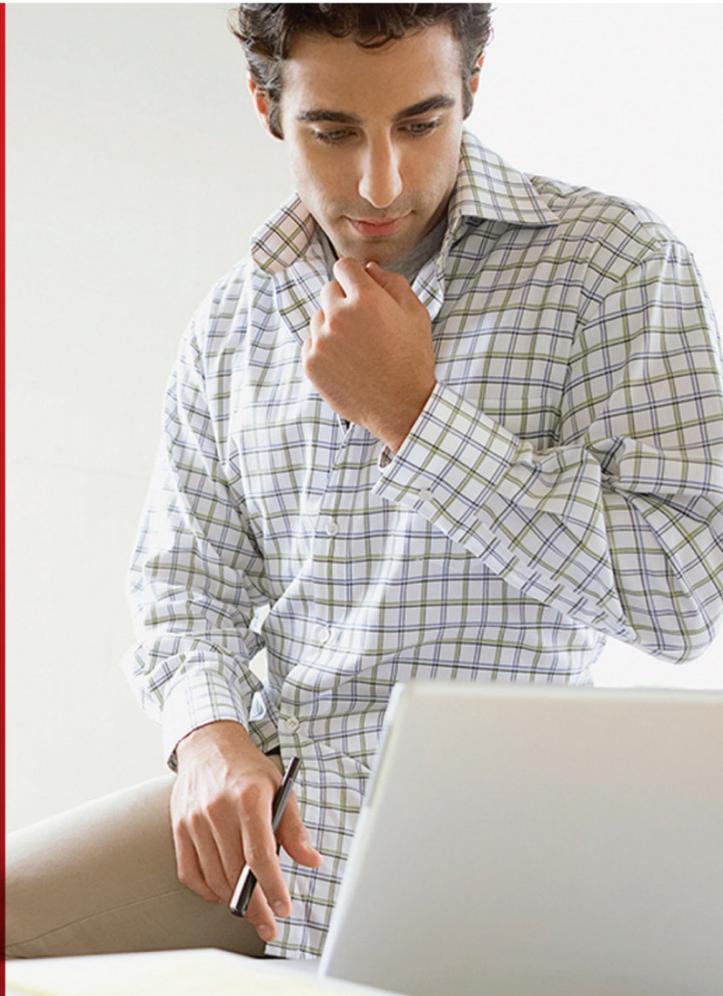




**Hardware and Software**  
Engineered to Work Together



Oracle University and you. You are not a Valid Partner use only

# Oracle Big Data Fundamentals

Activity Guide

D86898GC10

Edition 1.0 | May 2015 | D91415

Learn more from Oracle University at [oracle.com/education/](http://oracle.com/education/)

**Authors**

Lauran K. Serhal  
Brian Pottle  
Suresh Mohan

**Technical Contributors and Reviewers**

Marty Gubar  
Melliyal Annamalai  
Sharon Stephen  
Jean-Pierre Dijcks  
Bruce Nelson  
Daniel W McClary  
Josh Spiegel  
Anuj Sahni  
Dave Segleau  
Ashwin Agarwal  
Salome Clement  
Donna Carver  
Alex Kotopoulos  
Marcos Arancibia  
Mark Hornick  
Charlie Berger  
Ryan Stark  
Swarnapriya Shridhar  
Branislav Valny  
Dmitry Lychagin  
Mirella Tumolo  
S. Matt Taylor  
Lakshmi Narapareddi  
Drishya Tm

**Graphic Editors**

Rajiv Chandrabhanu  
Maheshwari Krishnamurthy

**Editors**

Malavika Jinka  
Smita Kommini  
Arijit Ghosh

**Publishers**

Veena Narasimhan  
Michael Sebastian Almeida  
Syed Ali

**Copyright © 2015, Oracle and/or its affiliates. All rights reserved.**

**Disclaimer**

This document contains proprietary information and is protected by copyright and other intellectual property laws. You may copy and print this document solely for your own use in an Oracle training course. The document may not be modified or altered in any way. Except where your use constitutes "fair use" under copyright law, you may not use, share, download, upload, copy, print, display, perform, reproduce, publish, license, post, transmit, or distribute this document in whole or in part without the express authorization of Oracle.

The information contained in this document is subject to change without notice. If you find any problems in the document, please report them in writing to: Oracle University, 500 Oracle Parkway, Redwood Shores, California 94065 USA. This document is not warranted to be error-free.

**Restricted Rights Notice**

If this documentation is delivered to the United States Government or anyone using the documentation on behalf of the United States Government, the following notice is applicable:

**U.S. GOVERNMENT RIGHTS**

The U.S. Government's rights to use, modify, reproduce, release, perform, display, or disclose these training materials are restricted by the terms of the applicable Oracle license agreement and/or the applicable U.S. Government contract.

**Trademark Notice**

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

## Table of Contents

---

<b>Practices for Lesson 1: Introduction.....</b>	<b>1-1</b>
Practices for Lesson 1: Introduction.....	1-2
Guided Practice 1-1: Oracle Big Data Documentation.....	1-3
Guided Practice 1-2: Oracle Big Data Tutorials on Oracle Learning Library (OLL) .....	1-11
Guided Practice 1-3: Accessing and Reviewing the Oracle Big Data Lite Virtual Machine Home Page .....	1-14
<b>Practices for Lesson 2: Big Data and the Oracle Information Management System.....</b>	<b>2-1</b>
Practices for Lesson 2.....	2-2
<b>Practices for Lesson 3: Using Oracle Big Data Lite Virtual Machine.....</b>	<b>3-1</b>
Practices for Lesson 3.....	3-2
Practice 3-1: Using the BigData Lite Virtual Machine (VM).....	3-3
Practice 3-2: Using the Oracle MoviePlex Application.....	3-12
<b>Practices for Lesson 4: Introduction to the Big Data Ecosystem.....</b>	<b>4-1</b>
Practices for Lesson 4.....	4-2
<b>Practices for Lesson 5: Introduction to the Hadoop Distributed File System (HDFS).....</b>	<b>5-1</b>
Practices for Lesson 5.....	5-2
Practice 5-1: Introduction to HDFS Commands .....	5-3
Solution 5-1: Introduction to HDFS Commands .....	5-4
<b>Practices for Lesson 6: Acquire Data Using CLI, Fuse DFS, and Flume .....</b>	<b>6-1</b>
Practice for Lesson 6 .....	6-2
Practice 6-1: Viewing Flume Commands and Configuration Options .....	6-3
<b>Practices for Lesson 7: Acquire and Access Data Using Oracle NoSQL Database .....</b>	<b>7-1</b>
Practices for Lesson 7 .....	7-2
Practice 7-1: Start and Run the MoviePlex Application .....	7-3
Practice 7-2: Start Oracle NoSQL and Load User Profile Data .....	7-6
Practice 7-3: Load Movie Data.....	7-11
Practice 7-4: Query Movie Data.....	7-13
<b>Practices for Lesson 8: Primary Administrative Tasks for Oracle NoSQL Database .....</b>	<b>8-1</b>
Practices for Lesson 8 .....	8-2
<b>Practices for Lesson 9: Introduction to MapReduce.....</b>	<b>9-1</b>
Practices for Lesson 9.....	9-2
Practice 9-1: Running a MapReduce Hadoop Job .....	9-3
Solution 9-1: Running a MapReduce Hadoop Job .....	9-6
<b>Practices for Lesson 10: Resource Management Using YARN.....</b>	<b>10-1</b>
Practices for Lesson 10.....	10-2
Practice 10-1: Resource Management Using YARN .....	10-3
Solution 10-1: Resource Management Using YARN .....	10-5
<b>Practices for Lesson 11: Overview of Hive and Pig .....</b>	<b>11-1</b>
Practices for Lesson 11: Overview of Hive and Pig.....	11-2
Practice 11-1: Manipulating Data by Using Hive .....	11-3
Solution 11-1: Manipulating Data by Using Hive .....	11-5
Practice 11-2: Extracting Facts by Using Hive .....	11-14
Solution 11-2: Extracting Facts by Using Hive .....	11-18
Practice 11-3: Working with Pig .....	11-33
Solution 11-3: Working with Pig .....	11-37
<b>Practices for Lesson 12: Overview of Cloudera Impala.....</b>	<b>12-1</b>

Practices for Lesson 12.....	12-2
<b>Practices for Lesson 13: Using Oracle XQuery for Hadoop.....</b>	<b>13-1</b>
Practices for Lesson 13.....	13-2
Practice 13-1: Using Oracle XQuery for Hadoop (OXH).....	13-3
Solution 13-1: Using Oracle XQuery for Hadoop (OXH).....	13-7
Practice 13-2: Working with Hive UDF and SerDe on XML Data .....	13-22
Solution 13-2: Working with Hive UDF and SerDe on XML Data .....	13-24
Practice 13-3: Loading Results from an XQuery into an Oracle Database .....	13-32
Solution 13-3: Loading Results from an XQuery into an Oracle Database .....	13-33
<b>Practices for Lesson 14: Overview of Solr.....</b>	<b>14-1</b>
Practices for Lesson 14.....	14-2
Guided Practice 14-1: Using Apache Solr .....	14-3
Guided Practice 14-2: Using Solr with Hue.....	14-13
<b>Practices for Lesson 15: Apache Spark .....</b>	<b>15-1</b>
Practices for Lesson 15.....	15-2
Practice 15-1: Using Apache Spark .....	15-3
<b>Practices for Lesson 16: Options for Integrating Your Big Data .....</b>	<b>16-1</b>
Practices for Lesson 16.....	16-2
<b>Practices for Lesson 17: Overview of Apache Sqoop .....</b>	<b>17-1</b>
Practices for Lesson 17.....	17-2
<b>Practices for Lesson 18: Using Oracle Loader for Hadoop (OLH).....</b>	<b>18-1</b>
Practices for Lesson 18.....	18-2
Guided Practice 18-1: Loading Data from HDFS Files into Oracle Database .....	18-3
Guided Practice 18-2: Loading Data from Hive Tables into Oracle Database.....	18-10
<b>Practices for Lesson 19: Using Copy to BDA .....</b>	<b>19-1</b>
Practices for Lesson 19.....	19-2
Practice 19-1: View Source Data and Identify Target Directory .....	19-3
Practice 19-2: Create External Tables and Copy the Data .....	19-5
Practice 19-3: Create Hive External Tables and Query the Data .....	19-8
<b>Practices for Lesson 20: Using Oracle SQL Connector for HDFS .....</b>	<b>20-1</b>
Practices for Lesson 20.....	20-2
Guided Practice 20-1: Accessing HDFS Files by Using OSCH.....	20-3
Guided Practice 20-2: Accessing Hive Tables by Using OSCH .....	20-8
Guided Practice 20-3: Accessing Partitioned Hive Tables by Using OSCH.....	20-14
<b>Practices for Lesson 21: Using Oracle Data Integrator and Oracle GoldenGate with Hadoop.....</b>	<b>21-1</b>
Practices for Lesson 21.....	21-2
Practice 21-1: Review Topology and Model Setup.....	21-3
Practice 21-2: Map and Load Data Into Hive Tables .....	21-5
<b>Practices for Lesson 22: Using Oracle Big Data SQL .....</b>	<b>22-1</b>
Practices for Lesson 22.....	22-2
Practice 22-1: Complete the Configuration of Big Data SQL .....	22-3
Practice 22-2: Review HDFS Data That You Want to Access.....	22-6
Practice 22-3: Leverage Hive Metadata for the Oracle External Tables - Then Query the Hadoop Data .....	22-12
Practice 22-4: Apply Oracle Security Policies Over Hadoop Data.....	22-15
Practice 22-5: Using Analytic SQL on Joined Data .....	22-18
<b>Practices for Lesson 23: Using Oracle Advanced Analytics: Oracle Data Mining and Oracle R Enterprise .....</b>	<b>23-1</b>

Practices for Lesson 23.....	23-2
Practice 23-1: Using Oracle Data Miner 4.0 with Big Data .....	23-3
Practice 23-2: Using Oracle R Enterprise with Big Data.....	23-39
<b>Practices for Lesson 24: Introducing Oracle Big Data Discovery.....</b>	<b>24-1</b>
Practices for Lesson 24.....	24-2
<b>Practices for Lesson 25: Introduction to the Oracle Big Data Appliance (BDA).....</b>	<b>25-1</b>
Practices for Lesson 25.....	25-2
Guided Practice 25-1: Introduction to Oracle BDA.....	25-3
<b>Practices for Lesson 26: Managing Oracle BDA .....</b>	<b>26-1</b>
Practices for Lesson 26.....	26-2
Guided Practice 26-1: Monitoring MapReduce Jobs .....	26-3
Guided Practice 26-2: Monitoring the Health of HDFS .....	26-5
Guided Practice 26-3: Using HUE.....	26-8
<b>Practices for Lesson 27: Balancing MapReduce Jobs.....</b>	<b>27-1</b>
Practices for Lesson 27.....	27-2
Practice 27-1: Balancing MapReduce Jobs.....	27-3
<b>Practices for Lesson 28: Securing Your Data .....</b>	<b>28-1</b>
Practices for Lesson 28.....	28-3

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and Error : You are not a Valid Partner use only

## Course Practice Environment: Security Credentials

---

For OS usernames and passwords, see the following:

- If you are attending a classroom-based or live virtual class, ask your instructor or LVC producer for OS credential information.
- If you are using a self-study format, refer to the communication that you received from Oracle University for this course.

For default credentials used in this course, see the following table:

Product-Specific Credentials		
User Type	Username	Password
Oracle Big Data Lite (BDLite)		welcome1
Oracle Database 12c		welcome1
Hue	oracle	welcome1
Oracle MoviePlex Application	guest1	welcome1
Oracle Data Integrator (ODI)	SUPERVISOR	welcome1
Hive Metastore (MySQL)	hive	welcome1
WebLogic	weblogic	welcome1
Cloudera Manager (CM)	admin	welcome1
Oracle SQL Developer Database Connection	BDA	welcome1

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and Error : You are not a Valid Partner use only

## **Practices for Lesson 1: Introduction**

**Chapter 1**

## Practices for Lesson 1: Introduction

---

### Practices Overview

In these practices, you will use your web browser to access and review some of the useful Oracle Big Data resources and documentation, and then bookmark such resources for easier access.

## Guided Practice 1-1: Oracle Big Data Documentation

### Overview

In this practice, you will explore some of the available Oracle Big Data Appliance documentation.

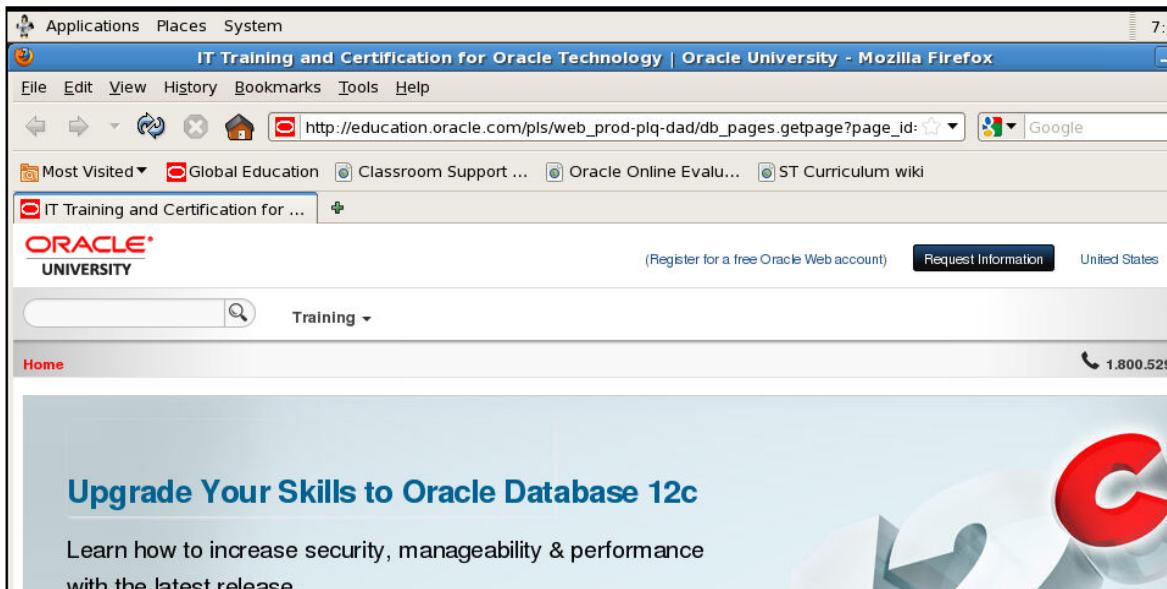
### Assumptions

### Tasks

1. Connect to your classroom machine assigned to you by your instructor. Your instructor will provide you with the credentials that you will need to connect to your machine. The desktop is displayed.



2. Start your Mozilla Firefox web browser. Click the Firefox Web Browser icon on your desktop.



Oracle University and Error : You are not a Valid Partner use only

- Access, review, and bookmark the Oracle Big Data Appliance documentation website at <http://www.oracle.com/technetwork/database/bigdata-appliance/documentation/index.html>.

The screenshot shows a Mozilla Firefox window with the title bar "Oracle Big Data Documentation - Mozilla Firefox" and the URL "http://www.oracle.com/technetwork/database/bigdata-appliance/documentation/index.html". The page content includes a navigation bar with links like "Products", "Solutions", "Downloads", "Store", "Support", "Training", "Partners", "About", and "OTN". A sidebar on the left lists various database products. The main content area features a section titled "Oracle Big Data Documentation" with a brief description and links to "Oracle Big Data, Release 4.1.0" (E57371\_01.zip), "Oracle Big Data, Release 4.0.0" (E55905\_01.zip), and "Oracle Big Data, Release 3.1.0". An advertisement for "COLLABORATE15" is visible on the right.

- Click the **Oracle Big Data, Release 4.1.0** link on the **Documentation** tab on the page.

The screenshot shows the "Documentation" tab selected on the Oracle Big Data Documentation page. The "Oracle Big Data, Release 4.1.0" link (E57371\_01.zip) is highlighted with a red box. The page also displays a brief description of Oracle Big Data and links to other releases.

5. Review and explore the page. Some of the useful links on this page for the course are: **The Owner's Guide, Software User's Guide, Oracle Loader for Hadoop, Cloudera's Distribution Including Apache Hadoop Library**, and others.

# Oracle Big Data Documentation

## Release 4.1

### Overview

#### Welcome

Companies have been making business decisions for decades based on transactional data stored in relational databases. Beyond that critical data is a potential treasure trove of less structured data: weblogs, social media, email, sensors, and photographs that can be mined for useful information.

Oracle offers a broad and integrated portfolio of products to help you acquire and organize these diverse data sources and analyze them alongside your existing data to find new insights and capitalize on hidden relationships. Learn how Oracle helps you acquire, organize, and analyze your big data.

#### Big Data Appliance Installation and Setup

[Safety and Compliance Guide](#)



[Site Checklists](#)



#### Related Software Libraries

[Oracle R Enterprise](#)



[Oracle NoSQL Database](#)



[Oracle Enterprise Manager Cloud Control](#)



[Oracle Audit Vault and Database Firewall](#)



#### Related Hardware Libraries

[Oracle Server X5-2L](#)



[Oracle Sun Rack II](#)



[Oracle Sun Datacenter InfiniBand Switch 36](#)



[Oracle Sun Network QDR InfiniBand Gateway Switch](#)



[Oracle Integrated Lights Out Manager](#)



and analyze them alongside your existing data to find new insights and capitalize on hidden relationships. Learn how Oracle helps you acquire, organize, and analyze your big data.

## Big Data Appliance Installation and Setup

- [Safety and Compliance Guide](#) ↗
- [Site Checklists](#)
- [Owner's Guide](#)

## Integrated Software and Big Data Connectors

- [Licensing Information](#)
- [Software User's Guide](#)
- [Enterprise Manager System Monitoring](#)
- [Plug-in Installation Guide for Oracle Big Data Appliance](#)
- [Connectors User's Guide](#)
- [Oracle Loader for Hadoop Java Example](#)
- [Perfect Balance Java API Reference](#)
- [Cloudera's Distribution Including Apache Hadoop Library](#) ↗

- [Oracle Server X5-2L](#) ↗
- [Oracle Sun Rack II](#) ↗
- [Oracle Sun Datacenter InfiniBand Switch 36](#) ↗
- [Oracle Sun Network QDR InfiniBand Gateway Switch](#) ↗
- [Oracle Integrated Lights Out Manager](#) ↗

6. Display the *Oracle Big Data Appliance Software User's Guide Release 4 (4.1)* documentation. Click the **Software User's Guide** link in the **Integrated Software and Big Data Connectors** section.

## Integrated Software and Big Data Connectors

- [Licensing Information](#)
- [Software User's Guide](#)

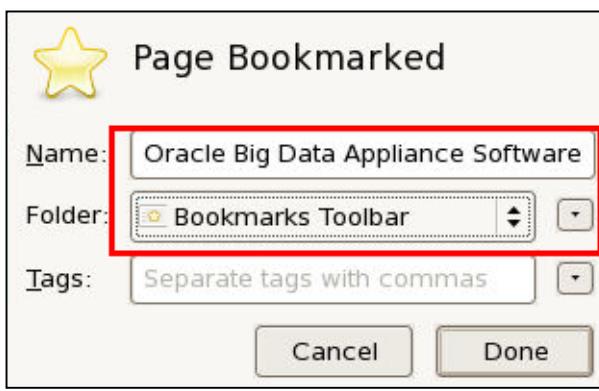
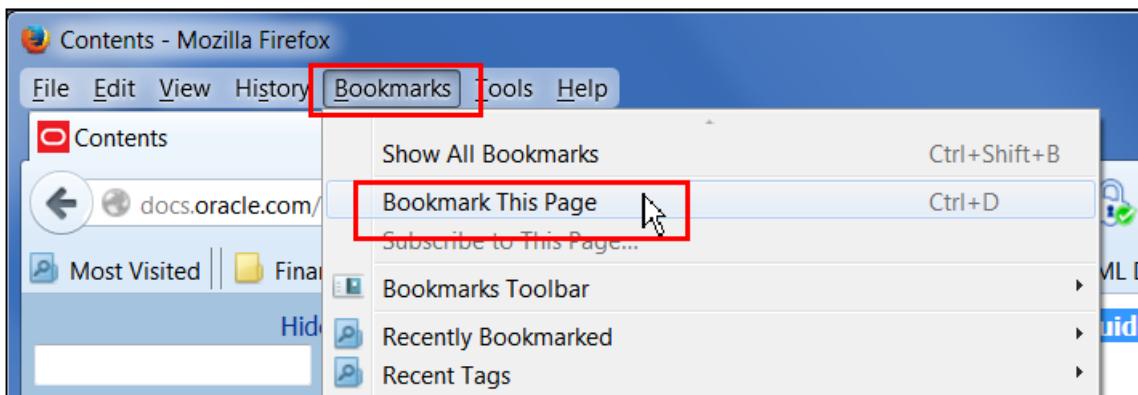
Home / Big Data / Oracle Big Data Documentation, Release 4.1

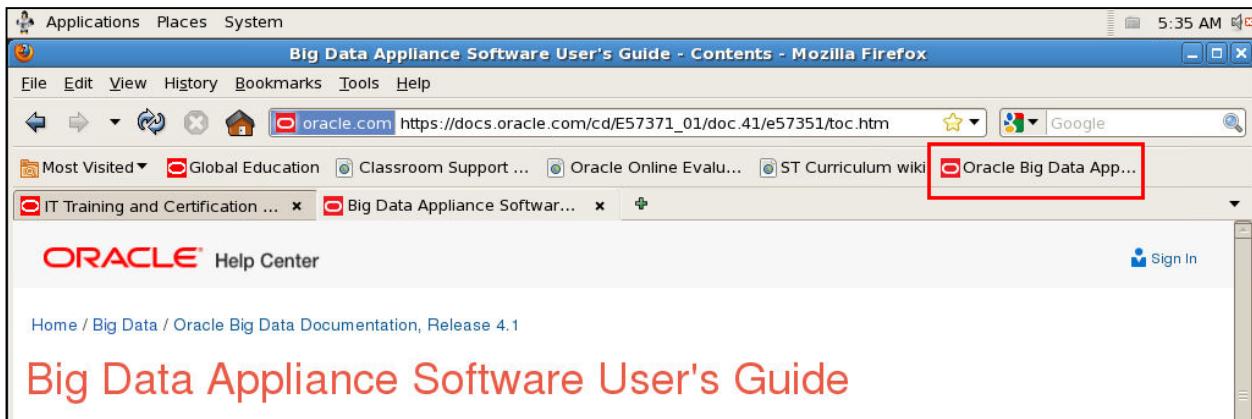
## Big Data Appliance Software User's Guide

The screenshot shows the 'Contents' page of the Oracle Big Data Appliance Software User's Guide. The left sidebar contains navigation buttons like 'Feedback' and 'Download'. The main content area lists several sections and chapters:

- Title and Copyright Information
- Preface
- Part I Administration
  - 1 Introducing Oracle Big Data Appliance
  - 2 Administering Oracle Big Data Appliance
  - 3 Supporting User Access to Oracle Big Data Appliance
  - 4 Configuring Oracle Exadata Database Machine for Use with Oracle Big Data Appliance
- Part II Oracle Big Data Appliance Software
  - 5 Optimizing MapReduce Jobs Using Perfect Balance
- Part III Oracle Big Data SQL
  - 6 Using Oracle Big Data SQL for Data Access
  - 7 Oracle Big Data SQL Reference
  - 8 Copying Oracle Tables to Hadoop
- Glossary
- Index

7. Review the contents briefly, and then bookmark this page using the **Bookmarks > Bookmark this Page** menu option in your browser. Enter **Oracle Big Data Appliance Software User's Guide Release 4 (4.1)** as the name for the bookmark, and select **Bookmarks Toolbar** for the folder.





8. Display the *Oracle Big Data Appliance Oracle Big Data Appliance Owner's Guide Release 4 (4.1)* documentation. Click the **Owner's Guide** HTML link in the **Big Data Appliance and Setup** section.

The screenshot shows a section titled "Big Data Appliance Installation and Setup". Below it are three links: "Safety and Compliance Guide", "Site Checklists", and "Owner's Guide". The "Owner's Guide" link is highlighted with a red box.

The screenshot shows the "Big Data Appliance Owner's Guide" page. At the top, there is a breadcrumb navigation: "Home / Big Data / Oracle Big Data Documentation, Release 4.1". Below the title are standard navigation icons for back, forward, and search.

# Contents

## Title and Copyright Information

- ▶ [Preface](#)
- ▶ [Changes in This Release for Oracle Big Data Appliance](#)

## Part I Preinstallation

- ▶ [1 Introduction to Oracle Big Data Appliance](#)
- ▶ [2 Site Requirements for Oracle Big Data Appliance](#)
- ▶ [3 Understanding the Network Requirements](#)
- ▶ [4 Using Oracle Big Data Appliance Configuration Generation Utility](#)
- ▶ [5 Setting Up Auto Service Request](#)

## Part II Hardware Installation and Configuration

- ▶ [6 Installing Oracle Big Data Appliance at the Site](#)
- ▶ [7 Configuring an Oracle Big Data Appliance Full Rack or Starter Rack](#)
- ▶ [8 Configuring an Oracle Big Data Appliance In-Rack Expansion Kit](#)
- ▶ [9 Connecting Multiple Oracle Big Data Appliance Racks](#)

## Guided Practice 1-2: Oracle Big Data Tutorials on Oracle Learning Library (OLL)

### Overview

In this practice, you explore the available tutorials, Oracle By Examples (OBEs), and other Big Data material on OLL.

### Assumptions

### Tasks

1. Access Oracle Learning Library (OLL) at <http://www.oracle.com/goto/oll>.

The screenshot shows the Oracle Learning Library homepage. At the top, there's a navigation bar with 'Home', 'Products', 'Search', and 'My Library' tabs. The 'Home' tab is selected. On the right, there are user status ('nobody'), help, and login links. A large banner in the center says 'Learn how to upgrade your Fusion Middleware products to the newly released 12c!...'. Below the banner, there's a sub-banner for 'ORACLE FUSION MIDDLEWARE' with text about upgrading to 12c. A 'Upgrade to FMW 12c!' button is present. A search bar at the bottom of the banner allows searching by title, description, tags, and role. To the left, a box titled 'Most Popular in Last Month' lists several Oracle Database-related articles. To the right, a box titled 'Latest Additions' lists recent articles. At the bottom right, a 'Tag Cloud' displays various tags like 'jd edwards', 'hyperion', 'application development', 'cloud', etc., with their respective hit counts.

2. Enter **Big Data** in the **Search** text box, and then press **Enter**. On the **Search Results** page, click the **Oracle Big Data Learning Library** link.

The screenshot shows a search results page for 'Big Data'. The search bar at the top contains the text 'Big Data'. Below the search bar, there is a sidebar titled 'Most Popular in Last 7 Days' with links to 'Importing and Exporting', 'Installing Oracle Database', 'Getting Started with Oracle Big Data', and 'Getting Started with Oracle NoSQL'. The main content area displays several search results:

- Integrate All Your Data with Oracle Big Data Connectors - Part 5 of 6
- Part 3: Using Big Data and NoSQL to Manage On-line Profiles
- Big Data Lite Movieplex Demo - SQL Pattern Matching for Sessionization Analysis
- Big Data Lite Movieplex Video Demo - SQL Pattern Matching for Sessionization Analysis
- Meeting the Challenge of Big Data
- Monitor Big Data Appliance (BDA) with Oracle Enterprise Manager Cloud Control
- Oracle Big Data Tutorial Video Series

A sidebar on the right shows navigation links: 'enterpriseone' (328) and '385'.

The screenshot shows a search results page for 'Big Data' on the Oracle Learning Library. The search bar at the top contains the text 'Big Data'. The main content area displays the following information:

**Search Results**

**View**

- Limit Types to:
  - Article
  - BLOG
  - Certification
  - Collection

**Search Results**

**Oracle Big Data Learning Library**

Learn about Oracle Big Data, Data Science, Learning Analytics, Oracle NoSQL Database, and more!

Tags: Big Data, Big Data Training, NoSQL Training

Type: Library

Release Date: 1 years ago

The screenshot shows the Oracle Big Data Learning Library homepage. The main title is 'Oracle Big Data Learning Library...'. Below it, a sub-headline reads: 'Learn about Oracle Big Data, Data Science, Learning Analytics, Oracle NoSQL Database, and more!'. There are three promotional cards:

- Oracle Big Data Essentials**: Attend this Oracle University Course!
- Using Oracle NoSQL Database**: Attend this Oracle University class!
- Oracle and Big Data on OTN**: See the latest resource on OTN.

A search bar at the bottom of the page allows users to 'Search content title, description and tags...'.

The navigation menu includes tabs for 'Welcome' (which is selected), 'Get Started', 'Learn by Role', 'Learn by Product', and 'Learning Paths'. Below the menu, there are buttons for 'Latest Additions' and 'Additional Resources'. A welcome message states: 'Welcome to the Oracle Big Data Appliance Learning Library. The library contains training information on Oracle's Big Data Appliance.' A note for new users suggests starting at the 'Get Started' tab. To the right, there is a 'Big Data and Innovation' section featuring a video player with two men and a play button, with the text 'Watch this webcast.'

3. Click the **Learn By Product** tab, and then review the available training under the various categories such as **Oracle Big Data Solution**, **Oracle NoSQL Database**, and so on.

The screenshot shows the Oracle Big Data Learning Library homepage. At the top, there are three promotional cards: 'Oracle Big Data Essentials', 'Using Oracle NoSQL Database', and 'Oracle and Big Data on OTN'. Below them is a search bar. The navigation bar includes 'Welcome', 'Get Started', 'Learn by Role', 'Learn by Product' (which is highlighted with a red box), 'Learning Paths', 'Latest Additions', and 'Additional Resources'. A sidebar on the right says 'Big Data and Innovation' and features a video thumbnail of two men talking.

This screenshot shows the 'Learn by Product' page. The main content area is divided into several sections: 'Oracle Big Data Solution' (with topics like 'Oracle Big Data and...', 'Meeting the Challenge...', 'Oracle Big Data Tutorials...', etc.); 'Oracle NoSQL Database' (with topics like 'Oracle NoSQL Data...', 'Deploying Oracle N...', 'Oracle NoSQL Data...', etc.); 'Data Mining' (with topics like 'Oracle Data Mining 12c...', 'Oracle Data Mining 11g...', 'Oracle Database 12c: ...'); 'Big Data and Innovation' (with a video thumbnail of two men and the text 'Watch this webcast.'); 'Featured Content' (with a thumbnail for 'Flume NG Basics' and the text 'Oracle Big Data and Data Science Basics'); 'Oracle R Enterprise' (with 'Oracle R Enterprise v ...'); 'Oracle Big Data Connectors' (with 'Integrate All Your Data ...', 'Using Oracle Direct Co...', 'Using Oracle R Conne...', etc.); and 'Oracle Data Integrator' (with 'Oracle Data Integrator ...', 'Oracle Data Integrator ...', 'Oracle Data Integrator ...', etc.).

## Guided Practice 1-3: Accessing and Reviewing the Oracle Big Data Lite Virtual Machine Home Page

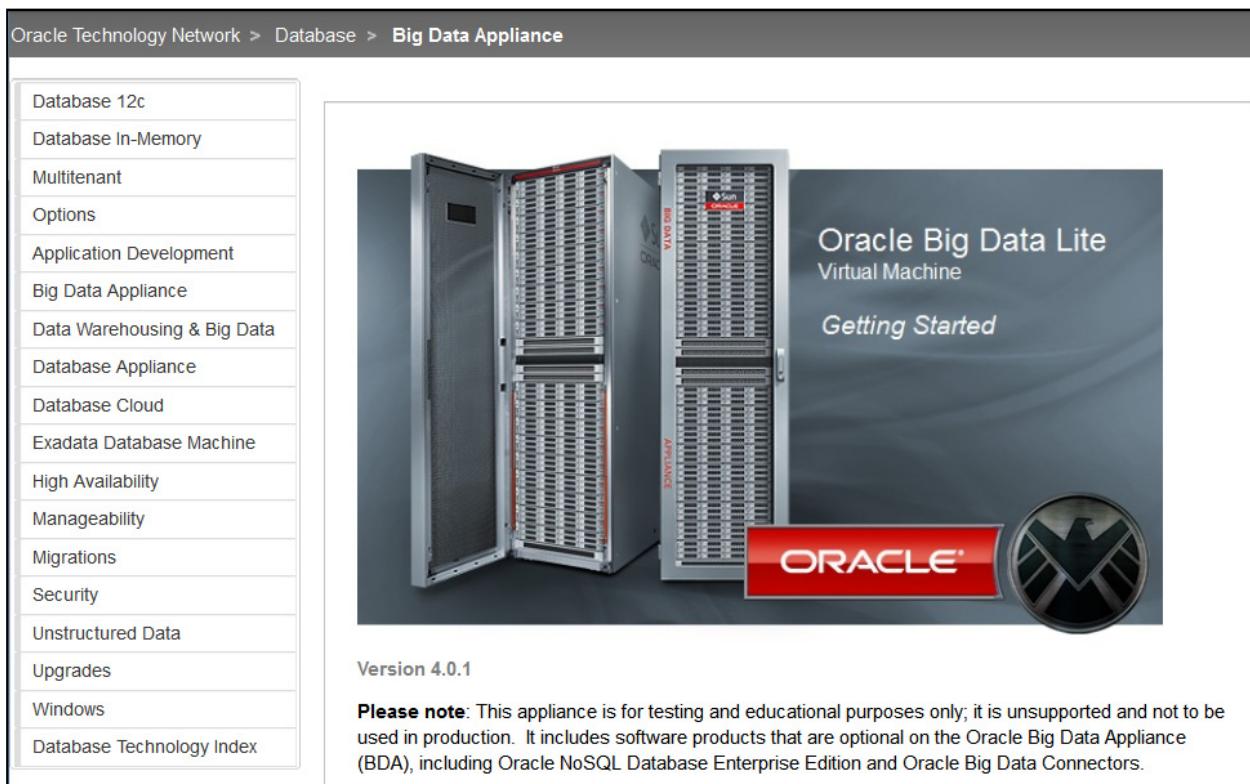
### Overview

In this practice, you access and explore the **Oracle Big Data Lite Virtual Machine** home page.

### Assumptions

### Tasks

1. Access and explore the Oracle Big Data Lite Virtual Machine home page at <http://www.oracle.com/technetwork/database/bigdata-appliance/bigdatalite-401-2405981.html>



The screenshot displays the Oracle Technology Network website under the 'Database' category, specifically the 'Big Data Appliance' section. On the left, there is a sidebar menu with various database-related links. The main content area features a large image of an Oracle Big Data Appliance server rack, which is a tall, grey server unit with multiple drive bays and components. To the right of the server image, the text 'Oracle Big Data Lite Virtual Machine' and 'Getting Started' is displayed. Below this, the 'ORACLE' logo is shown in red and white, along with the Oracle shield emblem. At the bottom of the page, the text 'Version 4.0.1' is visible, followed by a note: 'Please note: This appliance is for testing and educational purposes only; it is unsupported and not to be used in production. It includes software products that are optional on the Oracle Big Data Appliance (BDA), including Oracle NoSQL Database Enterprise Edition and Oracle Big Data Connectors.'

<p>Unstructured Data</p> <p>Upgrades</p> <p>Windows</p> <p>Database Technology Index</p>	<p><b>Version 4.0.1</b></p> <p><b>Please note:</b> This appliance is for testing and educational purposes only; it is unsupported and not to be used in production. It includes software products that are optional on the Oracle Big Data Appliance (BDA), including Oracle NoSQL Database Enterprise Edition and Oracle Big Data Connectors.</p> <ul style="list-style-type: none"><li>▪ <a href="#">Introduction</a></li><li>▪ <a href="#">Download Oracle Big Data Lite Virtual Machine</a></li><li>▪ <a href="#">Getting Started</a><ul style="list-style-type: none"><li>▪ <a href="#">Oracle MoviePlex</a></li><li>▪ <a href="#">Hands-on Labs</a></li><li>▪ <a href="#">Web Sites /White Papers / EBook / Blogs</a></li></ul></li></ul> <p><b>Introduction</b></p> <p>Oracle Big Data Lite Virtual Machine provides an integrated environment to help you get started with the Oracle Big Data platform. Many Oracle Big Data platform components have been installed and configured - allowing you to begin using the system right away. The following components are included on Oracle Big Data Lite:</p> <ul style="list-style-type: none"><li>▪ Oracle Enterprise Linux 6.4</li><li>▪ Oracle Database 12c Release 1 Enterprise Edition (12.1.0.2) - including Oracle Big Data SQL-enabled external tables</li><li>▪ Cloudera Distribution including Apache Hadoop (CDH5.1.2)</li><li>▪ Cloudera Manager (5.1.2)</li><li>▪ Oracle Big Data Connectors 4.0<ul style="list-style-type: none"><li>▪ Oracle SQL Connector for HDFS 3.1.0</li><li>▪ Oracle Loader for Hadoop 3.2.0</li></ul></li></ul>
	<ul style="list-style-type: none"><li>▪ Oracle SQL Connector for HDFS 3.1.0</li><li>▪ Oracle Loader for Hadoop 3.2.0</li><li>▪ Oracle Data Integrator 12c</li><li>▪ Oracle R Advanced Analytics for Hadoop 2.4.1</li><li>▪ Oracle XQuery for Hadoop 4.0.1</li><li>▪ Oracle NoSQL Database Enterprise Edition 12cR1 (3.0.14)</li><li>▪ Oracle JDeveloper 12c (12.1.3)</li><li>▪ Oracle SQL Developer and Data Modeler 4.0.3</li><li>▪ Oracle Data Integrator 12cR1 (12.1.3)</li><li>▪ Oracle GoldenGate 12c</li><li>▪ Oracle R Distribution 3.1.1</li><li>▪ Oracle Perfect Balance 2.2</li></ul>

## Download Oracle Big Data Lite Virtual Machine

You must accept the [License Agreement for Oracle Big Data Lite](#) to download this software.

**Accept** License Agreement  **Decline** License Agreement

File	Description
Deployment Guide	<p><b>Start Here!</b></p> <p>Deployment Guide provides step-by-step instructions for download and deployment.</p> <p><b>Technical Requirements:</b></p> <ul style="list-style-type: none"><li>▪ Dedicate 2 cores, 5 GB memory and 30GB disk space to the virtual machine</li><li>▪ Install will require ~45GB disk space including temporary files</li></ul>
 <a href="#">BigDataLite.7z.001</a> (2147483648 bytes)  <a href="#">BigDataLite.7z.002</a> (2147483648 bytes)  <a href="#">BigDataLite.7z.003</a> (2147483648 bytes)  <a href="#">BigDataLite.7z.004</a> (2147483648 bytes)  <a href="#">BigDataLite.7z.005</a> (2147483648 bytes)  <a href="#">BigDataLite.7z.006</a> (2147483648 bytes)  <a href="#">BigDataLite.7z.007</a> (607107322 bytes) <a href="#">md5sum.txt</a> (346 bytes)	<p><b>To get started:</b></p> <ul style="list-style-type: none"><li>▪ Download and install <a href="#">Oracle VM VirtualBox</a> and <a href="#">7-zip</a></li><li>▪ Download each of the 7-zip files</li><li>▪ Run the 7-zip extractor on the BigDataLite.7z.001 file only. This will create the BigDataLite-4.0.1.ova VirtualBox appliance file</li><li>▪ In VirtualBox, import BigDataLite-4.0.1.ova</li><li>▪ Start BigDataLite-4.0.1</li><li>▪ Log in as <code>oracle/welcome1</code></li></ul> <p>See the <a href="#">Deployment Guide</a> for details.</p>

## Cloudera JDBC Drivers

Download and install the Cloudera JDBC drivers to enable Oracle SQL Developer and Data Modeler to connect to Hive.

