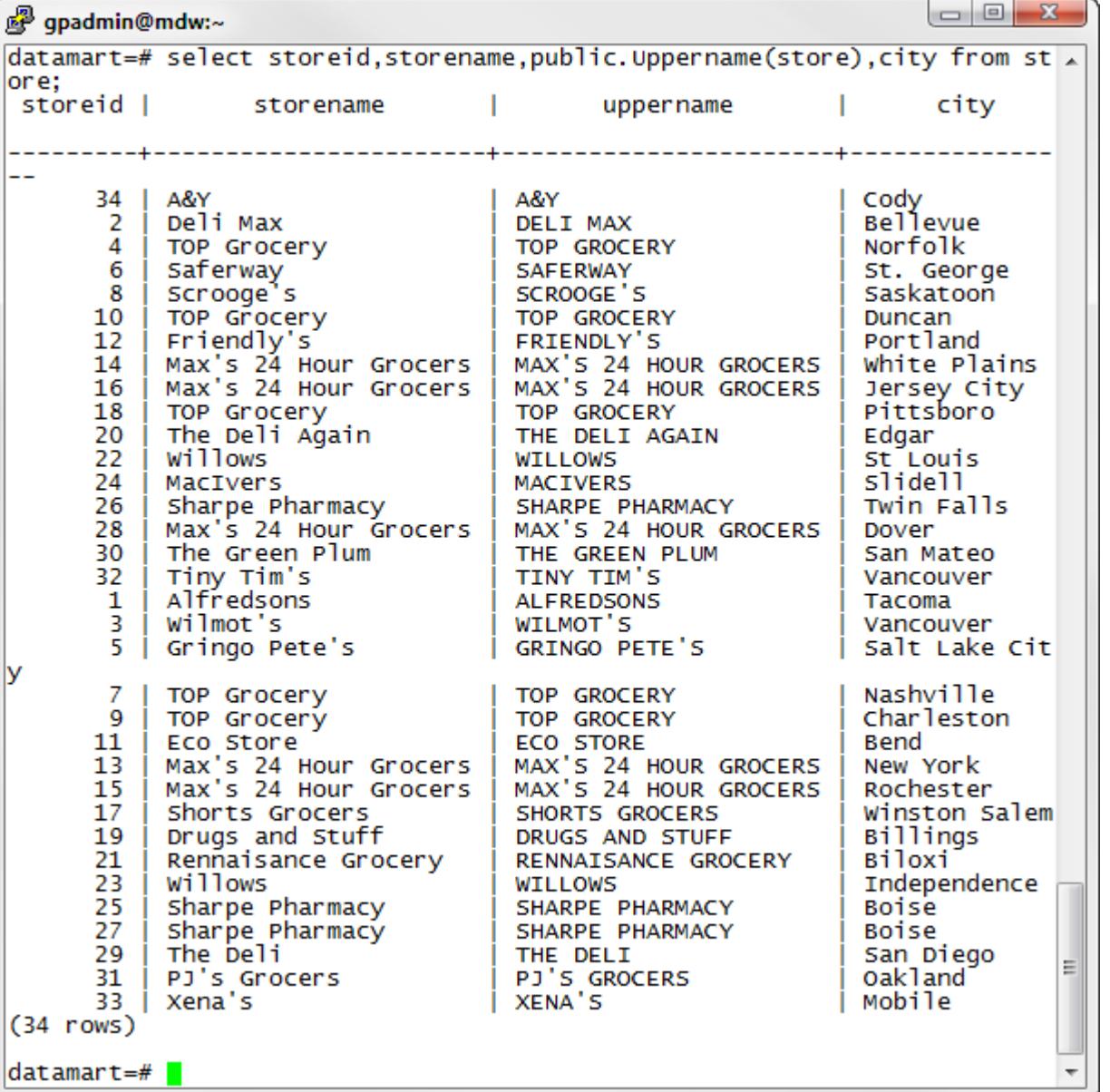
## Lab 23. PostgreSQL Functions

<b>Purpose:</b>	In this lab, you will create functions to perform simple, repeatable queries and tasks. You will create functions that use the SQL and the PL/pgSQL language. You will be required to create one user defined data type in order to correctly return data.