## Getting Started

You can load your own data into the VM or use the [Oracle MoviePlex](#) demo data that's provided. Oracle has created videos, sample code and hands-on labs based on Oracle MoviePlex that will help you learn how to develop big data applications using Oracle's big data platform. All of the collateral used to develop this application is included in the VM.

### Oracle MoviePlex

Oracle MoviePlex is a fictitious on-line movie streaming company. Customers log into Oracle MoviePlex where they are presented with a targeted list of movies based on their past viewing behavior. Because of this personalized experience and reliable and fast performance, customers spend a lot of money with the company and it has become extremely profitable :).

How do you harness big data to create a personalized experience for customers? Check out the videos listed below that highlight how Oracle MoviePlex tackled this challenge using Oracle's big data solution:

- Part 1. Overview: Improve the Customer Experience (10 min)
- Part 2. Deliver a Personalized Service - Oracle MoviePlex Application (5 min)
- Part 3. Manage Online Profiles w/Oracle NoSQL DB (5 min)
- Part 4. Turn Clicks into Value - Flume & Hive (5 min)
- Part 5. Integrate All Your Data with Oracle Big Data Connectors (8 min)
- Part 6. Maximize the Business Impact with Oracle Advanced Analytics (8 min)

## Hands-on Labs

There are several hands-on-labs available to help you get started with the platform:

Training Collateral	Description
<a href="#">Oracle Big Data Learning Library</a>	Numerous hands on labs and videos for capabilities that span the Oracle Big Data platform.
<a href="#">Analyze All Your Data with Oracle Big Data SQL</a>	Learn how to securely analyze all your data - across both Hadoop and Oracle Database 12c - using Oracle Big Data SQL
<a href="#">Introduction to Oracle NoSQL Database</a>	Use Oracle NoSQL Database in the context of the Oracle MoviePlex application. Create schemas, load data and then utilize that data in the online movie application.
<a href="#">Oracle NoSQL Database - Installation/cluster topology deployment (pdf   scripts)</a>	Learn how simple and intuitive it is to deploy a highly available (production ready) Oracle NoSQL Database cluster.
<a href="#">Access Data in Oracle NoSQL Database from Oracle Database</a>	Access data in Oracle NoSQL Database from Oracle Database 12c using external tables.
<a href="#">Data Manipulation with Hive and Pig (pdf   script)</a>	Quick introduction to HDFS, Pig and Hive.
<a href="#">Tame Big Data with Oracle Data Integration</a>	Learn about how to design Hadoop data integration using Oracle Data Integrator and Oracle GoldenGate.

<a href="#">Integrate Hadoop Data with Oracle Database using Oracle Big Data Connectors</a>	Use Oracle Loader for Hadoop to efficiently load data into the Oracle Database using MapReduce jobs. Access data in HDFS directly from the Oracle Database using Oracle SQL Connector for Hadoop.
<a href="#">Using SQL Pattern Matching</a>	This series features both OBEs and recorded webcasts. Learn how to use SQL for Pattern Matching. Row pattern matching in native SQL improves application and development productivity and query efficiency for row-sequence analysis.
<a href="#">Oracle Data Mining 12c Tutorial Series</a>	The OBE's in this series provide you with instructions on how to perform data mining with Oracle Database 12c, by using Oracle Data Miner 4.0. Oracle Data Miner 4.0 is included as an extension of Oracle SQL Developer, version 4.0.
<a href="#">Oracle R Enterprise v 1.4 - Tutorial Series</a>	Oracle R Enterprise (ORE), a component of the Oracle Advanced Analytics Option, makes the open source R statistical programming language and environment ready for the enterprise and big data.  This series teaches you how to use Oracle R Enterprise, version 1.4.

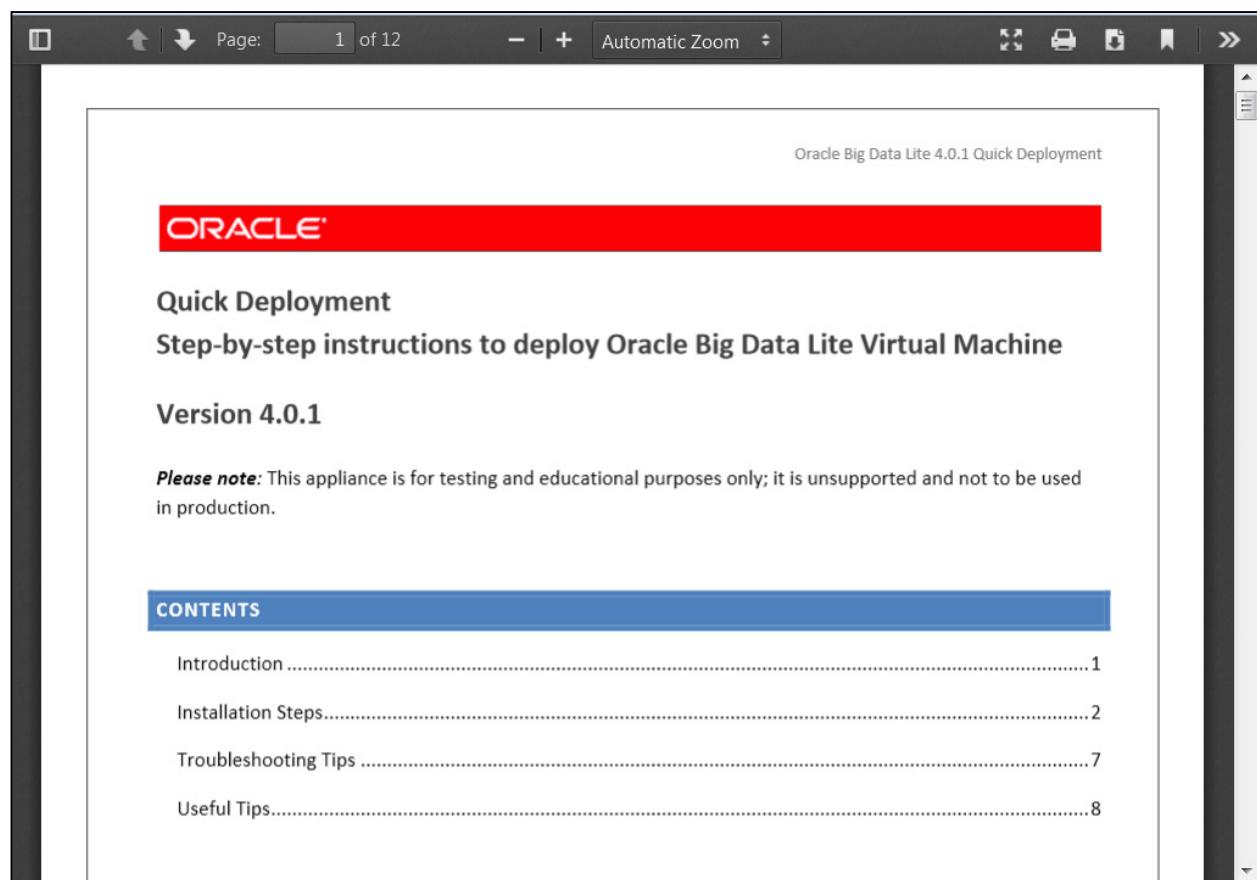
### Web Sites / White Papers / EBook / Blogs

Listed below are some resources to help you learn more about the Oracle big data platform:

- [Big Data on oracle.com](#) and [Oracle Technology Network](#)
- [The Data Warehouse Insider Blog](#) - Technical details, ideas and news on data warehousing and big data from the Oracle Team
- [Connecting Hadoop with Oracle Database Blog](#) - learn how to optimize connectivity between Hadoop and Oracle Database
- [Oracle NoSQL Blog](#) -this blog is about "everything NoSQL"
- [Oracle DW-Big Data Weekly Roundup](#)
- [White Paper: Oracle: Big Data for the Enterprise \(PDF\)](#)
- [Oracle Big Data Interactive E-Book](#)
- [Oracle Big Data Handbook \(Oracle Press\)](#)
- [Big Data Appliance Networking for the Data Center](#)

2. Review the Deployment Guide. Click the Deployment Guide link in the File column in the Download Oracle Big Data Lite Virtual Machine section.

Download Oracle Big Data Lite Virtual Machine	
You must accept the License Agreement for Oracle Big Data Lite to download this software.	
<input checked="" type="radio"/> Accept License Agreement <input type="radio"/> Decline License Agreement	
File	Description
<a href="#">Deployment Guide</a>	<b>Start Here!</b> Deployment Guide provides step-by-step instructions for download and deployment.  <b>Technical Requirements:</b> <ul style="list-style-type: none"><li>▪ Dedicate 2 cores, 5 GB memory and 30GB disk space to the virtual machine</li><li>▪ Install will require ~45GB disk space including temporary files</li></ul>



The screenshot shows a PDF document titled "Oracle Big Data Lite 4.0.1 Quick Deployment". The document has a red header bar with the word "ORACLE". The main content includes the title "Quick Deployment", subtitle "Step-by-step instructions to deploy Oracle Big Data Lite Virtual Machine", and version "Version 4.0.1". A note states: "Please note: This appliance is for testing and educational purposes only; it is unsupported and not to be used in production." Below this is a "CONTENTS" section with the following table of contents:

Introduction .....	1
Installation Steps.....	2
Troubleshooting Tips .....	7
Useful Tips.....	8

# **Practices for Lesson 2: Big Data and the Oracle Information Management System**

## **Chapter 2**

## Practices for Lesson 2

---

There are no practices for this lesson.

# **Practices for Lesson 3: Using Oracle Big Data Lite Virtual Machine**

## **Chapter 3**

## Practices for Lesson 3

---

### Practices Overview

In the first practice, you connect to your VM and log in to various tools that you will use in this course. You use a terminal window to practice some basic hadoop HDFS commands. You also log into Hue and R. You then load the library for Oracle R Enterprise and the library for Oracle R Connector for Hadoop. You also log in to pig and hive.

In the second practice, you explore some of the available scripts that you can use to reset and start the Movieplex application. You also explore the Movieplex application and the available tools on your desktop such as the Start/Stop Services tool which enables you to stop and start specific services on your BDA Lite.

## Practice 3-1: Using the BigData Lite Virtual Machine (VM)

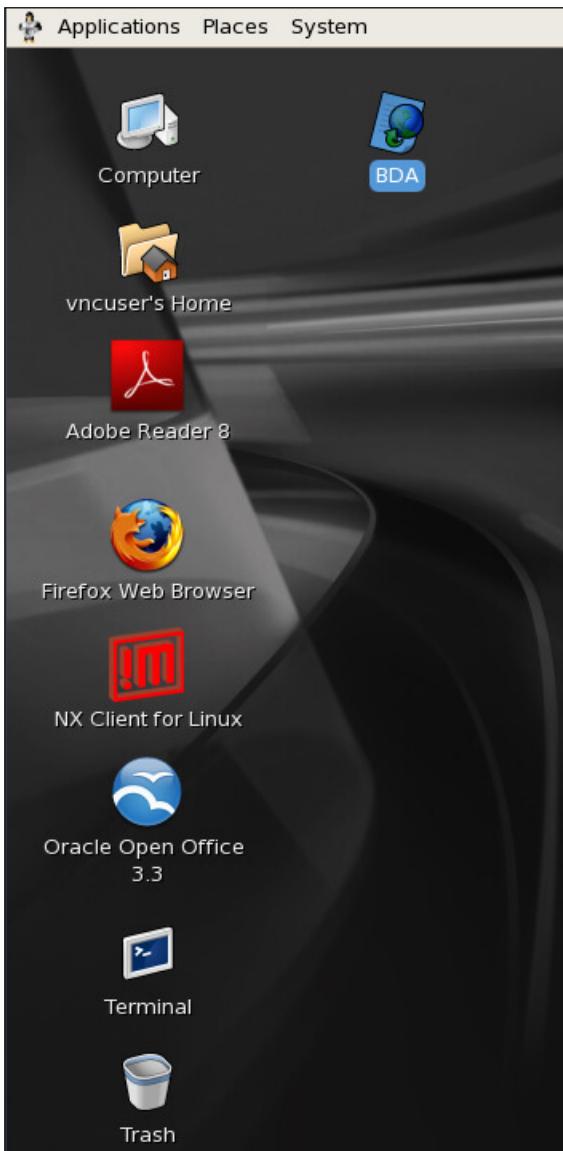
### Overview

In this practice you create an Oracle NoSQL Database instance and register the schemas that are used for the MoviePlex application.

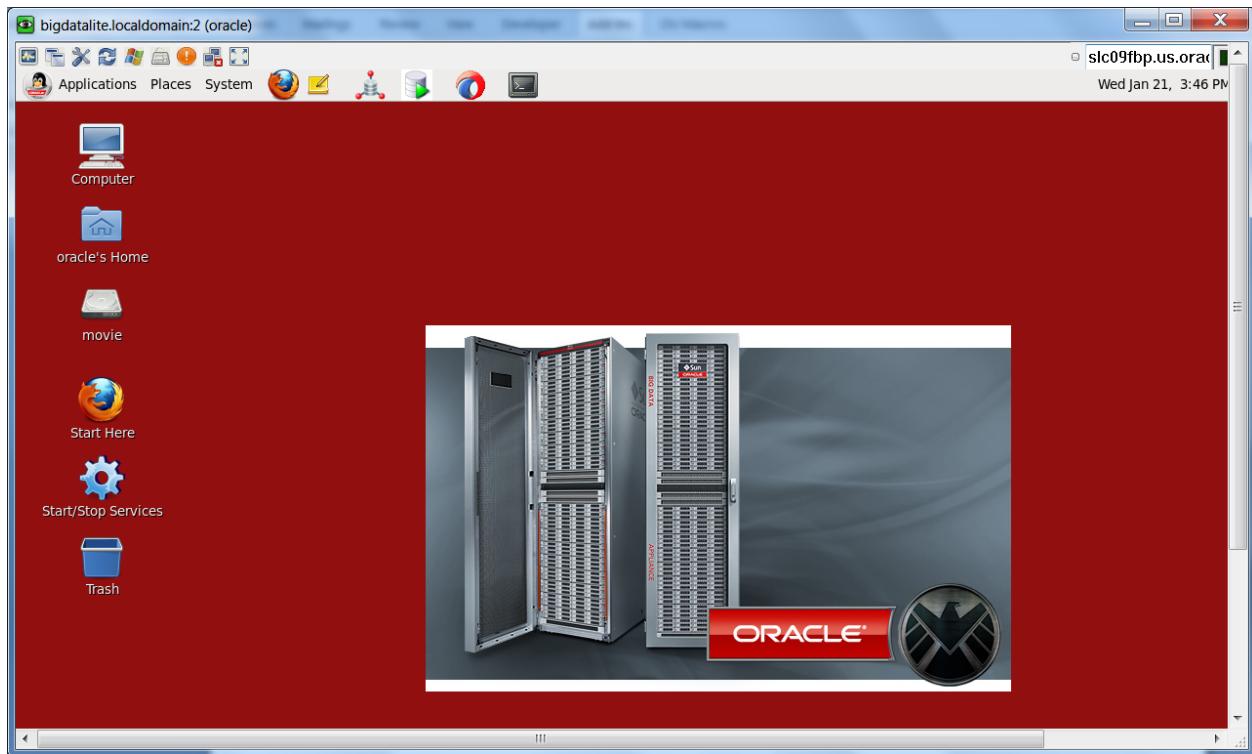
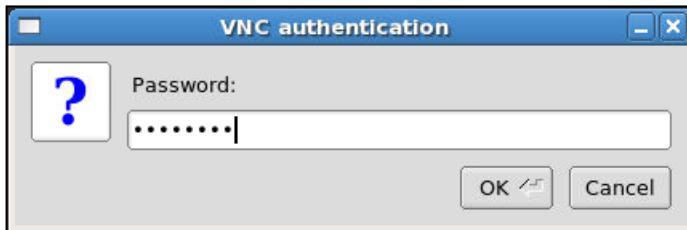
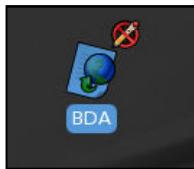
### Assumptions

### Tasks

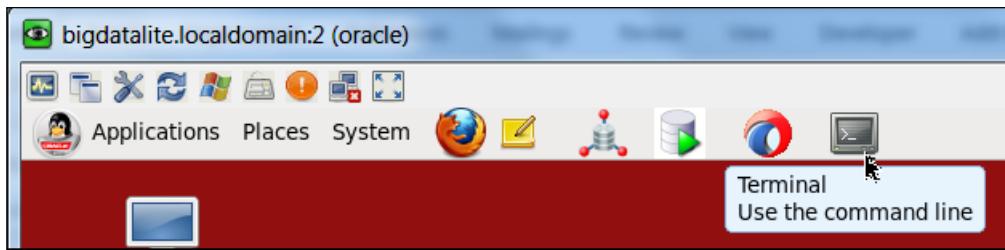
1. Connect to your classroom machine assigned to you by your instructor. Your instructor will provide you with the credentials that you will need to connect to your machine. Your host machine Desktop is displayed.



2. To log in to the Oracle Big Data Lite (BDLite) Virtual Machine (VM), double-click the **BDA** icon on your desktop. The Oracle Big Data Lite VM is displayed. When prompted to enter a password, enter `welcome1`, and then click **OK**. The BDLite VM Desktop is displayed.



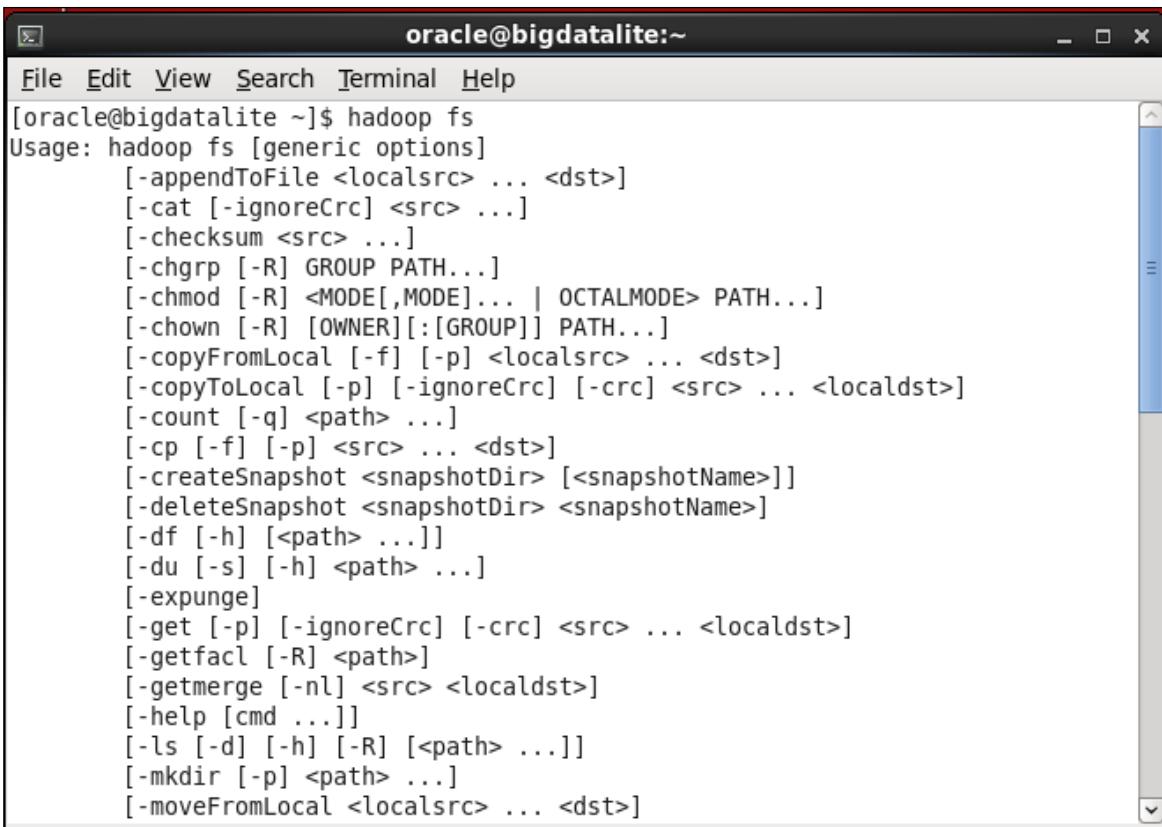
3. Open a terminal window.



A screenshot of a terminal window titled "oracle@bigdatalite:~". The window has a standard Linux-style interface with a menu bar (File, Edit, View, Search, Terminal, Help) and a command line area. The command line shows the prompt "[oracle@bigdatalite ~]\$".

4. You will use the `hadoop` command to interact with HDFS. You will review some of the basic available commands in HDFS. Notice that the commands are similar to your local Linux file system commands. Open a terminal window, and then enter the following command at the \$ command prompt:

```
hadoop fs
```

A screenshot of a terminal window titled "oracle@bigdatalite:~". The command entered is "hadoop fs". The terminal displays the usage information for the "fs" command, which includes various sub-commands such as -appendToFile, -cat, -checksum, -chgrp, -chmod, -chown, -copyFromLocal, -copyToLocal, -count, -cp, -createSnapshot, -deleteSnapshot, -df, -du, -expunge, -get, -getfacl, -getmerge, -help, -ls, -mkdir, and -moveFromLocal.

```
[-moveFromLocal <localsrc> ... <dst>]
[-moveToLocal <src> <localdst>]
[-mv <src> ... <dst>]
[-put [-f] [-p] <localsrc> ... <dst>]
[-renameSnapshot <snapshotDir> <oldName> <newName>]
[-rm [-f] [-r|-R] [-skipTrash] <src> ...]
[-rmdir [--ignore-fail-on-non-empty] <dir> ...]
[-setfacl [-R] [{-b|-k} {-m|-x} <acl_spec>] <path>|[--set <acl_spec> <path>]]
[-setrep [-R] [-w] <rep> <path> ...]
[-stat [format] <path> ...]
[-tail [-f] <file>]
[-test -[defsz] <path>]
[-text [-ignoreCrc] <src> ...]
[-touchz <path> ...]
[-usage [cmd ...]]]

Generic options supported are
-conf <configuration file>          specify an application configuration file
-D <property=value>                  use value for given property
-fs <local|namenode:port>            specify a namenode
-jt <local|jobtracker:port>          specify a job tracker
-files <comma separated list of files>    specify comma separated files to be copied to the map reduce cluster
-libjars <comma separated list of jars>    specify comma separated jar files to include in the classpath.
-archives <comma separated list of archives>  specify comma separated archives to be unarchived on the compute machines.

The general command line syntax is
bin/hadoop command [genericOptions] [commandOptions]

[oracle@bigdatalite ~]$ █
```

5. Use the `hadoop fs -ls` command to list the contents of the `/user/oracle` HDFS directory.

```
hadoop fs -ls /user/oracle
```

```
[oracle@bigdatalite ~]$ hadoop fs -ls /user/oracle
Found 6 items
drwx-----  - oracle oracle          0 2014-08-25 05:55 /user/oracle/.Trash
drwx-----  - oracle oracle          0 2015-01-21 15:13 /user/oracle/.staging
drwxr-xr-x  - oracle oracle          0 2014-01-12 18:15 /user/oracle/moviedemo
drwxr-xr-x  - oracle oracle          0 2014-09-24 09:38 /user/oracle/moviework
drwxr-xr-x  - oracle oracle          0 2014-09-08 15:50 /user/oracle/oggdemo
drwxr-xr-x  - oracle oracle          0 2014-09-20 13:59 /user/oracle/oozie-oozi
[oracle@bigdatalite ~]$ █
```

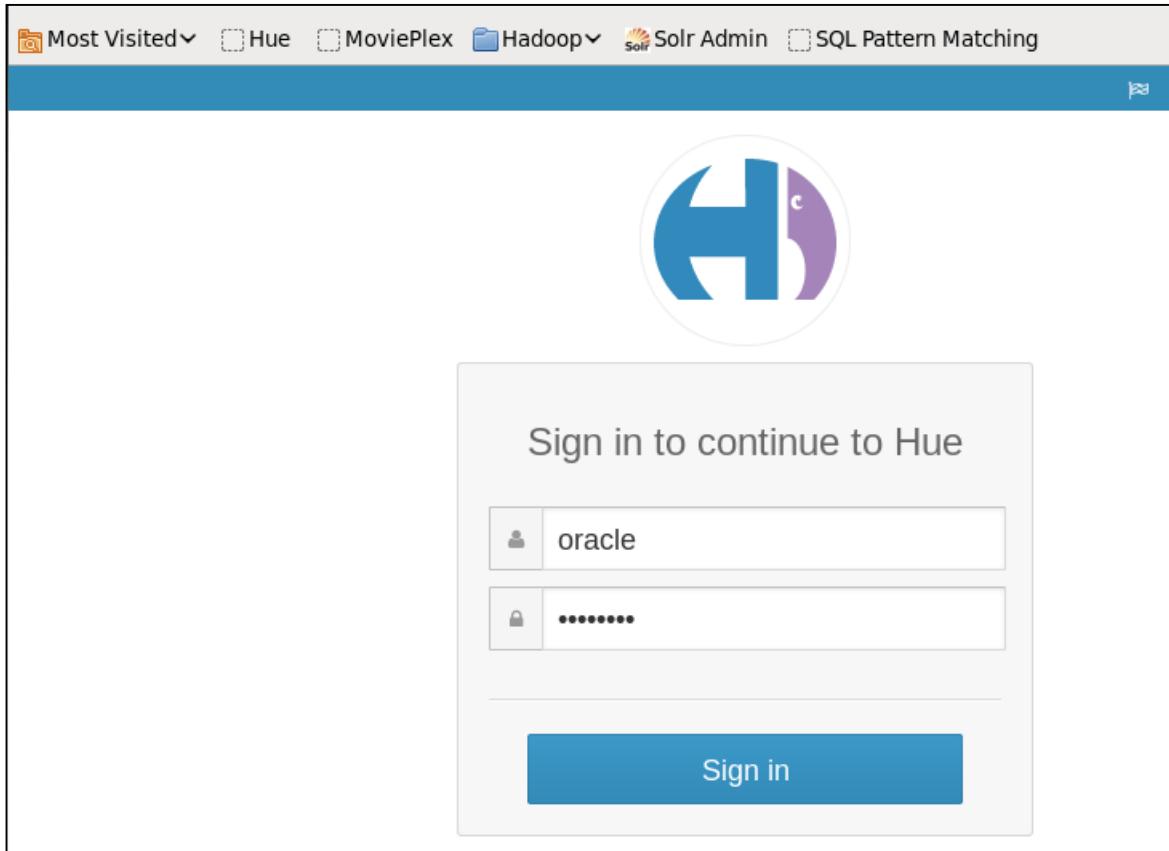
- ## 6. Run the command to check the hadoop version:

## **hadoop version**

```
[oracle@bigdatalite ~]$ hadoop version
Hadoop 2.3.0-cdh5.1.2
Subversion git://github.sf.cloudera.com/CDH/cdh.git -r 8e266e052e423af592871e2dfe09d54c03f6a0e8
Compiled by jenkins on 2014-08-26T01:36Z
Compiled with protoc 2.5.0
From source with checksum ec11b8ec19ca2bf3e7cb1bbe4ee182
This command was run using /usr/lib/hadoop/hadoop-common-2.3.0-cdh5.1.2.jar
[oracle@bigdatalite ~]$ █
```

7. Start up Hue.

- a. Start your Mozilla Firefox Web browser and access Hue using the following url: <http://localhost:8888> or you can use the saved **Hue** bookmark on the Bookmarks Toolbar.
- b. Enter oracle for the User name, if not already entered.
- c. Enter welcome1 for the password, if not already entered.
- d. Click **Sign in**



8. Start R. Enter the following command in a terminal window:

```
R
```

The screenshot shows a terminal window titled "oracle@bigdatalite:~". The user has typed "R" at the prompt, which is highlighted with a red box. The terminal displays the standard Oracle R startup message, including the version (3.1.1), copyright information, platform (x86\_64-unknown-linux-gnu), and various usage instructions. The message ends with a prompt "> █".

```
oracle@bigdatalite:~$ R
Oracle Distribution of R version 3.1.1  (--) -- "Sock it to Me"
Copyright (C)  The R Foundation for Statistical Computing
Platform: x86_64-unknown-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

You are using Oracle's distribution of R. Please contact
Oracle Support for any problems you encounter with this
distribution.

> █
```

9. Load the library for Oracle R Enterprise using the following command. This library enables you to connect R with the Oracle Database and move data between the two environments, as well as calling R functions from the database and vice versa.

```
> library (ORE)
```

```
> library (ORE)
Loading required package: OREbase

Attaching package: 'OREbase'

The following objects are masked from 'package:base':

  cbind, data.frame, eval, interaction, order, paste, pmax, pmin,
  rbind, table

Loading required package: OREembed
Loading required package: OREstats
Loading required package: MASS
Loading required package: OREgraphics
Loading required package: OREeda
Loading required package: OREmodels
Loading required package: OREdm
Loading required package: lattice
Loading required package: OREPredict
Loading required package: ORExml
> █
```

10. Load the library for Oracle R Connector for Hadoop using the following command. This library enables the movement of data from and to HFDS. It has many different functions for combining R with Hadoop. When the command is executed successfully, close your terminal window.

```
> library (ORCH)
```

```
> library (ORCH)
Loading required package: ORCHcore
Oracle R Connector for Hadoop 2.4.1 (rev. 307)
Info: using native C base64 encoding implementation
Info: Hadoop distribution is Cloudera's CDH v5.1.2
Info: using auto-detected ORCH HAL v4.2
Info: HDFS workdir is set to "/user/oracle"
Info: mapReduce is functional
Info: HDFS is functional
Info: Hadoop 2.3.0-cdh5.1.2 is up
Info: Sqoop 1.4.4-cdh5.1.2 is up
Info: OLE 3.2.0 is up
Info: Hive 0.12.0-cdh5.1.2 is up
Loading required package: ORCHstats
> █
```

11. Start up Pig in a new terminal window. Enter the following command to login into the PIG interpreter. Enter `quit` the `grunt>` command prompt when you are done to exit pig.

```
pig
```

A screenshot of a terminal window titled "oracle@bigdatalite:~". The window shows the output of the "pig" command. The first line of output is "[oracle@bigdatalite ~]\$ pig", with "pig" highlighted by a red box. The subsequent lines are Apache Pig version logs from January 26, 2015, at 13:58:56. The logs mention org.apache.pig.Main, org.apache.hadoop.conf.Configuration.deprecation, and org.apache.pig.backend.hadoop.executionengine.HExecutionEngine. The logs also mention the hadoop file system at: hdfs://bigdatalite.localdomain:8020. The final line is "grunt>".

```
[oracle@bigdatalite ~]$ pig
2015-01-26 13:58:56,355 [main] INFO  org.apache.pig.Main - Apache Pig version 0.
12.0-cdh5.1.2 (rexported) compiled Aug 25 2014, 19:51:44
2015-01-26 13:58:56,355 [main] INFO  org.apache.pig.Main - Logging error messages to: /home/oracle/pig_1422298736353.log
2015-01-26 13:58:56,375 [main] INFO  org.apache.pig.impl.util.Utils - Default bootstrap file /home/oracle/.pigbootup not found
2015-01-26 13:58:56,751 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.address
2015-01-26 13:58:56,751 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.defaultFS
2015-01-26 13:58:56,751 [main] INFO  org.apache.pig.backend.hadoop.executionengine.HExecutionEngine - Connecting to hadoop file system at: hdfs://bigdatalite.lo
caldomain:8020
2015-01-26 13:58:57,693 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.defaultFS
grunt>
```

```
quit
```

12. Use the `pig -help` command to print a list of Pig commands.

```
pig -help
```

```
[oracle@bigdatalite ~]$ pig -help

Apache Pig version 0.12.0-cdh5.1.2 (rexported)
compiled Aug 25 2014, 19:51:44

USAGE: Pig [options] [-] : Run interactively in grunt shell.
      Pig [options] -e[xcute] cmd [cmd ...] : Run cmd(s).
      Pig [options] [-f[file]] file : Run cmds found in file.

options include:
  -4, -log4jconf - Log4j configuration file, overrides log conf
  -b, -brief - Brief logging (no timestamps)
  -c, -check - Syntax check
  -d, -debug - Debug level, INFO is default
  -e, -execute - Commands to execute (within quotes)
  -f, -file - Path to the script to execute
  -g, -embedded - ScriptEngine classname or keyword for the ScriptEngine
  -h, -help - Display this message. You can specify topic to get help for that topic.
    properties is the only topic currently supported: -h properties.
  -i, -version - Display version information
  -l, -logfile - Path to client side log file; default is current working directory.
  -m, -param_file - Path to the parameter file
  -p, -param - Key value pair of the form param=val
  -r, -dryrun - Produces script with substituted parameters. Script is not executed.
  -t, -optimizer_off - Turn optimizations off. The following values are supported:
    SplitFilter - Split filter conditions
    PushUpFilter - Filter as early as possible
    MergeFilter - Merge filter conditions
    PushDownForEachFlatten - Join or explode as late as possible
    LimitOptimizer - Limit as early as possible
    ColumnMapKeyPrune - Remove unused data
    AddForEach - Add ForEach to remove unneeded columns
    MergeForEach - Merge adjacent ForEach
    GroupByConstParallelSetter - Force parallel 1 for "group all" statement
    All - Disable all optimizations
  All optimizations listed here are enabled by default. Optimization values are case insensitive.
  -v, -verbose - Print all error messages to screen
  -w, -warning - Turn warning logging on; also turns warning aggregation off
  -x, -execype - Set execution mode: local|mapreduce, default is mapreduce.
  -F, -stop_on_failure - Aborts execution on the first failed job; default is off
  -M, -no_multiquery - Turn multiquery optimization off; default is on
  -P, -propertyFile - Path to property file
  -printCmdDebug - Overrides anything else and prints the actual command used to run P
```

### 13. Exit the terminal window.

```
$ exit
```

## Practice 3-2: Using the Oracle MoviePlex Application

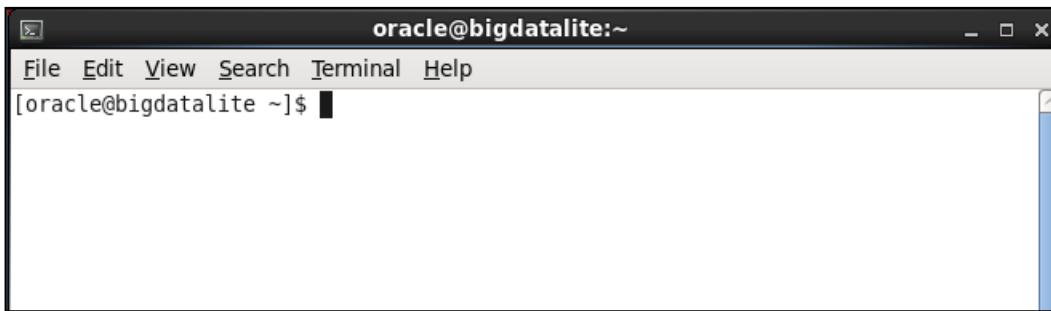
### Overview

In this practice, you run the script to start the Oracle MoviePlex application, and then log in to the application and explore it.

### Assumptions

### Tasks

1. Explore the `1_start_movieapp.sh` script in the `/home/oracle/movie/moviedemo/scripts` folder.
  - a. Open a new terminal window.