<b>Tasks:</b>	Students perform the following tasks: <ul style="list-style-type: none"><li>• Create SQL functions</li><li>• Create PL/pgSQL functions</li></ul>
<b>References:</b>	Module 9 – Developing Reports Using Advanced SQL <ul style="list-style-type: none"><li>• Lesson: PostgreSQL Functions</li></ul>

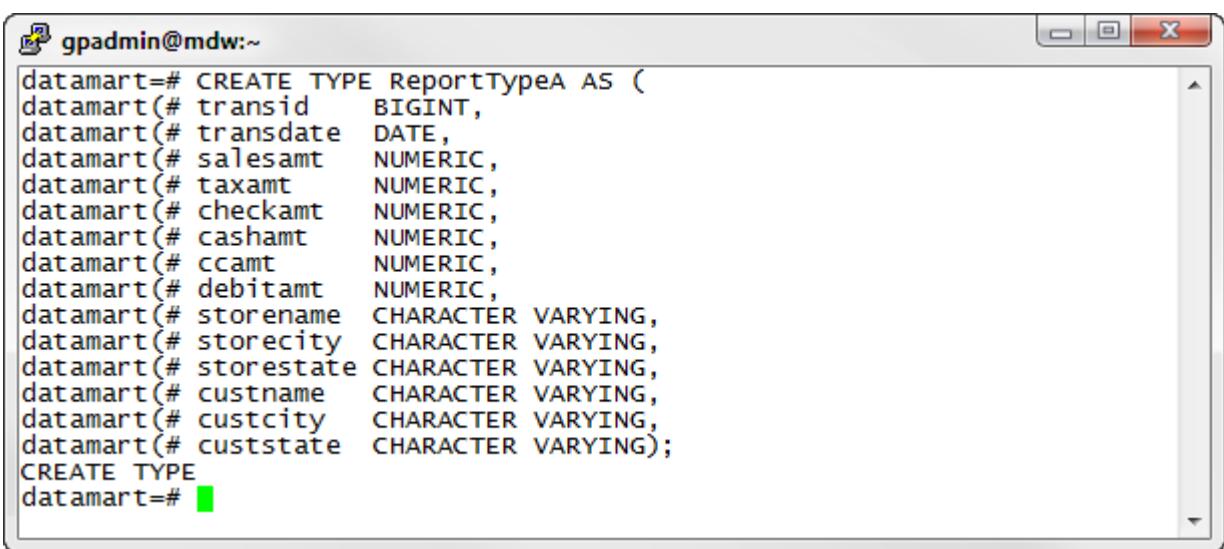
## Lab 23: Task 1 – Create SQL Functions

Step	Action
1.	<p>Connect to the datamart database as gpadmin.</p>
2.	<p>Create an overloaded SQL function that returns a row or rows from the store dimension table with a parameter of either the storeid or the state. You can create the procedure in the public schema.</p> <p>For hints on how to create the function, refer to the lab example in the /home/gp/sql/load_files/adv_sql_files/lab10query1.sql file.</p> <p>Remember that you want to return a SETOF to get more than one row.</p> <pre>datamart=# CREATE OR REPLACE FUNCTION public.GetStore (whichstate CHAR(2))     RETURNS SETOF store AS \$\$%         SELECT *         FROM dimensions.store     WHERE state=\$1; \$\$% LANGUAGE SQL;</pre>  <pre>gpadmin@mdw:~ datamart=# CREATE OR REPLACE FUNCTION public.GetStore (whichstate CHAR(2)) ) datamart-#     RETURNS SETOF store AS \$\$% datamart\$#     SELECT * datamart\$#         FROM dimensions.store datamart\$# WHERE state=\$1; datamart\$# \$\$% datamart-# LANGUAGE SQL; CREATE FUNCTION datamart=# </pre> <p>datamart=# CREATE OR REPLACE FUNCTION public.GetStore (whichid integer) integer)     RETURNS SETOF store AS \$\$% SELECT *     FROM dimensions.store WHERE storeid=\$1; \$\$% LANGUAGE SQL;</p>  <pre>gpadmin@mdw:~ datamart=# CREATE OR REPLACE FUNCTION public.GetStore (whichid integer) datamart-#     RETURNS SETOF store AS \$\$% datamart\$#     SELECT * datamart\$#         FROM dimensions.store datamart\$# WHERE storeid=\$1; datamart\$# \$\$% datamart-# LANGUAGE SQL; CREATE FUNCTION datamart=# </pre>

Step	Action
3.	<p>Test your function by selecting for the state of Missouri (MO) or any store id less than 30:</p> <pre>datamart=# SELECT * FROM public.GetStore('MO'); datamart=# SELECT * FROM public.GetStore(11);</pre>  <pre>gpadmin@mdw:~\$ datamart=# SELECT * FROM public.GetStore('MO'); storeid   storename   address   city   state   zip code   zipplusfour   countrycd   phone   pharmacy   grocery   deli   b eli   butcher   bakery -----+-----+-----+-----+-----+-----+-----+-----+ 18      22   willows   2534 Rib Sticker Way   St Louis   MO   631         3151              USA   (314)967-0533   f           f     t         55      23   willows   7555 Oregon Trail   Independence   MO   640         9680              USA   (816)699-7841   f           f     f         (2 rows)  datamart=# SELECT * FROM public.GetStore(11); storeid   storename   address   city   state   zipcode zipplusfour   countrycd   phone   pharmacy   grocery   deli   b utcher   bakery -----+-----+-----+-----+-----+-----+-----+-----+ 9771    11   Eco Store   901 Mountain View Way   Bend   OR   97707         USA              (541)424-1649   t           t     (1 row)  datamart=# </pre>