- b. Navigate to the `/home/oracle/movie/moviedemo/scripts` folder and display the contents of the `scripts` folder.

```
cd /home/oracle/movie/moviedemo/scripts  
ls -ls
```

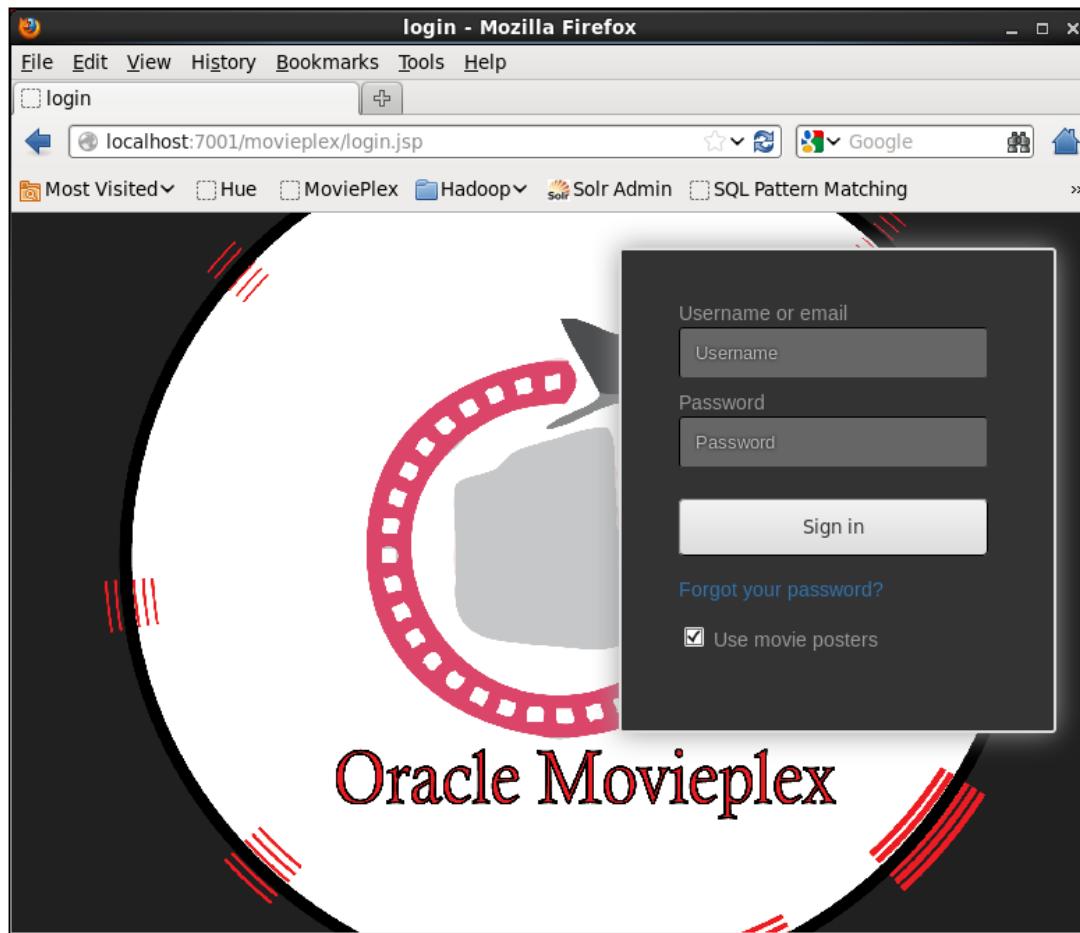
- c. Use the `more` command to display the contents of the `1_start_movieapp.sh` script.

```
[oracle@bigdatalite scripts]$ more 1_start_movieapp.sh  
echo Starting MoviePlex application using KVStore latest state  
cd /home/oracle/movie/moviedemo/nosqlbd/scripts  
. ./startDemoKeepKVState.sh  
echo Go to Firefox and wait for application to start.  
echo http://127.0.0.1:7001/bigdatademo-UI-context-root/index.jsp  
[oracle@bigdatalite scripts]$
```

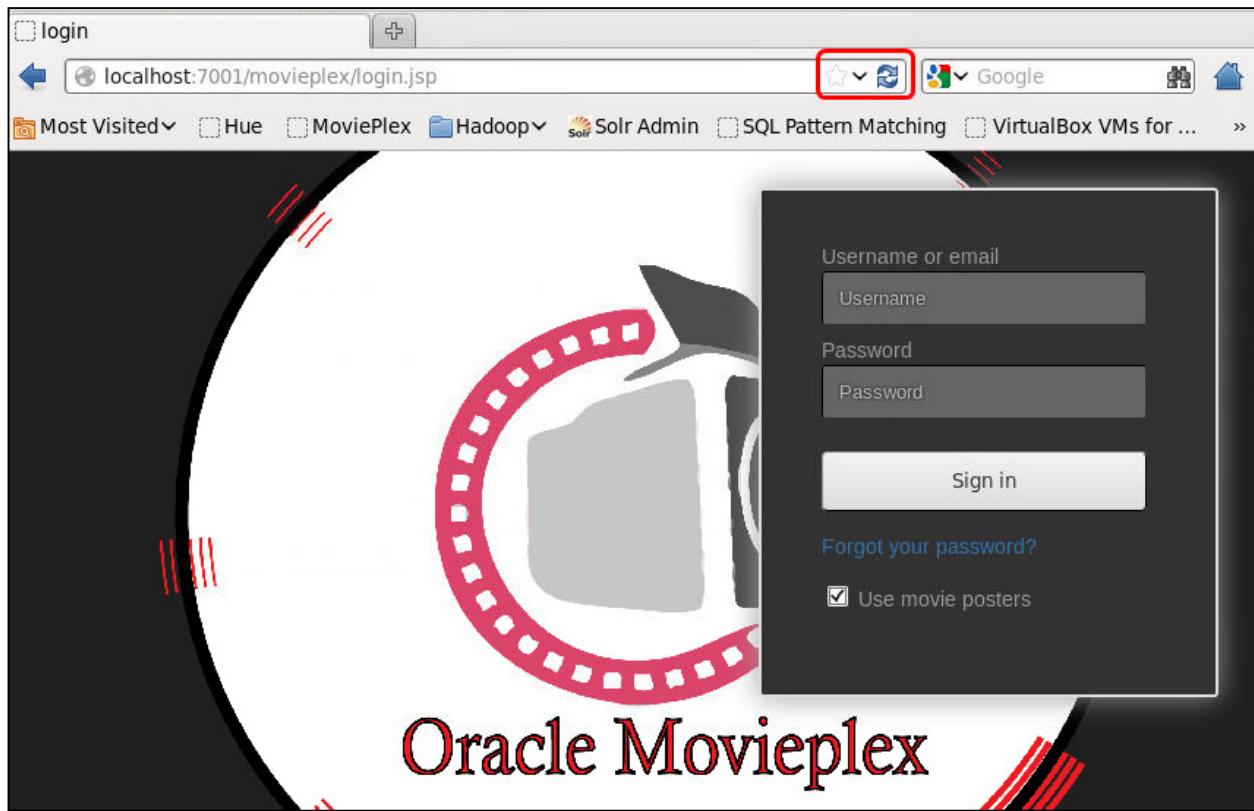
- d. Run the `1_start_movieapp.sh` script.

```
./1_start_movieapp.sh
```

2. Open your Mozilla Firefox Web browser and connect to the Oracle MoviePlex demo application by using the following URL:<http://localhost:7001/movieplex/index.jsp>

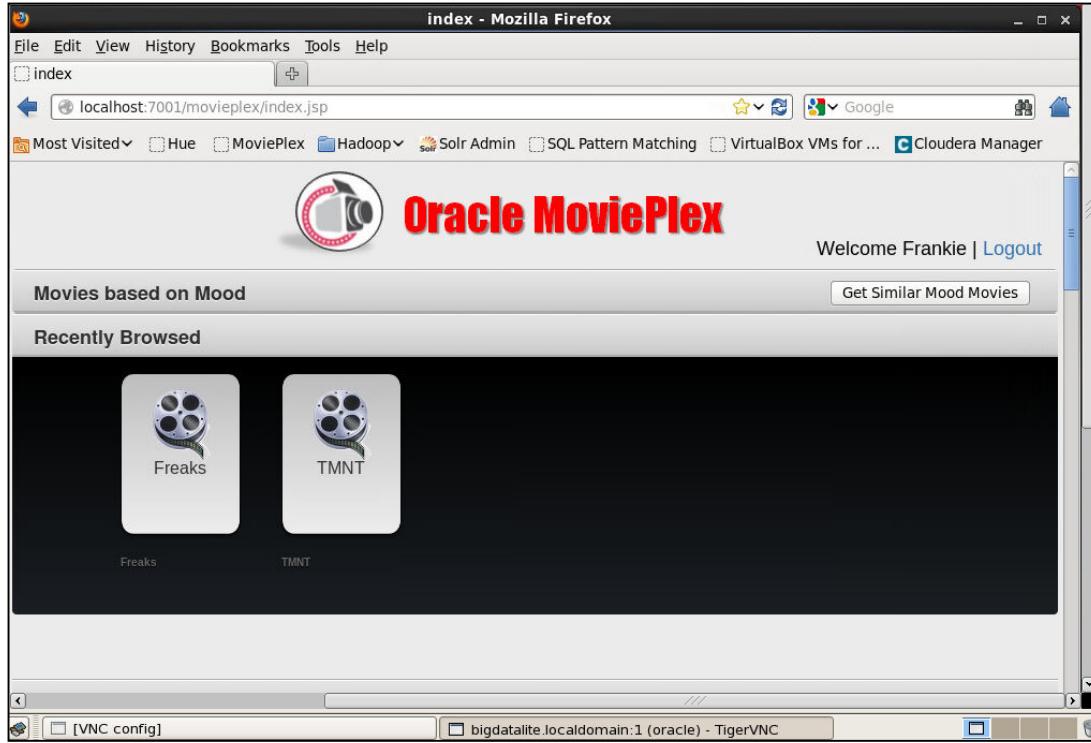
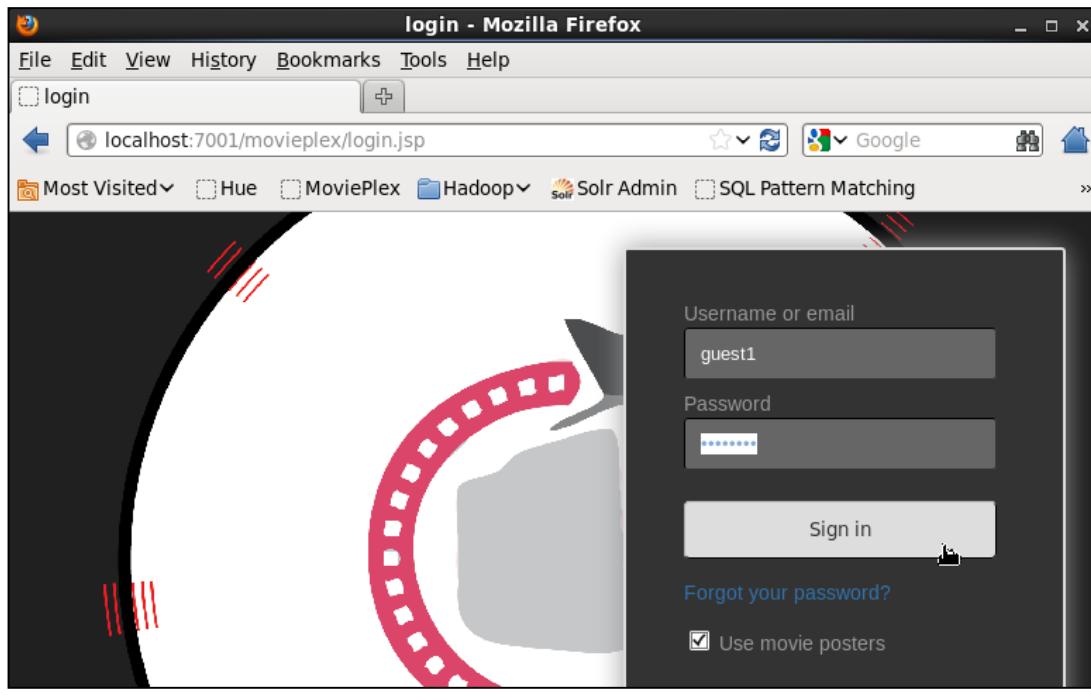


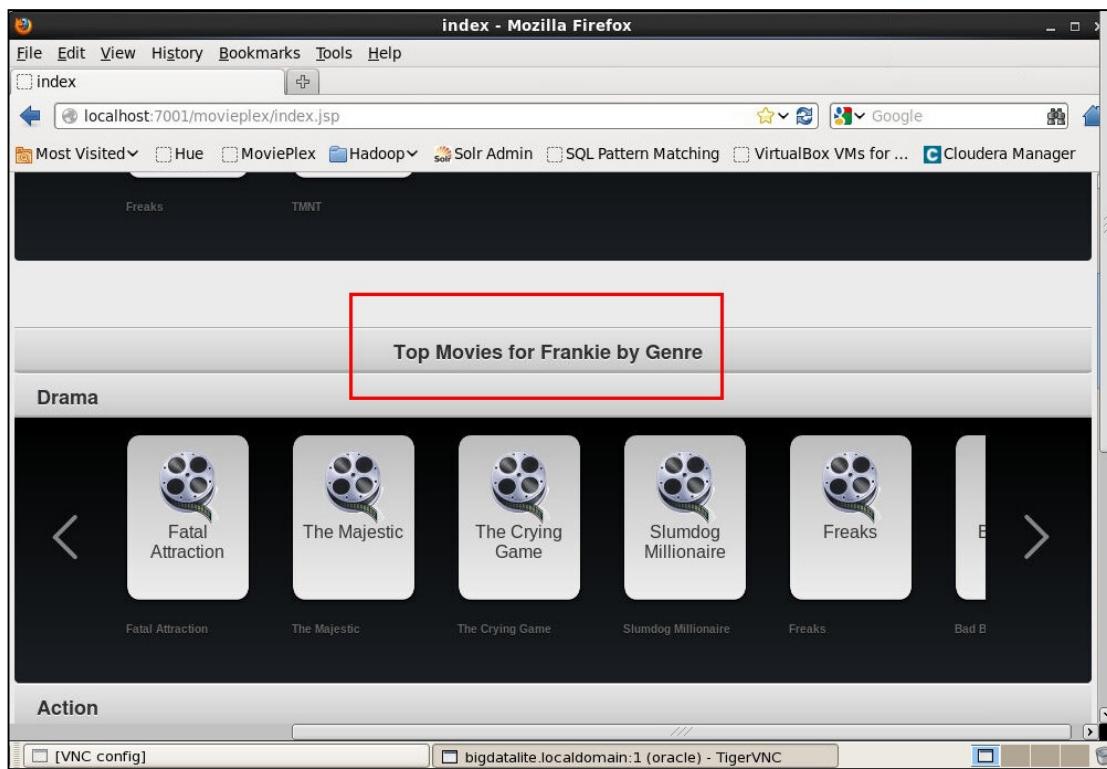
**Note:** You might need to wait 10-20 seconds after you start the script before the Oracle Movieplex application login screen is displayed. You can click the "**Reload current page**" icon on the address bar in your browser few times.



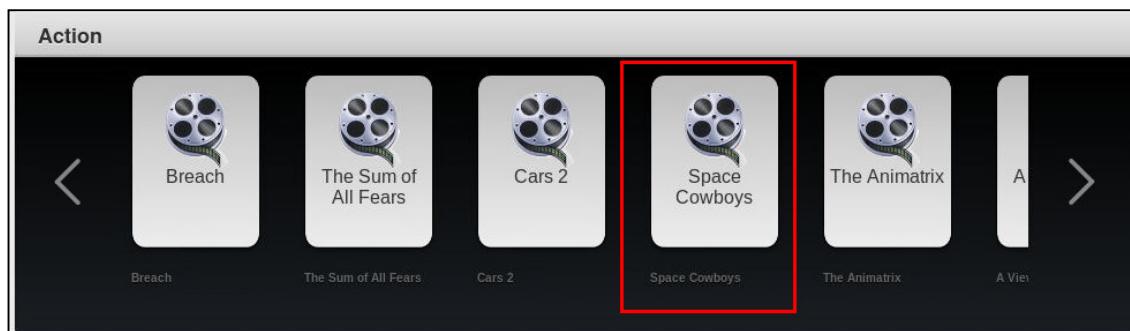
3. Enter guest1 in the "Username or email" text box, and welcome1 in the Password text box, and then click "Sign in." The Oracle Movieplex application is displayed. You can also use guest2 through guest100 in the "Username or email" text box with the welcome1 password.

**Note: You will not be able to play the movies or view the movies' icons in the classroom environment because of the Oracle University firewall settings.**





4. The application displays the "Top Movies for Frankie (guest1) by Genere". You can scroll down to see the various available movies in each genre.
5. On the left side of each genre, you will see a ">" arrow which enables you to see more movies. In the Action genre, scroll to the right and click the "Space Cowboys" movie. This displays the movie's synopsis. Click the play button on the icon. You will get the following expected message because of the Oracle University firewall: "This video does not exist." Close the window or a red X is displayed. Close the "Space Cowboys (2000)" window to return to the Movieplex home page.



**Space Cowboys (2000)**

Rent for \$1.99



Space Cowboys

**Overview:** Space Cowboys is a 2000 space drama film directed and produced by Clint Eastwood. Eastwood also stars in the film alongside Tommy Lee Jones, Donald Sutherland, and James Garner as four older "ex-test pilots" who are sent into space to repair an old Soviet satellite. The original music score was composed by Eastwood and Lennie Niehaus.

**Cast:** Tommy Lee Jones, Courtney B. Vance, Donald Sutherland, William Devane, Clint Eastwood, Loren Dean

**Director:** Clint Eastwood

**Writer:** Ken Kaufman, Howard Klausner

★ ★ ★ ★

**Space Cowboys (2000)**

Rent for \$1.99



Space Cowboys

**Overview:** Space Cowboys is a 2000 space drama film directed and produced by Clint Eastwood. Eastwood also stars in the film alongside Tommy Lee Jones, Donald Sutherland, and James Garner as four older "ex-test pilots" who are sent into space to repair an old Soviet satellite. The original music score was composed by Eastwood and Lennie Niehaus.

**Cast:** Tommy Lee Jones, Courtney B. Vance, Donald Sutherland, William Devane, Clint Eastwood, Loren Dean

**Director:** Clint Eastwood

**Writer:** Ken Kaufman, Howard Klausner

★ ★ ★ ★

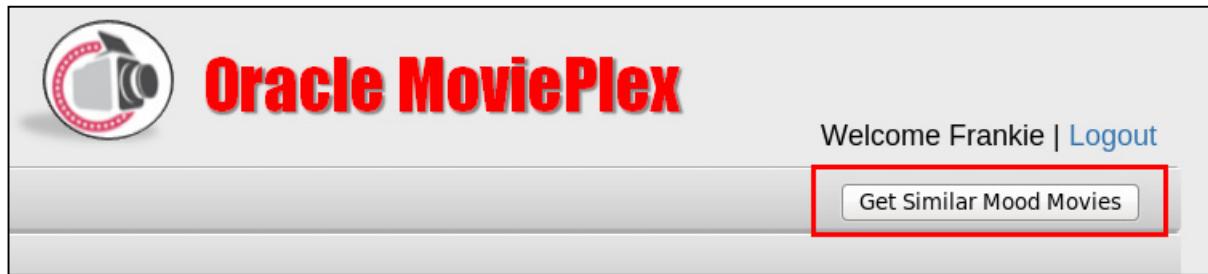
6. Notice that there is a new "Recently Browsed" section showing the movie that you just browsed. If you run this Movieplex application locally on your own machine, then you can play the movie and if you stop watching the movie, you will see a new "Continue Watching" section which will show you the movies that you started but did not complete. This will enable you to continue watching any of those movies. At the bottom of the screen, you should see a section of the movies that you have watched and completed.

The screenshot shows the Oracle MoviePlex website. At the top is a camera icon and the text "Oracle MoviePlex". To the right are links for "Welcome Frankie | Logout". Below the header is a section titled "Movies based on Mood" with a "Get Similar Mood Movies" button. Underneath is a "Recently Browsed" section. Three movie cards are displayed: "Space Cowboys" (highlighted with a red box), "Freaks", and "TMNT". Each card has a thumbnail of a film reel and the movie title below it.

7. You can rate the movies that you browsed or watched. Click the movie that you browsed, and then rate it.

The screenshot shows the movie details for "Space Cowboys (2000)". The title is at the top with a close button. Below it is a rental offer "Rent for \$1.99" with a play button icon. To the right is the "Overview": "Space Cowboys is a 2000 space drama film directed and produced by Clint Eastwood. Eastwood also stars in the film alongside Tommy Lee Jones, Donald Sutherland, and James Garner as four older "ex-test pilots" who are sent into space to repair an old Soviet satellite. The original music score was composed by Eastwood and Lennie Niehaus." Below the overview are the "Cast", "Director", and "Writer". At the bottom is a rating section with five yellow stars and a thumbs-up icon, which is highlighted with a red box.

8. Finally, scroll up to the top of the screen and look at movies based on your current mood. Click "Get Similar Mood Movies". Even if you have a relatively consistent taste, things can be different each time you log in to the application. You might be interested in different movies if you are watching alone or with someone else. The application can start to figure out what you are interested in now as opposed to in general and it would recommend something that is better suited for your current mood. This is an example on how making the right offers at the right time can greatly improve the likely hood for success. Another key for success for any high volume service is to deliver information economically and with minimum latency.



9. A new "Movies based on Mood" section is displayed with the movies recommendation for your current mood.

The screenshot shows the Oracle MoviePlex application running in a browser. The URL in the address bar is "localhost:7001/movieplex/index.jsp". The page displays the "Oracle MoviePlex" logo and a "Welcome Frankie | Logout" link. A prominent section titled "Movies based on Mood" is shown, with a red rectangular box highlighting its title. Below this section is a grid of movie cards, each featuring a film reel icon and the movie title: "Mrs. Doubtfire", "The King's Speech", "Ocean's Twelve", "Chicago", and "The Full Monty". Navigation arrows ("<" and ">") are positioned on either side of the grid. Below the grid, under the heading "Recently Browsed", are three small film reel icons.

10. Exit the Movieplex application. Click Logout.



## **Practices for Lesson 4: Introduction to the Big Data Ecosystem**

**Chapter 4**

## Practices for Lesson 4

---

There are no practices for this lesson.

## **Practices for Lesson 5: Introduction to the Hadoop Distributed File System (HDFS)**

**Chapter 5**

## Practices for Lesson 5

---

### Practices Overview

In this practice, you review HDFS commands. You also load an AVRO log file that tracked activity in an online movie application into HDFS.

## Practice 5-1: Introduction to HDFS Commands

---

### Overview

In this practice, you review HDFS commands. You also load an Avro log file that tracked activity in an online movie application into HDFS.

### Assumptions

### Tasks

1. Start a terminal window.
2. Review the commands that are available for the Hadoop Distributed File System. You will find that its composition is similar to the local Linux file system. You will use the `hadoop fs` command when interacting with HDFS. Spend a few minutes reviewing the syntax for some of the useful commands that you will use in HDFS such as `-cat`, `-copyFromLocal`, `-copyToLocal`, `-cp`, `-get`, `-getfacl`, `-help`, `-ls`, `-moveFromLocal`, `-moveToLocal`, `-mv`, `-rm`, `-put`, `-mkdir`, and `-tail`.
3. Display the `hadoop fs` help system.
4. Display the `hadoop fs` help system for only the `get` and `put` commands.
5. Navigate to the `/home/oracle/movie/moviework/mapreduce` directory.
6. Use the `hadoop fs` command to list the contents of `/user/oracle`.
7. Create a subdirectory named `my_stuff` in the `/user/oracle` folder, and then confirm that the directory is created.
8. Remove the `my_stuff` directory, and then confirm that the directory is deleted.
9. Review the content of the `read_avro_file.sh` compressed Avro application log.
10. Review the commands that are available for the Hadoop Distributed File System, and then copy the file into HDFS.
11. The `/user/oracle/movework/applog_avro` HDFS directory contains a compressed `movieapp_3months.avro` Avro log file. This log file contains the tracked activity for an online movie application. The Avro data represents individual clicks from an online movie rental site. If the file didn't exist in the `/user/oracle/movework/applog_avro` HDFS directory, you could use the basic `put` command to copy the file from the local file system into the specified directory in HDFS. Construct the `put` command that would perform such operation.
12. Confirm that the file already exists in `/user/oracle/movework/applog_avro`.

## Solution 5-1: Introduction to HDFS Commands

### Overview

In this solution, you review some of the HDFS commands you learned about in this lesson. You also load an Avro log file that contains tracked activity for an online movie application from the local file system into HDFS.

### Steps

1. Start a terminal window.



2. Review the commands that are available for the Hadoop Distributed File System. You will find that its composition is similar to the local Linux file system. You will use the `hadoop fs` command when interacting with HDFS. Spend a few minutes reviewing the syntax for some of the useful commands that you will use in HDFS such as `-cat`, `-copyFromLocal`, `-copyToLocal`, `-cp`, `-get`, `-getfacl`, `-help`, `-ls`, `-moveFromLocal`, `-moveToLocal`, `-mv`, `-rmdir`, `-put`, `-mkdir`, and `-tail`.

```
hadoop fs
```

```
[oracle@bigdatalite ~]$ hadoop fs
Usage: hadoop fs [generic options]
      [-appendToFile <localsrc> ... <dst>]
      [-cat [-ignoreCrc] <src> ...]
      [-checksum <src> ...]
      [-chgrp [-R] GROUP PATH...]
      [-chmod [-R] <MODE[,MODE]... | OCTALMODE> PATH...]
      [-chown [-R] [OWNER][:[GROUP]] PATH...]
      [-copyFromLocal [-f] [-p] <localsrc> ... <dst>]
      [-copyToLocal [-p] [-ignoreCrc] [-crc] <src> ... <localdst>]
      [-count [-q] <path> ...]
      [-cp [-f] [-p] <src> ... <dst>]
      [-createSnapshot <snapshotDir> [<snapshotName>]]
      [-deleteSnapshot <snapshotDir> <snapshotName>]
      [-df [-h] [<path> ...]]
      [-du [-s] [-h] <path> ...]
      [-expunge]
      [-get [-p] [-ignoreCrc] [-crc] <src> ... <localdst>]
      [-getfacl [-R] <path>]
      [-getmerge [-nl] <src> <localdst>]
      [-help [cmd ...]]
      [-ls [-d] [-h] [-R] [<path> ...]]
      [-mkdir [-p] <path> ...]
      [-moveFromLocal <localsrc> ... <dst>]
      [-moveToLocal <src> <localdst>]
      [-mv <src> ... <dst>]
      [-put [-f] [-p] <localsrc> ... <dst>]
      [-renameSnapshot <snapshotDir> <oldName> <newName>]
      [-rm [-f] [-r|-R] [-skipTrash] <src> ...]
      [-rmdir [--ignore-fail-on-non-empty] <dir> ...]
```

```
[-setfacl [-R] [{-b|-k} {-m|-x <acl_spec>} <path>]|[--set <acl_spec> <path>]
[-setrep [-R] [-w] <rep> <path> ...]
[-stat [format] <path> ...]
[-tail [-f] <file>]
[-test [-defsz] <path>]
[-text [-ignoreCrc] <src> ...]
[-touchz <path> ...]
[-usage [cmd ...]]]

Generic options supported are
-conf <configuration file>          specify an application configuration file
-D <property=value>                  use value for given property
-fs <local|namenode:port>           specify a namenode
-jt <local|jobtracker:port>          specify a job tracker
-files <comma separated list of files>  specify comma separated files to be copied to the map reduce cluster
-libjars <comma separated list of jars>  specify comma separated jar files to include in the classpath.
-archives <comma separated list of archives>  specify comma separated archives to be unarchived on the compute machines.
```

The general command line syntax is  
bin/hadoop command [genericOptions] [commandOptions]

[oracle@bigdatalite ~]\$ █

3. Display the hadoop fs help system for all commands.

```
hadoop fs -help
```

```
[oracle@bigdatalite ~]$ hadoop fs -help
Usage: hadoop fs [generic options]
      [-appendToFile <localsrc> ... <dst>]
      [-cat [-ignoreCrc] <src> ...]
      [-checksum <src> ...]
      [-chgrp [-R] GROUP PATH...]
      [-chmod [-R] <MODE[,MODE]... | OCTALMODE> PATH...]
      [-chown [-R] [OWNER][:[GROUP]] PATH...]
      [-copyFromLocal [-f] [-p] <localsrc> ... <dst>]
      [-copyToLocal [-p] [-ignoreCrc] [-crc] <src> ... <localdst>]
      [-count [-q] <path> ...]
      [-cp [-f] [-p] <src> ... <dst>]
      [-createSnapshot <snapshotDir> [<snapshotName>]]
      [-deleteSnapshot <snapshotDir> <snapshotName>]
      [-df [-h] [<path> ...]]
      [-du [-s] [-h] <path> ...]
      [-expunge]
      [-get [-p] [-ignoreCrc] [-crc] <src> ... <localdst>]
      [-getfacl [-R] <path>]
      [-getmerge [-nl] <src> <localdst>]
      [-help [cmd ...]]
      [-ls [-d] [-h] [-R] [<path> ...]]
      [-mkdir [-p] <path> ...]
      [-moveFromLocal <localsrc> ... <dst>]
      [-moveToLocal <src> <localdst>]
      [-mv <src> ... <dst>]
      [-put [-f] [-p] <localsrc> ... <dst>]
      [-renameSnapshot <snapshotDir> <oldName> <newName>]
      [-rm [-f] [-r|-R] [-skipTrash] <src> ...]
```

4. Display the hadoop fs help system for only the get and put commands.

```
hadoop fs -help get put
```

```
[oracle@bigdatalite ~]$ hadoop fs -help get put
-get [-p] [-ignoreCrc] [-crc] <src> ... <localdst>:      Copy files that match the file pa
ttern <src>
              to the local name. <src> is kept. When copying multiple,
              files, the destination must be a directory. Passing
              -p preserves access and modification times,
              ownership and the mode.
-put [-f] [-p] <localsrc> ... <dst>:      Copy files from the local file system
              into fs. Copying fails if the file already
              exists, unless the -f flag is given. Passing
              -p preserves access and modification times,
              ownership and the mode. Passing -f overwrites
              the destination if it already exists.
[oracle@bigdatalite ~]$ █
```

5. Navigate to the /home/oracle/movie/moviework/mapreduce directory.

```
cd /home/oracle/movie/moviework/mapreduce
```

```
[oracle@bigdatalite reset]$ cd /home/oracle/movie/moviework/mapreduce
[oracle@bigdatalite mapreduce]$ pwd
/home/oracle/movie/moviework/mapreduce
[oracle@bigdatalite mapreduce]$
```

6. Use the hadoop fs command to list the contents of /user/oracle.

```
hadoop fs -ls /user/oracle
```

```
[oracle@bigdatalite mapreduce]$ hadoop fs -ls /user/oracle
Found 6 items
drwx-----  - oracle oracle      0 2014-08-25 05:55 /user/oracle/.Trash
drwx-----  - oracle oracle      0 2014-09-23 13:25 /user/oracle/.staging
drwxr-xr-x  - oracle oracle      0 2014-01-12 18:15 /user/oracle/moviedemo
drwxr-xr-x  - oracle oracle      0 2014-09-24 09:38 /user/oracle/movework
drwxr-xr-x  - oracle oracle      0 2014-09-08 15:50 /user/oracle/oggdemo
drwxr-xr-x  - oracle oracle      0 2014-09-20 13:59 /user/oracle/oozie-oozi
[oracle@bigdatalite mapreduce]$
```

7. Create a subdirectory named my\_stuff in the /user/oracle folder, and then confirm that the directory is created.

```
hadoop fs -mkdir /user/oracle/my_stuff
hadoop fs -ls /user/oracle
```

**The my\_stuff directory is created.**

```
[oracle@bigdatalite mapreduce]$ hadoop fs -mkdir /user/oracle/my_stuff
[oracle@bigdatalite mapreduce]$ hadoop fs -ls /user/oracle
Found 7 items
drwx-----  - oracle oracle      0 2014-08-25 05:55 /user/oracle/.Trash
drwx-----  - oracle oracle      0 2014-09-23 13:25 /user/oracle/.staging
drwxr-xr-x  - oracle oracle      0 2014-01-12 18:15 /user/oracle/moviedemo
drwxr-xr-x  - oracle oracle      0 2014-09-24 09:38 /user/oracle/movework
drwxr-xr-x  - oracle oracle      0 2015-04-06 02:40 /user/oracle/my_stuff
drwxr-xr-x  - oracle oracle      0 2014-09-08 15:50 /user/oracle/oggdemo
drwxr-xr-x  - oracle oracle      0 2014-09-20 13:59 /user/oracle/oozie-oozi
[oracle@bigdatalite mapreduce]$
```

8. Remove the `my_stuff` directory, and then confirm that the directory is deleted.

```
hadoop fs -rm -r my_stuff  
hadoop fs -ls
```

The `my_stuff` directory is removed.

```
[oracle@bigdatalite mapreduce]$ hadoop fs -rm -r my_stuff  
15/04/06 02:42:55 INFO fs.TrashPolicyDefault: Namenode trash configuration: Deletion interval = 0 minutes  
, Emtier interval = 0 minutes.  
Deleted my_stuff  
[oracle@bigdatalite mapreduce]$ hadoop fs -ls  
Found 6 items  
drwx----- . oracle oracle 0 2014-08-25 05:55 .Trash  
drwx----- . oracle oracle 0 2014-09-23 13:25 .staging  
drwxr-xr-x - oracle oracle 0 2014-01-12 18:15 moviedemo  
drwxr-xr-x - oracle oracle 0 2014-09-24 09:38 moviework  
drwxr-xr-x - oracle oracle 0 2014-09-08 15:50 oggdemo  
drwxr-xr-x - oracle oracle 0 2014-09-20 13:59 oozie-oozi  
[oracle@bigdatalite mapreduce]$
```

9. Review the content of the `read_avro_file.sh` compressed Avro application log.

```
cat read_avro_file.sh
```

```
[oracle@bigdatalite mapreduce]$ pwd  
/home/oracle/movie/moviework/mapreduce  
[oracle@bigdatalite mapreduce]$ ls -l  
total 21340  
-rw-r--r--. 1 oracle oinstall 1302 Jan 12 2014 ddl_hive_moviefact.sql  
-rwxr-x---. 1 oracle oinstall 6395 Jan 25 2014 HoLMapReduce_Commands.txt  
-rwxr-x---. 1 oracle oinstall 19242668 Feb 4 2014 movieapp_3months.avro  
-rwxr-x---. 1 oracle oinstall 2591736 Jan 25 2014 movieapp_3months.log.gz  
-rwxr-x---. 1 oracle oinstall 346 May 3 2014 read_avro_file.sh  
lrwxrwxrwx. 1 oracle oinstall 53 Apr 25 2014 reset_mapreduce.sh -> /home/  
oracle/movie/moviework/reset/reset_mapreduce.sh  
[oracle@bigdatalite mapreduce]$ cat read_avro_file.sh  
java -classpath /home/oracle/movie/moviework/flume/java/JsonToAvro/classes:/usr/  
lib/flume-ng/lib/avro.jar:/usr/lib/flume-ng/lib/jackson-mapper-asl-1.9.3.jar:/us  
r/lib/flume-ng/lib/jackson-core-asl-1.9.3.jar:/usr/lib/flume-ng/lib/snappy-java-  
1.0.4.1.jar oracle.avro.ReadActivityFile /home/oracle/movie/moviework/mapreduce/  
movieapp_3months.avro 10  
[oracle@bigdatalite mapreduce]$
```

10. Display the hadoop fs help system for only the put command. You can use the history feature in your terminal window, up arrow, to recall the same command that was used earlier.

```
hadoop fs -help put
```

```
[oracle@bigdatalite mapreduce]$ hadoop fs -help put
-put [-f] [-p] <localsrc> ... <dst>: Copy files from the local file system
      into fs. Copying fails if the file already
      exists, unless the -f flag is given. Passing
      -p preserves access and modification times,
      ownership and the mode. Passing -f overwrites
      the destination if it already exists.
[oracle@bigdatalite mapreduce]$ █
```

11. The /user/oracle/moviework/applog\_avro HDFS directory contains a compressed movieapp\_3months.avro Avro log file. This log file contains the tracked activity for an online movie application. The Avro data represents individual clicks from an online movie rental site. If the file didn't exist in the /user/oracle/moviework/applog\_avro HDFS directory, you could use the basic put command to copy the file for the local file system into the specified directory in HDFS. Construct the put command that would perform such operation.

```
hadoop fs -put movieapp_3months.avro
/user/oracle/moviework/applog_avro
```

```
the destination it already exists.
[oracle@bigdatalite mapreduce]$ pwd
/home/oracle/movie/moviework/mapreduce
[oracle@bigdatalite mapreduce]$ ls -l
total 21340
-rw-r--r--. 1 oracle oinstall 1302 Jan 12 2014 ddl_hive_moviefact.sql
-rwxr-x---. 1 oracle oinstall 6395 Jan 25 2014 HoLMaPReduce_Commands.txt
-rwxr-x---. 1 oracle oinstall 19242668 Feb 4 2014 movieapp_3months.avro
-rwxr-x---. 1 oracle oinstall 2591736 Jan 25 2014 movieapp_3months.log.gz
-rwxr-x---. 1 oracle oinstall 346 May 3 2014 read_avro_file.sh
lrwxrwxrwx. 1 oracle oinstall 53 Apr 25 2014 reset_mapreduce.sh -> /home/
oracle/movie/moviework/reset/reset_mapreduce.sh
[oracle@bigdatalite mapreduce]$ hadoop fs -put movieapp_3months.avro /user/oracle/moviework/applog_avro
```

12. Confirm that the file already exists in /user/oracle/moviework/applog\_avro.

```
[oracle@bigdatalite mapreduce]$ hadoop fs -ls /user/oracle/moviework/applog_avro
Found 1 items
-rw-r--r--. 1 oracle oracle 19242668 2015-02-13 09:58 /user/oracle/moviework/
applog_avro/movieapp_3months.avro
[oracle@bigdatalite mapreduce]$ █
```



## **Practices for Lesson 6: Acquire Data Using CLI, Fuse DFS, and Flume**

**Chapter 6**

## Practice for Lesson 6

---

### Practice Overview

In this practice, you connect to Flume, review Flume commands, and then view Movieplex application configuration files.

## Practice 6-1: Viewing Flume Commands and Configuration Options

### Overview

In this practice, you start flume-ng and review generic Flume commands options using the help facility. Then, you view the configuration and agent files for the MoviePlex application.

### Tasks

1. Open a terminal window and enter the following command to view Flume options.

```
flume-ng help
```



The screenshot shows a terminal window titled "oracle@bigdatalite:~". The window displays the usage information for the flume-ng command, including global options, agent options, and avro-client options. It also includes notes about config file usage and a warning about host and port specification.

```
oracle@bigdatalite:~$ flume-ng help
Usage: /usr/lib/flume-ng/bin/flume-ng <command> [options]...

commands:
  help          display this help text
  agent         run a Flume agent
  avro-client   run an avro Flume client
  version       show Flume version info

global options:
  --conf,-c <conf>      use configs in <conf> directory
  --classpath,-C <cp>    append to the classpath
  --dryrun,-d            do not actually start Flume, just print the command
  --plugins-path <dirs>  colon-separated list of plugins.d directories. See the
                        plugins.d section in the user guide for more details.
                        Default: $FLUME_HOME/plugins.d
  -Dproperty=value     sets a Java system property value
  -Xproperty=value     sets a Java -X option

agent options:
  --conf-file,-f <file> specify a config file (required)
  --name,-n <name>      the name of this agent (required)
  --help,-h              display help text

avro-client options:
  --rpcProps,-P <file>  RPC client properties file with server connection param
  s
  --host,-H <host>      hostname to which events will be sent
  --port,-p <port>       port of the avro source
  --dirname <dir>        directory to stream to avro source
  --filename,-F <file>   text file to stream to avro source (default: std input)
  --headerFile,-R <file> File containing event headers as key/value pairs on each new line
  --help,-h              display help text

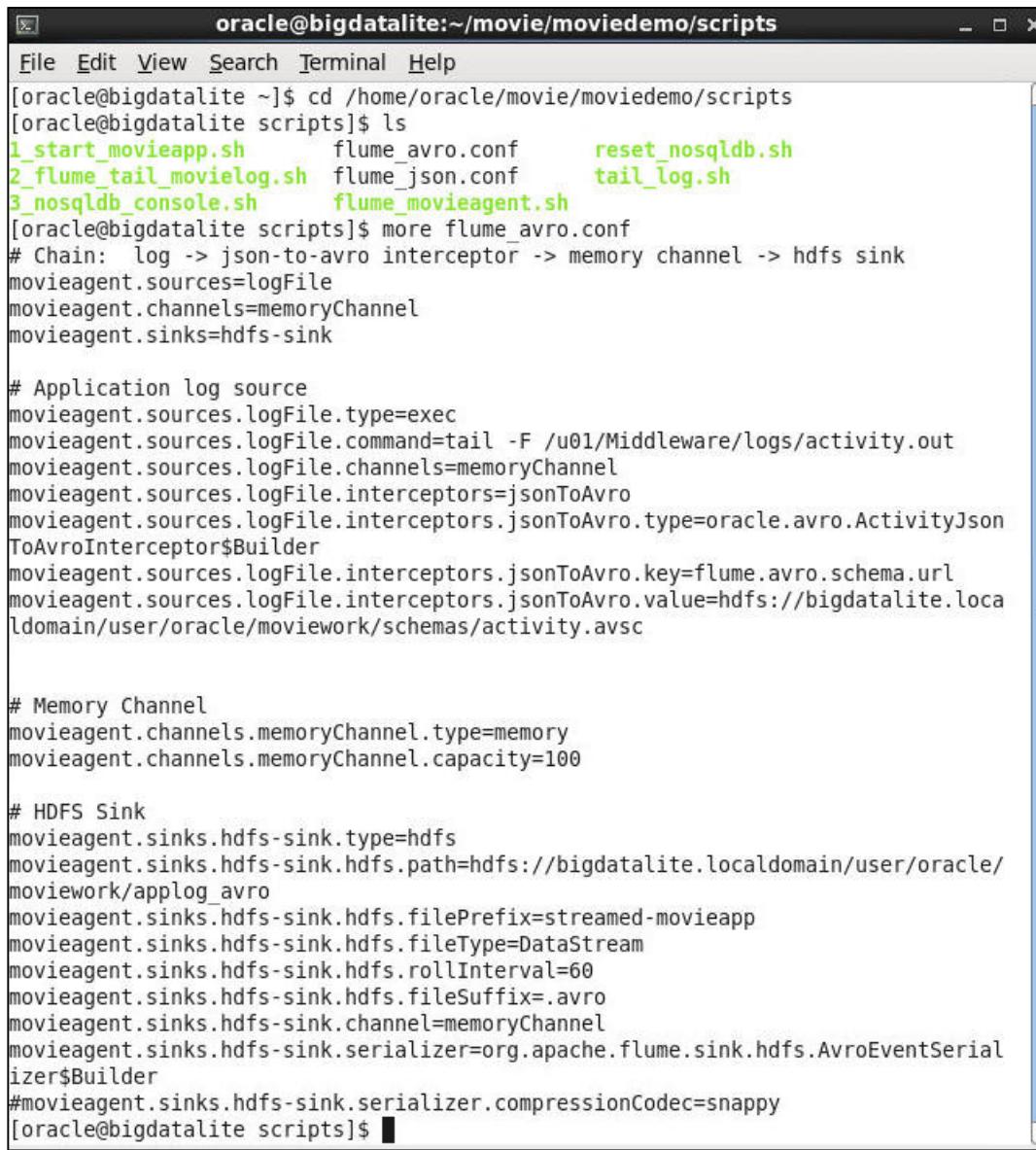
Either --rpcProps or both --host and --port must be specified.

Note that if <conf> directory is specified, then it is always included first
in the classpath.

[oracle@bigdatalite ~]$
```

2. Execute the following commands to review the Flume AVRO configuration file for the MoviePlex application.

```
cd /home/oracle/movie/moviedemo/scripts  
ls  
more flume_avro.conf
```



The screenshot shows a terminal window titled "oracle@bigdatalite:~/movie/moviedemo/scripts". The window displays the contents of the "flume\_avro.conf" file. The file defines a Flume agent named "movieagent" with the following configuration:

```
[oracle@bigdatalite scripts]$ cd /home/oracle/movie/moviedemo/scripts  
[oracle@bigdatalite scripts]$ ls  
1_start_movieapp.sh      flume_avro.conf      reset_nosqldb.sh  
2_flume_tail_movielog.sh flume_json.conf    tail_log.sh  
3_nosqlldb_console.sh   flume_movieagent.sh  
[oracle@bigdatalite scripts]$ more flume_avro.conf  
# Chain: log -> json-to-avro interceptor -> memory channel -> hdfs sink  
movieagent.sources=logFile  
movieagent.channels=memoryChannel  
movieagent.sinks=hdfs-sink  
  
# Application log source  
movieagent.sources.logFile.type=exec  
movieagent.sources.logFile.command=tail -F /u01/Middleware/logs/activity.out  
movieagent.sources.logFile.channels=memoryChannel  
movieagent.sources.logFile.interceptors=jsonToAvro  
movieagent.sources.logFile.interceptors.jsonToAvro.type=oracle.avro.ActivityJson  
ToAvroInterceptor$Builder  
movieagent.sources.logFile.interceptors.jsonToAvro.key=flume.avro.schema.url  
movieagent.sources.logFile.interceptors.jsonToAvro.value=hdfs://bigdatalite.loca  
ldomain/user/oracle/moviework/schemas/activity.avsc  
  
# Memory Channel  
movieagent.channels.memoryChannel.type=memory  
movieagent.channels.memoryChannel.capacity=100  
  
# HDFS Sink  
movieagent.sinks.hdfs-sink.type=hdfs  
movieagent.sinks.hdfs-sink.hdfs.path=hdfs://bigdatalite.loca  
ldomain/user/oracle/movework/applog_avro  
movieagent.sinks.hdfs-sink.hdfs.filePrefix=streamed-movieapp  
movieagent.sinks.hdfs-sink.hdfs.fileType=DataStream  
movieagent.sinks.hdfs-sink.hdfs.rollInterval=60  
movieagent.sinks.hdfs-sink.hdfs.fileSuffix=.avro  
movieagent.sinks.hdfs-sink.channel=memoryChannel  
movieagent.sinks.hdfs-sink.serializer=org.apache.flume.sink.hdfs.AvroEventSerial  
izer$Builder  
#movieagent.sinks.hdfs-sink.serializer.compressionCodec=snappy  
[oracle@bigdatalite scripts]$
```

3. Review the agent file for the MoviePlex application.

```
more flume_movieagent.sh
```

```
[oracle@bigdatalite scripts]$ more flume_movieagent.sh
flume-ng agent --conf-file flume_avro.conf --name movieagent
[oracle@bigdatalite scripts]$ █
```

4. Review the Flume JSON configuration file for the MoviePlex application.

```
more flume_json.conf
```

```
[oracle@bigdatalite scripts]$ more flume_json.conf
# Chain: json log -> memory channel -> hdfs sink
movieagent.sources=logFile
movieagent.channels=memoryChannel
movieagent.sinks=hdfs-sink

# Application log source
movieagent.sources.logFile.type=exec
movieagent.sources.logFile.command=/home/oracle/movie/moviedemo/scripts/tail_log.sh
movieagent.sources.logFile.channels=memoryChannel

# Memory channel
movieagent.channels.memoryChannel.type=memory
movieagent.channels.memoryChannel.batchSize=1
movieagent.channels.memoryChannel.capacity=100

# HDFS Sink
movieagent.sinks.hdfs-sink.type=hdfs
movieagent.sinks.hdfs-sink.hdfs.path=hdfs://bigdatalite.localdomain/user/oracle/movie
work/applog_json
movieagent.sinks.hdfs-sink.hdfs.filePrefix=streamed-movieapp
movieagent.sinks.hdfs-sink.hdfs.fileType=DataStream
movieagent.sinks.hdfs-sink.hdfs.writeFormat=Text
movieagent.sinks.hdfs-sink.hdfs.rollInterval=60
movieagent.sinks.hdfs-sink.channel=memoryChannel
[oracle@bigdatalite scripts]$ █
```

5. Exit the terminal window.



## **Practices for Lesson 7: Acquire and Access Data Using Oracle NoSQL Database**

**Chapter 7**

## Practices for Lesson 7

---

### Practices Overview

In these guided practices, you perform the following tasks to load data into NoSQL database and query that data:

- Run the MoviePlex application
- Start the Oracle NoSQL Database instance
- Load User Profile data
- Load new movie data
- Query the new movie data

## Practice 7-1: Start and Run the MoviePlex Application

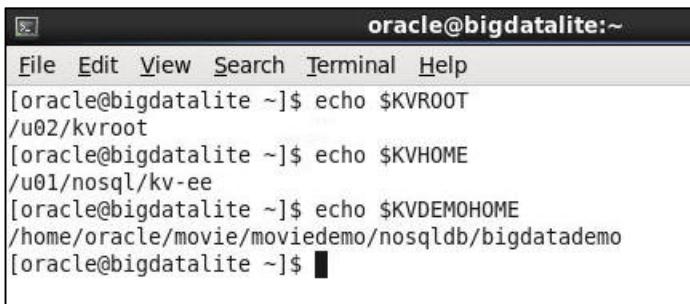
### Overview

In this guided practice, you open and run the MovieDemo application using Oracle JDeveloper.