4.	<p>Create an SQL function that takes as a parameter a table name (you can use any of the dimension tables) and returns the appropriate name column in uppercase. Call the function <code>uppercase</code>. Create it in the <code>public</code> schema.</p> <p>For hints on how to create the function, refer to the lab example in the <code>/home/gp/sql/load_files/adv_sql_files/lab10query2.sql</code> file.</p> <pre>datamart=# CREATE FUNCTION public.UpperName (store) RETURNS text AS \$\$</pre> <ol style="list-style-type: none"> <li>SELECT UPPER(\$1.storename);</li> <li>\$\$</li> <li>LANGUAGE SQL;</li> </ol>  <pre>gpadmin@mdw:~\$ datamart=# CREATE FUNCTION public.UpperName (store) datamart-# RETURNS text AS \$\$ datamart\$#     SELECT UPPER(\$1.storename); datamart\$# \$\$ datamart-# LANGUAGE SQL; CREATE FUNCTION datamart=# </pre>

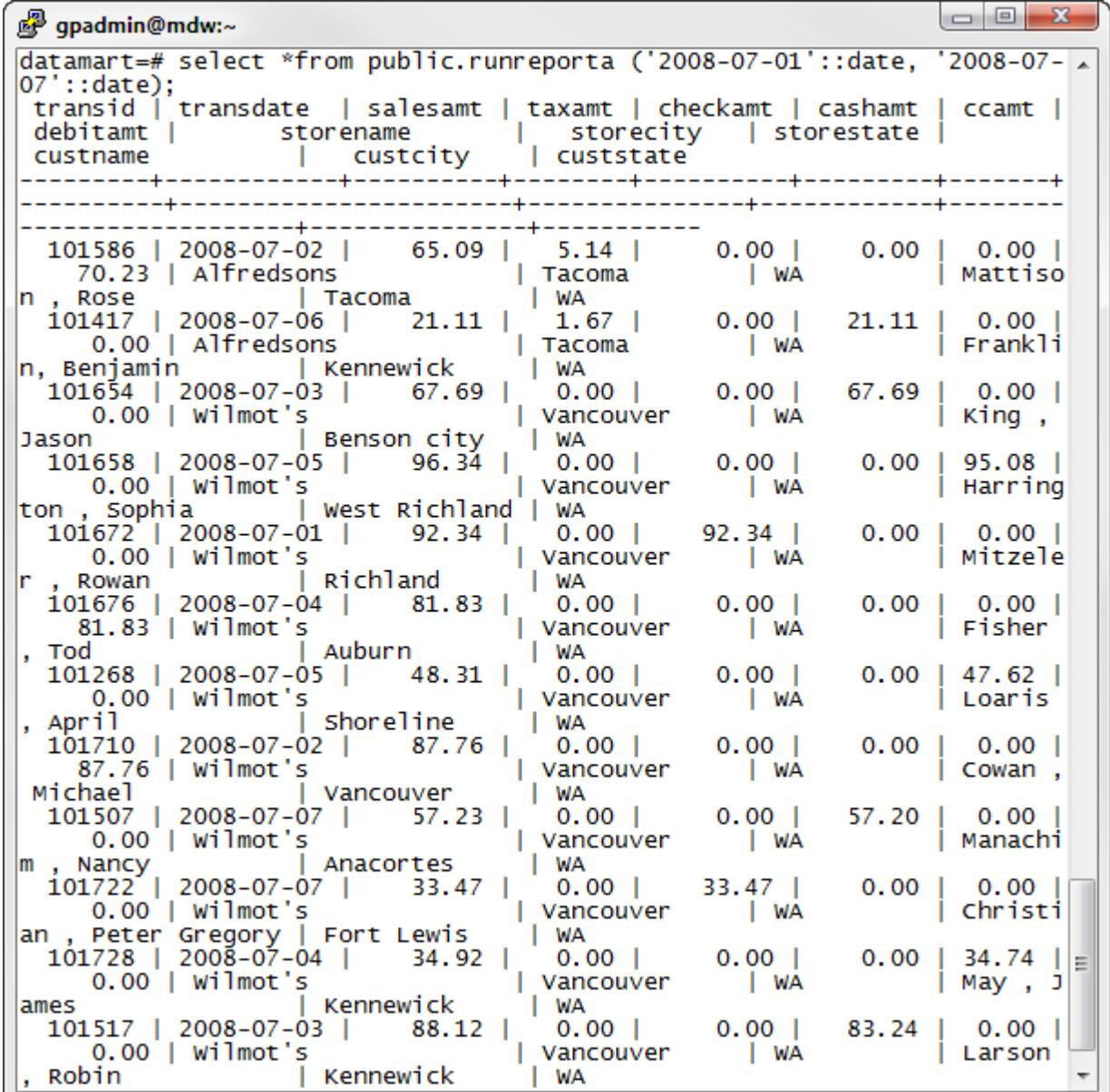
Step	Action																																																																																																																																																
5.	<p>Test the function by selecting the storeid, storename, public.Uppercname(store), city columns from store.</p> <pre>datamart=# select storeid,storename,public.Uppercname(store),city from store;</pre>  <p>The screenshot shows a terminal window titled "gpadmin@mdw:~". The command "datamart=# select storeid,storename,public.Uppercname(store),city from store;" is entered. The output is a table with four columns: storeid, storename, uppercase, and city. The table contains 34 rows of data, showing various grocery store names and their corresponding uppercase versions and cities.</p> <table border="1"> <thead> <tr> <th>storeid</th> <th>storename</th> <th>uppercase</th> <th>city</th> </tr> </thead> <tbody> <tr><td>34</td><td>A&amp;Y</td><td>A&amp;Y</td><td>Cody</td></tr> <tr><td>2</td><td>Deli Max</td><td>DELI MAX</td><td>Bellevue</td></tr> <tr><td>4</td><td>TOP Grocery</td><td>TOP GROCERY</td><td>Norfolk</td></tr> <tr><td>6</td><td>Safeway</td><td>SAFERWAY</td><td>St. George</td></tr> <tr><td>8</td><td>Scrooge's</td><td>SCROOGE'S</td><td>Saskatoon</td></tr> <tr><td>10</td><td>TOP Grocery</td><td>TOP GROCERY</td><td>Duncan</td></tr> <tr><td>12</td><td>Friendly's</td><td>FRIENDLY'S</td><td>Portland</td></tr> <tr><td>14</td><td>Max's 24 Hour Grocers</td><td>MAX'S 24 HOUR GROCERS</td><td>white Plains</td></tr> <tr><td>16</td><td>Max's 24 Hour Grocers</td><td>MAX'S 24 HOUR GROCERS</td><td>Jersey City</td></tr> <tr><td>18</td><td>TOP Grocery</td><td>TOP GROCERY</td><td>Pittsboro</td></tr> <tr><td>20</td><td>The Deli Again</td><td>THE DELI AGAIN</td><td>Edgar</td></tr> <tr><td>22</td><td>Willows</td><td>WILLOWS</td><td>St Louis</td></tr> <tr><td>24</td><td>MacIvers</td><td>MACIVERS</td><td>Slidell</td></tr> <tr><td>26</td><td>Sharpe Pharmacy</td><td>SHARPE PHARMACY</td><td>Twin Falls</td></tr> <tr><td>28</td><td>Max's 24 Hour Grocers</td><td>MAX'S 24 HOUR GROCERS</td><td>Dover</td></tr> <tr><td>30</td><td>The Green Plum</td><td>THE GREEN PLUM</td><td>San Mateo</td></tr> <tr><td>32</td><td>Tiny Tim's</td><td>TINY TIM'S</td><td>Vancouver</td></tr> <tr><td>1</td><td>Alfredsons</td><td>ALFREDSONS</td><td>Tacoma</td></tr> <tr><td>3</td><td>Wilmot's</td><td>WILMOT'S</td><td>Vancouver</td></tr> <tr><td>5</td><td>Gringo Pete's</td><td>GRINGO PETE'S</td><td>Salt Lake Cit</td></tr> <tr><td>7</td><td>TOP Grocery</td><td>TOP GROCERY</td><td>Nashville</td></tr> <tr><td>9</td><td>TOP Grocery</td><td>TOP GROCERY</td><td>Charleston</td></tr> <tr><td>11</td><td>Eco Store</td><td>ECO STORE</td><td>Bend</td></tr> <tr><td>13</td><td>Max's 24 Hour Grocers</td><td>MAX'S 24 HOUR GROCERS</td><td>New York</td></tr> <tr><td>15</td><td>Max's 24 Hour Grocers</td><td>MAX'S 24 HOUR GROCERS</td><td>Rochester</td></tr> <tr><td>17</td><td>Shorts Grocers</td><td>SHORTS GROCERS</td><td>Winston Salem</td></tr> <tr><td>19</td><td>Drugs and Stuff</td><td>DRUGS AND STUFF</td><td>Billings</td></tr> <tr><td>21</td><td>Renaissance Grocery</td><td>RENAISSANCE GROCERY</td><td>Biloxi</td></tr> <tr><td>23</td><td>Willows</td><td>WILLOWS</td><td>Independence</td></tr> <tr><td>25</td><td>Sharpe Pharmacy</td><td>SHARPE PHARMACY</td><td>Boise</td></tr> <tr><td>27</td><td>Sharpe Pharmacy</td><td>SHARPE PHARMACY</td><td>Boise</td></tr> <tr><td>29</td><td>The Deli</td><td>THE DELI</td><td>San Diego</td></tr> <tr><td>31</td><td>PJ's Grocers</td><td>PJ'S GROCERS</td><td>Oakland</td></tr> <tr><td>33</td><td>Xena's</td><td>XENA'S</td><td>Mobile</td></tr> <tr><td colspan="4">(34 rows)</td></tr> </tbody> </table> <pre>datamart=#</pre>	storeid	storename	uppercase	city	34	A&Y	A&Y	Cody	2	Deli Max	DELI MAX	Bellevue	4	TOP Grocery	TOP GROCERY	Norfolk	6	Safeway	SAFERWAY	St. George	8	Scrooge's	SCROOGE'S	Saskatoon	10	TOP Grocery	TOP GROCERY	Duncan	12	Friendly's	FRIENDLY'S	Portland	14	Max's 24 Hour Grocers	MAX'S 24 HOUR GROCERS	white Plains	16	Max's 24 Hour Grocers	MAX'S 24 HOUR GROCERS	Jersey City	18	TOP Grocery	TOP GROCERY	Pittsboro	20	The Deli Again	THE DELI AGAIN	Edgar	22	Willows	WILLOWS	St Louis	24	MacIvers	MACIVERS	Slidell	26	Sharpe Pharmacy	SHARPE PHARMACY	Twin Falls	28	Max's 24 Hour Grocers	MAX'S 24 HOUR GROCERS	Dover	30	The Green Plum	THE GREEN PLUM	San Mateo	32	Tiny Tim's	TINY TIM'S	Vancouver	1	Alfredsons	ALFREDSONS	Tacoma	3	Wilmot's	WILMOT'S	Vancouver	5	Gringo Pete's	GRINGO PETE'S	Salt Lake Cit	7	TOP Grocery	TOP GROCERY	Nashville	9	TOP Grocery	TOP GROCERY	Charleston	11	Eco Store	ECO STORE	Bend	13	Max's 24 Hour Grocers	MAX'S 24 HOUR GROCERS	New York	15	Max's 24 Hour Grocers	MAX'S 24 HOUR GROCERS	Rochester	17	Shorts Grocers	SHORTS GROCERS	Winston Salem	19	Drugs and Stuff	DRUGS AND STUFF	Billings	21	Renaissance Grocery	RENAISSANCE GROCERY	Biloxi	23	Willows	WILLOWS	Independence	25	Sharpe Pharmacy	SHARPE PHARMACY	Boise	27	Sharpe Pharmacy	SHARPE PHARMACY	Boise	29	The Deli	THE DELI	San Diego	31	PJ's Grocers	PJ'S GROCERS	Oakland	33	Xena's	XENA'S	Mobile	(34 rows)			
storeid	storename	uppercase	city																																																																																																																																														
34	A&Y	A&Y	Cody																																																																																																																																														
2	Deli Max	DELI MAX	Bellevue																																																																																																																																														
4	TOP Grocery	TOP GROCERY	Norfolk																																																																																																																																														