### Tasks

1. Open a terminal window.
2. View the three Oracle NoSQL Database-specific environment variables:
  - KVHOME is where binaries are installed
  - KVROOT is where data and configuration files are saved
  - KVDEMOHOME is where the source of the practice project is saved

```
echo $KVROOT  
echo $KVHOME  
echo $KVDEMOHOME
```



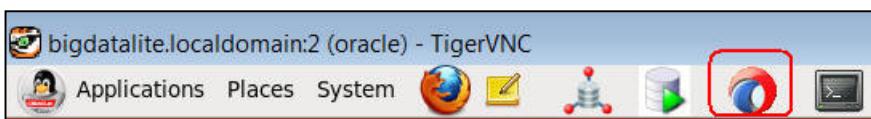
A screenshot of a terminal window titled "oracle@bigdatalite:~". The window shows the following command history:

```
File Edit View Search Terminal Help  
[oracle@bigdatalite ~]$ echo $KVROOT  
/u02/kvroot  
[oracle@bigdatalite ~]$ echo $KVHOME  
/u01/nosql/kv-ee  
[oracle@bigdatalite ~]$ echo $KVDEMOHOME  
/home/oracle/movie/moviedemo/nosqldb/bigdatademo  
[oracle@bigdatalite ~]$ █
```

3. Remove the \$KVROOT value by executing the following command:

```
rm -rf $KVROOT
```

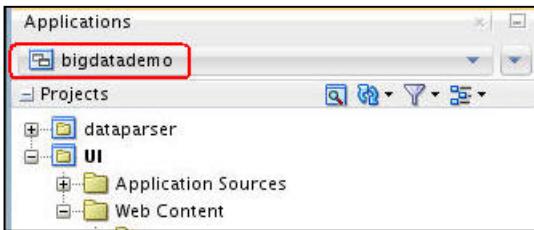
4. Leaving the terminal window open, single-click the JDeveloper icon on the toolbar to launch Oracle JDeveloper.



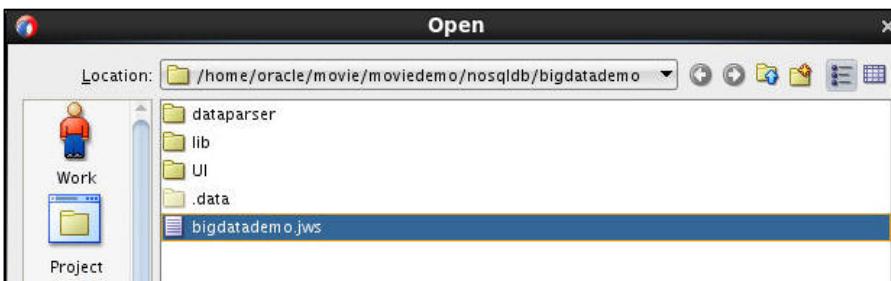
Note: If prompted, select the default **Studio Developer** user and click **OK**.

5. Close any open documents in the JDeveloper viewer pane, including the Start Page and Log window. (Documents are organized by tabs in the viewer pane.)

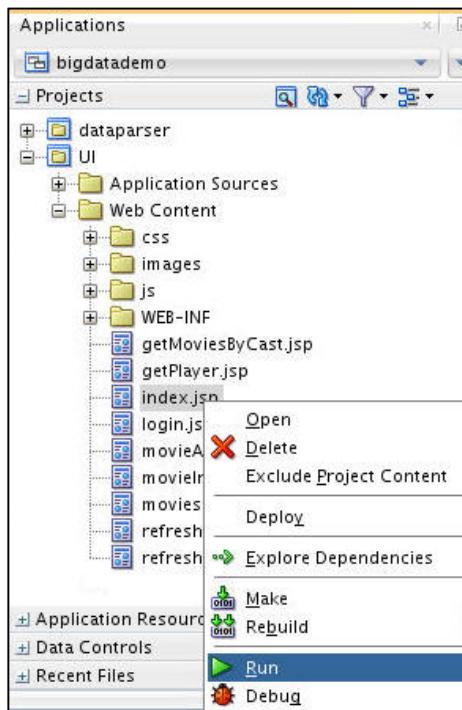
**Note:** If the `bigdatademo` application is already open in JDeveloper's Applications navigator (as shown here), skip step 6 and go directly to step 7.



- Using the File menu, open `bigdatademo.jws` from the `/home/oracle/movie/moviedemo/nosqldb/bigdatademo` directory.



- In the Applications navigator, drill on **UI > Web Content**. Then, right-click `index.jsp` and select **Run**, as shown here:



Result: JDeveloper builds and deploys the application.

**Note:** The application will take a few moments to load. Disregard any application message that NoSQL database not running. This is expected.

8. In the pre-filled login page, change the username to guest2 and click **Sign in**.



Result: The Movieplex application appears, as shown here:

The screenshot shows the Oracle Movieplex application interface. At the top, there's a navigation bar with links like File, Edit, View, History, Bookmarks, Tools, Help, and a search bar. Below the header, the main content area features the Oracle MoviePlex logo (a camera icon inside a red film strip circle) and the text "Oracle MoviePlex". A "Welcome Kelli | Logout" link is visible. The interface includes sections for "Movies based on Mood" and "Top Movies for Kelli by Genre" (Drama). The "Top Movies for Kelli by Genre" section displays five movie titles: Supernatural, Happy-Go-Lucky, Tigerland, I Am Sam, and La mala educación. Navigation arrows are present on both sides of the movie list.

## Practice 7-2: Start Oracle NoSQL and Load User Profile Data

### Overview

In this guided practice, you start an Oracle NoSQL Database instance and then load the user profile information. You step through some of the code while loading the profile information.

At the end of this practice, you should be able to log on successfully. KVLite will be used as the Oracle NoSQL Database Instance.

### Tasks

1. Switch back to the terminal window. Then, change the directory to KVHOME using the following command:

```
cd $KVHOME
```

2. Start KVLite from the current working directory:

```
java -jar $KVHOME/lib/kvstore.jar kvlite -host localhost -root $KVROOT
```

```
[oracle@bigdatalite ~]$ cd $KVHOME  
[oracle@bigdatalite kv-ee]$ java -jar $KVHOME/lib/kvstore.jar kvlite -host localhost -root $KVROOT  
KVLite: exception in start: java.rmi.server.ExportException: Port already in use: 5000; nested exce  
ption is:  
java.net.BindException: Address already in use
```

**Note:** You might get the error shown above if KVLite is already started. You can ignore the list of exceptions.

3. In the terminal window, select **File > Open Tab**. Then, in the new tab, change to the schemas directory using the following command:

```
cd $KVDEMOHOME/dataparser/schemas
```

```
oracle@bigdatalite:u01/nosql/kv-ee      X oracle@bigdatalite:~/movie/moviedemo/nosq.  
[oracle@bigdatalite ~]$ cd $KVDEMOHOME/dataparser/schemas  
[oracle@bigdatalite schemas]$ █
```

4. Execute the following command:

```
ls -altr
```

Result: The schemas directory contains six AVRO schema files (\*.avsc).

```
[oracle@bigdatalite schemas]$ ls -altr
total 32
-rw-r--r--. 1 oracle oinstall 1121 Jul 17 2013 movie.avsc
-rw-r--r--. 1 oracle oinstall 238 Jul 17 2013 genre.avsc
-rw-r--r--. 1 oracle oinstall 472 Jul 17 2013 customer.avsc
-rw-r--r--. 1 oracle oinstall 614 Jul 17 2013 crew.avsc
-rw-r--r--. 1 oracle oinstall 732 Jul 17 2013 cast.avsc
-rw-r--r--. 1 oracle oinstall 693 Jul 17 2013 activity.avsc
drwxr-xr-x. 2 oracle oinstall 4096 Oct 23 2013 .
drwxr-xr-x. 10 oracle oinstall 4096 Sep 3 15:16 ..
[oracle@bigdatalite schemas]$ █
```

5. Start an admin session from the same terminal window tab.

```
java -jar $KVHOME/lib/kvstore.jar runadmin -host localhost -port 5000
```

```
[oracle@bigdatalite schemas]$ java -jar $KVHOME/lib/kvstore.jar runadmin -host localhost -port 5000
kv-> █
```

Result: You should now be logged in to the KV shell.

6. Register the customer schema:

```
ddl add-schema -file customer.avsc
```

```
kv-> ddl add-schema -file customer.avsc
Cannot add schema, already exists: oracle.avro.Customer
kv-> █
```

**Note:** If you receive the message shown above, the schema is already registered.

7. If you did *not* receive the error shown above, use the same `ddl` command to register the following schemas (Note: use the arrow key to recall previous commands):

- `movie.avsc`
- `cast.avsc`
- `crew.avsc`
- `genre.avsc`
- `activity.avsc`

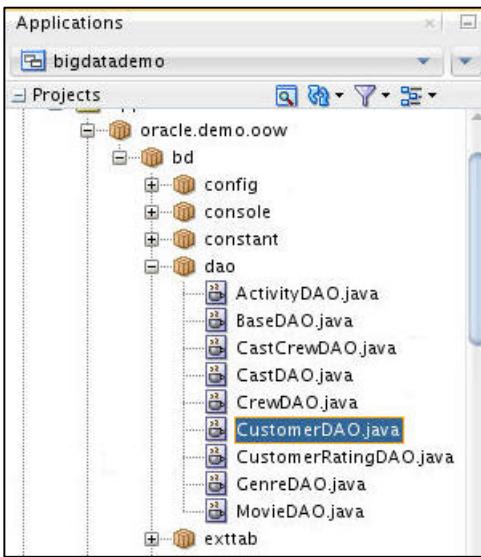
**Note:** If you received the `schema already exists` message in the previous step, the additional schemas also exist, and you do not have to register them.

- Run the `show schemas` command to confirm that all six schemas are registered.

```
show schemas
```

```
kv-> show schemas
oracle.avro.Activity
  ID: 6 Modified: 2014-09-03 16:36:09 UTC, From: bigdatalite.locaLdomain
oracle.avro.Cast
  ID: 3 Modified: 2014-09-03 16:35:28 UTC, From: bigdatalite.locaLdomain
oracle.avro.Crew
  ID: 4 Modified: 2014-09-03 16:35:42 UTC, From: bigdatalite.locaLdomain
oracle.avro.Customer
  ID: 1 Modified: 2014-09-03 16:34:59 UTC, From: bigdatalite.locaLdomain
oracle.avro.Genre
  ID: 5 Modified: 2014-09-03 16:35:56 UTC, From: bigdatalite.locaLdomain
oracle.avro.Movie
  ID: 2 Modified: 2014-09-03 16:35:13 UTC, From: bigdatalite.locaLdomain
kv-> █
```

- At the `kv->` prompt, execute the `exit` command to close the kv admin session. Then, execute the `exit` command at the `$` prompt to close the extra Terminal window tab. Finally, execute the `exit` command at the `$` prompt to close the Terminal window.
- In the MoviePlex application (Firefox browser), log out of the application. However, leave Login window open. Then, switch back to JDeveloper.
- In the Applications navigator, drill on **dataparser > Application Sources > oracle.demo.oww > bd > dao**. Then, double-click **CustomerDAO.java** to open the java class file.



**Note:**

- In this script, the Data Access Class named CustomerDAO (shown in the next screenshot) interacts with the Oracle NoSQL Database.
- This class has all of the access methods for customer-related operations, such as creating a customer profile, reading a profile using customer Id, and authenticating a customer based on username and password.

```

package oracle.demo.oow.bd.dao;

import ...;

public class CustomerDAO extends BaseDAO {

    private static int MOVIE_MAX_COUNT = 25;
    private static int GENRE_MAX_COUNT = 10;

    private static final String PASSWORD = StringUtil.getMessageDigest("welcome1");
    private static final String USERNAME = "guest";

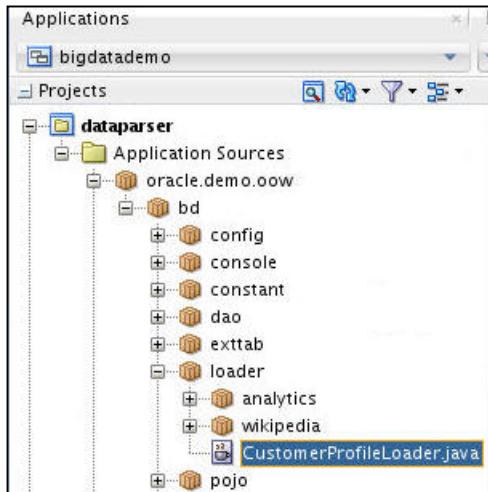
    /** Variables for JSONAvroBinding ***/
    private Schema customerSchema = null;
    private JsonAvroBinding customerBinding = null;

    public CustomerDAO() {
        super();
        customerSchema = parser.getTypes().get("oracle.avro.Customer");
        customerBinding = catalog.getJsonBinding(customerSchema);
    }
}

```

12. Now, upload the new the customer profile data into Oracle NoSQL Database.

In the Applications navigator, collapse the **dao** node. Then drill on the **loader** node. Finally, right-click the **CustomerProfileLoader.java** class file, and select **Run** from the menu.



Results: An output pane at the bottom of the JDeveloper window (dataparser.jpr – Log) shows the output of the process. Notice that 101 new user profiles have been added.

```
Running: dataparser.jpr - Log
97 {"id":1023088,"name":"Lily","email":"lily.wang@oraclemail.com","username":"guest97"
98 {"id":1023089,"name":"Sheri","email":"sheri.henderson@oraclemail.com","username":"guest97"
99 {"id":1045446,"name":"Clarence","email":"clarence.dejesus@oraclemail.com","username":"gues
100 {"id":1156799,"name":"Hatha","email":"hatha.durufl@oraclemail.com","username":"guest99",
101 {"id":1081065,"name":"Jin-Hua","email":"jin-hua.lawson@oraclemail.com","username":"guest1
Process exited with exit code 0.
```

13. In the Navigator pane, close the **Processes** tab.

## Practice 7-3: Load Movie Data

### Overview

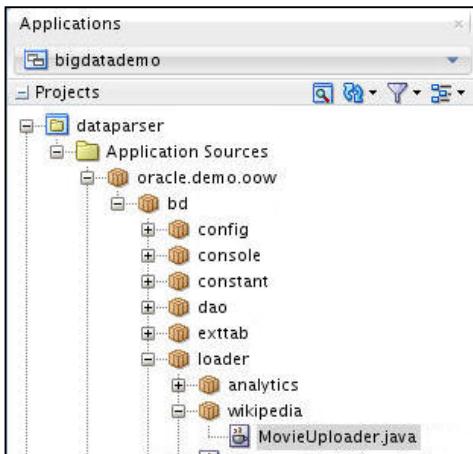
In this guided practice, you load the movie data by running a program. This program also creates a new movie-to-genre association using major-minor keys.

### Assumptions

You have completed Practice 7-2

### Tasks

1. In the Applications navigator, drill down to **Application Sources > oracle.demo.oww > bd > loader > wikipedia**. Then, open the **MovieUploader.java** file by double-clicking.



**Note:** This class reads the movie-info.out file and loads content into Oracle NoSQL DB.

2. In the Find box above the code, type '**main()**'. The `main()` method is found, shown here:

```

public static void main(String[] args) {
    MovieUploader mu = new MovieUploader();
    MovieDAO movieDAO = new MovieDAO();
    boolean isHOL = true;
    int movieCount = 5000;

    /**
     * If running this class you pass any argument that would mean, you are
     * setting up the environment for DEMO (not for the HOL)
     */
    if (args.length > 0) {
        isHOL = false;
        movieCount = 0;
    }

    try {
        /**
         * Step 1 - Upload movie info.
         * You can set how many movies do you want to upload into kv-store.
         * There close to quarter million movies in all. Default is set to
         * 5000 movies, which if set to 0, would mean upload all the movies.
         */
        mu.uploadMovieInfo(Constant.TARGET_NOSQL, movieCount);
        // mu.uploadMovieInfo(Constant.TARGET_RDBMS);
    }
}

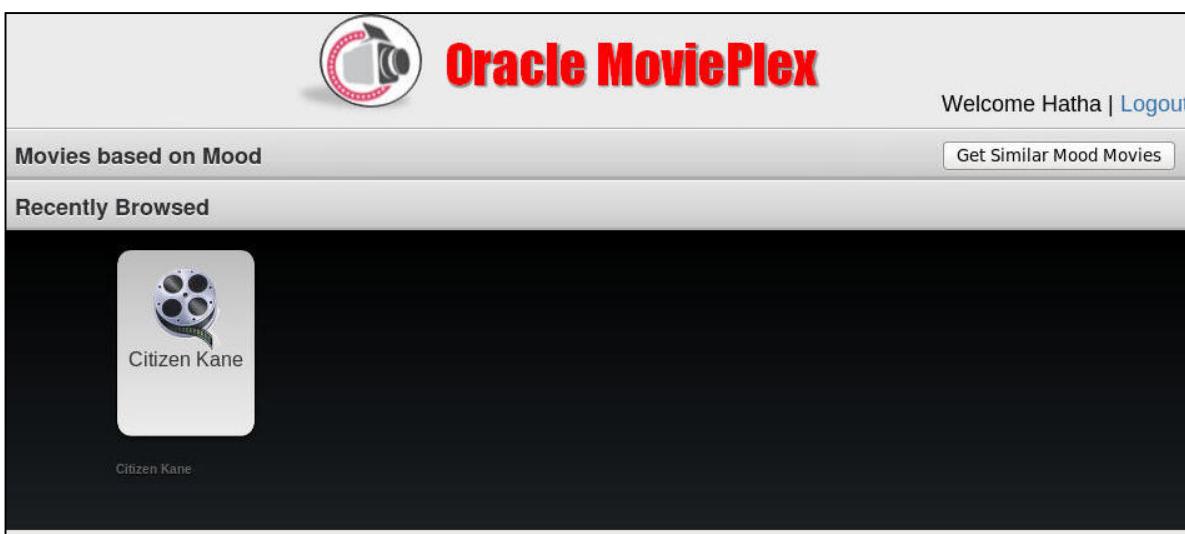
```

**Note:** Inside the `main()` method, the `uploadMovieInfo()` method writes data for the movies within a `try` loop.

3. In the navigator, right-click `MovieUploader.java` and select **Run**.  
**Note:** The program takes a few moments to load genres and data for all 5000 movies.
4. When the process finishes (see the log pane), toggle back to the MoviePlex application and sign in as `guest99`, using the same filled-in password.
5. In the **Drama** genre section, click the **Citizen Kane** film icon. A description for the movie opens.



6. Close the pop-up description pane.  
Result: The movie is now shown in the new **Recently Browsed** category.



7. Log out of the MoviePlex application, and then close the Login (Firefox browser) window.

## Practice 7-4: Query Movie Data

### Overview

In this guided practice, you query movie data by running a Java program that incorporates NoSQL coding.

### Assumptions

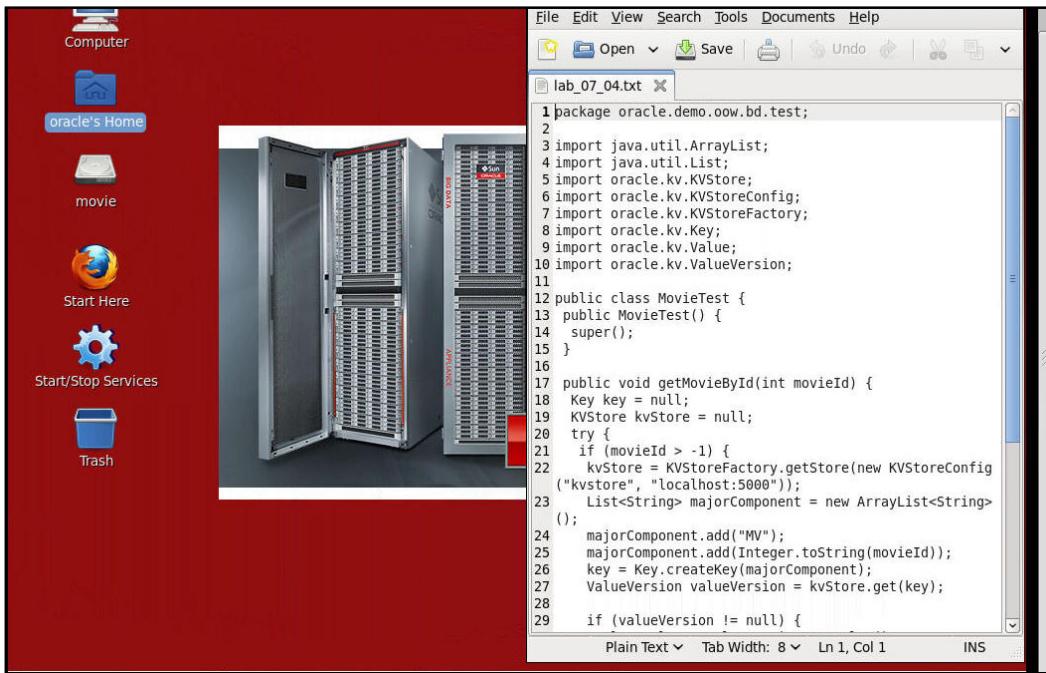
You have completed Practice 7-3

### Tasks

1. First, minimize the JDeveloper window (do not exit JDeveloper). Then, using the **Oracle's Home** icon on the desktop, navigate to the **/home/oracle/practice\_commands** folder and open the **lab\_07\_04.txt** using **gedit**.

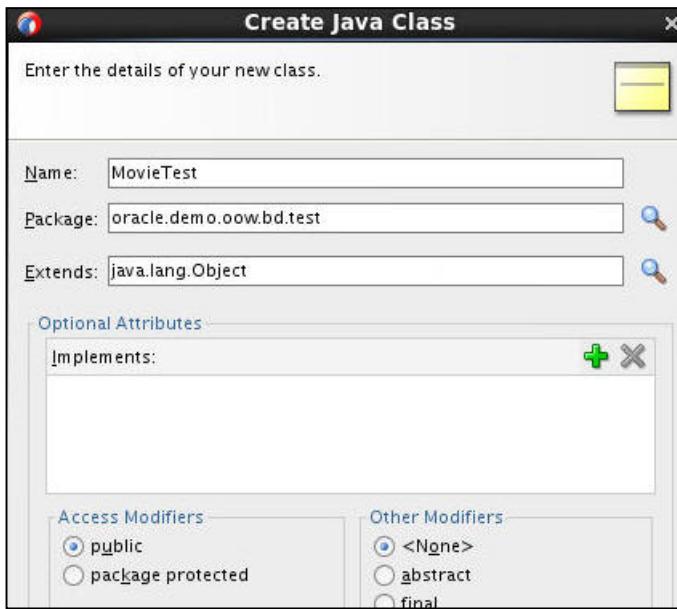
Result: The file opens in a gedit window.

2. Then, resize the gedit window and move it to the right side of your screen. Your screen should look something like this:



**Note:** You will copy the appropriate NoSQL database code from this file to a Java class.

3. Switch to JDeveloper and create a new Java class by performing these steps:
  - A. From the main menu, select **File > New > Java Class**. The Create Java Class dialog appears.
  - B. In the Create Java Class dialog, enter **MovieTest** as the Name and **oracle.demo.oow.bd.test** as the Package (shown below).



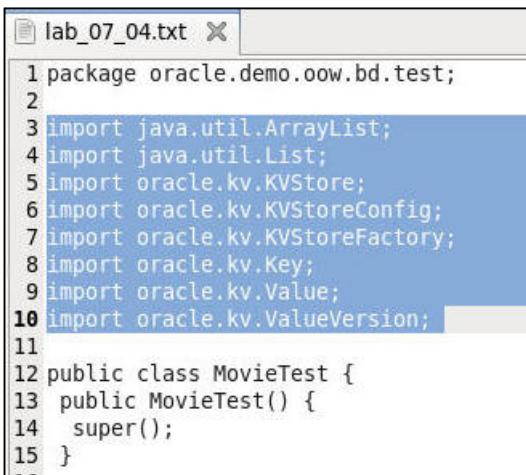
- C. Then click **OK**.
- D. In the Choose Source Folder dialog, select the first option under dataparser.jpr and click **OK**.



Result: The MovieTest.java class file looks like this:

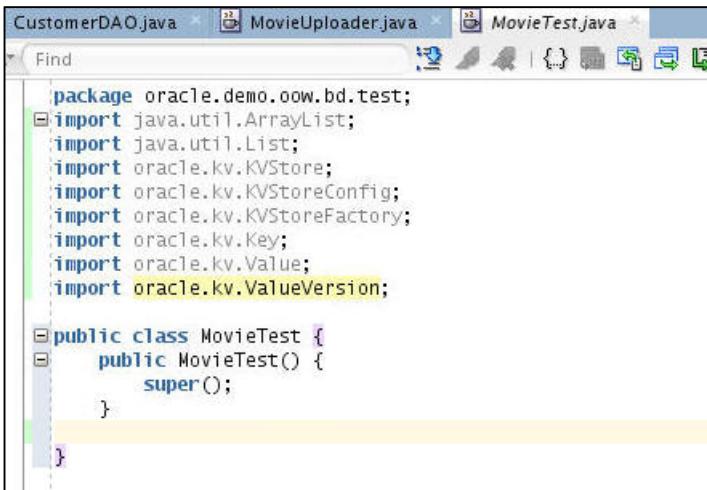
Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

4. Next, switch back to the lab\_07\_04.txt file, select all of the import statements (lines 3-10), and copy them to the clipboard.



```
1 package oracle.demo.oow.bd.test;
2
3 import java.util.ArrayList;
4 import java.util.List;
5 import oracle.kv.KVStore;
6 import oracle.kv.KVStoreConfig;
7 import oracle.kv.KVStoreFactory;
8 import oracle.kv.Key;
9 import oracle.kv.Value;
10 import oracle.kv.ValueVersion;
11
12 public class MovieTest {
13     public MovieTest() {
14         super();
15     }
16 }
```

5. In JDeveloper, paste the import statements on the blank line after the initial package statement. Then, add a blank line after the last import statement, and another blank line just before the last "}" closure. Result: Your Java class should now look like this:

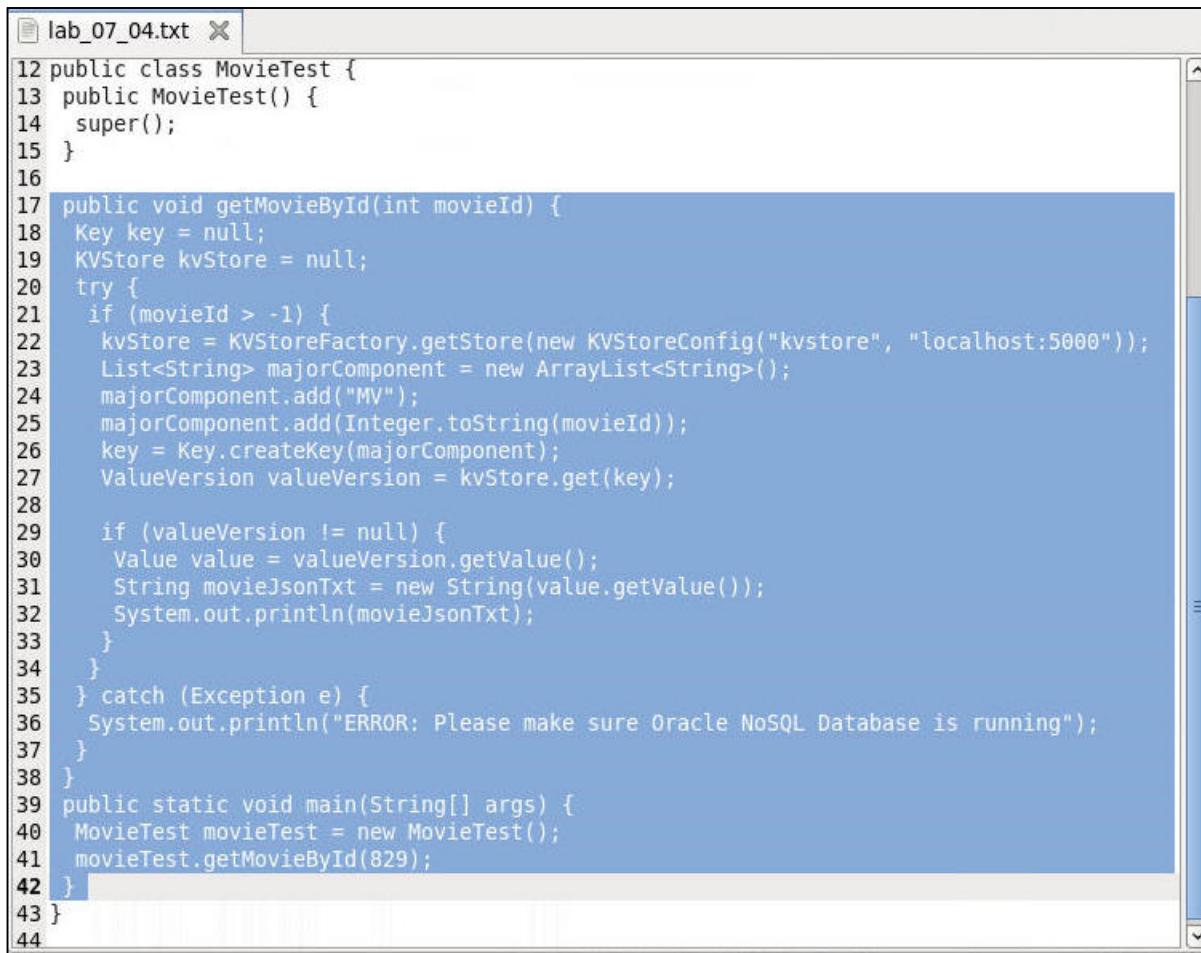


```
package oracle.demo.oow.bd.test;
import java.util.ArrayList;
import java.util.List;
import oracle.kv.KVStore;
import oracle.kv.KVStoreConfig;
import oracle.kv.KVStoreFactory;
import oracle.kv.Key;
import oracle.kv.Value;
import oracle.kv.ValueVersion;

public class MovieTest {
    public MovieTest() {
        super();
    }
}
```

6. Back in the lab\_07\_04.txt file, widen the window so that no code lines wrap.

A. Then, select all lines from the getMovieById(int movieId) method (line 17) through the second to last “}” closure (line 42), as shown below.

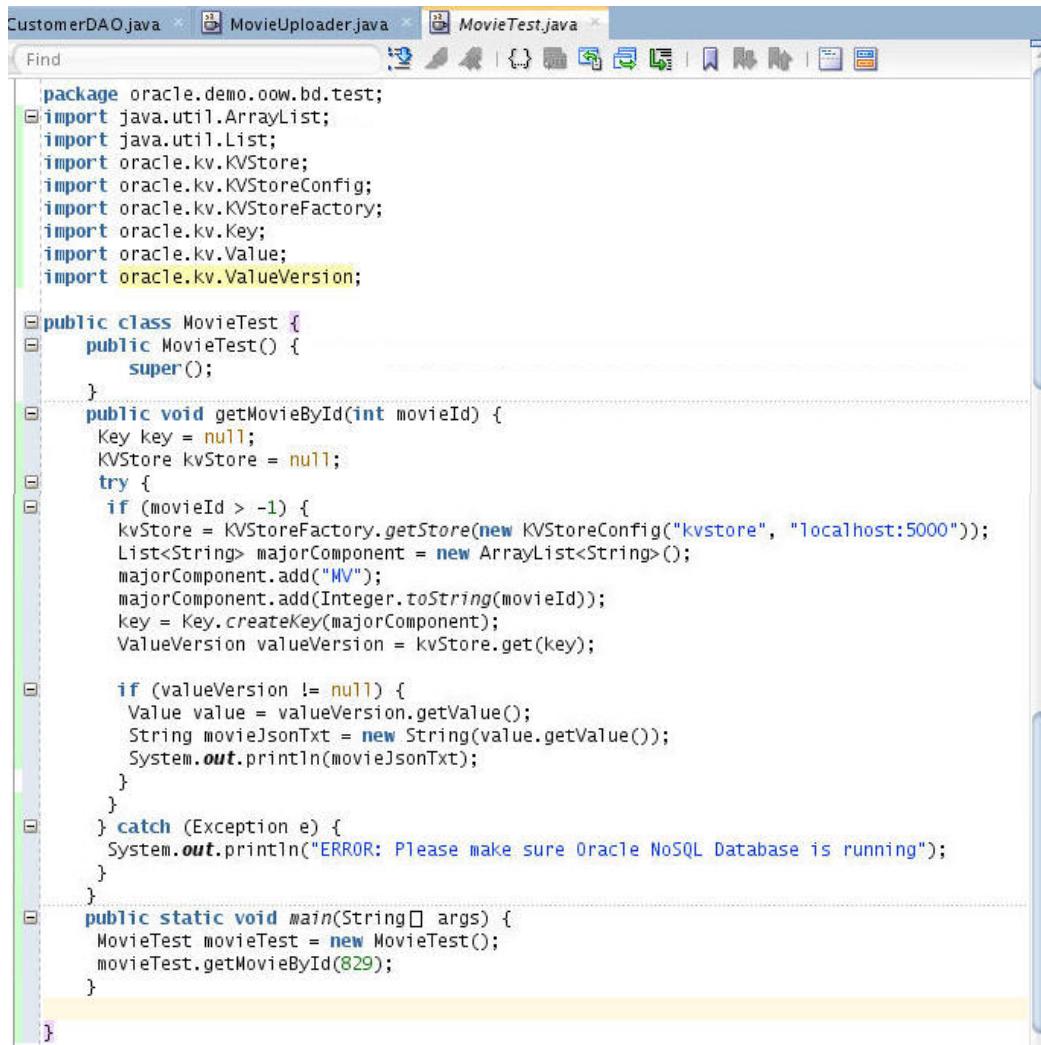


The screenshot shows a text editor window titled "lab\_07\_04.txt". The code is a Java class named "MovieTest". The "getMovieById" method is highlighted with a blue selection. The code uses Java 8 features like try-with-resources and streams. It connects to an Oracle NoSQL database to retrieve a movie by its ID and prints the result as JSON.

```
12 public class MovieTest {  
13     public MovieTest() {  
14         super();  
15     }  
16  
17     public void getMovieById(int movieId) {  
18         Key key = null;  
19         KVStore kvStore = null;  
20         try {  
21             if (movieId > -1) {  
22                 kvStore = KVStoreFactory.getStore(new KVStoreConfig("kvstore", "localhost:5000"));  
23                 List<String> majorComponent = new ArrayList<String>();  
24                 majorComponent.add("MV");  
25                 majorComponent.add(Integer.toString(movieId));  
26                 key = Key.createKey(majorComponent);  
27                 ValueVersion valueVersion = kvStore.get(key);  
28  
29                 if (valueVersion != null) {  
30                     Value value = valueVersion.getValue();  
31                     String movieJsonTxt = new String(value.getValue());  
32                     System.out.println(movieJsonTxt);  
33                 }  
34             }  
35         } catch (Exception e) {  
36             System.out.println("ERROR: Please make sure Oracle NoSQL Database is running");  
37         }  
38     }  
39     public static void main(String[] args) {  
40         MovieTest movieTest = new MovieTest();  
41         movieTest.getMovieById(829);  
42     }  
43 }  
44 }
```

B. Copy the selection to the clipboard.

7. In JDeveloper, paste the selection on the blank line between the last two “}” closures, as shown below.



The screenshot shows the JDeveloper IDE with the MovieTest.java file open. The code is as follows:

```
CustomerDAO.java x MovieUploader.java x MovieTest.java x
Find
package oracle.demo.ooow.bd.test;
import java.util.ArrayList;
import java.util.List;
import oracle.kv.KVStore;
import oracle.kv.KVStoreConfig;
import oracle.kv.KVStoreFactory;
import oracle.kv.Key;
import oracle.kv.Value;
import oracle.kv.ValueVersion;

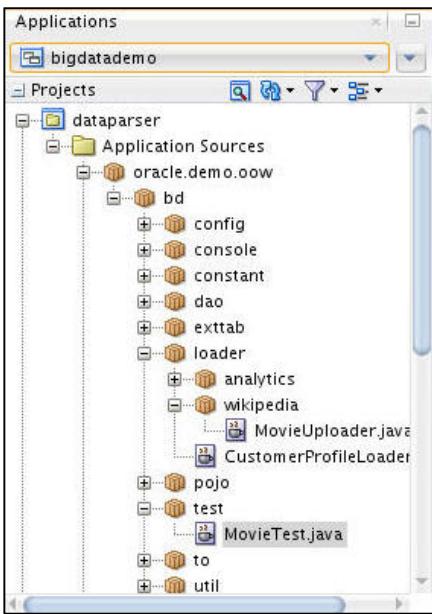
public class MovieTest {
    public MovieTest() {
        super();
    }
    public void getMovieById(int movieId) {
        Key key = null;
        KVStore kvStore = null;
        try {
            if (movieId > -1) {
                kvStore = KVStoreFactory.getStore(new KVStoreConfig("kvstore", "localhost:5000"));
                List<String> majorComponent = new ArrayList<String>();
                majorComponent.add("MV");
                majorComponent.add(Integer.toString(movieId));
                key = Key.createKey(majorComponent);
                ValueVersion valueVersion = kvStore.get(key);

                if (valueVersion != null) {
                    Value value = valueVersion.getValue();
                    String movieJsonTxt = new String(value.getValue());
                    System.out.println(movieJsonTxt);
                }
            }
        } catch (Exception e) {
            System.out.println("ERROR: Please make sure Oracle NoSQL Database is running");
        }
    }
    public static void main(String[] args) {
        MovieTest movieTest = new MovieTest();
        movieTest.getMovieById(829);
    }
}
```

**Note:** The pasted code performs the following.

- The `getMovieById()` method takes `movieId` as input.
- Then, in a `try/catch` block, the first `if` loop:
  - Creates a connection to a running instance of `kvstore`
  - Builds the key
  - Uses the `get()` method on `kvStore` to return the value
- The second `if` loop:
  - Uses the returned value from the first loop and converts it to a string
  - Prints out the associated descriptive movie text from the `movieJsonTxt` file
- The class's `main()` method is used to specify the Movie ID in the NoSQL database that is to be queried.

- Save the **MovieTest.java** class file, and then right-click on the same file in the Navigator (shown below) and select **Run** from the menu.