6	Safeway	SAFERWAY	St. George																																																																																																																																														
8	Scrooge's	SCROOGE'S	Saskatoon																																																																																																																																														
10	TOP Grocery	TOP GROCERY	Duncan																																																																																																																																														
12	Friendly's	FRIENDLY'S	Portland																																																																																																																																														
14	Max's 24 Hour Grocers	MAX'S 24 HOUR GROCERS	white Plains																																																																																																																																														
16	Max's 24 Hour Grocers	MAX'S 24 HOUR GROCERS	Jersey City																																																																																																																																														
18	TOP Grocery	TOP GROCERY	Pittsboro																																																																																																																																														
20	The Deli Again	THE DELI AGAIN	Edgar																																																																																																																																														
22	Willows	WILLOWS	St Louis																																																																																																																																														
24	MacIvers	MACIVERS	Slidell																																																																																																																																														
26	Sharpe Pharmacy	SHARPE PHARMACY	Twin Falls																																																																																																																																														
28	Max's 24 Hour Grocers	MAX'S 24 HOUR GROCERS	Dover																																																																																																																																														
30	The Green Plum	THE GREEN PLUM	San Mateo																																																																																																																																														
32	Tiny Tim's	TINY TIM'S	Vancouver																																																																																																																																														
1	Alfredsons	ALFREDSONS	Tacoma																																																																																																																																														
3	Wilmot's	WILMOT'S	Vancouver																																																																																																																																														
5	Gringo Pete's	GRINGO PETE'S	Salt Lake Cit																																																																																																																																														
7	TOP Grocery	TOP GROCERY	Nashville																																																																																																																																														