- After the program finishes, right-click on the Log window and select **Wrap** from the menu. Scroll down to see the Movie Description, which was returned from the NoSQL query.

```
Running: dataparser.jpr - Log
Actions ▾
Chinatown
Chinatown is a 1974 American neo-noir film, directed by Roman Polanski from a screenplay by Robert Towne and starring Jack Nicholson, Faye Dunaway, and John Huston. The film features many elements of the film noir genre, particularly a multi-layered story that is part mystery and part psychological drama. It was released by Paramount Pictures. The story, set in Los Angeles in 1937, was inspired by the California Water Wars, the historical disputes over land and water rights that had raged in southern California during the 1910s and 1920s, in which William Mulholland acted on behalf of Los Angeles interests to secure water rights in the Owens Valley. Chinatown was the last film Roman Polanski made in the United States before returning to Europe. Chinatown is frequently included in lists of the greatest films in world cinema. It holds second place on the American Film Institute list of Best Mystery Films. Chinatown was nominated for eleven Academy Awards, ultimately winning only Best Original Screenplay for Robert Towne. It also won Golden Globe Awards for Best Film, Best Director, Best Actor, and Best Screenplay. In 1991, Chinatown was selected for preservation in the United States National Film Registry by the Library of Congress as being "culturally, historically, or aesthetically significant." A sequel, The Two Jakes, was released in 1990, again starring Nicholson, who also directed, with Robert Towne returning to write the screenplay. The film failed to generate the acclaim of its predecessor.
@/6ybT8RbSbd4AltIDABuv39dgqMU.jpg@1974@...@...
Crime@Thriller@...
Drama
```

The screenshot shows the JDeveloper Log window with the title "Running: dataparser.jpr - Log". The log output is displayed in a monospaced font. It starts with the word "Chinatown" followed by a detailed description of the film, including its release year (1974), director (Roman Polanski), and stars (Jack Nicholson, Faye Dunaway, John Huston). The description highlights its status as a neo-noir film with elements of mystery and psychological drama. It mentions the film's inspiration from the California Water Wars and the role of William Mulholland. The log also notes that Chinatown was the last film Polanski made in the US before returning to Europe. It is described as frequently included in lists of greatest films and as holding second place on the AFI list of Best Mystery Films. The film was nominated for eleven Academy Awards, winning Best Original Screenplay for Robert Towne. It also won Golden Globe Awards for Best Film, Best Director, Best Actor, and Best Screenplay. In 1991, it was selected for preservation in the United States National Film Registry by the Library of Congress as being "culturally, historically, or aesthetically significant." A sequel, "The Two Jakes", was released in 1990, again starring Nicholson, who also directed, with Robert Towne returning to write the screenplay. The log concludes with file paths and genre information: "@/6ybT8RbSbd4AltIDABuv39dgqMU.jpg@1974@...@...", "Crime@Thriller@...", and "Drama".

- Exit JDeveloper. When prompted to terminate the running processes, select **Yes**.
- Select **File > Quit** to close the lab\_07\_04.txt file.

## **Practices for Lesson 8: Primary Administrative Tasks for Oracle NoSQL Database**

**Chapter 8**

## Practices for Lesson 8

---

There are no practices for this lesson.

## **Practices for Lesson 9: Introduction to MapReduce**

**Chapter 9**

## Practices for Lesson 9

---

### Practices Overview

In these practices, you will:

- Compile a `WordCount.java` program, which will run on a Hadoop cluster
- Upload the files on which to run the `WordCount` into Hadoop Distributed File System (HDFS)
- Run the `WordCount.java` program and view the results

## Practice 9-1: Running a MapReduce Hadoop Job

### Overview

In this practice, you run a word count Java program in the Hadoop cluster to count the number of word occurrences in two text files in HDFS.

### Assumptions

XYZ is a banking and insurance company. The company needs to create a marketing plan for the car insurance business to identify and target the current customers who are more likely to purchase car insurance (instead of using a scatter-gun approach). This involves analyzing a sample of customers who already decided whether or not to purchase car insurance. The company plans to create a customer profile from the customers who already purchased the car insurance. Based on the customer profiles identified through analysis, the company plans to target the other customers who have not yet decided to purchase the company's car insurance.

### Tasks

1. Open a terminal window.
2. Navigate to the folder that contains the scripts that you will use in this practice as follows:  

```
cd /home/oracle/exercises/wordCount
```
3. Review the WordCount.java program that runs the word count MapReduce job on a Hadoop cluster as follows. Exit gedit when you are done reviewing the program.  

```
gedit WordCount.java
```

**Note:** Line 14 in the WordCount.java program implements the Mapper interface while line 28 implements the Reducer interface.

4. Create a new directory named wordcount\_classes in the /home/oracle/exercises/wordCount local file system directory.  

```
mkdir wordcount_classes
```
5. Compile the WordCount.java program. Enter the following command to compile the WordCount.java program and to set the correct classpath and the output directory. Next, press **Enter**. The -d <directory> specifies the directory where to place the resulting class files in.  

```
javac -cp /usr/lib/hadoop/*:/usr/lib/hadoop/client-0.20/* -d wordcount_classes WordCount.java
```
6. Create a JAR file from the compile directory of WordCount. This JAR file is required because the code for word count will be sent to all of the nodes in the cluster and the code will run simultaneously on all nodes that have appropriate data. Enter the following command at the command prompt, and then press **Enter**.  

```
jar -cvf WordCount.jar -C wordcount_classes/ .
```

- View the contents of the two files on which you will perform the word count using the WordCount.java program. This enables you to see the customers' feedback about the car insurance services that the XYZ Company offers. The two files contain the customers' comments that will be used as the basis for the sentiment analysis. Enter the following command at the command prompt, and then press **Enter**.

```
cat file01 file02
```

- Create the /user/oracle/wordcount and the /user/oracle/wordcount/input directories in HDFS as follows:

```
hadoop fs -mkdir /user/oracle/wordcount  
hadoop fs -mkdir /user/oracle/wordcount/input
```

```
[oracle@bigdatalite wordCount]$ hadoop fs -mkdir /user/oracle/wordcount  
[oracle@bigdatalite wordCount]$ hadoop fs -mkdir /user/oracle/wordcount/input
```

- Confirm that the two HDFS directories are created by using the following commands:

```
hadoop fs -ls /user/oracle  
hadoop fs -ls /user/oracle/wordcount
```

- Copy the file01 and file02 files from the /home/oracle/exercises/wordCount local file system into the /user/oracle/wordcount/input directory in HDFS using the copyFromLocal command in Hadoop. Enter the following commands at the command prompt, and then press **Enter**.

```
hadoop fs -copyFromLocal file01  
/user/oracle/wordcount/input/file01
```

```
hadoop fs -copyFromLocal file02  
/user/oracle/wordcount/input/file02
```

**Remember that files in HDFS are split into multiple blocks or “chunks” and are stored on separate nodes in the Hadoop cluster for parallel parsing.**

- Confirm that the two files are copied successfully into the /user/oracle/wordcount/input directory in HDFS.

```
hadoop fs -ls /user/oracle/wordcount/input
```

- Run the MapReduce job to perform a word count on the files.

```
hadoop jar WordCount.jar WordCount /user/oracle/wordcount/input  
/user/oracle/wordcount/output
```

13. Display the contents of the /user/oracle/wordcount directory. This directory should contain two directories. Enter the following command at the command prompt, and then press **Enter**.

```
hadoop fs -ls /user/oracle/wordcount
```

14. Display the contents of the /user/oracle/wordcount/output directory. Enter the following command at the command prompt, and then press **Enter**.

```
hadoop fs -ls /user/oracle/wordcount/output
```

15. Display the contents of part-r-0000 results file from HDFS by using the cat command from Hadoop. Enter the following command at the command prompt, and then press **Enter**.

```
hadoop fs -cat /user/oracle/wordcount/output/part-r-00000
```

16. Re-run the MapReduce job again. Enter the following command at the command prompt, and then press **Enter**.

```
hadoop jar WordCount.jar WordCount /user/oracle/wordcount/input  
/user/oracle/wordcount/output
```

17. An error message is displayed and no MapReduce job is executed. You cannot update the data in the results directory because Hadoop does not allow updating of data files; only read and write operations are allowed. Therefore, the execution has nowhere to place the output. To re-run the MapReduce job, you must either point the job to another output directory or remove the files from the current output directory.

Remove the contents of the output directory. Enter the following command at the command prompt, and then press **Enter**.

```
hadoop fs -rm -r /user/oracle/wordcount/output
```

18. Re-run the MapReduce job.

```
hadoop jar WordCount.jar WordCount /user/oracle/wordcount/input  
/user/oracle/wordcount/output
```

19. Exit the terminal window.

```
exit
```

## Solution 9-1: Running a MapReduce Hadoop Job

### Overview

In this solution, you run a word count Java program in the Hadoop cluster to count the number of word occurrences in two text files in HDFS.

### Steps

1. Open a terminal window.



2. Navigate to the folder that contains the scripts that you will use in this practice as follows:

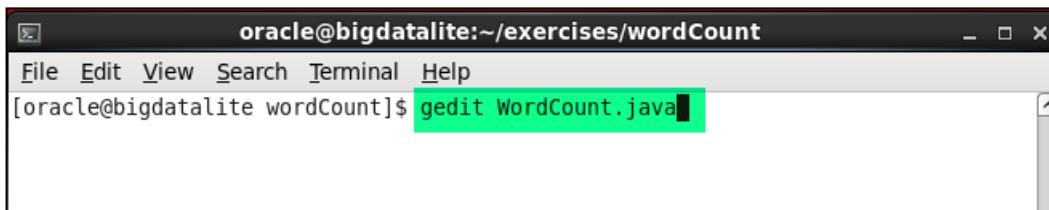
```
cd /home/oracle/exercises/wordCount
```



```
[oracle@bigdatalite ~]$ pwd  
/home/oracle  
[oracle@bigdatalite ~]$ cd /home/oracle/exercises/wordCount  
[oracle@bigdatalite wordCount]$ pwd  
/home/oracle/exercises/wordCount  
[oracle@bigdatalite wordCount]$ ls -l  
total 44  
-rw-r--r--. 1 oracle oinstall 554 Dec  3 08:53 cheat.sh  
-rw-r--r--. 1 oracle oinstall 165 Dec  3 08:53 cleanup.sh  
-rw-r--r--. 1 oracle oinstall 105 Dec  3 08:53 compile.sh  
-rw-r--r--. 1 oracle oinstall 147 Dec  3 08:53 copyFiles.sh  
-rw-r--r--. 1 oracle oinstall  58 Dec  3 08:53 createJar.sh  
-rw-r--r--. 1 oracle oinstall  58 Dec  3 08:53 deleteOutput.sh  
-rw-r--r--. 1 oracle oinstall 518 Dec  3 08:53 file01  
-rw-r--r--. 1 oracle oinstall 154 Dec  3 08:53 file02  
-rw-r--r--. 1 oracle oinstall 105 Dec  3 08:53 runWordCount.sh  
-rw-r--r--. 1 oracle oinstall  67 Dec  3 08:53 viewResults.sh  
-rw-r--r--. 1 oracle oinstall 1960 Dec  3 08:53 WordCount.java  
[oracle@bigdatalite wordCount]$
```

3. Review the WordCount.java program that runs the word count MapReduce job on a Hadoop cluster as follows:

```
gedit WordCount.java
```



```
[oracle@bigdatalite ~]$ gedit WordCount.java
```

```
WordCount.java X
1 import org.apache.hadoop.io.*;
2 import org.apache.hadoop.conf.Configuration;
3 import org.apache.hadoop.fs.Path;
4 import org.apache.hadoop.mapreduce.*;
5 import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
6 import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
7 import org.apache.hadoop.util.GenericOptionsParser;
8 import java.io.IOException;
9 import java.lang.InterruptedException;
10 import java.util.StringTokenizer;
11
12 public class WordCount {
13
14     public static class MyMapper extends Mapper<Object, Text, Text, IntWritable> {
15         private final static IntWritable one = new IntWritable(1);
16         private Text word = new Text();
17
18         public void map(Object key, Text value, Context context)
19             throws IOException, InterruptedException {
20             StringTokenizer itr = new StringTokenizer(value.toString());
21             while (itr.hasMoreTokens()) {
22                 word.set(itr.nextToken());
23                 context.write(word, one);
24             }
25         }
26     }
27 }
```

```
27
28     public static class MyReducer extends Reducer<Text, IntWritable, Text, IntWritable> {
29         public void reduce(Text key, Iterable<IntWritable> values, Context context)
30             throws IOException, InterruptedException {
31             int sum = 0;
32             for (IntWritable value : values) {
33                 sum += value.get();
34             }
35             context.write(key, new IntWritable(sum));
36         }
37     }
38
39     public static void main(String[] args) throws Exception {
40         Configuration conf = new Configuration();
41         String[] otherArgs = new GenericOptionsParser(conf, args).getRemainingArgs();
42         Job job = new Job(conf, "WordCount");
43         job.setJarByClass(WordCount.class);
44         job.setMapperClass(MyMapper.class);
45         job.setReducerClass(MyReducer.class);
46         job.setOutputKeyClass(Text.class);
47         job.setOutputValueClass(IntWritable.class);
48         FileInputFormat.addInputPath(job, new Path(otherArgs[0]));
49         FileOutputFormat.setOutputPath(job, new Path(otherArgs[1]));
50         System.exit(job.waitForCompletion(true) ? 0 : 1);
51     }
52 }
```

**Note:** Line 14 in the WordCount.java program implements the Mapper interface while line 28 implements the Reducer interface.

Exit gedit when you are done reviewing the program.

4. Create a new directory named wordcount\_classes in the /home/oracle/exercises/wordCount local file system directory.

```
mkdir wordcount_classes
```

```
[oracle@bigdatalite wordCount]$ pwd  
/home/oracle/exercises/wordCount  
[oracle@bigdatalite wordCount]$ mkdir wordcount_classes  
[oracle@bigdatalite wordCount]$ ls -l  
total 48  
-rw-r--r--. 1 oracle oinstall 554 Dec  3 08:53 cheat.sh  
-rw-r--r--. 1 oracle oinstall 165 Dec  3 08:53 cleanup.sh  
-rw-r--r--. 1 oracle oinstall 105 Dec  3 08:53 compile.sh  
-rw-r--r--. 1 oracle oinstall 147 Dec  3 08:53 copyFiles.sh  
-rw-r--r--. 1 oracle oinstall 58 Dec  3 08:53 createJar.sh  
-rw-r--r--. 1 oracle oinstall 58 Dec  3 08:53 deleteOutput.sh  
-rw-r--r--. 1 oracle oinstall 518 Dec  3 08:53 file01  
-rw-r--r--. 1 oracle oinstall 154 Dec  3 08:53 file02  
-rw-r--r--. 1 oracle oinstall 105 Dec  3 08:53 runWordCount.sh  
-rw-r--r--. 1 oracle oinstall 67 Dec  3 08:53 viewResults.sh  
drwxr-xr-x. 2 oracle oinstall 4096 Feb 17 07:22 wordcount_classes  
-rw-r--r--. 1 oracle oinstall 1960 Dec  3 08:53 WordCount.java  
[oracle@bigdatalite wordCount]$
```

```
[oracle@bigdatalite wordCount]$ cd wordcount_classes  
[oracle@bigdatalite wordcount_classes]$ ls -l  
total 0  
[oracle@bigdatalite wordcount_classes]$
```

5. Compile the WordCount.java program. Enter the following command to compile the WordCount.java program and to set the correct classpath and the output directory. Next, press **Enter**. The -d <directory> specifies the directory in which to place the resulting class files.

```
javac -cp /usr/lib/hadoop/*:/usr/lib/hadoop/client-0.20/* -d wordcount_classes WordCount.java
```

```
[oracle@bigdatalite wordCount]$ javac -cp /usr/lib/hadoop/*:/usr/lib/hadoop/client-0.20/* -d wordcount_classes WordCount.java  
[oracle@bigdatalite wordCount]$
```

```
[oracle@bigdatalite wordCount]$ cd wordcount_classes  
[oracle@bigdatalite wordcount_classes]$ ls -l  
total 12  
-rw-r--r--. 1 oracle oinstall 1604 Feb 17 07:24 WordCount.class  
-rw-r--r--. 1 oracle oinstall 1722 Feb 17 07:24 WordCount$MyMapper.class  
-rw-r--r--. 1 oracle oinstall 1633 Feb 17 07:24 WordCount$MyReducer.class  
[oracle@bigdatalite wordcount_classes]$
```

6. Create a JAR file from the compile directory of WordCount. This JAR file is required because the code for word count will be sent to all of the nodes in the cluster and the code will run simultaneously on all nodes that have appropriate data. Enter the following command at the command prompt, and then press **Enter**.

```
jar -cvf WordCount.jar -C wordcount_classes/ .
```

```
[oracle@bigdatalite wordcount_classes]$ cd ..
[oracle@bigdatalite wordCount]$ pwd
/home/oracle/exercises/wordCount
[oracle@bigdatalite wordCount]$ jar -cvf WordCount.jar -C wordcount_classes/ .
added manifest
adding: WordCount.class(in = 1604) (out= 845)(deflated 47%)
adding: WordCount$MyReducer.class(in = 1633) (out= 687)(deflated 57%)
adding: WordCount$MyMapper.class(in = 1722) (out= 753)(deflated 56%)
[oracle@bigdatalite wordCount]$
```

7. View the contents of the two files on which you will perform the word count by using the WordCount.java program. This enables you to see the customers' feedback about the car insurance services that the XYZ Company offers. The two files contain the customers' comments that will be used as the basis for the sentiment analysis. Enter the following command at the command prompt, and then press **Enter**.

```
cat file01 file02
```

```
[oracle@bigdatalite wordCount]$ cat file01 file02
very disappointed and very expensive
expensive and unreliable insurance
worthless insurance and expensive
worst customer service
worst insurance company
worst professional staff and unreliable insurance company
insurance is very expensive
worst insurance cover
terrible service
disappointed with the expensive insurance service
worthless and expensive
awful customer service
terrible worst service
worst bank and worst customer service
worst insurance
disappointed with protocols
unreliable insurance
best service I recommend it
good professionals and efficient insurance
I will recommend it
good customer service
best insurance I found I recommend it[oracle@bigdatalite wordCount]$
```

8. Create the /user/oracle/wordcount and the /user/oracle/wordcount/input directories in HDFS as follows:

```
hadoop fs -mkdir /user/oracle/wordcount
hadoop fs -mkdir /user/oracle/wordcount/input
```

```
[oracle@bigdatalite wordCount]$ hadoop fs -mkdir /user/oracle/wordcount
[oracle@bigdatalite wordCount]$ hadoop fs -mkdir /user/oracle/wordcount/input
```

9. Confirm that the two HDFS directories are created by using the following commands:

```
hadoop fs -ls /user/oracle
```

```
[oracle@bigdatalite wordCount]$ hadoop fs -ls /user/oracle
Found 7 items
drwx-----  - oracle oracle      0 2014-08-25 05:55 /user/oracle/.Trash
drwx-----  - oracle oracle      0 2014-09-23 13:25 /user/oracle/.staging
drwxr-xr-x  - oracle oracle      0 2014-01-12 18:15 /user/oracle/moviedemo
drwxr-xr-x  - oracle oracle      0 2014-09-24 09:38 /user/oracle/moviework
drwxr-xr-x  - oracle oracle      0 2014-09-08 15:50 /user/oracle/oggdemo
drwxr-xr-x  - oracle oracle      0 2014-09-20 13:59 /user/oracle/oozie-oozi
drwxr-xr-x  - oracle oracle      0 2015-02-17 08:20 /user/oracle/wordcount
```

```
hadoop fs -ls /user/oracle/wordcount
```

```
[oracle@bigdatalite wordCount]$ hadoop fs -ls /user/oracle/wordcount
Found 1 items
drwxr-xr-x  - oracle oracle      0 2015-02-17 08:20 /user/oracle/wordcount/
input
[oracle@bigdatalite wordCount]$
```

10. Copy the `file01` and `file02` files from the `/home/oracle/exercises/wordCount` local file system into the `/user/oracle/wordcount/input` directory in HDFS using the `copyFromLocal` command in Hadoop. Enter the following two commands at the command prompt, and then press **Enter**.

```
hadoop fs -copyFromLocal file01
/user/oracle/wordcount/input/file01
```

```
hadoop fs -copyFromLocal file02
/user/oracle/wordcount/input/file02
```

```
[oracle@bigdatalite wordCount]$ hadoop fs -copyFromLocal file01 /user/oracle/wordcount/input/file01
[oracle@bigdatalite wordCount]$ hadoop fs -copyFromLocal file02 /user/oracle/wordcount/input/file02
```

**Remember that files in HDFS are split into multiple blocks or “chunks” and are stored on separate nodes in the Hadoop cluster for parallel parsing.**

11. Confirm that the two files are copied successfully into the `/user/oracle/wordcount/input` directory in HDFS.

```
hadoop fs -ls /user/oracle/wordcount/input
```

```
[oracle@bigdatalite wordCount]$ hadoop fs -ls /user/oracle/wordcount/input
Found 2 items
-rw-r--r--  1 oracle oracle      518 2015-02-17 08:52 /user/oracle/wordcount/
input/file01
-rw-r--r--  1 oracle oracle      154 2015-02-17 08:52 /user/oracle/wordcount/
input/file02
[oracle@bigdatalite wordCount]$
```

12. Run the MapReduce job to perform a word count on the files.

```
hadoop jar WordCount.jar WordCount /user/oracle/wordcount/input  
/user/oracle/wordcount/output
```

```
[oracle@bigdatalite wordCount]$ hadoop jar WordCount.jar WordCount /user/oracle/  
wordcount/input /user/oracle/wordcount/output
```

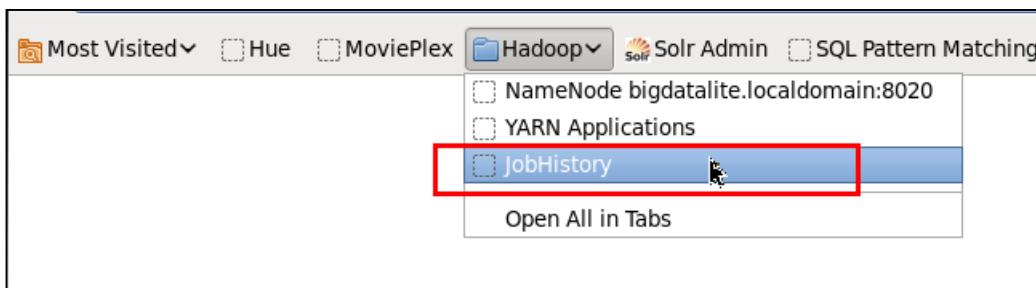
**Informational text from the Hadoop infrastructure is displayed on the screen to help you track the status of the job. When the job is completed, the command prompt is displayed. The highlighted information shows the MapReduce job ID you will need if you want to kill the job while it is running. This is covered in more detail in practice 10.**

In addition, you can track the job using the provided URL. Alternatively, you can access the YARN or JobHistory GUI from the Hadoop > YARN applications or the Hadoop > JobHistory bookmarks in your BDLite Web browser toolbar. The job\_id and output shown in the screen capture might be different than your output. This is expected.

```
[oracle@bigdatalite wordCount]$ hadoop jar WordCount.jar WordCount /user/oracle/  
wordcount/input /user/oracle/wordcount/output  
15/02/17 09:02:45 INFO client.RMProxy: Connecting to ResourceManager at localhos  
t/127.0.0.1:8032  
15/02/17 09:02:46 INFO input.FileInputFormat: Total input paths to process : 2  
15/02/17 09:02:46 INFO mapreduce.JobSubmitter: number of splits:2  
15/02/17 09:02:46 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_14  
24084899599_0001  
15/02/17 09:02:47 INFO impl.YarnClientImpl: Submitted application application_14  
24084899599_0001  
15/02/17 09:02:47 INFO mapreduce.Job: The url to track the job: http://bigdatali  
te.localdomain:8088/proxy/application_1424084899599_0001/  
15/02/17 09:02:47 INFO mapreduce.Job: Running job: job_1424084899599_0001  
15/02/17 09:02:55 INFO mapreduce.Job: Job job_1424084899599_0001 running in uber  
mode : false  
15/02/17 09:02:55 INFO mapreduce.Job: map 0% reduce 0%  
15/02/17 09:03:02 INFO mapreduce.Job: map 50% reduce 0%  
15/02/17 09:03:03 INFO mapreduce.Job: map 100% reduce 0%  
15/02/17 09:03:08 INFO mapreduce.Job: map 100% reduce 100%  
15/02/17 09:03:09 INFO mapreduce.Job: Job job_1424084899599_0001 completed succe  
ssfully  
15/02/17 09:03:09 INFO mapreduce.Job: Counters: 49  
File System Counters  
FILE: Number of bytes read=1180  
FILE: Number of bytes written=276409  
FILE: Number of read operations=0  
FILE: Number of large read operations=0  
FILE: Number of write operations=0  
HDFS: Number of bytes read=942  
HDFS: Number of bytes written=282  
HDFS: Number of read operations=9  
HDFS: Number of large read operations=0  
HDFS: Number of write operations=2  
Job Counters
```

```
Total time spent by all reduces in occupied slots (ms)=3816
Total time spent by all map tasks (ms)=10947
Total time spent by all reduce tasks (ms)=3816
Total vcore-seconds taken by all map tasks=10947
Total vcore-seconds taken by all reduce tasks=3816
Total megabyte-seconds taken by all map tasks=2802432
Total megabyte-seconds taken by all reduce tasks=976896
Map-Reduce Framework
  Map input records=22
  Map output records=87
  Map output bytes=1000
  Map output materialized bytes=1186
  Input split bytes=270
  Combine input records=0
  Combine output records=0
  Reduce input groups=30
  Reduce shuffle bytes=1186
  Reduce input records=87
  Reduce output records=30
  Spilled Records=174
  Shuffled Maps =2
  Failed Shuffles=0
  Merged Map outputs=2
  GC time elapsed (ms)=177
  CPU time spent (ms)=1870
  Physical memory (bytes) snapshot=700084224
  Virtual memory (bytes) snapshot=2069086208
  Total committed heap usage (bytes)=572522496
Shuffle Errors
  BAD_ID=0
  CONNECTION=0
  IO_ERROR=0
  WRONG_LENGTH=0
  WRONG_MAP=0
  WRONG_REDUCE=0
File Input Format Counters
  Bytes Read=672
File Output Format Counters
  Bytes Written=282
[oracle@bigdatalite wordCount]$
```

You can view the completed MapReduce job details by using the JobHistory service on your web browser link. Open your web browser and click Hadoop > JobHistory from the Bookmarks toolbar as follows:



The screenshot shows the Hadoop JobHistory interface in Mozilla Firefox. The title bar says "JobHistory - Mozilla Firefox". The address bar shows the URL "bigdatalite.localdomain:19888/jobhistory". The page header includes links for "JobHistory", "bigdatalite.localdomain:19888/jobhistory", "Most Visited", "Hue", "MoviePlex", "Hadoop", "Solr Admin", "SQL Pattern Matching", "VirtualBox VMs for ...", and "Cloudera Manager". On the left, there's a sidebar with "Application" dropdown (About Jobs selected), "Tools" button, and a logo of a yellow elephant with the word "hadoop". The main content area is titled "Retired Jobs" and shows a table with 20 entries. One row is highlighted with a red border, showing a job ID: "job\_1424084899599\_0001". The table columns include: Submit Time, Start Time, Finish Time, Job ID, Name, User, Queue, State, Maps Total, and Co. The "Maps Total" column for the highlighted row shows the value "2".

13. Display the contents of the /user/oracle/wordcount directory. This directory should contain two directories. Enter the following command at the command prompt, and then press **Enter**.

```
hadoop fs -ls /user/oracle/wordcount
```

```
[oracle@bigdatalite wordCount]$ hadoop fs -ls /user/oracle/wordcount
Found 2 items
drwxr-xr-x - oracle oracle          0 2015-02-17 08:52 /user/oracle/wordcount/
input
drwxr-xr-x - oracle oracle          0 2015-02-17 09:03 /user/oracle/wordcount/
output
[oracle@bigdatalite wordCount]$
```

The output directory was created in step 12. The input directory contains the file01 and file02 files.

14. Display the contents of the /user/oracle/wordcount/output directory. Enter the following command at the command prompt, and then press **Enter**.

```
hadoop fs -ls /user/oracle/wordcount/output
```

```
[oracle@bigdatalite wordCount]$ hadoop fs -ls /user/oracle/wordcount/output
Found 2 items
-rw-r--r-- 1 oracle oracle          0 2015-02-17 09:03 /user/oracle/wordcount/
output/_SUCCESS
-rw-r--r-- 1 oracle oracle        282 2015-02-17 09:03 /user/oracle/wordcount/
output/part-r-00000
[oracle@bigdatalite wordCount]$
```

15. Display the contents of the part-r-0000 results file from HDFS by using the cat command from Hadoop. Enter the following command at the command prompt, and then press **Enter**.

```
hadoop fs -cat /user/oracle/wordcount/output/part-r-00000
```

```
[oracle@bigdatalite wordCount]$ hadoop fs -cat /user/oracle/wordcount/output/part-r-00000
I          4
and        7
awful      1
bank        1
best        2
company    2
cover      1
customer    4
disappointed 3
efficient    1
expensive    6
found       1
good        2
insurance   11
is          1
it          3
professional 1
professionals 1
protocols    1
recommend    3
service     8
staff       1
terrible     2
the         1
unreliable   3
very        3
will        1
with        2
worst       8
worthless    2
[oracle@bigdatalite wordCount]$
```

The results of the WordCount program are displayed. The WordCount program counted the number of occurrences of each word in the file01 and file02 text files in the /user/oracle/wordcount/input HDFS directory. Most of the customers' comments are negative:

- Worst: 8 times
- Unreliable: 3 times
- Expensive: 6 times,
- Worthless: 2 times
- Good: 2 times.
- Terrible: 2 times

16. Re-run the MapReduce job again. Enter the following command at the command prompt, and then press **Enter**.

```
hadoop jar WordCount.jar WordCount /user/oracle/wordcount/input  
/user/oracle/wordcount/output
```

```
[oracle@bigdatalite wordCount]$ hadoop jar WordCount.jar WordCount /user/oracle/  
wordcount/input /user/oracle/wordcount/output  
15/02/17 11:19:10 INFO client.RMProxy: Connecting to ResourceManager at localhos  
t/127.0.0.1:8032  
15/02/17 11:19:10 WARN security.UserGroupInformation: PrivilegedActionException  
as:oracle (auth:SIMPLE) cause:org.apache.hadoop.mapred.FileAlreadyExistsExcepti  
on: Output directory hdfs://bigdatalite.localdomain:8020/user/oracle/wordcount/o  
utput already exists  
Exception in thread "main" org.apache.hadoop.mapred.FileAlreadyExistsException:  
Output directory hdfs://bigdatalite.localdomain:8020/user/oracle/wordcount/outpu  
t already exists  
    at org.apache.hadoop.mapreduce.lib.output.FileOutputFormat.checkOutputSp  
ecs(FileOutputFormat.java:146)  
    at org.apache.hadoop.mapreduce.JobSubmitter.checkSpecs(JobSubmitter.java  
:458)  
    at org.apache.hadoop.mapreduce.JobSubmitter.submitJobInternal(JobSubmitte  
r.java:343)  
    at org.apache.hadoop.mapreduce.Job$10.run(Job.java:1295)  
    at org.apache.hadoop.mapreduce.Job$10.run(Job.java:1292)  
    at java.security.AccessController.doPrivileged(Native Method)  
    at javax.security.auth.Subject.doAs(Subject.java:415)  
    at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInforma  
tion.java:1554)  
    at org.apache.hadoop.mapreduce.Job.submit(Job.java:1292)  
    at org.apache.hadoop.mapreduce.Job.waitForCompletion(Job.java:1313)  
    at WordCount.main(WordCount.java:50)  
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)  
    at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.  
java:57)  
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAcces  
sorImpl.java:43)  
    at java.lang.reflect.Method.invoke(Method.java:606)  
    at org.apache.hadoop.util.RunJar.main(RunJar.java:212)  
[oracle@bigdatalite wordCount]$ █
```

17. An error message is displayed and no MapReduce job is executed. You cannot update the data in the results directory because Hadoop does not allow updating of data files; only read and write operations are allowed. Therefore, the execution has nowhere to place the output. To re-run the MapReduce job, you must either point the job to another output directory or remove the files from the current output directory.  
Remove the contents of the output directory. Enter the following command at the command prompt, and then press **Enter**.

```
hadoop fs -rm -r /user/oracle/wordcount/output
```

```
[oracle@bigdatalite wordCount]$ hadoop fs -rm -r /user/oracle/wordcount/output  
15/02/17 11:27:17 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele  
tion interval = 0 minutes, Emptier interval = 0 minutes.  
Deleted /user/oracle/wordcount/output  
[oracle@bigdatalite wordCount]$ hadoop fs -ls /user/oracle/wordcount/output  
ls: `/user/oracle/wordcount/output': No such file or directory  
[oracle@bigdatalite wordCount]$ █
```

18. Re-run the MapReduce job.

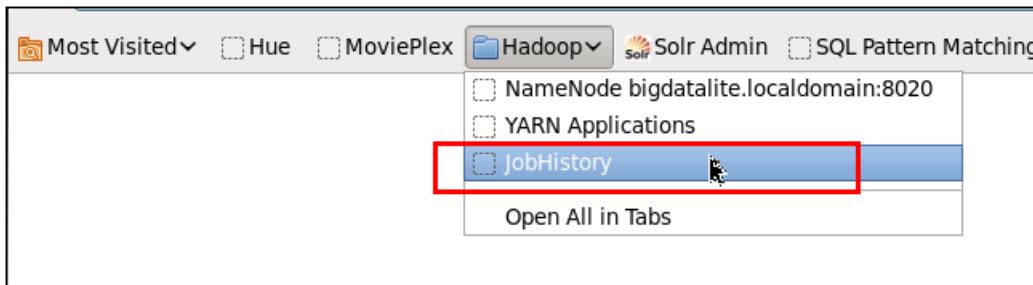
```
hadoop jar WordCount.jar WordCount /user/oracle/wordcount/input  
/user/oracle/wordcount/output
```

The MapReduce job runs successfully and the results are displayed as follows:

```
[oracle@bigdatalite wordCount]$ hadoop jar WordCount.jar WordCount /user/oracle/  
wordcount/input /user/oracle/wordcount/output  
15/02/17 11:29:56 INFO client.RMProxy: Connecting to ResourceManager at localhos  
t/127.0.0.1:8032  
15/02/17 11:29:57 INFO input.FileInputFormat: Total input paths to process : 2  
15/02/17 11:29:57 INFO mapreduce.JobSubmitter: number of splits:2  
15/02/17 11:29:57 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_14  
24182679355_0001  
15/02/17 11:29:57 INFO impl.YarnClientImpl: Submitted application application_14  
24182679355_0001  
15/02/17 11:29:57 INFO mapreduce.Job: The url to track the job: http://bigdatali  
te.localdomain:8088/proxy/application_1424182679355_0001/  
15/02/17 11:29:57 INFO mapreduce.Job: Running job: job_1424182679355_0001  
15/02/17 11:30:06 INFO mapreduce.Job: Job job_1424182679355_0001 running in uber  
mode : false  
15/02/17 11:30:06 INFO mapreduce.Job: map 0% reduce 0%  
15/02/17 11:30:14 INFO mapreduce.Job: map 100% reduce 0%  
15/02/17 11:30:20 INFO mapreduce.Job: map 100% reduce 100%  
15/02/17 11:30:20 INFO mapreduce.Job: Job job_1424182679355_0001 completed succe  
ssfully  
15/02/17 11:30:20 INFO mapreduce.Job: Counters: 49  
File System Counters  
    FILE: Number of bytes read=1180  
    FILE: Number of bytes written=276409  
    FILE: Number of read operations=0  
    FILE: Number of large read operations=0  
    FILE: Number of write operations=0  
    HDFS: Number of bytes read=942  
    HDFS: Number of bytes written=282  
    HDFS: Number of read operations=9  
    HDFS: Number of large read operations=0  
    HDFS: Number of write operations=2  
Job Counters  
    Launched map tasks=2  
    Launched reduce tasks=1  
    Data-local map tasks=2  
    Total time spent by all maps in occupied slots (ms)=10660
```

```
Total time spent by all reduces in occupied slots (ms)=3694
Total time spent by all map tasks (ms)=10660
Total time spent by all reduce tasks (ms)=3694
Total vcore-seconds taken by all map tasks=10660
Total vcore-seconds taken by all reduce tasks=3694
Total megabyte-seconds taken by all map tasks=2728960
Total megabyte-seconds taken by all reduce tasks=945664
Map-Reduce Framework
    Map input records=22
    Map output records=87
    Map output bytes=1000
    Map output materialized bytes=1186
    Input split bytes=270
    Combine input records=0
    Combine output records=0
    Reduce input groups=30
    Reduce shuffle bytes=1186
    Reduce input records=87
    Reduce output records=30
    Spilled Records=174
    Shuffled Maps =2
    Failed Shuffles=0
    Merged Map outputs=2
    GC time elapsed (ms)=122
    CPU time spent (ms)=1880
    Physical memory (bytes) snapshot=690888704
    Virtual memory (bytes) snapshot=2062200832
    Total committed heap usage (bytes)=560988160
Shuffle Errors
    BAD_ID=0
    CONNECTION=0
    IO_ERROR=0
    WRONG_LENGTH=0
    WRONG_MAP=0
    WRONG_REDUCE=0
File Input Format Counters
    Bytes Read=672
File Output Format Counters
    Bytes Written=282
[oracle@bigdatalite wordCount]$
```

You can view the completed MapReduce job details by using the JobHistory service on your web browser link. Open your web browser and click Hadoop > JobHistory from the Bookmarks toolbar as follows:



The second highlighted MapReduce job is the job that you ran in step 18 in this practice.

The screenshot shows the Hadoop JobHistory interface. At the top left is the Hadoop logo. To its right is the title "JobHistory". On the left side, there's a sidebar with "Application" expanded, showing "About Jobs" and "Tools". The main area is titled "Retired Jobs" and contains a table of completed jobs. The table has columns: Submit Time, Start Time, Finish Time, Job ID, Name, User, Queue, State, Maps Total, and Maps Completed. There are two rows of data:

Submit Time	Start Time	Finish Time	Job ID	Name	User	Queue	State	Maps Total	Maps Completed
2015.02.17 11:29:57 EST	2015.02.17 11:30:05 EST	2015.02.17 11:30:18 EST	job_1424182679355_0001	WordCount	oracle	root.oracle	SUCCEEDED	2	2
2015.02.17 09:02:46 EST	2015.02.17 09:02:53 EST	2015.02.17 09:03:07 EST	job_1424084899599_0001	WordCount	oracle	root.oracle	SUCCEEDED	2	2

At the bottom of the table, it says "Showing 1 to 2 of 2 entries".

19. Exit the terminal window.

```
exit
```

## **Practices for Lesson 10: Resource Management Using YARN**

**Chapter 10**

## Practices for Lesson 10

---

### Practices Overview

In this practice, you monitor a running job by using the CLI and the Resource Manager UI.

## Practice 10-1: Resource Management Using YARN

### Overview

In this practice, you monitor a running job by using the CLI and the Resource Manager UI.

### Assumptions

### Tasks

1. Start up your Mozilla Firefox web browser.
2. Click Hue on the browser's toolbar.
3. If the Hive Editor is not displayed, select **Query Editors > Hive** from the Hue's menu bar to display the Hive Editor.
4. Click **Saved Queries**.

**Note:** Do not close this tab. You will use HUE in later steps in this practice.

5. Open a new tab in your web browser, and then click **Hadoop > YARN Applications**. Note that no applications were run up to this point on your VM with today's date. Your screen will be different than the screen capture shown in this step. In this example, you assume that today is Thursday, February 14. Explore the page. Scroll to the right to see additional information about the jobs displayed in the **Cluster Metrics** section.
6. Open a new terminal window in order to track jobs at the command prompts.
7. You can use the following `yarn application` command line to view the status of the applications. Enter the following command at the command prompt to check and see if there are any applications that are currently running. The screen capture shows that there are no jobs currently running.

```
$ yarn applications -list
```

**Note:** Do not close your terminal window. In the next step, you will run a HiveQL query in Hue, and you will switch back quickly to this terminal window, and use the up arrow key on your keyboard (command history) to display and run the command you used in this step to see if there are any applications that are running.

8. Return to the Hue tab in your browser. Click the **Saved Queries** link, and then click the **Insert into staging table** link. The HiveQL query is displayed. Click **Execute** to run the query.

**Note:** When you click **Execute**, switch back quickly to the terminal window that you used in step 7, and use the up arrow key on your keyboard (command history) to re-display and run the `yarn applications -list` command you used in step 7 to see if there are any applications that are running.

The HiveQL query that you executed started a MapReduce job as shown in the preceding example.

9. Return to the **YARN All Applications** tab in your Browser from step 5 in this practice. Refresh the screen. Click the **Reload current page** in the URL or address bar in your web browser. The MapReduce job that was started when you executed the HiveQL should be displayed.

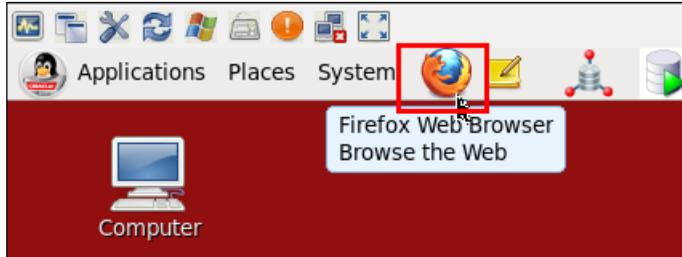
## Solution 10-1: Resource Management Using YARN

### Overview

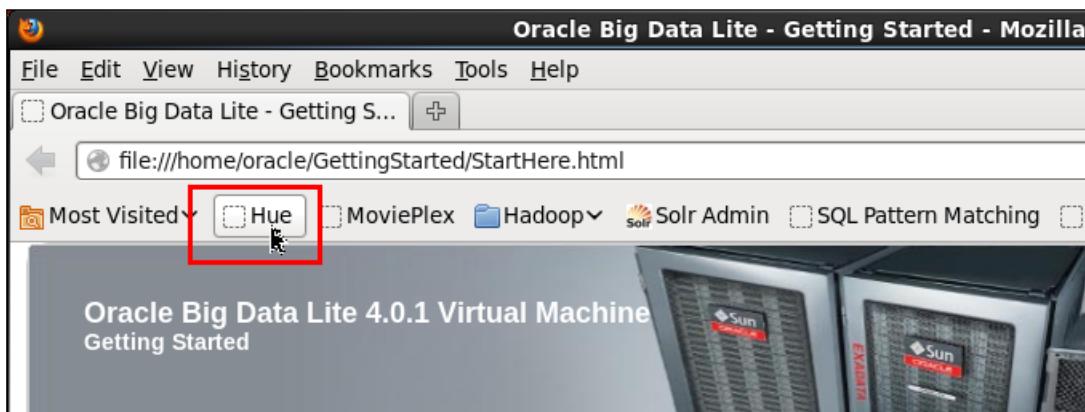
In this solution, you monitor a running job by using the CLI and the Resource Manager UI.