9	TOP Grocery	TOP GROCERY	Charleston																																																																																																																																														
11	Eco Store	ECO STORE	Bend																																																																																																																																														
13	Max's 24 Hour Grocers	MAX'S 24 HOUR GROCERS	New York																																																																																																																																														
15	Max's 24 Hour Grocers	MAX'S 24 HOUR GROCERS	Rochester																																																																																																																																														
17	Shorts Grocers	SHORTS GROCERS	Winston Salem																																																																																																																																														
19	Drugs and Stuff	DRUGS AND STUFF	Billings																																																																																																																																														
21	Renaissance Grocery	RENAISSANCE GROCERY	Biloxi																																																																																																																																														
23	Willows	WILLOWS	Independence																																																																																																																																														
25	Sharpe Pharmacy	SHARPE PHARMACY	Boise																																																																																																																																														
27	Sharpe Pharmacy	SHARPE PHARMACY	Boise																																																																																																																																														
29	The Deli	THE DELI	San Diego																																																																																																																																														
31	PJ's Grocers	PJ'S GROCERS	Oakland																																																																																																																																														
33	Xena's	XENA'S	Mobile																																																																																																																																														
(34 rows)																																																																																																																																																	

## Lab 23: Task 2 – Create PL/pgSQL Functions

Step	Action
1.	<p>Create a PL/pgSQL function to join the transaction fact table with the store and customer dimensions to return the transaction columns (transid, transdate, salesamt, taxamt, checkamt, cashamt, ccamt, debitamt), the store columns (storename, city, state) and the customer columns (custname, city, state).</p> <p>Parameters are the starting and ending transaction dates. (HINT: The output will be a user defined data type.)</p> <p>First, create the user defined data type and call it ReportTypeA.