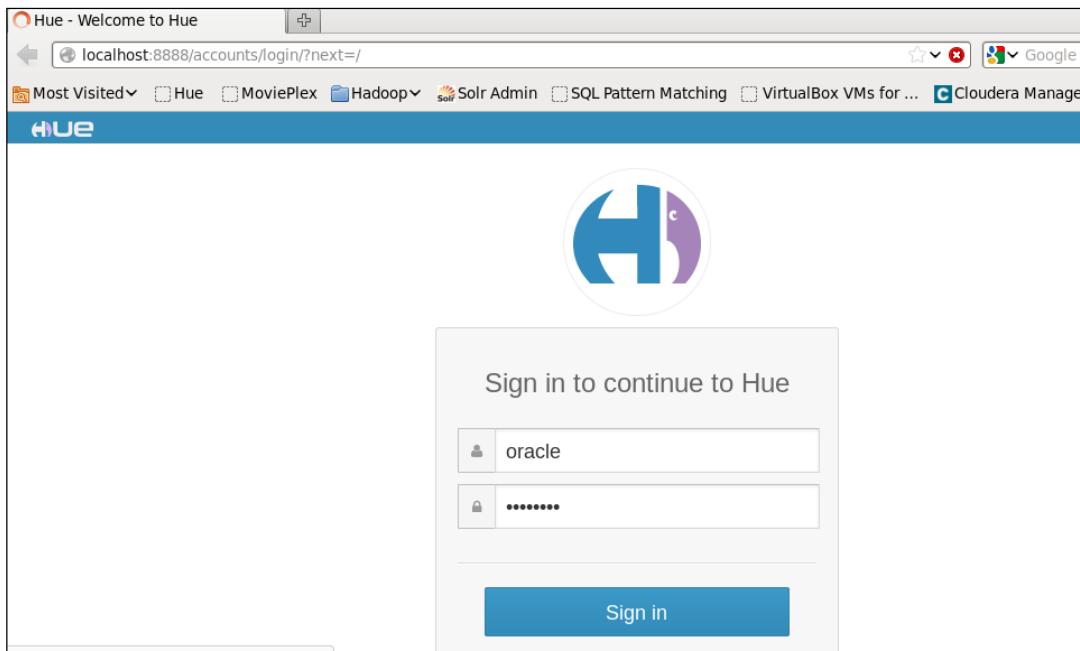
### Steps

1. Start up your Mozilla Firefox web browser.

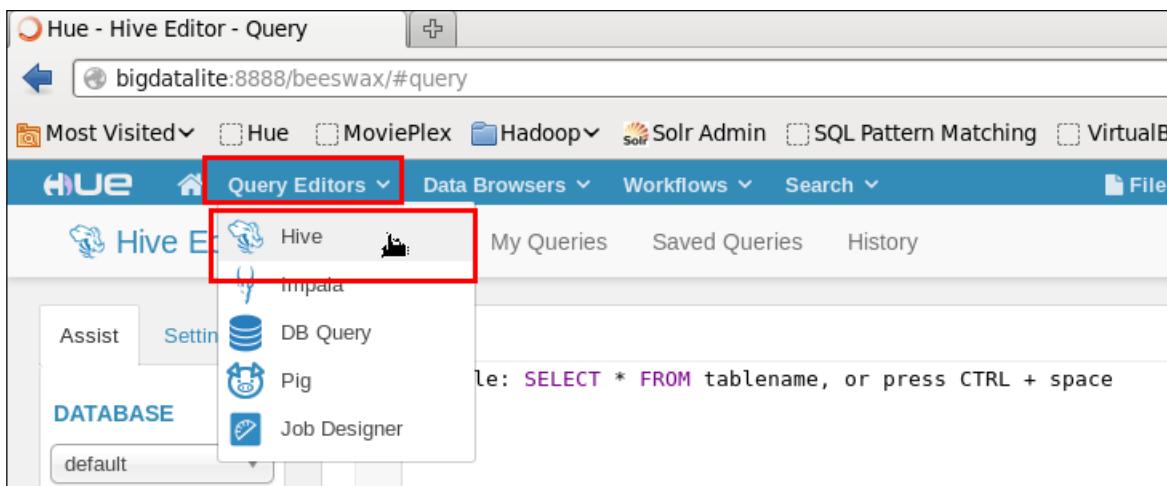


2. Click Hue on the browser's toolbar. The username and password fields should be already populated; if not, use the credentials oracle/welcome1. Click **Sign in**.





3. If the Hive Editor is not displayed, select **Query Editors > Hive** from Hue's menu bar to display the Hive Editor.



4. Click **Saved Queries**.

The screenshot shows the Hue interface with the 'Query Editor' tab selected. A red box highlights the 'Saved Queries' tab in the top navigation bar. Below the tabs, there are buttons for 'Assist' and 'Settings', and a search bar with the placeholder 'Example: SELECT \* FROM tablename, or press CTRL + space'. The URL in the address bar is 'bigdatalite:8888/beeswax/#query'.

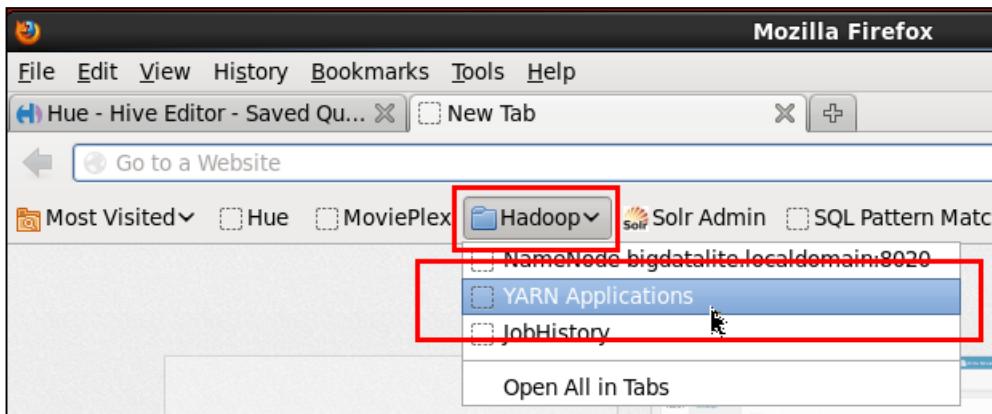
The screenshot shows the 'Saved Queries' page. A red box highlights the 'Saved Queries' tab in the top navigation bar. Below it, the page title is 'Saved Queries'. There is a search bar labeled 'Search for query' and a row of buttons: 'Edit', 'Copy', 'Usage history', and 'Move to trash'. On the left, there is a list of saved queries with checkboxes next to them, and a red box highlights the first item: 'Name'. To the right, there is a 'Description' column with three entries:

Name	Description
2. Insert into staging table	Inserts filtered and transformed log data into staging table
3. Query Staging Table	Queries the staging table that is the source for Oracle Database
1. Query Application Log	Queries application log containing AVRO data

**Note:** Do not close this tab. You will use HUE in later steps in this practice.

5. Open a new tab in your web browser, and then click **Hadoop > YARN Applications**.

**Note:** If you ran the two MapReduce jobs in practice 9, then you should see those jobs. Your screen will be different than the screen capture shown in this step. In this example, you assume that today is Tuesday, February 17. Explore the page. Scroll to the right to see additional information about the job(s) displayed in the **Cluster Metrics** section.



The screenshot shows the "Hue - Hive Editor - Query" interface. The title bar says "localhost:8088/cluster". The main content is titled "All Applications" with a Hadoop logo. On the left, there's a sidebar with "Cluster Metrics" and a "User Metrics for dr.who" table. The "User Metrics" table has one entry:

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	Active Nodes	Decommissioned Nodes	Lost Nodes
1	0	0	1	0	0 B	4 GB	0 B	1	0	0

Below the table, there's a search bar and a table showing application details:

ID	User	Name	Application Type	Queue	StartTime	FinishTime	State	FinalSt.
application_1424182679355_0001	oracle	WordCount	MAPREDUCE	root.oracle	Tue, 17 Feb 2015 16:29:57 GMT	Tue, 17 Feb 2015 16:30:19 GMT	FINISHED	SUCCESS

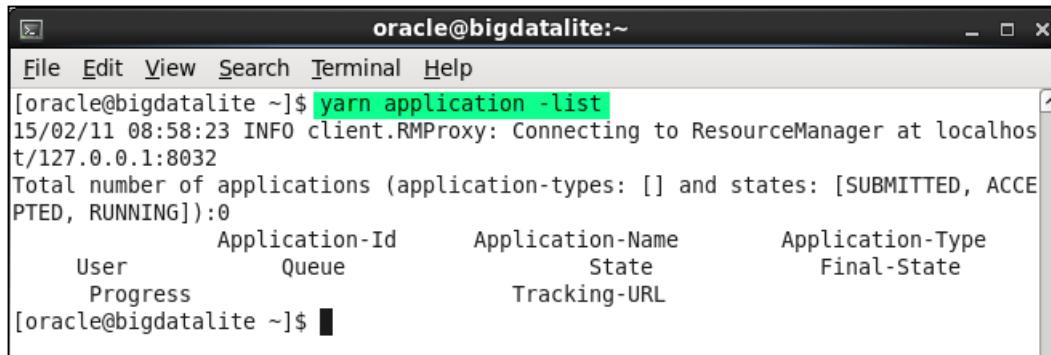
At the bottom, it says "Showing 1 to 1 of 1 entries".

6. Open a new terminal window to track jobs at the command prompt.



7. You can use the following `yarn application` command line to view the status of applications. Enter the following command at the command prompt to check and see if there are any applications that are currently running.

```
$ yarn application -list
```



The screenshot shows a terminal window titled "oracle@bigdatalite:~". The window displays the command `yarn application -list` and its output. The output includes a header for the application list and a table with columns for Application-ID, Application-Name, Application-Type, User, Queue, State, Final-State, Progress, and Tracking-URL. The table is empty, indicating no applications are currently running.

Application-ID	Application-Name	Application-Type	
User	Queue	State	Final-State
Progress		Tracking-URL	

The screen capture shows that there are no jobs currently running.

**Note:** Do not close your terminal window. In the next step, you will run a HiveQL query in Hue, and you will switch back quickly to this terminal window, and use the up arrow key on your keyboard (command history) to display and run the command you used in this step to see if there are any applications that are running.

8. Return to the Hue tab in your browser. Click the **Saved Queries** link, and then click the **Insert into staging table** link. The HiveQL query is displayed. Click **Execute** to run the query.

**Note:** When you click **Execute**, switch back quickly to the terminal window that you used in step 7, and use the up arrow key on your keyboard (command history) to re-display and run the `yarn applications -list` command you used in step 7 to see if there are any applications that are running. **In addition, the results shown in this practice will not exactly match your results.**

The screenshot shows the Hue web interface with the title bar "Hue - Hive Editor - Saved Queries". The navigation bar includes links for Most Visited, Hue, MoviePlex, Hadoop, Solr Admin, SQL Pattern Matching, and VirtualBox. Below the navigation bar, there are tabs for "HUE", "Query Editors", "Data Browsers", "Workflows", and "Search". The "Saved Queries" tab is currently active and highlighted with a red box. The main content area is titled "Saved Queries" and contains a search bar and several query entries. The first entry, "2. Insert into staging table", is highlighted with a red box. The other two entries are "3. Query Staging Table" and "1. Query Application Log". Each entry has a "Description" field below it.

Name	Description
2. Insert into staging table	Inserts filtered and transformed log data into staging table
3. Query Staging Table	Queries the staging table that is the source for Oracle Database
1. Query Application Log	Queries application log containing AVRO data

The screenshot shows the details of the "2. Insert into staging table" query. The title is "2. Insert into staging table" and the description is "Inserts filtered and transformed log data into staging table". The query code is displayed in a code editor:

```

1 INSERT OVERWRITE TABLE moviework.movieapp_log_stage
2 SELECT * FROM (
3   SELECT custid,
4     movieid,
5     CASE WHEN genreid > 0 THEN genreid ELSE -1 END genreid,
6     time,
7     CAST((CASE recommended WHEN 'Y' THEN 1 ELSE 0 END) AS INT) recommended,
8     activity,
9     cast(null AS INT) rating,
10    price
11   FROM moviework.movieapp_log_avro
12   WHERE activity IN (2,4,5,11)
13 UNION ALL
14   SELECT
15     m1.custid,
16     m1.movieid,
17     CASE WHEN m1.genreid > 0 THEN m1.genreid ELSE -1 END genreid,
18     m1.time,
19     CAST((CASE m1.recommended WHEN 'Y' THEN 1 ELSE 0 END) AS INT) recommended,
20     m1.activity,
21     m1.rating

```

At the bottom of the query editor, there are several buttons: "Execute" (highlighted with a red box), "Save", "Save as...", "Explain", and "or create a New query".

```
[oracle@bigdatalite ~]$ yarn application -list
15/02/17 14:35:37 INFO client.RMProxy: Connecting to ResourceManager at localhos
t/127.0.0.1:8032
Total number of applications (application-types: [] and states: [SUBMITTED, ACCE
PTED, RUNNING]):1
      Application-Id      Application-Name      Application-Type
      User        Queue        State        Final-State
      Progress           Tracking-URL
application_1424182679355_0003  INSERT OVERWRITE TABLE moview...union_result(Sta
ge-1)          MAPREDUCE      oracle      root.oracle      RUNNI
NG          UNDEFINED           50% http://bigdatalite.localdomain:1
4674
[oracle@bigdatalite ~]$ yarn application -list
15/02/17 14:35:42 INFO client.RMProxy: Connecting to ResourceManager at localhos
t/127.0.0.1:8032
Total number of applications (application-types: [] and states: [SUBMITTED, ACCE
PTED, RUNNING]):0
      Application-Id      Application-Name      Application-Type
      User        Queue        State        Final-State
      Progress           Tracking-URL
[oracle@bigdatalite ~]$ c
```

The HiveQL query that you executed started a MapReduce job as shown in the preceding example.

9. Return to the **YARN All Applications** tab in your Browser from step 5 in this practice. Refresh the screen. Click the **Reload current page** in the URL or address bar in your web browser. The MapReduce job that was started when you executed the HiveQL should be displayed.

Scheduler	ID	User	Name	Application Type	Queue	StartTime	FinishTime
NEW_SAVING	application_1424182679355_0006	oracle	INSERT OVERWRITE TABLE moview...union_result(Stage-4)	MAPREDUCE	root.oracle	Tue, 17 Feb 2015 19:36:28 GMT	Tue, 17 Feb 2015 19:36:41 GMT
SUBMITTED	application_1424182679355_0005	oracle	INSERT OVERWRITE TABLE moview...union_result(Stage-3)	MAPREDUCE	root.oracle	Tue, 17 Feb 2015 19:36:07	Tue, 17 Feb 2015 19:36:26 GMT
ACCEPTED	application_1424182679355_0004	oracle	INSERT OVERWRITE TABLE moview...union_result(Stage-11)	MAPREDUCE	root.oracle	Tue, 17 Feb 2015 19:35:48	Tue, 17 Feb 2015 19:36:04 GMT
RUNNING	application_1424182679355_0003	oracle	INSERT OVERWRITE TABLE moview...union_result(Stage-1)	MAPREDUCE	root.oracle	Tue, 17 Feb 2015 19:35:12	Tue, 17 Feb 2015 19:35:41 GMT
FINISHED	application_1424182679355_0002	oracle	INSERT OVERWRITE TABLE moview...union_result(Stage-1)	MAPREDUCE	root.oracle	Tue, 17 Feb 2015 19:15:21	Tue, 17 Feb 2015 19:16:12 GMT
KILLED	application_1424182679355_0001	oracle	WordCount	MAPREDUCE	root.oracle	Tue, 17 Feb 2015 19:15:21	Tue, 17 Feb 2015 19:16:12 GMT



## **Practices for Lesson 11: Overview of Hive and Pig**

### **Chapter 11**

## Practices for Lesson 11: Overview of Hive and Pig

---

### Practices Overview

Hadoop is mainly written in Java. Therefore, Java is the language of MapReduce. However, the Hadoop community understands that Java is not always the quickest or most natural way to describe data processing. Because of that, the ecosystem has evolved to support a wide variety of APIs and secondary languages. From scripting languages to SQL, the Hadoop ecosystem allows developers to express their data processing jobs in the language they deem most suitable. Hive and Pig are a pair of these secondary languages for interacting with data stored in HDFS.

- Hive is a data warehousing system that exposes a SQL-like language called HiveQL.
- Pig is an analysis platform that provides a dataflow language called Pig Latin.

In this practice, you will learn the basics of both the Hive and Pig languages.

## Practice 11-1: Manipulating Data by Using Hive

### Overview

In this practice you create a database to store your hive tables and then create a simple external table in Hive that allows you to view the contents of the file.

You also create a more sophisticated external table that parses the JSON fields and maps them to columns in the table. Next, you select the minimum and maximum time periods from the table by using HiveQL.

### Assumptions

None

### Tasks

1. Open a new terminal window.
2. Reset the hive environment used in this practice to make sure that the tables and the database that you will create in the next step does not already exist. Navigate to the /home/oracle/movie/moviework/reset directory, and then run the reset\_mapreduce.sh script. The commands in the reset\_reset.sh and reset\_hive.sql reset the hive environment used in this practice. If your tables and database do not exist, and you get error messages, ignore those messages. If you also get the following message: "/user/oracle/my\_stuff No such file or directory," ignore that message.

```
cd /home/oracle/movie/moviework/reset  
./reset_mapreduce.sh
```

3. Copy the /home/oracle/movie/moviework/mapreduce/movieapp\_3months.avro local file to the /user/oracle/moviework/applog\_avro HDFS directory. You will need this file in this practice. Enter the following command at the command prompt, and then press **Enter**.

```
hadoop fs -put  
/home/oracle/movie/moviework/mapreduce/movieapp_3months.avro  
/user/oracle/moviework/applog_avro
```

4. Enter **hive** at the command prompt to display the Hive command line.

```
$ hive
```

5. Create a new Hive database named moviework. Ensure that the database has been successfully created:

```
hive> create database moviework;  
hive> show databases;
```

- To create a table in a database, you can either fully qualify the table name, prefix the database name to the name of the table, or you can designate that you want all DDL and DML operations to apply to a specific database. For simplicity in this practice, apply the subsequent operations to the moviework database by using the `use` command followed by the name of the database:

```
hive> use moviework;
```

- Open a new terminal window. Review the schema for the Avro file. This schema definition has already been saved in HDFS in the `/user/oracle/moviework/schemas/` directory. Enter the following command:

```
$ hadoop fs -cat moviework/schemas/activity.avsc
```

**Note:** The schema contains the field names, data types, and default values for each of the fields.

- Return to the first window where you were running the Hive commands.
- Create a Hive external table. You do not need to specify the column names or data types when defining the table. The Avro serializer-deserializer (or SERDE) will parse the schema definition to determine these values.

```
hive> CREATE EXTERNAL TABLE movieapp_log_avro ROW FORMAT SERDE
      'org.apache.hadoop.hive.serde2.avro.AvroSerDe' WITH
      SERDEPROPERTIES
      ('avro.schema.url'='hdfs://bigdatalite.localdomain/user/oracle/m
      ovework/schemas/activity.avsc') STORED AS INPUTFORMAT
      'org.apache.hadoop.hive.ql.io.avro.AvroContainerInputFormat'
      OUTPUTFORMAT
      'org.apache.hadoop.hive.ql.io.avro.AvroContainerOutputFormat'
      LOCATION '/user/oracle/moviework/applog_avro';
```

- After the table is created, review the results by selecting the first 20 rows from the newly created `movieapp_log_avro` table. Use the `LIMIT` clause to limit the results to the first 20 rows.

```
hive> SELECT * FROM movieapp_log_avro LIMIT 20;
```

- HiveQL supports many standard SQL operations. Find the minimum and maximum time periods that are available in the log file by using the `min` and `max` functions.

```
hive> SELECT MIN(time), MAX(time) FROM movieapp_log_avro;
```

## Solution 11-1: Manipulating Data by Using Hive

### Overview

In this practice, you create a database to store your Hive tables and then create a simple external table in Hive that allows you to view the contents of the file.

You also create a more sophisticated external table that parses the JSON fields and maps them to the columns in the table. Next, you select the minimum and maximum time periods from the table using HiveQL.

### Assumptions

None

### Tasks

1. Open a new terminal window.



2. Reset the Hive environment used in this practice to make sure that the tables and the database that you create in the next step do not already exist. Navigate to the /home/oracle/movie/moviework/reset directory, and then run the reset\_mapreduce.sh script. The commands in the reset\_reset.sh and reset\_hive.sql reset the Hive environment used in this practice. If your tables and database do not exist and you get error messages, ignore those messages. If you also get the following message: "/user/oracle/my\_stuff No such file or directory," ignore that message.

```
cd /home/oracle/movie/moviework/reset  
./reset_mapreduce.sh
```



```
[oracle@bigdatalite reset]$ pwd
/home/oracle/moviework/reset
[oracle@bigdatalite reset]$ ls -l
total 20
-rw-r--r--. 1 oracle oinstall 1044 May  3  2014 derby.log
-rwxrwxrwx. 1 oracle oinstall 1086 Apr 28  2014 reset_conn.sh
-rw-r--r--. 1 oracle oinstall 136 Jan 12  2014 reset_hive.sql
-rwxrwxrwx. 1 oracle oinstall 213 Jan 25  2014 reset_mapreduce.sh
drwxr-xr-x. 4 oracle oinstall 4096 May  3  2014 TempStatsStore
[oracle@bigdatalite reset]$ ./reset_mapreduce.sh
Dropping files
rm: `/user/oracle/my_stuff': No such file or directory
15/02/17 18:36:16 INFO fs.TrashPolicyDefault: Namenode trash configuration: Deletion interval = 0 minutes, Emttier interval = 0 minutes.
Deleted /user/oracle/moviework/applog_avro/movieapp_3months.avro
15/02/17 18:36:16 INFO fs.TrashPolicyDefault: Namenode trash configuration: Deletion interval = 0 minutes, Emttier interval = 0 minutes.
Deleted /user/oracle/moviework/applog_avro/streamed-movieapp.1424093245691.avro
15/02/17 18:36:16 INFO fs.TrashPolicyDefault: Namenode trash configuration: Deletion interval = 0 minutes, Emttier interval = 0 minutes.
Deleted /user/oracle/moviework/applog_avro/streamed-movieapp.1424093631239.avro
15/02/17 18:36:16 INFO fs.TrashPolicyDefault: Namenode trash configuration: Deletion interval = 0 minutes, Emttier interval = 0 minutes.
Deleted /user/oracle/moviework/applog_avro/streamed-movieapp.1424093706803.avro
15/02/17 18:36:16 INFO fs.TrashPolicyDefault: Namenode trash configuration: Deletion interval = 0 minutes, Emttier interval = 0 minutes.
Deleted /user/oracle/moviework/applog_avro/streamed-movieapp.1424093797158.avro
Dropping Hive database moviework
15/02/17 18:36:18 INFO Configuration.deprecation: mapred.input.dir.recursive is deprecated. Instead, use mapreduce.input.fileinputformat.input.dir.recursive
15/02/17 18:36:18 INFO Configuration.deprecation: mapred.max.split.size is deprecated. Instead, use mapreduce.input.fileinputformat.split.maxsize
15/02/17 18:36:18 INFO Configuration.deprecation: mapred.min.split.size is deprecated. Instead, use mapreduce.input.fileinputformat.split.minsize
15/02/17 18:36:18 INFO Configuration.deprecation: mapred.min.split.size.per.rack is deprecated. Instead, use mapreduce.input.fileinputformat.split.minsize.per.rack
```

```
15/02/17 18:36:18 INFO Configuration.deprecation: mapred.min.split.size.per.node is deprecated. Instead, use mapreduce.input.fileinputformat.split.minsize.per.node
15/02/17 18:36:18 INFO Configuration.deprecation: mapred.reduce.tasks is deprecated. Instead, use mapreduce.job.reduces
15/02/17 18:36:18 INFO Configuration.deprecation: mapred.reduce.tasks.speculative.execution is deprecated. Instead, use mapreduce.reduce.speculative
15/02/17 18:36:19 WARN conf.HiveConf: DEPRECATED: Configuration property hive.metastore.local no longer has any effect. Make sure to provide a valid value for hive.metastore.uris if you are connecting to a remote metastore.

Logging initialized using configuration in jar:file:/usr/lib/hive/lib/hive-common-0.12.0-cdh5.1.2.jar!/hive-log4j.properties
OK
Time taken: 2.987 seconds
OK
```

3. Copy the

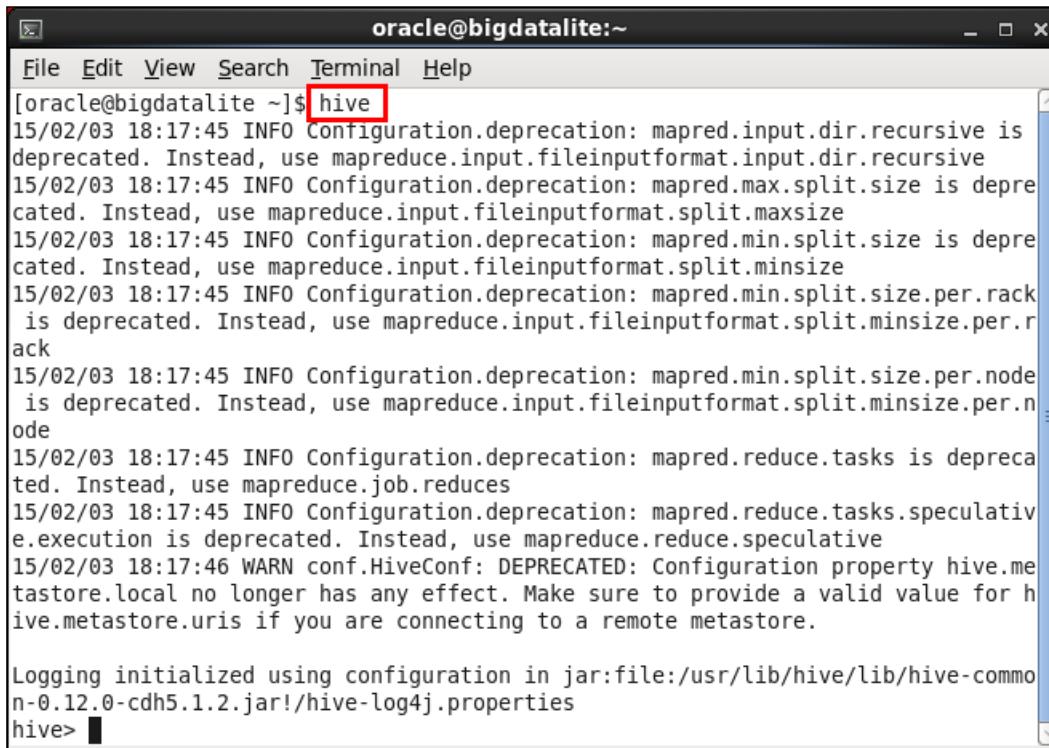
/home/oracle/movie/moviework/mapreduce/movieapp\_3months.avro local file to the /user/oracle/movework/applog\_avro HDFS directory. You will need this file in this practice. Enter the following command at the command prompt, and then press **Enter**.

```
hadoop fs -put  
/home/oracle/movie/movework/mapreduce/movieapp_3months.avro  
/user/oracle/movework/applog_avro
```

```
[oracle@bigdatalite reset]$ hadoop fs -put /home/oracle/movie/movework/mapreduc  
e/movieapp_3months.avro /user/oracle/movework/applog_avro  
[oracle@bigdatalite reset]$ hadoop fs -ls /user/oracle/movework/applog_avro  
Found 1 items  
-rw-r--r-- 1 oracle oracle 19242668 2015-02-18 09:16 /user/oracle/movework/  
applog avro/movieapp_3months.avro
```

4. Enter **hive** at the command prompt to display the Hive command line.

```
hive
```



The screenshot shows a terminal window titled "oracle@bigdatalite:~". The user has typed "hive" into the command line. The terminal displays several informational messages from the Hive configuration, including deprecation warnings for various parameters like "mapred.input.dir.recursive", "mapred.max.split.size", and "mapred.min.split.size". It also shows a warning about the "metastore.local" property. Finally, it logs the initialization of the Hive configuration from the jar file "/usr/lib/hive/lib/hive-common-0.12.0-cdh5.1.2.jar!/hive-log4j.properties".

```
[oracle@bigdatalite ~]$ hive  
15/02/03 18:17:45 INFO Configuration.deprecation: mapred.input.dir.recursive is  
deprecated. Instead, use mapreduce.input.fileinputformat.input.dir.recursive  
15/02/03 18:17:45 INFO Configuration.deprecation: mapred.max.split.size is depre  
cated. Instead, use mapreduce.input.fileinputformat.split.maxsize  
15/02/03 18:17:45 INFO Configuration.deprecation: mapred.min.split.size is depre  
cated. Instead, use mapreduce.input.fileinputformat.split.minsize  
15/02/03 18:17:45 INFO Configuration.deprecation: mapred.min.split.size.per.rack  
is deprecated. Instead, use mapreduce.input.fileinputformat.split.minsize.per.r  
ack  
15/02/03 18:17:45 INFO Configuration.deprecation: mapred.min.split.size.per.node  
is deprecated. Instead, use mapreduce.input.fileinputformat.split.minsize.per.n  
ode  
15/02/03 18:17:45 INFO Configuration.deprecation: mapred.reduce.tasks is depre  
cated. Instead, use mapreduce.job.reduces  
15/02/03 18:17:45 INFO Configuration.deprecation: mapred.reduce.tasks.speculativ  
e.execution is deprecated. Instead, use mapreduce.reduce.speculative  
15/02/03 18:17:46 WARN conf.HiveConf: DEPRECATED: Configuration property hive.me  
tastore.local no longer has any effect. Make sure to provide a valid value for h  
ive.metastore.uris if you are connecting to a remote metastore.  
  
Logging initialized using configuration in jar:file:/usr/lib/hive/lib/hive-commo  
n-0.12.0-cdh5.1.2.jar!/hive-log4j.properties  
hive> ■
```

5. Create a new Hive database named moviework. Ensure that the database has been successfully created:

```
hive> create database moviework;  
hive> show databases;
```

```
hive> create database moviework;  
OK  
Time taken: 0.15 seconds  
hive> show databases;  
OK  
default  
moviedemo  
moviework  
Time taken: 0.087 seconds, Fetched: 3 row(s)  
hive> ■
```

6. To create a table in a database, you can either fully qualify the table name, prefix the database name to the name of the table, or you can designate that you want all DDL and DML operations to apply to a specific database. For simplicity in this practice, apply the subsequent operations to the moviework database by using the use command followed by the name of the database:

```
hive> use moviework;
```

```
hive> use moviework;  
OK  
Time taken: 0.022 seconds  
hive> ■
```

7. Open a new terminal window. Review the schema for the Avro file. This schema definition has already been saved in HDFS in the /user/oracle/moviework/schemas/ directory. Enter the following command:

```
hadoop fs -cat moviework/schemas/activity.avsc
```



**Note:** The schema contains the field names, data types, and default values for each of the fields.

```
[oracle@bigdatalite ~]$ hadoop fs -cat /user/oracle/ moviework/schemas/activity.avsc
cat: `/user/oracle': Is a directory
{
  "type" : "record",
  "name" : "Activity",
  "namespace" : "oracle.avro",
  "fields" : [ {
    "name" : "custId",
    "type" : ["null","int"],
    "default" : null
  }, {
    "name" : "movieId",
    "type" : ["null","int"],
    "default" : null
  }, {
    "name" : "activity",
    "type" : ["null","int"],
    "default" : null
  }, {
    "name" : "genreId",
    "type" : ["null","int"],
    "default" : null
  }, {
    "name" : "recommended",
    "type" : ["null","string"],
    "default" : null
  }, {
    "name" : "time",
    "type" : ["null","string"],
    "default" : null
  }, {
    "name" : "rating",
    "type" : ["null","int"],
    "default" : null
  }, {
    "name" : "price",
    "type" : ["null","double"],
    "default" : null
  }, {
    "name" : "position",
    "type" : ["null","int"],
    "default" : null
  } ]
}
```

8. Return to the first window where you were running the Hive commands.

```
hive> show databases;
OK
default
moviedemo
moviework
Time taken: 0.498 seconds, Fetched: 3 row(s)
hive> use moviework;
OK
Time taken: 0.043 seconds
hive> ■
```

9. Create a Hive external table. You do not need to specify the column names or data types when defining the table. The Avro serializer-deserializer (or SERDE) will parse the schema definition to determine these values.

```
hive> CREATE EXTERNAL TABLE movieapp_log_avro ROW FORMAT SERDE  
'org.apache.hadoop.hive.serde2.avro.AvroSerDe' WITH  
SERDEPROPERTIES  
( 'avro.schema.url'='hdfs://bigdatalite.localdomain/user/oracle/m  
oviework/schemas/activity.avsc') STORED AS INPUTFORMAT  
'org.apache.hadoop.hive.ql.io.avro.AvroContainerInputFormat'  
OUTPUTFORMAT  
'org.apache.hadoop.hive.ql.io.avro.AvroContainerOutputFormat'  
LOCATION '/user/oracle/moviework/applog_avro';
```

```
hive> CREATE EXTERNAL TABLE movieapp_log_avro ROW FORMAT SERDE 'org.apache.hadoo  
p.hive.serde2.avro.AvroSerDe' WITH SERDEPROPERTIES ('avro.schema.url'='hdfs://bi  
gdatalite.localdomain/user/oracle/moviework/schemas/activity.avsc') STORED AS IN  
PUTFORMAT 'org.apache.hadoop.hive.ql.io.avro.AvroContainerInputFormat' OUTPUTFOR  
MAT 'org.apache.hadoop.hive.ql.io.avro.AvroContainerOutputFormat' LOCATION '/use  
r/oracle/moviework/applog_avro';  
OK  
Time taken: 0.917 seconds  
hive>
```

10. After the table is created, review the results by selecting the first 20 rows from the newly created `movieapp_log_avro` table. Use the `LIMIT` clause to limit the results to the first 20 rows.

```
hive> SELECT * FROM movieapp_log_avro LIMIT 20;
```

**Notice that no MapReduce job is created when you select everything in the table instead of selecting specific columns, and also when you do not have a where clause. You are scanning the entire file. Contrast this with the select query in the next step.**

hive> <b>SELECT * FROM movieapp_log_avro LIMIT 20;</b>								
OK								
1185972	0	8	0	null	2012-07-01:00:00:07	NULL	NULL	N
ULL								
1354924	1948	7	9	N	2012-07-01:00:00:22	NULL	NULL	N
ULL								
1083711	0	9	0	null	2012-07-01:00:00:26	NULL	NULL	N
ULL								
1234182	11547	7	6	Y	2012-07-01:00:00:32	NULL	NULL	N
ULL								
1010220	11547	6	6	Y	2012-07-01:00:00:42	NULL	NULL	N
ULL								
1143971	0	8	0	null	2012-07-01:00:00:43	NULL	NULL	N
ULL								
1253676	0	9	0	null	2012-07-01:00:00:50	NULL	NULL	N
ULL								
1351777	608	7	6	N	2012-07-01:00:01:03	NULL	NULL	N
ULL								
1143971	0	9	0	null	2012-07-01:00:01:07	NULL	NULL	N
ULL								
1363545	27205	7	9	Y	2012-07-01:00:01:18	NULL	NULL	N
ULL								
1067283	1124	7	9	Y	2012-07-01:00:01:26	NULL	NULL	N
ULL								
1126174	16309	7	46	N	2012-07-01:00:01:35	NULL	NULL	N
ULL								
1234182	11547	7	6	Y	2012-07-01:00:01:39	NULL	NULL	N
ULL								
1067283	0	9	0	null	2012-07-01:00:01:55	NULL	NULL	N
ULL								
1377537	0	9	0	null	2012-07-01:00:01:58	NULL	NULL	N
ULL								
1347836	0	8	0	null	2012-07-01:00:02:03	NULL	NULL	N
ULL								
1137285	0	8	0	null	2012-07-01:00:03:39	NULL	NULL	N
ULL								
1354924	0	9	0	null	2012-07-01:00:03:51	NULL	NULL	N
ULL								
1036191	0	8	0	null	2012-07-01:00:03:55	NULL	NULL	N
ULL								
1363545	27205	5	9	Y	2012-07-01:00:04:03	NULL	NULL	N
ULL								
Time taken: 0.57 seconds, Fetched: 20 row(s)								
hive> █								

11. HiveQL supports many standard SQL operations. Find the minimum and maximum time periods that are available in the log file by using the `min` and `max` functions.

```
hive> SELECT MIN(time), MAX(time) FROM movieapp_log_avro;
```

```
hive> SELECT MIN(time), MAX(time) FROM movieapp_log_avro;
Total MapReduce jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapred.reduce.tasks=<number>
Starting Job = job_1424182679355_0008, Tracking URL = http://bigdatalite.localdo
main:8088/proxy/application_1424182679355_0008/
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1424182679355_0008
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2015-02-18 09:19:39,399 Stage-1 map = 0%,  reduce = 0%
2015-02-18 09:19:48,226 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 5.03 se
c
2015-02-18 09:19:49,272 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 5.03 se
c
2015-02-18 09:19:50,303 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 5.03 se
c
2015-02-18 09:19:51,357 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 5.03 se
c
2015-02-18 09:19:52,394 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 5.03 se
c
2015-02-18 09:19:53,427 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 5.03 se
c
2015-02-18 09:19:54,492 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 5.03 se
c
2015-02-18 09:19:55,533 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 6.62
sec
2015-02-18 09:19:56,574 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 6.62
sec
MapReduce Total cumulative CPU time: 6 seconds 620 msec
Ended Job = job_1424182679355_0008
MapReduce Jobs Launched:
Job 0: Map: 1  Reduce: 1  Cumulative CPU: 6.62 sec  HDFS Read: 19254455 HDFS W
rite: 40 SUCCESS
Total MapReduce CPU Time Spent: 6 seconds 620 msec
OK
2012-07-01:00:00:07    2012-10-01:03:19:24
Time taken: 27.523 seconds, Fetched: 1 row(s)
hive> █
```

**When the MapReduce job is completed successfully, you can check on the details of this job by using either the YARN or JobHistory services, using the bookmarks on your web browser toolbar.**

**Note:** The number of jobs displayed in YARN or JobHistory might be different than your results.

The screenshot shows the Hadoop JobHistory interface. At the top, there's a navigation bar with tabs for JobHistory, Most Visited, Hue, MoviePlayer, Hadoop (which is selected and highlighted with a red box), Solr Admin, SQL Pattern Matching, VirtualBox VMs for ..., Cloudera Manager, NameNode bigdatalite.localdomain:8020, and YARN Applications. Below the navigation bar is a logo for 'hadoop' and a search bar containing 'JobHistory'. A link 'Open All in Tabs' is also present. The main area is titled 'Retired Jobs' and contains a table with columns: Submit Time, Start Time, Finish Time, Job ID, Name, User, Queue, and State. One row in the table is highlighted with a red box, corresponding to the job shown in the detailed view below.

Submit Time	Start Time	Finish Time	Job ID	Name	User	Queue	State
2015.02.18 09:19:31 EST	2015.02.18 09:19:38 EST	2015.02.18 09:19:54 EST	job_1424182679355_0008	SELECT MIN(time), MAX(ti...movieapp_log_avro(Stage	oracle	root.oracle	SUCCEEDED

This screenshot shows the detailed view of a MapReduce job. On the left, a sidebar menu includes Application, Job (with Overview, Counters, Configuration, Map tasks, Reduce tasks), and Tools. The main content area is titled 'MapReduce Job job\_1424182679355\_0008'. It displays the following details:

- Job Name:** SELECT MIN(time), MAX(ti...movieapp\_log\_avro(Stage-1)
- User Name:** oracle
- Queue:** root.oracle
- State:** SUCCEEDED
- Uberized:** false
- Submitted:** Wed Feb 18 09:19:31 EST 2015
- Started:** Wed Feb 18 09:19:38 EST 2015
- Finished:** Wed Feb 18 09:19:54 EST 2015
- Elapsed:** 16sec
- Diagnostics:**
  - Average Map Time: 7sec
  - Average Reduce Time: 1sec
  - Average Shuffle Time: 3sec
  - Average Merge Time: 0sec

Below this, there's a table for the ApplicationMaster:

Attempt Number	Start Time	Node
1	Wed Feb 18 09:19:34 EST 2015	bigdatalite.localdomain:8042

## Practice 11-2: Extracting Facts by Using Hive

### Overview

Hive allows for the manipulation of data in HDFS by using a variant of SQL. This makes it an excellent choice for transforming and consolidating data for loading into a relational database. In this practice, you will use HiveQL to filter and aggregate click data to build facts about users' movie preferences.

The query results will be saved in a staging table used to populate the Oracle Database.