</p> <pre>datamart=# CREATE TYPE ReportTypeA AS ( transid      BIGINT, transdate    DATE, salesamt     NUMERIC, taxamt       NUMERIC, checkamt     NUMERIC, cashamt      NUMERIC, ccamt        NUMERIC, debitamt     NUMERIC, storename    CHARACTER VARYING, storecity   CHARACTER VARYING, storestate  CHARACTER VARYING, custname    CHARACTER VARYING, custcity    CHARACTER VARYING, custstate   CHARACTER VARYING);</pre>  <p>The screenshot shows a terminal window titled 'gpadmin@mdw:~'. The command entered is the same as the one shown above, creating the 'ReportTypeA' UDT with various fields like transid, transdate, and storename.</p>

Step	Action
2.	<p>Create the function, RunReportA. Create the function in the public schema.</p> <pre> datamart=# CREATE OR REPLACE FUNCTION public.RunReportA   (IN StartDate DATE, IN EndDate    DATE) RETURNS SETOF ReportTypeA AS \$BODY\$ DECLARE r ReportTypeA%rowtype; BEGIN FOR r IN SELECT  t.transid                   ,t.transdate                   ,t.salesamt                   ,t.taxamt                   ,t.checkamt                   ,t.cashamt                   ,t.ccamt                   ,t.debitamt                   ,s.storename                   ,s.city                   ,s.state                   ,c.custname                   ,c.city                   ,c.state              FROM facts.transaction t             INNER JOIN dimensions.store s             ON s.storeid = t.storeid             INNER JOIN dimensions.customer c             ON c.customerid = t.customerid             WHERE t.transdate BETWEEN StartDate AND EndDate LOOP       RETURN NEXT r;       -- return current row of SELECT END LOOP; RETURN; END \$BODY\$ LANGUAGE plpgsql; </pre>

Step	Action
	<pre>gpadmin@mdw:~\$  datamart=# CREATE OR REPLACE FUNCTION public.RunReportA datamart-#   (IN StartDate DATE, IN EndDate    DATE) datamart-# RETURNS SETOF ReportTypeA AS datamart-\$# \$BODY\$ datamart\$# DECLARE r ReportTypeA%rowtype; datamart\$# BEGIN datamart\$# FOR r IN SELECT t.transid datamart\$#           ,t.transdate datamart\$#           ,t.salesamt datamart\$#           ,t.taxamt datamart\$#           ,t.checkamt datamart\$#           ,t.cashamt datamart\$#           ,t.ccamt datamart\$#           ,t.debitamt datamart\$#           ,s.storename datamart\$#           ,s.city datamart\$#           ,s.state datamart\$#           ,c.custname datamart\$#           ,c.city datamart\$#           ,c.state datamart\$# datamart\$#           FROM facts.transaction t datamart\$#           INNER JOIN dimensions.store s datamart\$#           ON s.storeid = t.storeid datamart\$#           INNER JOIN dimensions.customer c datamart\$#           ON c.customerid = t.customerid datamart\$#           WHERE t.transdate BETWEEN StartDate AND EndDate LO OP datamart\$#     RETURN NEXT r; datamart\$#     -- return current row of SELECT datamart\$# END LOOP; datamart\$# RETURN; datamart\$# END datamart\$# \$BODY\$ datamart-# LANGUAGE plpgsql; CREATE FUNCTION datamart=# gpadmin@mdw:~\$</pre>

Step	Action