### Tasks

1. The `movieapp_log_avro` table contains an activity column. Activity states are as follows:

1. RATE\_MOVIE
2. COMPLETED\_MOVIE
3. PAUSE\_MOVIE
4. START\_MOVIE
5. BROWSE\_MOVIE
6. LIST\_MOVIE
7. SEARCH\_MOVIE
8. LOGIN
9. LOGOUT
10. INCOMPLETE\_MOVIE
11. PURCHASE\_MOVIE

Hive maps queries into MapReduce jobs, simplifying the process of querying large data sets in HDFS. HiveQL statements can be mapped to phases of the MapReduce framework. As illustrated in the following figure, selection and transformation operations occur in map tasks, while aggregation is handled by reducers. Join operations are flexible: they can be performed in the reducer or mappers depending on the size of the left-most table

2. Write a query to select only those clicks that correspond to `START_MOVIE` (4), `BROWSE_MOVIE` (5), `COMPLETED_MOVIE` (2), or `PURCHASE_MOVIE` (11). Use a `CASE` statement to transform the `RECOMMENDED` column into integers where Y is 1 and N is 0. Also, ensure `GENREID` is not null. Include only the first 25 rows:

```
hive> SELECT custid, movieid,
CASE WHEN genreid > 0 THEN genreid ELSE -1 END genreid,
time,
CASE recommended WHEN 'Y' THEN 1 ELSE 0 END recommended,
activity,
price
FROM movieapp_log_avro
WHERE activity IN (2,4,5,11) LIMIT 25;
```

3. Select the movie ratings made by a user. Consider the following: what if a user rates the same movie multiple times? In this scenario, you should load only the user's most recent movie rating. In Oracle Database 12c, you can use a windowing function. However, HiveQL does not provide sophisticated analytic functions. Instead, you must use an inner join to compute the result. Run the following query to select the custid, movieid, genreid, time, recommended state, activity, and most recent rating for each movie.

```
hive> SELECT
  m1.custid, m1.movieid,
CASE WHEN m1.genreid > 0 THEN m1.genreid ELSE -1 END genreid,
m1.time,
CASE m1.recommended WHEN 'Y' THEN 1 ELSE 0 END recommended,
m1.activity,
m1.rating
FROM
movieapp_log_avro m1
JOIN
(SELECT
custid,
movieid,
CASE WHEN genreid > 0 THEN genreid ELSE -1 END genreid,
MAX(time) max_time,
activity
FROM
movieapp_log_avro
GROUP BY custid, movieid, genreid, activity
)
m2
ON
(
m1.custid=m2.custid
AND
m1.movieid=m2.movieid
AND
m1.genreid=m2.genreid
AND
m1.time=m2.max_time
AND
m1.activity=1
AND
m2.activity=1
)
LIMIT 25;
```

4. Run the following code to create the `movieapp_log_stage` staging table. You will load the results of the previous two queries into this staging table.

```
hive> CREATE TABLE  
movieapp_log_stage  
(  
custId INT,  
movieId INT,  
genreId INT,  
time STRING,  
recommended INT,  
activity INT,  
rating INT,  
sales FLOAT  
)  
ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t';
```

5. Load the results of the previous two queries into the `movieapp_log_stage` table by using the following code:

```
hive> INSERT OVERWRITE TABLE movieapp_log_stage  
SELECT * FROM (  
SELECT custid, movieid,  
CASE WHEN genreid > 0 THEN genreid ELSE -1 END genreid,  
time,  
CAST((CASE recommended WHEN 'Y' THEN 1 ELSE 0 END) AS INT)  
recommended,  
activity,  
cast(null AS INT) rating,  
price  
FROM movieapp_log_avro  
WHERE activity IN (2,4,5,11)  
UNION ALL  
SELECT m1.custid, m1.movieid,  
CASE WHEN m1.genreid > 0 THEN m1.genreid ELSE -1 END genreid,  
m1.time,  
CAST((CASE m1.recommended WHEN 'Y' THEN 1 ELSE 0 END) AS  
INT) recommended,  
m1.activity,  
m1.rating,  
cast(null as float) price  
FROM movieapp_log_avro m1  
JOIN  
(SELECT custid, movieid,
```

```
CASE WHEN genreid > 0 THEN genreid ELSE -1 END genreid,
MAX(time) max_time,
activity
FROM movieapp_log_avro
GROUP BY custid, movieid, genreid, activity ) m2
ON
( m1.custid = m2.custid
AND m1.movieid = m2.movieid
AND m1.genreid = m2.genreid
AND m1.time = m2.max_time
AND m1.activity = 1
AND m2.activity = 1 )
) union_result;
```

## Solution 11-2: Extracting Facts by Using Hive

### Overview

In this practice, you use HiveQL to filter and aggregate click data to build facts about users' movie preferences. The query results will be saved in a staging table used to populate the Oracle Database.

### Tasks

1. The `movieapp_log_avro` table contains an activity column. Activity states are as follows:

1. RATE\_MOVIE
2. COMPLETED\_MOVIE
3. PAUSE\_MOVIE
4. START\_MOVIE
5. BROWSE\_MOVIE
6. LIST\_MOVIE
7. SEARCH\_MOVIE
8. LOGIN
9. LOGOUT
10. INCOMPLETE\_MOVIE
11. PURCHASE\_MOVIE

Hive maps queries into MapReduce jobs, simplifying the process of querying large data sets in HDFS. HiveQL statements can be mapped to phases of the MapReduce framework. As illustrated in the following figure, selection and transformation operations occur in map tasks, while aggregation is handled by reducers. Join operations are flexible: they can be performed in the reducer or mappers depending on the size of the left-most table.

2. Write a query to select only those clicks that correspond to `START_MOVIE` (4), `BROWSE_MOVIE` (5), `COMPLETED_MOVIE` (2), or `PURCHASE_MOVIE` (11). Use a `CASE` statement to transform the `RECOMMENDED` column into integers where Y is 1 and N is 0. Also, ensure `GENREID` is not null. Include only the first 25 rows:

```
hive> SELECT custid, movieid,
CASE WHEN genreid > 0 THEN genreid ELSE -1 END genreid,
time,
CASE recommended WHEN 'Y' THEN 1 ELSE 0 END recommended,
activity,
price
FROM movieapp_log_avro
WHERE activity IN (2,4,5,11) LIMIT 25;
```

```

hive> SELECT custid, movieid,
    > CASE WHEN genreid > 0 THEN genreid ELSE -1 END genreid,
    > time,
    > CASE recommended WHEN 'Y' THEN 1 ELSE 0 END recommended,
    > activity,
    > price
    > FROM movieapp_log_avro
    > WHERE activity IN (2,4,5,11) LIMIT 25;
Total MapReduce jobs = 1
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1422550693204_0010, Tracking URL = http://bigdatalite.localdomain:8088/pro
y/application_1422550693204_0010/
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1422550693204_0010
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2015-02-04 18:16:37,500 Stage-1 map = 0%,  reduce = 0%
2015-02-04 18:16:44,802 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.93 sec
2015-02-04 18:16:45,831 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.93 sec
MapReduce Total cumulative CPU time: 1 seconds 930 msec
Ended Job = job_1422550693204_0010
MapReduce Jobs Launched:
Job 0: Map: 1  Cumulative CPU: 1.93 sec   HDFS Read: 28844 HDFS Write: 1067 SUCCESS
Total MapReduce CPU Time Spent: 1 seconds 930 msec
OK

```

OK						
1363545	27205	9	2012-07-01:00:04:03	1	5	NULL
1346299	424	18	2012-07-01:00:05:02	1	4	NULL
1126174	16309	46	2012-07-01:00:05:45	0	5	NULL
1354924	1948	9	2012-07-01:00:07:21	0	11	1.99
1126174	275	8	2012-07-01:00:12:40	0	5	NULL
1363545	7211	6	2012-07-01:00:13:47	0	5	NULL
1036191	11450	3	2012-07-01:00:16:04	0	5	NULL
1363545	11393	30	2012-07-01:00:18:44	0	5	NULL
1126174	1647	3	2012-07-01:00:20:43	0	4	NULL
1129727	14	24	2012-07-01:00:28:30	1	5	NULL
1036191	9346	6	2012-07-01:00:28:44	1	5	NULL
1129727	500	3	2012-07-01:00:30:38	1	5	NULL
1363545	27205	9	2012-07-01:00:35:09	1	11	3.99
1036191	1149812	17	2012-07-01:00:37:35	0	4	NULL
1152235	9346	6	2012-07-01:00:39:49	1	2	NULL
1103597	22954	30	2012-07-01:00:44:02	1	5	NULL
1144051	768	9	2012-07-01:00:46:53	0	5	NULL
1152235	9346	3	2012-07-01:00:47:05	1	5	NULL
1047082	77	20	2012-07-01:00:47:32	1	2	NULL
1152235	6977	3	2012-07-01:00:48:36	1	5	NULL
1258710	1624	6	2012-07-01:00:52:04	0	2	NULL
1129727	36586	9	2012-07-01:00:53:22	0	5	NULL
1085645	11547	6	2012-07-01:00:53:25	1	5	NULL
1138832	9522	6	2012-07-01:00:55:30	0	5	NULL
1144051	1954	45	2012-07-01:00:56:55	0	5	NULL

Time taken: 16.517 seconds, Fetched: 25 row(s)

3. Select the movie ratings made by a user. Consider the following: what if a user rates the same movie multiple times? In this scenario, you should only load the user's most recent movie rating. In Oracle Database 12c, you can use a windowing function. However, HiveQL does not provide sophisticated analytic functions. Instead, you must use an inner join to compute the result. Run the following query to select the custid, movieid, genreid, time, recommended state, activity, and most recent rating for each movie.

```
hive> SELECT
m1.custid,
m1.movieid,
CASE WHEN m1.genreid > 0 THEN m1.genreid ELSE -1 END genreid,
m1.time,
CASE m1.recommended WHEN 'Y' THEN 1 ELSE 0 END recommended,
m1.activity,
m1.rating
FROM movieapp_log_avro m1
JOIN
(SELECT
custid,
movieid,
CASE WHEN genreid > 0 THEN genreid ELSE -1 END genreid,
MAX(time) max_time,
activity
FROM movieapp_log_avro
GROUP BY custid,
movieid,
genreid,
activity
) m2
ON (
m1.custid = m2.custid
AND m1.movieid = m2.movieid
AND CASE WHEN m1.genreid > 0 THEN m1.genreid ELSE -1 END = CASE
WHEN m2.genreid > 0 THEN m2.genreid ELSE -1 END
AND m1.time = m2.max_time
AND m1.activity = 1
AND m2.activity = 1
) LIMIT 25;
```

```
hive> SELECT
> m1.custid,
> m1.movieid,
> CASE WHEN m1.genreid > 0 THEN m1.genreid ELSE -1 END genreid,
> m1.time,
> CASE m1.recommended WHEN 'Y' THEN 1 ELSE 0 END recommended,
> m1.activity,
> m1.rating
> FROM movieapp_log_avro m1
> JOIN
> (SELECT
> custid,
> movieid,
> CASE WHEN genreid > 0 THEN genreid ELSE -1 END genreid,
> MAX(time) max_time,
> activity
> FROM movieapp_log_avro
> GROUP BY custid,
> movieid,
> genreid,
> activity
> ) m2
> ON (
> m1.custid = m2.custid
> AND m1.movieid = m2.movieid
> AND CASE WHEN m1.genreid > 0 THEN m1.genreid ELSE -1 END = CASE WHEN m2.genreid > 0 THE
N m2.genreid ELSE -1 END
> AND m1.time = m2.max_time
> AND m1.activity = 1
> AND m2.activity = 1
> ) LIMIT 25;
```

```
Total MapReduce jobs = 4
Launching Job 1 out of 4
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapred.reduce.tasks=<number>
Starting Job = job_1422550693204_0011, Tracking URL = http://bigdatalite.localdomain:8088/proxy/application_1422550693204_0011/
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1422550693204_0011
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2015-02-04 18:26:05,416 Stage-1 map = 0%,  reduce = 0%
2015-02-04 18:26:15,798 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 5.48 sec
2015-02-04 18:26:16,829 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 5.48 sec
2015-02-04 18:26:17,880 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 5.48 sec
2015-02-04 18:26:18,925 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 5.48 sec
2015-02-04 18:26:19,955 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 5.48 sec
2015-02-04 18:26:21,008 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 5.48 sec
2015-02-04 18:26:22,069 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 5.48 sec
2015-02-04 18:26:23,101 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 8.94 sec
2015-02-04 18:26:24,137 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 8.94 sec
MapReduce Total cumulative CPU time: 8 seconds 940 msec
Ended Job = job_1422550693204_0011
Stage-7 is selected by condition resolver.
Stage-8 is filtered out by condition resolver.
Stage-2 is filtered out by condition resolver.
15/02/04 18:26:26 WARN conf.Configuration: file:/tmp/oracle/hive_2015-02-04_18-25-57_464_6780
829651312582678-1/-local-10010/jobconf.xml:an attempt to override final parameter: mapreduce.
job.end-notification.max.retry.interval; Ignoring.
15/02/04 18:26:26 WARN conf.Configuration: file:/tmp/oracle/hive_2015-02-04_18-25-57_464_6780
```

```
15/02/04 18:26:26 WARN conf.Configuration: file:/tmp/oracle/hive_2015-02-04_18-25-57_464_6780^
829651312582678-1/-local-10010/jobconf.xml:an attempt to override final parameter: mapreduce.
job.end-notification.max.attempts; Ignoring.
15/02/04 18:26:27 INFO Configuration.deprecation: mapred.input.dir.recursive is deprecated. I
nstead, use mapreduce.input.fileinputformat.input.dir.recursive
15/02/04 18:26:27 INFO Configuration.deprecation: mapred.max.split.size is deprecated. Instea
d, use mapreduce.input.fileinputformat.split.maxsize
15/02/04 18:26:27 INFO Configuration.deprecation: mapred.min.split.size is deprecated. Instea
d, use mapreduce.input.fileinputformat.split.minsize
15/02/04 18:26:27 INFO Configuration.deprecation: mapred.min.split.size.per.rack is depreca
d. Instead, use mapreduce.input.fileinputformat.split.minsize.per.rack
15/02/04 18:26:27 INFO Configuration.deprecation: mapred.min.split.size.per.node is depreca
d. Instead, use mapreduce.input.fileinputformat.split.minsize.per.node
15/02/04 18:26:27 INFO Configuration.deprecation: mapred.reduce.tasks is deprecated. Instead,
use mapreduce.job.reduces
15/02/04 18:26:27 INFO Configuration.deprecation: mapred.reduce.tasks.speculative.execution i
s deprecated. Instead, use mapreduce.reduce.speculative
15/02/04 18:26:27 WARN conf.HiveConf: DEPRECATED: Configuration property hive.metastore.local
no longer has any effect. Make sure to provide a valid value for hive.metastore.uris if you
are connecting to a remote metastore.
Execution log at: /tmp/oracle/oracle_20150204182525_4ae1f099-3a42-45c5-8c39-3a215590a447.log
2015-02-04 06:26:27      Starting to launch local task to process map join;      maximum memor
y = 257949696
2015-02-04 06:26:29      Dump the side-table into file: file:/tmp/oracle/hive_2015-02-04_18-25
-57_464_6780829651312582678-1/-local-10003/HashTable-Stage-4/MapJoin-mapfile01--.hashtable
2015-02-04 06:26:29      Upload 1 File to: file:/tmp/oracle/hive_2015-02-04_18-25-57_464_67808
29651312582678-1/-local-10003/HashTable-Stage-4/MapJoin-mapfile01--.hashtable
2015-02-04 06:26:29      End of local task; Time Taken: 1.832 sec.
Execution completed successfully
MapredLocal task succeeded
Launching Job 3 out of 4
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1422550693204_0012, Tracking URL = http://bigdatalite.localdomain:8088/pro
xy/application_1422550693204_0012/
```

```
Ended Job = job_1422550693204_0012
MapReduce Jobs Launched:
Job 0: Map: 1 Reduce: 1   Cumulative CPU: 8.94 sec   HDFS Read: 19254455 HDFS Write: 274313
SUCCESS
Job 1: Map: 1   Cumulative CPU: 2.68 sec   HDFS Read: 94554 HDFS Write: 1032 SUCCESS
Total MapReduce CPU Time Spent: 11 seconds 620 msec
OK
1126174 1647    9      2012-07-01:00:20:11    0      1      5
1152235 642     19     2012-07-01:00:54:41    1      1      2
1144051 8321    3      2012-07-01:01:11:46    0      1      4
1303830 11547   17     2012-07-01:01:14:57    1      1      3
1138832 134     16     2012-07-01:01:18:15    0      1      3
1075119 272     7      2012-07-01:01:26:39    1      1      4
1161010 15121   25     2012-07-01:01:35:04    1      1      5
1161010 10193   14     2012-07-01:01:44:18    1      1      2
1161861 752     45     2012-07-01:01:55:40    1      1      5
1161010 289     15     2012-07-01:01:57:43    1      1      3
1085964 1128682 20     2012-07-01:02:04:36    0      1      2
1036191 9346    6      2012-07-01:02:22:09    1      1      1
1021684 5       6      2012-07-01:02:34:51    1      1      4
1429963 756     12     2012-07-01:02:52:29    0      1      2
1036191 9346    3      2012-07-01:02:57:57    1      1      3
1085645 2135    7      2012-07-01:03:04:10    1      1      3
1036191 1149812 17     2012-07-01:03:10:21    0      1      1
1303830 242     8      2012-07-01:03:11:37    0      1      5
1307015 9346    3      2012-07-01:03:16:57    1      1      3
1303830 253     3      2012-07-01:03:26:32    0      1      4
1273442 2022    15     2012-07-01:03:46:00    0      1      4
1446850 278     24     2012-07-01:03:54:20    1      1      5
1161010 522     46     2012-07-01:03:56:55    1      1      4
1027738 14      24     2012-07-01:04:10:55    1      1      3
1144051 813     6      2012-07-01:04:37:00    0      1      1
Time taken: 48.75 seconds, Fetched: 25 row(s)
hive> █
```

- Run the following code to create the `movieapp_log_stage` staging table. You will load the results of the previous two queries into this staging table.

```
hive> CREATE TABLE
movieapp_log_stage
(
  custId INT,
  movieId INT,
  genreId INT,
  time STRING,
  recommended INT,
  activity INT,
  rating INT,
  sales FLOAT
)
ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t';
```

```
hive> CREATE TABLE
  > movieapp_log_stage
  > (
  > custId INT,
  > movieId INT,
  > genreId INT,
  > time STRING,
  > recommended INT,
  > activity INT,
  > rating INT,
  > sales FLOAT
  > )
  > ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t';
OK
Time taken: 0.084 seconds
hive> █
```

```
hive> describe movieapp_log_stage;
OK
custid          int           None
movieid         int           None
genreid         int           None
time            string        None
recommended     int           None
activity         int           None
rating           int           None
sales            float         None
Time taken: 0.078 seconds, Fetched: 8 row(s)
hive> █
```

5. Load the results of the previous two queries into the `movieapp_log_stage` table by using the following code:

```
hive> INSERT OVERWRITE TABLE movieapp_log_stage
SELECT * FROM (
SELECT custid,
movieid,
CASE WHEN genreid > 0 THEN genreid ELSE -1 END genreid,
time,
CAST((CASE recommended WHEN 'Y' THEN 1 ELSE 0 END) AS INT)
recommended,
activity,
cast(null AS INT) rating,
price
FROM movieapp_log_avro
WHERE activity IN (2,4,5,11)
UNION ALL
SELECT
m1.custid,
m1.movieid,
```

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

```
CASE WHEN m1.genreid > 0 THEN m1.genreid ELSE -1 END genreid,
m1.time,
CAST((CASE m1.recommended WHEN 'Y' THEN 1 ELSE 0 END) AS INT)
recommended,
m1.activity,
m1.rating,
cast(null as float) price
FROM movieapp_log_avro m1
JOIN
(SELECT
custid,
movieid,
CASE WHEN genreid > 0 THEN genreid ELSE -1 END genreid,
MAX(time) max_time,
activity
FROM movieapp_log_avro
GROUP BY custid,
movieid,
genreid,
activity
) m2
ON (
m1.custid = m2.custid
AND m1.movieid = m2.movieid
AND CASE WHEN m1.genreid > 0 THEN m1.genreid ELSE -1 END = CASE
WHEN m2.genreid > 0 THEN m2.genreid ELSE -1 END
AND m1.time = m2.max_time
AND m1.activity = 1
AND m2.activity = 1
)
) union_result;
```

```
hive> INSERT OVERWRITE TABLE movieapp_log_stage
  > SELECT * FROM (
  >   SELECT custid,
  >   movieid,
  >   CASE WHEN genreid > 0 THEN genreid ELSE -1 END genreid,
  >   time,
  >   CAST((CASE recommended WHEN 'Y' THEN 1 ELSE 0 END) AS INT) recommended,
  >   activity,
  >   cast(null AS INT) rating,
  >   price
  >   FROM movieapp_log_avro
  >   WHERE activity IN (2,4,5,11)
  > UNION ALL
  >   SELECT
  >     m1.custid,
  >     m1.movieid,
  >     CASE WHEN m1.genreid > 0 THEN m1.genreid ELSE -1 END genreid,
  >     m1.time,
  >     CAST((CASE m1.recommended WHEN 'Y' THEN 1 ELSE 0 END) AS INT) recommended,
  >     m1.activity,
  >     m1.rating,
  >     cast(null as float) price
  >     FROM movieapp_log_avro m1
  > JOIN
  >   (SELECT
  >     custid,
  >     movieid,
  >     CASE WHEN genreid > 0 THEN genreid ELSE -1 END genreid,
  >     MAX(time) max_time,
  >     activity
  >     FROM movieapp_log_avro
  >     GROUP BY custid,
  >     movieid,
```

```
> GROUP BY custid,
> movieid,
> genreid,
> activity
> ) m2
> ON (
> m1.custid = m2.custid
> AND m1.movieid = m2.movieid
> AND CASE WHEN m1.genreid > 0 THEN m1.genreid ELSE -1 END = CASE WHEN m2.genreid > 0 THE
N m2.genreid ELSE -1 END
> AND m1.time = m2.max_time
> AND m1.activity = 1
> AND m2.activity = 1
>
> ) union_result;
Total MapReduce jobs = 7
Launching Job 1 out of 7
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapred.reduce.tasks=<number>
Starting Job = job_1422550693204_0013, Tracking URL = http://bigdatalite.localdomain:8088/pro
xy/application_1422550693204_0013/
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1422550693204_0013
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2015-02-04 20:16:31,209 Stage-1 map = 0%,  reduce = 0%
2015-02-04 20:16:40,493 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 5.74 sec
2015-02-04 20:16:41,521 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 5.74 sec
2015-02-04 20:16:42,556 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 5.74 sec
2015-02-04 20:16:43,608 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 5.74 sec
2015-02-04 20:16:44,668 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 5.74 sec
```

```
2015-02-04 20:16:43,608 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 5.74 sec
2015-02-04 20:16:44,668 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 5.74 sec
2015-02-04 20:16:45,703 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 5.74 sec
2015-02-04 20:16:46,764 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 5.74 sec
2015-02-04 20:16:47,808 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 5.74 sec
2015-02-04 20:16:48,841 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 9.19 sec
2015-02-04 20:16:49,875 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 9.19 sec
MapReduce Total cumulative CPU time: 9 seconds 190 msec
Ended Job = job_1422550693204_0013
Stage-14 is selected by condition resolver.
Stage-15 is filtered out by condition resolver.
Stage-2 is filtered out by condition resolver.
15/02/04 20:16:52 WARN conf.Configuration: file:/tmp/oracle/hive_2015-02-04_20-16-22_402_4058
913684699097916-1/-local-10012/jobconf.xml:an attempt to override final parameter: mapreduce.
job.end-notification.max.retry.interval; Ignoring.
15/02/04 20:16:52 WARN conf.Configuration: file:/tmp/oracle/hive_2015-02-04_20-16-22_402_4058
913684699097916-1/-local-10012/jobconf.xml:an attempt to override final parameter: mapreduce.
job.end-notification.max.attempts; Ignoring.
15/02/04 20:16:52 INFO Configuration.deprecation: mapred.input.dir.recursive is deprecated. I
nstead, use mapreduce.input.fileinputformat.input.dir.recursive
15/02/04 20:16:52 INFO Configuration.deprecation: mapred.max.split.size is deprecated. Instea
d, use mapreduce.input.fileinputformat.split.maxsize
15/02/04 20:16:52 INFO Configuration.deprecation: mapred.min.split.size is deprecated. Instea
d, use mapreduce.input.fileinputformat.split.minsize
15/02/04 20:16:52 INFO Configuration.deprecation: mapred.min.split.size.per.rack is deprecate
d. Instead, use mapreduce.input.fileinputformat.split.minsize.per.rack
15/02/04 20:16:52 INFO Configuration.deprecation: mapred.min.split.size.per.node is deprecate
d. Instead, use mapreduce.input.fileinputformat.split.minsize.per.node
15/02/04 20:16:52 INFO Configuration.deprecation: mapred.reduce.tasks is deprecated. Instead,
use mapreduce.job.reduces
15/02/04 20:16:52 INFO Configuration.deprecation: mapred.reduce.tasks.speculative.execution i
s deprecated. Instead, use mapreduce.reduce.speculative
15/02/04 20:16:52 WARN conf.HiveConf: DEPRECATED: Configuration property hive.metastore.local
no longer has any effect. Make sure to provide a valid value for hive.metastore.uris if you
```

```
Stage-2 is filtered out by condition resolver.  
15/02/04 20:16:52 WARN conf.Configuration: file:/tmp/oracle/hive_2015-02-04_20-16-22_402_4058  
913684699097916-1/-local-10012/jobconf.xml:an attempt to override final parameter: mapreduce.  
job.end-notification.max.retry.interval; Ignoring.  
15/02/04 20:16:52 WARN conf.Configuration: file:/tmp/oracle/hive_2015-02-04_20-16-22_402_4058  
913684699097916-1/-local-10012/jobconf.xml:an attempt to override final parameter: mapreduce.  
job.end-notification.max.attempts; Ignoring.  
15/02/04 20:16:52 INFO Configuration.deprecation: mapred.input.dir.recursive is deprecated. I  
nstead, use mapreduce.input.fileinputformat.input.dir.recursive  
15/02/04 20:16:52 INFO Configuration.deprecation: mapred.max.split.size is deprecated. Instea  
d, use mapreduce.input.fileinputformat.split.maxsize  
15/02/04 20:16:52 INFO Configuration.deprecation: mapred.min.split.size is deprecated. Instea  
d, use mapreduce.input.fileinputformat.split.minsize  
15/02/04 20:16:52 INFO Configuration.deprecation: mapred.min.split.size.per.rack is deprecate  
d. Instead, use mapreduce.input.fileinputformat.split.minsize.per.rack  
15/02/04 20:16:52 INFO Configuration.deprecation: mapred.min.split.size.per.node is deprecate  
d. Instead, use mapreduce.input.fileinputformat.split.minsize.per.node  
15/02/04 20:16:52 INFO Configuration.deprecation: mapred.reduce.tasks is deprecated. Instead,  
use mapreduce.job.reduces  
15/02/04 20:16:52 INFO Configuration.deprecation: mapred.reduce.tasks.speculative.execution i  
s deprecated. Instead, use mapreduce.reduce.speculative  
15/02/04 20:16:52 WARN conf.HiveConf: DEPRECATED: Configuration property hive.metastore.local  
no longer has any effect. Make sure to provide a valid value for hive.metastore.uris if you  
are connecting to a remote metastore.  
Execution log at: /tmp/oracle/oracle_20150204201616_1190995b-9948-4576-8c59-f1095763e025.log  
2015-02-04 08:16:53      Starting to launch local task to process map join;      maximum memor  
y = 257949696  
2015-02-04 08:16:55      Dump the side-table into file: file:/tmp/oracle/hive_2015-02-04_20-16  
-22_402_4058913684699097916-1/-local-10005/HashTable-Stage-11/MapJoin-mapfile21--.hashtable  
2015-02-04 08:16:55      Upload 1 File to: file:/tmp/oracle/hive_2015-02-04_20-16-22_402_40589  
13684699097916-1/-local-10005/HashTable-Stage-11/MapJoin-mapfile21--.hashtable  
2015-02-04 08:16:55      End of local task; Time Taken: 1.81 sec.  
Execution completed successfully  
MapredLocal task succeeded
```

```
Stage-6 is filtered out by condition resolver.
Launching Job 6 out of 7
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1422550693204_0016, Tracking URL = http://bigdatalite.localdomain:8088/proxy/application_1422550693204_0016/
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1422550693204_0016
Hadoop job information for Stage-4: number of mappers: 1; number of reducers: 0
2015-02-04 20:17:47,562 Stage-4 map = 0%, reduce = 0%
2015-02-04 20:17:54,095 Stage-4 map = 100%, reduce = 0%, Cumulative CPU 1.93 sec
2015-02-04 20:17:55,129 Stage-4 map = 100%, reduce = 0%, Cumulative CPU 1.93 sec
MapReduce Total cumulative CPU time: 1 seconds 930 msec
Ended Job = job_1422550693204_0016
Loading data to table moviework.movieapp_log_stage
chgrp: changing ownership of '/user/hive/warehouse/moviework.db/movieapp_log_stage': User does not belong to hive
MapReduce Jobs Launched:
Job 0: Map: 1 Reduce: 1 Cumulative CPU: 9.19 sec HDFS Read: 19254455 HDFS Write: 274313
SUCCESS
Job 1: Map: 1 Cumulative CPU: 6.08 sec HDFS Read: 19254455 HDFS Write: 292702 SUCCESS
Job 2: Map: 2 Cumulative CPU: 9.92 sec HDFS Read: 19550032 HDFS Write: 1813410 SUCCESS
Job 3: Map: 1 Cumulative CPU: 1.93 sec HDFS Read: 1813825 HDFS Write: 1813410 SUCCESS
Total MapReduce CPU Time Spent: 27 seconds 120 msec
OK
Time taken: 92.938 seconds
hive> █
```

You can also click the Tracking URL to track the job progress:

```
Stage-6 is filtered out by condition resolver.
Launching Job 6 out of 7
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1422550693204_0016, Tracking URL = http://bigdatalite.localdomain:8088/proxy/application_1422550693204_0016/
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1422550693204_0016
Hadoop job information for Stage-4: number of mappers: 1; number of reducers: 0
2015-02-04 20:17:47,562 Stage-4 map = 0%, reduce = 0%
```

The screenshot shows a web browser window displaying the Hadoop MapReduce Job details. The URL is `bigdatalite.localdomain:19888/jobhistory/job/job_1422550693204_0016/`. The page title is "MapReduce Job job\_1422550693204\_0016". On the left, there's a sidebar with a "hadoop" logo and navigation links for Application, Job (Overview, Counters, Configuration, Map tasks, Reduce tasks), and Tools. The main content area displays job details and application master statistics.

**Job Details:**

- Job Name:** INSERT OVERWRITE TABLE moviea...union\_result(Stage-4)
- User Name:** oracle
- Queue:** root.oracle
- State:** SUCCEEDED
- Uberized:** false
- Submitted:** Wed Feb 04 20:17:40 EST 2015
- Started:** Wed Feb 04 20:17:46 EST 2015
- Finished:** Wed Feb 04 20:17:53 EST 2015
- Elapsed:** 7sec
- Diagnostics:** Average Map Time 4sec

**ApplicationMaster Statistics:**

Attempt Number	Start Time	Node
1	Wed Feb 04 20:17:42 EST 2015	bigdatalite.localdomain:8042

Task Type	Total	Complete
<b>Map</b>	1	1
<b>Reduce</b>	0	0

Attempt Type	Failed	Killed	Suc
<b>Maps</b>	0	0	1
<b>Reduces</b>	0	0	0

## Practice 11-3: Working with Pig

### Overview

In this practice, you will process the monthly cash accounts data for customers in the XYZ Company. The company keeps a comma-delimited file that contains five columns: customer id, checking account balance, bank funds, number of monthly checks written, and number of automatic payments completed. This data is retrieved from one of the company's internal systems on a monthly basis and it contains the monthly cash situation for each customer. The file that you process contains data for three months; therefore, you will see three rows for each customer where each row corresponds to one month. When you load Pig, you will perform some aggregations on the data and then store the data in a Hive table to obtain the monthly average values per customer, for each measured dimension.

In this practice you perform the following:

- Load the monthly cash account data into the company's HDFS.
- Run a PIG script that aggregates the data to determine the average values of the accounts for each customer.
- View the results and save them into an HDFS file that will be imported to the Oracle Database.
- Upload files into HDFS.

### Tasks

1. Open a new terminal window.
2. Enter the following command at the command prompt, and then press **Enter**.

```
$ cd /home/oracle/exercises/pig
```

3. Use the `head` command to display the first 10 lines of the `export_monthly_cash_accounts.csv` file.

```
head export_monthly_cash_accounts.csv
```

The `head` command prints the first 10 lines of the file. The first 10 rows of the data file are displayed. The first column represents the customer id, followed by the monthly checking amount, bank funds, number of checks written in a month, and number of automatic payments completed.

4. Use the `copyFromLocal` command to load data into the HDFS for processing. Confirm that the file is loaded into the HDFS

```
hadoop fs -copyFromLocal export_monthly_cash_accounts.csv
```

5. Run the PIG script in interactive mode so that you can see each step of the process. Open the PIG interpreter, called `grunt`, using the following command:

```
Pig
```

When you are in the `grunt` shell mode, you can start typing Pig scripts.

6. Load the data file from HDFS into Pig for processing. The data is not actually copied; instead, a handler is created for the file so that Pig knows how to interpret the data. Enter the following on the command line, and then press Enter.

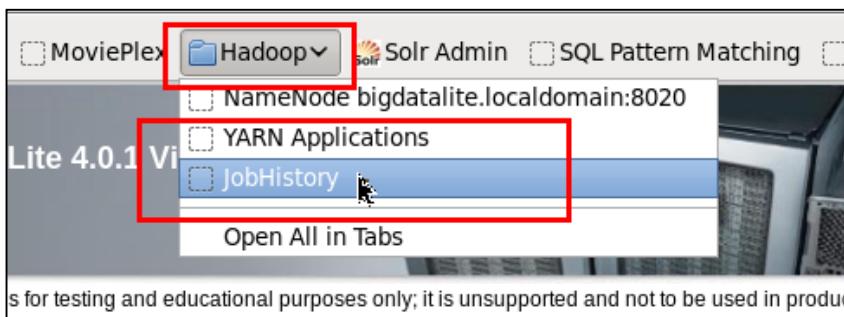
```
grunt> monthly_cash_accounts = load  
'export_monthly_cash_accounts.csv' using PigStorage(',')  
as(customer_id,checking_amount,bank_funds,monthly_checks_written,  
,t_amount_autom_payments);
```

7. View the data loaded as a five-column table.

```
grunt> dump monthly_cash_accounts;
```

Note that the output is similar to the WordCount practice in Lesson 9, which you performed earlier. This is normal because Pig is merely a high-level language. All commands that process data simply run MapReduce tasks in the background, so the dump command simply becomes a MapReduce job that is run. This applies to all the commands you will run in Pig. The output on the screen shows you all the rows of the file in tuple form. **A tuple is an ordered set of fields.**

8. View the details of the completed MapReduce job by using either **YARN Applications** or the **JobHistory** GUI. Open your web browser and then click the Hadoop saved bookmark on the toolbar. Select the **JobHistory** menu option.



The screenshot shows the Hadoop JobHistory interface. At the top, there's a navigation bar with links like 'JobHistory', 'bigdatalite.localdomain:19888/jobhistory', 'Most Visited', 'Hue', 'MoviePlex', 'Hadoop', 'Solr Admin', 'SQL Pattern Matching', 'VirtualBox VMs for ...', and 'Cloudera Manager'. Below the navigation bar is the Hadoop logo. The main title is 'JobHistory'. On the left, there's a sidebar with 'Application' (About, Jobs), 'Tools', and a 'Retired Jobs' section. Under 'Retired Jobs', it says 'Show 20 entries'. A table lists two jobs:

Submit Time	Start Time	Finish Time	Job ID	Name	User	Queue	State
2015.02.05 08:28:01 EST	2015.02.05 08:28:06 EST	2015.02.05 08:28:12 EST	job_1422550693204_0017	PigLatin:DefaultJobName	oracle	root.oracle	SUCCEEDED
2015.02.04 20:17:40 EST	2015.02.04 20:17:46 EST	2015.02.04 20:17:53 EST	job_1422550693204_0016	INSERT OVERWRITE TABLE moviea...union_result(Stage	oracle	root.oracle	SUCCEEDED

JobHistory displays information about jobs that ran. It shows information about when a job was submitted, started, and completed. It also shows the job id, the user who ran the job, the state of the job, and the number of Map and Reduce jobs.

- Analyze the data; you should group the data by customer id so that you have all of the account details of one customer grouped together.

```
grunt> grouped= group monthly_cash_accounts by customer_id;
```

- Dump the grouped variable to check its contents.

```
grunt> dump grouped;
```

All of the groups are displayed in tuple form.

- Go through each group tuple and get the customer id (from the grouped variable) and the average values for the checking\_amount, bank\_funds, monthly\_checks\_written, and the t\_amount\_autom\_payments.

```
grunt> average= foreach grouped generate group,
AVG(monthly_cash_accounts.checking_amount),
AVG(monthly_cash_accounts.bank_funds),
AVG(monthly_cash_accounts.monthly_checks_written),
AVG(monthly_cash_accounts.t_amount_autom_payments);
grunt> dump average;
```

You can see a dump of all customer ids with their respective average account values.

- Sort the average monthly values for each customer id in order from highest to lowest checking amount.

```
grunt> sorted = order average by $1 DESC;
grunt> dump sorted;
```

The list is sorted in descending order. The accounts are displayed with the lowest checking amounts (column highlighted in cyan). You can scroll up to see the rest of the values.

13. Write these results out to HDFS, because you will need them in a file that will be processed by the Oracle SQL Connector for HDFS (OSCH). The OSCH will take the data out of the HDFS file and load it into the Oracle Database, where you will create a 360 degree view of the customers.

```
grunt> store sorted into 'customer_averages' using  
PigStorage(' , ');
```

14. The new calculated data is now permanently stored in HDFS. Exit the grunt shell.

```
grunt> quit;
```

15. View the contents of the customer\_averages directory.

16. View some of the contents of the HDFS file.

```
$ hadoop fs -cat /user/oracle/customer_averages/part-r-00000 |  
head
```

17. Close the terminal.

```
$ exit
```

## Solution 11-3: Working with Pig

### Overview

In this practice, you will process the monthly cash accounts data for the customers in the XYZ company. The company keeps a comma-delimited file that contains five columns: customer id, checking account balance, bank funds, number of monthly checks written, and number of automatic payments completed. This data is retrieved from one of the company's internal systems on a monthly basis and it contains the monthly cash situation for each customer. The file that you will process contains data for three months; therefore, you will see three rows for each customer where each row corresponding to one month. After you load the Pig, you will perform some aggregations on the data and then store the data in a Hive table to obtain the monthly average values per customer, for each measured dimension.

In this practice you perform the following:

- Load the monthly cash account data into the company's HDFS.
- Run a PIG script that aggregates the data to determine the average values of the accounts for each customer.
- View the results and save them into an HDFS file that will be imported to the Oracle Database.
- Upload files into HDFS.