3.	<p>Test the function by running a report for the dates, 2008-07-01 to 2008-07-07:</p> <pre>datamart=# select *from public.runreporta ('2008-07-01'::date, '2008-07-07'::date);</pre>  <pre>gpadmin@mdw:~\$ datamart=# select *from public.runreporta ('2008-07-01'::date, '2008-07-07'::date);       transid   transdate   salesamt   taxamt   checkamt   cashamt   ccamt         debitamt   storename              storecity              storestate   custname         custname               custcity   -----+-----+-----+-----+-----+-----+-----+  101586   2008-07-02      65.09     5.14       0.00       0.00       0.00    70.23   Alfredsons              Tacoma              WA         Mattiso   n , Rose               Tacoma   WA   101417   2008-07-06      21.11     1.67       0.00      21.11       0.00    0.00   Alfredsons              Tacoma              WA         Franklin   n , Benjamin               Kennewick   WA   101654   2008-07-03      67.69     0.00       0.00      67.69       0.00    0.00   Wilmot's   Benson city   Vancouver              WA         King , Jason                                      WA   101658   2008-07-05      96.34     0.00       0.00       0.00      95.08    0.00   Wilmot's   West Richland   Vancouver              WA         Harring ton                                      WA   101672   2008-07-01      92.34     0.00      92.34       0.00       0.00    0.00   Wilmot's   Richland   Vancouver              WA         Mitzele r                                      WA   101676   2008-07-04      81.83     0.00       0.00       0.00       0.00    81.83   Wilmot's   Auburn   Vancouver              WA         Fisher , Tod                                      WA   101268   2008-07-05      48.31     0.00       0.00       0.00      47.62    0.00   Wilmot's   Shoreline   Vancouver              WA         Loaris , April                                      WA   101710   2008-07-02      87.76     0.00       0.00       0.00       0.00    87.76   Wilmot's   Vancouver   Vancouver              WA         Cowan , Michael                                      WA   101507   2008-07-07      57.23     0.00       0.00      57.20       0.00    0.00   Wilmot's   Vancouver   Vancouver              WA         Manachi m                                      WA   101722   2008-07-07      33.47     0.00      33.47       0.00       0.00    0.00   Wilmot's   Anacortes   Vancouver              WA         Christi an                                      WA   101728   2008-07-04      34.92     0.00       0.00       0.00      34.74    0.00   Wilmot's   Fort Lewis   Vancouver              WA         May , James                                      WA   101517   2008-07-03      88.12     0.00       0.00      83.24       0.00    0.00   Wilmot's   Kennewick   Vancouver              WA         Larson , Robin                                      WA  </pre>
	<p><b>Summary</b></p> <p>You may find that using functions to generate reports is a simple way to store SQL in the database to ensure that any user executing the query cannot change it and the query will produce consistent results.</p> <p>Functions provide a versatile way to perform complex transformation logic. Use functions in conjunction with temporary tables whenever you need to modify rows during ETL processing. Remember that you can also create temporary tables in the PL/pgSQL function!</p> <p>Use dynamic functions whenever you need to create ad-hoc SQL. This will again keep consistent and repeatable SQL stored in the database.</p>

*End of Lab Exercise*

**END OF LAB**

This page intentionally left blank