### Tasks

1. Open a new terminal window.



2. Enter the following command at the command prompt, and then press **Enter**.

```
$ cd /home/oracle/exercises/pig
```



3. Use the `head` command to display the first 10 lines of the `export_monthly_cash_accounts.csv` file.

```
head export_monthly_cash_accounts.csv
```

The `head` command prints the first 10 lines of the file. The first 10 rows of the data file are displayed. The first column represents the customer id, followed by the monthly

checking amount, bank funds, number of checks written in a month, and number of automatic payments completed.

```
[oracle@bigdatalite pig]$ head export_monthly_cash_accounts.csv  
CU8983,27.5,0,0,0  
CU8983,25,0,0,0  
CU8983,22.5,0,0,0  
CU1655,176,501,0,668  
CU1655,160,450.9,1,568  
CU1655,144,551.1,2,768  
CU13483,27.5,750,13,161  
CU13483,25,675,14,61  
CU13483,22.5,825,15,261  
CU9863,27.5,0,0,0  
[oracle@bigdatalite pig]$ █
```

4. Use the `copyFromLocal` command to load data into the HDFS for processing. Confirm that the file is loaded into the HDFS.

```
$ hadoop fs -copyFromLocal export_monthly_cash_accounts.csv
```

```
[oracle@bigdatalite pig]$ hadoop fs -copyFromLocal export_monthly_cash_accounts.csv  
[oracle@bigdatalite pig]$ hadoop fs -ls export_monthly_cash_accounts.csv  
Found 1 items  
-rw-r--r-- 1 oracle oracle 67522 2015-02-05 08:14 export_monthly_cash_accounts.csv  
[oracle@bigdatalite pig]$ hadoop fs -ls  
Found 8 items  
drwx----- - oracle oracle 0 2014-08-25 05:55 .Trash  
drwx----- - oracle oracle 0 2015-02-04 20:17 .staging  
-rw-r--r-- 1 oracle oracle 67522 2015-02-05 08:14 export_monthly_cash_accounts.csv  
drwxr-xr-x - oracle oracle 0 2014-01-12 18:15 moviedemo  
drwxr-xr-x - oracle oracle 0 2014-09-24 09:38 moviework  
drwxr-xr-x - oracle oracle 0 2014-09-08 15:50 oggdemo  
drwxr-xr-x - oracle oracle 0 2014-09-20 13:59 oozie-oozi  
drwxr-xr-x - oracle oracle 0 2015-02-02 11:20 wordcount  
[oracle@bigdatalite pig]$ █
```

5. Run the PIG script in interactive mode so that you can see each step of the process. Open the PIG interpreter, called grunt, by using the following command:

```
pig
```

When you are in the grunt shell mode, you can start typing Pig scripts.

```
[oracle@bigdatalite pig]$ pig
2015-02-05 08:18:43,762 [main] INFO org.apache.pig.Main - Apache Pig version 0.
compiled Aug 25 2014, 19:51:44
2015-02-05 08:18:43,762 [main] INFO org.apache.pig.Main - Logging error messages
to /pig/pig_1423142323760.log
2015-02-05 08:18:43,780 [main] INFO org.apache.pig.impl.util.Utils - Default boot
up not found
2015-02-05 08:18:44,199 [main] INFO org.apache.hadoop.conf.Configuration.deprecate
is deprecated. Instead, use mapreduce.jobtracker.address
2015-02-05 08:18:44,200 [main] INFO org.apache.hadoop.conf.Configuration.deprecate
is deprecated. Instead, use fs.defaultFS
2015-02-05 08:18:44,200 [main] INFO org.apache.pig.backend.hadoop.executionengine
switching to hadoop file system at: hdfs://bigdatalite.localdomain:8020
2015-02-05 08:18:45,099 [main] INFO org.apache.hadoop.conf.Configuration.deprecate
is deprecated. Instead, use fs.defaultFS
grunt> ■
```

6. Load the data file from HDFS into Pig for processing. The data is not actually copied; instead, a handler is created for the file so that Pig knows how to interpret the data. Enter the following on the command line, and then press Enter.

```
grunt> monthly_cash_accounts = load
'export_monthly_cash_accounts.csv' using PigStorage(',')
as(customer_id,checking_amount,bank_funds,monthly_checks_written
,t_amount_autom_payments);
```

```
grunt> monthly_cash_accounts = load 'export_monthly_cash_accounts.csv' using PigStorage(',')
,checking_amount,bank_funds,monthly_checks_written,t_amount_autom_payments);
grunt> ■
```

7. View the data loaded as a five-column table.

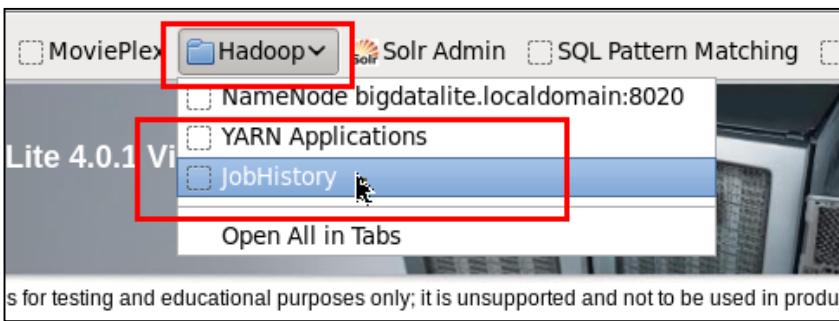
```
grunt> dump monthly_cash_accounts;
```

```
grunt> monthly_cash_accounts = load 'export_monthly_cash_accounts.csv' using Pig
Storage(',') as(customer_id,checking_amount,bank_funds,monthly_checks_written,t_
amount_autom_payments);
grunt> dump monthly cash accounts;
```

Note that the output is similar to the WordCount practice in Lesson 9, which you performed earlier. This is normal because Pig is merely a high-level language. All commands that process data simply run MapReduce tasks in the background, so the `dump` command simply becomes a MapReduce job that is run. This applies to all of the commands you will run in Pig. The output on the screen shows you all of the rows of the file in tuple form. **A tuple is an ordered set of fields.** The partial output is as follows:

```
(CU8647,24.3,5390,4,1412)
(CU5473,27.5,3601,0,550)
(CU5473,25,3240.9,1,450)
(CU5473,22.5,3961.1,2,650)
(CU10161,27.5,700,10,242)
(CU10161,25,630,11,142)
(CU10161,22.5,770,12,342)
(CU6234,695.2,9850,3,3005)
(CU6234,632,8865,4,2905)
(CU6234,568.8,10835,5,3105)
(CU12312,27.5,500,6,125)
(CU12312,25,450,7,25)
(CU12312,22.5,550,8,225)
(CU975,809.6,6600,1,2321)
(CU975,736,5940,2,2221)
(CU975,662.4,7260,3,2421)
(CU7799,27.5,500,14,196)
(CU7799,25,450,15,96)
(CU7799,22.5,550,16,296)
(CU2753,27.5,1800,1,2086)
(CU2753,25,1620,2,1986)
(CU2753,22.5,1980,3,2186)
(CU11667,27.5,1050,2,211)
grunt> █
```

8. View the details of the completed MapReduce job by using either **YARN Applications** or the **JobHistory** GUI. Open your web browser, and then click the Hadoop saved bookmark on the toolbar.



The screenshot shows the Hadoop JobHistory interface. At the top, there's a navigation bar with links like 'JobHistory', 'bigdatalite.localdomain:19888/jobhistory', 'Most Visited', 'Hue', 'MoviePlex', 'Hadoop', 'Solr Admin', 'SQL Pattern Matching', 'VirtualBox VMs for ...', and 'Cloudera Manager'. Below the navigation bar is the Hadoop logo. The main title is 'JobHistory'. On the left, there's a sidebar with 'Application' (About, Jobs), 'Tools', and a 'Retired Jobs' section. Under 'Retired Jobs', it says 'Show 20 entries'. A table follows, with columns: Submit Time, Start Time, Finish Time, Job ID, Name, User, Queue, and State. Two rows are highlighted with a red border:

Submit Time	Start Time	Finish Time	Job ID	Name	User	Queue	State
2015.02.05 08:28:01 EST	2015.02.05 08:28:06 EST	2015.02.05 08:28:12 EST	job_1422550693204_0017	PigLatin:DefaultJobName	oracle	root.oracle	SUCCEEDED
2015.02.04 20:17:40 EST	2015.02.04 20:17:46 EST	2015.02.04 20:17:53 EST	job_1422550693204_0016	INSERT OVERWRITE TABLE moviea...union_result(Stage	oracle	root.oracle	SUCCEEDED

JobHistory displays information about jobs that ran. It shows information about when a job was submitted, started, and completed. It also shows the job id, the user who ran the job, the state of the job, and the number of Map and Reduce jobs.

- Analyze the data; you should group the data by customer id so that you have all of the account details of one customer grouped together.

```
grunt> grouped= group monthly_cash_accounts by customer_id;
```

```
grunt> grouped= group monthly_cash_accounts by customer_id;
2015-02-05 09:02:03,333 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - . Instead, use fs.defaultFS
grunt> ■
```

- Dump the grouped variable to check its contents.

```
grunt> dump grouped;
```

```
grunt> grouped= group monthly_cash_accounts by customer_id;
2014-08-07 01:59:36,041 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.defaultFS
grunt> dump grouped;■
```

All of the groups are displayed in tuple form. Because the output might look a bit confusing, only some tuples are highlighted in the following partial screenshot for clarity.

```
(CU15509,{(CU15509,27.5,500,2,134),(CU15509,25,450,3,34),(CU15509,22.5,550,4,234)})  
(CU15532,{(CU15532,25,2160,4,7839),(CU15532,22.5,2640,5,8039),(CU15532,27.5,2400,3,7939)})  
(CU15559,{(CU15559,22.5,0,0,0),(CU15559,25,0,0,0),(CU15559,27.5,0,0,0)})  
(CU15599,{(CU15599,283,675,5,693),(CU15599,254.7,825,6,893),(CU15599,311.3,750,4,793)})  
(CU15635,{(CU15635,25434.2,2100,1,23180),(CU15635,20809.8,2310,3,23280),(CU15635,23122,1890,1),(CU15671,{(CU15671,4080,2700,2,3995),(CU15671,4488,3000,1,4095),(CU15671,3672,3300,3,4195)})  
(CU15694,{(CU15694,22.5,0,3,0),(CU15694,27.5,0,1,0),(CU15694,25,0,2,0)})  
(CU15745,{(CU15745,8397.4,0,1,14368),(CU15745,7634,0,2,14268),(CU15745,6870.6,0,3,14468)})  
(CU15766,{(CU15766,22.5,13750,5,38308),(CU15766,25,11250,4,38108),(CU15766,27.5,12500,3,3820)}  
(CU15780,{(CU15780,27.5,1900,3,44737),(CU15780,25,1710,4,44637),(CU15780,22.5,2090,5,44837)})  
(CU15782,{(CU15782,27.5,1000,0,0),(CU15782,25,900,0,0),(CU15782,22.5,1100,0,0)})  
(CU15784,{(CU15784,10798.2,9350,4,12912),(CU15784,13197.8,8500,2,12812),(CU15784,11998,7650,1),(CU15786,25,675,5,137),(CU15786,22.5,825,6,337),(CU15786,27.5,750,4,237)})  
(CU15798,{(CU15798,25,0,2,0),(CU15798,27.5,0,1,0),(CU15798,22.5,0,3,0)})  
(CU15800,{(CU15800,25,450,4,84),(CU15800,22.5,550,5,284),(CU15800,27.5,500,3,184)})  
(CU15809,{(CU15809,25,0,0,0),(CU15809,27.5,0,0,0),(CU15809,22.5,0,0,0)})  
(CU15821,{(CU15821,143.1,0,9,272),(CU15821,159,0,8,72),(CU15821,174.9,0,7,172)})  
(CU15828,{(CU15828,27.5,0,0,505),(CU15828,25,0,0,405),(CU15828,22.5,0,0,605)})  
(CU15839,{(CU15839,22.5,0,0,0),(CU15839,27.5,0,0,0),(CU15839,25,0,0,0)})  
(CU15852,{(CU15852,27.5,0,0,0),(CU15852,25,0,0,0),(CU15852,22.5,0,0,0)})  
(CU15853,{(CU15853,795.3,500,5,1114),(CU15853,723,450,6,1014),(CU15853,650.7,550,7,1214)})  
(CU15854,{(CU15854,25,585,15,88),(CU15854,22.5,715,16,288),(CU15854,27.5,650,14,188)})  
(CU15866,{(CU15866,27.5,250,1,0),(CU15866,25,225,2,0),(CU15866,22.5,275,3,0)})  
(CU15879,{(CU15879,22.5,0,0,0),(CU15879,25,0,0,0),(CU15879,27.5,0,0,0)})  
(CU15886,{(CU15886,25,225,4,0),(CU15886,22.5,275,5,0),(CU15886,27.5,250,3,0)})  
(CU15889,{(CU15889,22.5,0,0,0),(CU15889,27.5,0,0,0),(CU15889,25,0,0,0)})  
(CU15927,{(CU15927,27.5,0,8,0),(CU15927,22.5,0,10,0),(CU15927,25,0,9,0)})  
(CU15942,{(CU15942,22.5,275,0,0),(CU15942,25,225,0,0),(CU15942,27.5,250,0,0)})  
(CU15957,{(CU15957,22.5,0,13,0),(CU15957,25,0,12,0),(CU15957,27.5,0,11,0)})  
(CU15960,{(CU15960,22.5,0,2,0),(CU15960,27.5,0,0,0),(CU15960,25,0,1,0)})  
(CU15979,{(CU15979,25,0,4,0),(CU15979,22.5,0,5,0),(CU15979,27.5,0,3,0)})  
(CU15988,{(CU15988,27.5,1200,2,135),(CU15988,25,1080,3,35),(CU15988,22.5,1320,4,235)})  
grunt> ■
```

11. Go through each group tuple and get the customer id (from the grouped variable) and the average values for the checking\_amount, bank\_funds, monthly\_checks\_written, and t\_amount\_autom\_payments.

```
grunt> average= foreach grouped generate group,  
AVG(monthly_cash_accounts.checking_amount),  
AVG(monthly_cash_accounts.bank_funds),  
AVG(monthly_cash_accounts.monthly_checks_written),  
AVG(monthly_cash_accounts.t_amount_autom_payments);  
grunt> dump average;
```

```
grunt> average= foreach grouped generate group, AVG(monthly_cash_accounts.checking_amount), AVG(monthly_cash_accounts.bank_funds), AVG(monthly_cash_accounts.monthly_checks_written), AVG(monthly_cash_accounts.t_amount_autom_payments);  
grunt> dump average;■
```

You can see a dump of all customer ids with their respective average account values.

```
2015-02-05 09:43:14,500 [JobControl] INFO org.apache.hadoop.yarn.client.api.impl.YarnClientImpl
ion application_1422550693204_0019
2015-02-05 09:43:14,509 [JobControl] INFO org.apache.hadoop.mapreduce.Job - The url to track the
lite.localdomain:8088/proxy/application_1422550693204_0019/
2015-02-05 09:43:14,775 [main] INFO org.apache.pig.backend.hadoop.executionengine.mapReduceLayer
HadoopJobId: job_1422550693204_0019
2015-02-05 09:43:14,775 [main] INFO org.apache.pig.backend.hadoop.executionengine.mapReduceLayer
Processing aliases average,grouped,monthly_cash_accounts
2015-02-05 09:43:14,775 [main] INFO org.apache.pig.backend.hadoop.executionengine.mapReduceLayer
detailed locations: M: monthly_cash_accounts[1,24],average[3,9],grouped[2,9] C: average[3,9],gr
[3,9]
2015-02-05 09:43:14,809 [main] INFO org.apache.pig.backend.hadoop.executionengine.mapReduceLayer
0% complete
2015-02-05 09:43:27,340 [main] INFO org.apache.pig.backend.hadoop.executionengine.mapReduceLayer
50% complete
■
```

```
- Setting Parallelism to 1
2015-02-05 09:43:11,516 [main] INFO org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.Jo
- creating jar file Job2877538282290083299.jar
2015-02-05 09:43:14,188 [main] INFO org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.Jo
- jar file Job2877538282290083299.jar created
2015-02-05 09:43:14,205 [main] INFO org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.Jo
- Setting up single store job
2015-02-05 09:43:14,220 [main] INFO org.apache.pig.data.SchemaTupleFrontend - Key [pig.schematuple]
ot generate code.
2015-02-05 09:43:14,220 [main] INFO org.apache.pig.data.SchemaTupleFrontend - Starting process to r
e to distributed cache
2015-02-05 09:43:14,221 [main] INFO org.apache.pig.data.SchemaTupleFrontend - Setting key [pig.sche
with classes to deserialize []
2015-02-05 09:43:14,274 [main] INFO org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.M
1 map-reduce job(s) waiting for submission.
2015-02-05 09:43:14,276 [JobControl] INFO org.apache.hadoop.yarn.client.RMProxy - Connecting to Res
ocalhost/127.0.0.1:8032
2015-02-05 09:43:14,286 [JobControl] INFO org.apache.hadoop.conf.Configuration.deprecation - fs.def
ecated. Instead, use fs.defaultFS
2015-02-05 09:43:14,404 [JobControl] INFO org.apache.hadoop.mapreduce.lib.input.FileInputFormat - I
to process : 1
2015-02-05 09:43:14,404 [JobControl] INFO org.apache.pig.backend.hadoop.executionengine.util.MapRe
t paths to process : 1
2015-02-05 09:43:14,406 [JobControl] INFO org.apache.pig.backend.hadoop.executionengine.util.MapRe
t paths (combined) to process : 1
2015-02-05 09:43:14,461 [JobControl] INFO org.apache.hadoop.mapreduce.JobSubmitter - number of spli
2015-02-05 09:43:14,493 [JobControl] INFO org.apache.hadoop.mapreduce.JobSubmitter - Submitting to
1422550693204_0019
2015-02-05 09:43:14,506 [JobControl] INFO org.apache.hadoop.yarn.client.api.impl.YarnClientImpl - S
ion application_1422550693204_0019
2015-02-05 09:43:14,509 [JobControl] INFO org.apache.hadoop.mapreduce.Job - The url to track the jo
lite.localdomain:8088/proxy/application_1422550693204_0019/
■
```

```
(CU15509,25.0,500.0,3.0,134.0)
(CU15532,25.0,2400.0,4.0,7939.0)
(CU15559,25.0,0.0,0.0,0.0)
(CU15599,283.0,750.0,5.0,793.0)
(CU15635,23122.0,2100.0,2.0,23180.0)
(CU15671,4080.0,3000.0,2.0,4095.0)
(CU15694,25.0,0.0,2.0,0.0)
(CU15745,7634.0,0.0,2.0,14368.0)
(CU15766,25.0,12500.0,4.0,38208.0)
(CU15780,25.0,1900.0,4.0,44737.0)
(CU15782,25.0,1000.0,0.0,0.0)
(CU15784,11998.0,8500.0,3.0,12812.0)
(CU15786,25.0,750.0,5.0,237.0)
(CU15798,25.0,0.0,2.0,0.0)
(CU15800,25.0,500.0,4.0,184.0)
(CU15809,25.0,0.0,0.0,0.0)
(CU15821,159.0,0.0,8.0,172.0)
(CU15828,25.0,0.0,0.0,505.0)
(CU15839,25.0,0.0,0.0,0.0)
(CU15852,25.0,0.0,0.0,0.0)
(CU15853,723.0,500.0,6.0,1114.0)
(CU15854,25.0,650.0,15.0,188.0)
(CU15866,25.0,250.0,2.0,0.0)
(CU15879,25.0,0.0,0.0,0.0)
(CU15886,25.0,250.0,4.0,0.0)
(CU15889,25.0,0.0,0.0,0.0)
(CU15927,25.0,0.0,9.0,0.0)
(CU15942,25.0,250.0,0.0,0.0)
(CU15957,25.0,0.0,12.0,0.0)
(CU15960,25.0,0.0,1.0,0.0)
(CU15979,25.0,0.0,4.0,0.0)
(CU15988,25.0,1200.0,3.0,135.0)
grunt> █
```

12. Sort the average monthly values for each customer id in order from highest to lowest checking amount.

```
grunt> sorted = order average by $1 DESC;
grunt> dump sorted;
```

```
grunt> sorted = order average by $1 DESC;
grunt> dump sorted;█
```

The list is sorted in descending order. The accounts are displayed with the lowest checking amounts (column highlighted in cyan). You can scroll up to see the rest of the values.

```
(CU1015,25.0,501.0,0.0,0.0)
(CU1005,25.0,3000.0,1.0,787.0)
(CU985,25.0,8350.0,1.0,567.0)
(CU938,25.0,0.0,0.0,507.0)
(CU929,25.0,0.0,0.0,507.0)
(CU895,25.0,9500.0,3.0,76330.0)
(CU878,25.0,550.0,2.0,607.0)
(CU868,25.0,12500.0,17.0,59437.0)
(CU860,25.0,0.0,0.0,503.0)
(CU847,25.0,7644.0,2.0,98590.0)
(CU840,25.0,16800.0,3.0,670.0)
(CU793,25.0,24600.0,4.0,14975.0)
(CU790,25.0,16141.0,2.0,560.0)
(CU726,25.0,23000.0,1.0,30329.0)
(CU685,25.0,2401.0,4.0,3702.0)
(CU544,25.0,0.0,0.0,503.0)
(CU513,25.0,0.0,0.0,504.0)
(CU491,25.0,3686.0,1.0,958.0)
(CU475,25.0,0.0,2.0,504.0)
(CU396,25.0,9000.0,3.0,2680.0)
(CU339,25.0,17001.0,14.0,21264.0)
(CU293,25.0,3200.0,2.0,2826.0)
(CU225,25.0,3100.0,2.0,1046.0)
(CU216,25.0,2201.0,4.0,55686.0)
(CU197,25.0,0.0,0.0,598.0)
(CU161,25.0,6500.0,4.0,2220.0)
(CU153,25.0,0.0,0.0,504.0)
(CU141,25.0,3500.0,12.0,2975.0)
(CU129,25.0,0.0,1.0,713.0)
(CU27,25.0,0.0,0.0,506.0)
(CU2,25.0,0.0,0.0,542.0)
grunt> ■
```

13. Write these results out to HDFS, because you will need them in a file that will be processed by the Oracle SQL Connector for HDFS (OSCH). The OSCH will take the data out of the HDFS file and load it into the Oracle Database, where you will create a 360 degree view of the customers.

```
grunt> store sorted into 'customer_averages' using
PigStorage(' ','');
```

```
grunt> store sorted into 'customer_averages' using PigStorage(' ','');
```

```
2015-02-05 10:33:18,904 [main] INFO org.apache.hadoop.ipc.Client - Retrying connect to server: bin/127.0.0.1:29741. Already tried 0 time(s); retry policy is RetryUpToMaximumCountWithFixedSleep(maxRetries=10, sleepTime=1000 MILLISCONDS)
2015-02-05 10:33:19,904 [main] INFO org.apache.hadoop.ipc.Client - Retrying connect to server: bin/127.0.0.1:29741. Already tried 1 time(s); retry policy is RetryUpToMaximumCountWithFixedSleep(maxRetries=10, sleepTime=1000 MILLISCONDS)
2015-02-05 10:33:20,905 [main] INFO org.apache.hadoop.ipc.Client - Retrying connect to server: bin/127.0.0.1:29741. Already tried 2 time(s); retry policy is RetryUpToMaximumCountWithFixedSleep(maxRetries=10, sleepTime=1000 MILLISCONDS)
2015-02-05 10:33:21,011 [main] INFO org.apache.hadoop.mapred.ClientServiceDelegate - Application FinalApplicationStatus=SUCCEEDED. Redirecting to job history server
2015-02-05 10:33:21,339 [main] INFO org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.Success!
grunt> ■
```

14. The new calculated data is now permanently stored in HDFS. Exit the grunt shell.

```
grunt> quit;
```

```
grunt> quit  
[oracle@bigdatalite pig]$ █
```

15. View the contents of the `customer_averages` directory.

```
[oracle@bigdatalite pig]$ hadoop fs -ls /user/oracle  
Found 9 items  
drwx-----  - oracle oracle      0 2014-08-25 05:55 /user/oracle/.Trash  
drwx-----  - oracle oracle      0 2015-02-05 10:33 /user/oracle/_staging  
drwxr-xr-x  - oracle oracle      0 2015-02-05 10:33 /user/oracle/customer_averages  
-rw-r--r--  1 oracle oracle  67522 2015-02-05 08:14 /user/oracle/export_monthly_cash_accounts.csv  
drwxr-xr-x  - oracle oracle      0 2014-01-12 18:15 /user/oracle/moviedemo  
drwxr-xr-x  - oracle oracle      0 2014-09-24 09:38 /user/oracle/moviework  
drwxr-xr-x  - oracle oracle      0 2014-09-08 15:50 /user/oracle/oggdemo  
drwxr-xr-x  - oracle oracle      0 2014-09-20 13:59 /user/oracle/oozie-oozi  
drwxr-xr-x  - oracle oracle      0 2015-02-02 11:20 /user/oracle/wordcount  
[oracle@bigdatalite pig]$ hadoop fs -ls /user/oracle/customer_averages  
Found 2 items  
-rw-r--r--  1 oracle oracle      0 2015-02-05 10:33 /user/oracle/customer_averages/_SUCCESS  
-rw-r--r--  1 oracle oracle  29193 2015-02-05 10:33 /user/oracle/customer_averages/part-r-00000  
[oracle@bigdatalite pig]$ █
```

16. View some of the content of the HDFS file.

```
$ hadoop fs -cat /user/oracle/customer_averages/part-r-00000 |  
head
```

```
[oracle@bigdatalite pig]$ hadoop fs -cat /user/oracle/customer_averages/part-r-00000 | head  
CU55,592.0,36000.0,6.0,499362.0  
CU5630,25.0,33500.0,4.0,165815.0  
CU1091,25.0,5501.0,0.0,112822.0  
CU9789,25.0,22001.0,2.0,107253.0  
CU5676,25.0,10200.0,2.0,105507.0  
CU12603,7170.0,7648.0,17.0,100578.0  
CU3167,21069.0,10525.0,1.0,98684.0  
CU847,25.0,7644.0,2.0,98590.0  
CU5807,2129.0,10049.0,2.0,87304.0  
CU5992,1428.0,9100.0,2.0,77470.0  
cat: Unable to write to output stream.  
[oracle@bigdatalite pig]$ █
```

17. Close the terminal.

```
$ exit
```

## **Practices for Lesson 12: Overview of Cloudera Impala**

**Chapter 12**

## Practices for Lesson 12

---

There are no practices for this lesson.

# **Practices for Lesson 13: Using Oracle XQuery for Hadoop**

**Chapter 13**

## Practices for Lesson 13

---

### Practices Overview

In these practices, you will perform the following:

- Review basic XML.
- Write and execute an XQuery against data in the Hadoop cluster on HDFS.
- Use OXH to transform an XML file in Hive.
- Load results from an XQuery into an Oracle Database.

## Practice 13-1: Using Oracle XQuery for Hadoop (OXH)

### Overview

In this practice, you write and execute an XQuery against data in the Hadoop cluster on HDFS.

### Assumptions

### Tasks

**Note:** The scripts for this practice are in the /home/oracle/exercises/OXH folder.

1. Open a terminal window.
2. Enter the following command at the command prompt, and then press **Enter**.

```
cd /home/oracle/exercises/OXH
```

3. Review the contents of the books.xml file, which you will use in this practice. Use gedit.

```
[oracle@bigdatalite OXH]$ gedit books.xml
```

```
1 <?xml version="1.0" encoding="ISO-8859-1"?>
2 <bookstore>
3
4 <book category="COOKING">
5   <title lang="en">Everyday Italian</title>
6   <author>Giada De Laurentiis</author>
7   <year>2005</year>
8   <price>30.00</price>
9 </book>
10
11 <book category="CHILDREN">
12   <title lang="en">Harry Potter</title>
13   <author>J. K. Rowling</author>
14   <year>2005</year>
15   <price>29.99</price>
16 </book>
17
18 <book category="WEB">
19   <title lang="en">XQuery Kick Start</title>
20   <author>James McGovern</author>
21   <author>Per Bothner</author>
22   <author>Kurt Cagle</author>
23   <author>James Linn</author>
24   <author>Vaidyanathan Nagarajan</author>
25   <year>2003</year>
26   <price>49.99</price>
27 </book>
28
29 <book category="WEB">
30   <title lang="en">Learning XML</title>
31   <author>Erik T. Ray</author>
32   <year>2003</year>
33   <price>39.95</price>
34 </book>
35
36 </bookstore>
37
```

4. View the contents of the books1a.xq file.

```
cat books1a.xq
```

5. View the contents of the books1b.xq file.

```
cat books1b.xq
```

6. View the contents of the books1-local.sh file. OXH XQuery is currently set to run locally (-jt local -fs local), which tests the queries against a small set of sample data before going to the Hadoop cluster and accessing large data sets.

```
cat books1-local.sh
```

7. Run the books1-local.sh file.

```
./books1-local.sh
```

8. View the contents of the books1.sh file.

```
cat books1.sh
```

9. Copy the books.xml file to the Hadoop HDFS file system and confirm that it is copied to HDFS.

```
hadoop fs -put books.xml  
hadoop fs -ls books.xml
```

10. Run the books1.sh script.

```
./books1.sh
```

11. Filter by price. Display only the books that cost less than \$30. Review the contents of the books2.xq script.

12. Run the books2.sh script.

```
./books2.sh
```

13. Add year to the filter. Display only the books that cost more than \$10 and that were published in 2005. Open the books2a.xq script and review the code.

- Edit the books2.xq script and review the code.
- Replace the where and return clauses as follows:

```
import module namespace xmlf = "oxh:xmlf";  
import module namespace text = "oxh:text";  
for $x in xmlf:collection("books.xml", "book")  
where $x/price>10 and $x/year eq '2005'  
return text:put-text($x/title || ' $' || $x/price)
```

- Save the file as books2a.xq in the OXH folder, and then exit gedit.
- View the new books2a.xq file.

14. Open books2.sh by using gedit, replace books2.xq with books2a.xq, save your changes, exit gedit, and then run the script.

```
./books2.sh
```

15. Filter by the category “COOKING”. Edit books2a.xq script, and change the code as follows. Save the file as books2b.xq in the OXH folder, and then exit gedit.

```
import module namespace xmlf = "oxh:xmlf";  
import module namespace text = "oxh:text";  
for $x in xmlf:collection("books.xml", "book")  
where $x/@category eq 'COOKING'
```

```
return text:put-text($x/title || ' Category : ' || $x/@category  
|| ' $' || $x/price)
```

16. Open books2.sh and replace books2a.xq with books2b.xq and run the script.

```
./books2.sh
```

## Solution 13-1: Using Oracle XQuery for Hadoop (OXH)

### Overview

In this solution, you write and execute an XQuery against data in the Hadoop cluster on HDFS.

### Steps

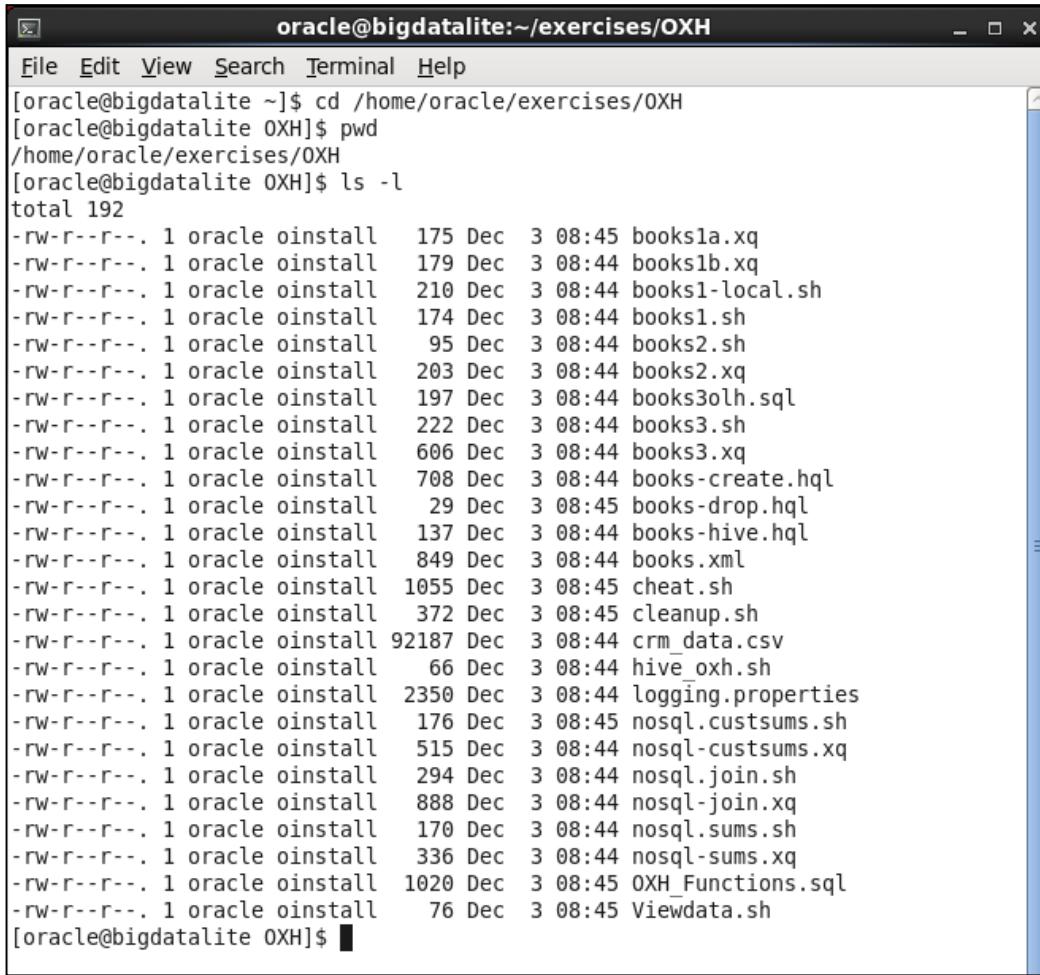
**Note:** The scripts for this practice are in the /home/oracle/exercises/OXH folder.

1. Open a terminal window.



2. Enter the following command at the command prompt, and then press **Enter**.

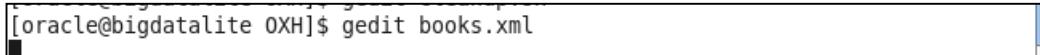
```
cd /home/oracle/exercises/OXH
```



oracle@bigdatalite:~/exercises/OXH

```
[oracle@bigdatalite ~]$ cd /home/oracle/exercises/OXH
[oracle@bigdatalite OXH]$ pwd
/home/oracle/exercises/OXH
[oracle@bigdatalite OXH]$ ls -l
total 192
-rw-r--r--. 1 oracle oinstall 175 Dec  3 08:45 books1a.xq
-rw-r--r--. 1 oracle oinstall 179 Dec  3 08:44 books1b.xq
-rw-r--r--. 1 oracle oinstall 210 Dec  3 08:44 books1-local.sh
-rw-r--r--. 1 oracle oinstall 174 Dec  3 08:44 books1.sh
-rw-r--r--. 1 oracle oinstall  95 Dec  3 08:44 books2.sh
-rw-r--r--. 1 oracle oinstall 203 Dec  3 08:44 books2.xq
-rw-r--r--. 1 oracle oinstall 197 Dec  3 08:44 books3olh.sql
-rw-r--r--. 1 oracle oinstall 222 Dec  3 08:44 books3.sh
-rw-r--r--. 1 oracle oinstall 606 Dec  3 08:44 books3.xq
-rw-r--r--. 1 oracle oinstall 708 Dec  3 08:44 books-create.hql
-rw-r--r--. 1 oracle oinstall  29 Dec  3 08:45 books-drop.hql
-rw-r--r--. 1 oracle oinstall 137 Dec  3 08:44 books-hive.hql
-rw-r--r--. 1 oracle oinstall 849 Dec  3 08:44 books.xml
-rw-r--r--. 1 oracle oinstall 1055 Dec  3 08:45 cheat.sh
-rw-r--r--. 1 oracle oinstall 372 Dec  3 08:45 cleanup.sh
-rw-r--r--. 1 oracle oinstall 92187 Dec  3 08:44 crm_data.csv
-rw-r--r--. 1 oracle oinstall   66 Dec  3 08:44 hive_oxh.sh
-rw-r--r--. 1 oracle oinstall 2350 Dec  3 08:44 logging.properties
-rw-r--r--. 1 oracle oinstall 176 Dec  3 08:45 nosql.custsums.sh
-rw-r--r--. 1 oracle oinstall 515 Dec  3 08:44 nosql-custsums.xq
-rw-r--r--. 1 oracle oinstall 294 Dec  3 08:44 nosql.join.sh
-rw-r--r--. 1 oracle oinstall 888 Dec  3 08:44 nosql-join.xq
-rw-r--r--. 1 oracle oinstall 170 Dec  3 08:44 nosql.sums.sh
-rw-r--r--. 1 oracle oinstall 336 Dec  3 08:44 nosql-sums.xq
-rw-r--r--. 1 oracle oinstall 1020 Dec  3 08:45 OXH_Functions.sql
-rw-r--r--. 1 oracle oinstall   76 Dec  3 08:45 Viewdata.sh
[oracle@bigdatalite OXH]$ █
```

3. Review the contents of the `books.xml` file, which you will use in this practice. Use `gedit`.



```
[oracle@bigdatalite OXH]$ gedit books.xml
```

```
books.xml X
1 <?xml version="1.0" encoding="ISO-8859-1"?>
2 <bookstore>
3
4 <book category="COOKING">
5   <title lang="en">Everyday Italian</title>
6   <author>Giada De Laurentiis</author>
7   <year>2005</year>
8   <price>30.00</price>
9 </book>
10
11 <book category="CHILDREN">
12   <title lang="en">Harry Potter</title>
13   <author>J. K. Rowling</author>
14   <year>2005</year>
15   <price>29.99</price>
16 </book>
17
18 <book category="WEB">
19   <title lang="en">XQuery Kick Start</title>
20   <author>James McGovern</author>
21   <author>Per Bothner</author>
22   <author>Kurt Cagle</author>
23   <author>James Linn</author>
24   <author>Vaidyanathan Nagarajan</author>
25   <year>2003</year>
26   <price>49.99</price>
27 </book>
28
29 <book category="WEB">
30   <title lang="en">Learning XML</title>
31   <author>Erik T. Ray</author>
32   <year>2003</year>
33   <price>39.95</price>
34 </book>
35
36 </bookstore>
37
```

4. View the contents of the books1a.xq file.

```
cat books1a.xq
```

```
[oracle@bigdatalite 0XH]$ cat books1a.xq
import module namespace xmlf = "oxh:xmlf";
import module namespace text = "oxh:text";

for $x in xmlf:collection("books.xml")/bookstore/book/title
return text:put-xml($x)

[oracle@bigdatalite 0XH]$ █
```

5. View the contents of the books1b.xq file.

```
cat books1b.xq
```

```
[oracle@bigdatalite OXH]$ cat books1b.xq
import module namespace xmlf = "oxh:xmlf";
import module namespace text = "oxh:text";

for $x in xmlf:collection("books.xml")/bookstore/book[price<30]
return text:put-xml($x)

[oracle@bigdatalite OXH]$ █
```

6. View the contents of the books1-local.sh file. OXH XQuery is currently set to run locally (-jt local -fs local), which tests the queries against a small set of sample data before going to the Hadoop cluster and accessing large data sets.

```
cat books1-local.sh
```

```
[oracle@bigdatalite OXH]$ cat books1-local.sh
rm -rf books1
hadoop jar $OXH_HOME/lib/oxh.jar -jt local -fs local ./books1a.xq -print -output
./books1

rm -rf books2
hadoop jar $OXH_HOME/lib/oxh.jar -jt local -fs local ./books1b.xq -print -output
./books2

[oracle@bigdatalite OXH]$ █
```

7. Run the books1-local.sh file.

**Note:** If you get a “permission denied” error when you attempt to run the .sh script, issue the following command on the OXH directory because all .sh files should have execute privileges: chmod 777 \*.sh

```
./books1-local.sh
```

The partial output is as follows:

```
[oracle@bigdatalite 0XH]$ ./books1-local.sh
15/02/09 09:52:00 WARN fs.FileSystem: "local" is a deprecated filesystem name. Use "file://" instead.
15/02/09 09:52:01 INFO hadoop.xquery: OXH: Oracle XQuery for Hadoop 4.0.1 (build 4.0.1-cdh5.0.0-mr1 @mr2). Copyright (c) 2015, Oracle. All rights reserved.
15/02/09 09:52:01 INFO hadoop.xquery: Executing query "./books1a.xq". Output path: "file:/home/oracle/exercises/OXH/books1"
15/02/09 09:52:03 INFO hadoop.xquery: Submitting map-reduce job "oxh:books1a.xq#0" id="c3f3cad8-8292-46b0-9569-0224df5ff652.0", inputs=[file:/home/oracle/exercises/OXH/books.xml], output=file:/home/oracle/exercises/OXH/books1
15/02/09 09:52:03 INFO Configuration.deprecation: session.id is deprecated. Instead, use dfs.metrics.session-id
15/02/09 09:52:03 INFO jvm.JvmMetrics: Initializing JVM Metrics with processName=JobTracker, sessionId=
15/02/09 09:52:04 INFO input.FileInputFormat: Total input paths to process : 1
Map output records=0
Input split bytes=106
Spilled Records=0
Failed Shuffles=0
Merged Map outputs=0
GC time elapsed (ms)=0
CPU time spent (ms)=0
Physical memory (bytes) snapshot=0
Virtual memory (bytes) snapshot=0
Total committed heap usage (bytes)=333447168
File Input Format Counters
Bytes Read=0
File Output Format Counters
Bytes Written=0
15/02/09 09:52:06 INFO hadoop.xquery: Finished executing "./books1a.xq". Output path: "file:/home/oracle/exercises/OXH/books1"
<title lang="en">Everyday Italian</title>
<title lang="en">Harry Potter</title>
<title lang="en">XQuery Kick Start</title>
<title lang="en">Learning XML</title>
15/02/09 09:52:07 WARN fs.FileSystem: "local" is a deprecated filesystem name. Use "file://" instead.
15/02/09 09:52:08 INFO hadoop.xquery: OXH: Oracle XQuery for Hadoop 4.0.1 (build 4.0.1-cdh5.0.0-mr1 @mr2). Copyright (c) 2015, Oracle. All rights reserved.
15/02/09 09:52:08 INFO hadoop.xquery: Executing query "./books1b.xq". Output path: "file:/home/oracle/exercises/OXH/books2"
```

```
15/02/09 09:52:13 INFO mapreduce.Job: Job job_local1373727105_0001 completed successfully
15/02/09 09:52:13 INFO mapreduce.Job: Counters: 18
  File System Counters
    FILE: Number of bytes read=14727021
    FILE: Number of bytes written=15092499
    FILE: Number of read operations=0
    FILE: Number of large read operations=0
    FILE: Number of write operations=0
  Map-Reduce Framework
    Map input records=1
    Map output records=0
    Input split bytes=106
    Spilled Records=0
    Failed Shuffles=0
    Merged Map outputs=0
    GC time elapsed (ms)=0
    CPU time spent (ms)=0
    Physical memory (bytes) snapshot=0
    Virtual memory (bytes) snapshot=0
    Total committed heap usage (bytes)=333447168
  File Input Format Counters
    Bytes Read=0
  File Output Format Counters
    Bytes Written=0
15/02/09 09:52:13 INFO hadoop.xquery: Finished executing "./books1b.xq". Output path: "file:/home/oracle/exercises/0XH/books2"
<book category="CHILDREN">&xA; <title lang="en">Harry Potter</title>&xA; <author>J. K. Rowling</author>&xA; <year>2005</year>&xA; <price>29.99</price>&xA;</book>
[oracle@bigdatalite 0XH]$
```

```
[oracle@bigdatalite OXH]$ ls -ls
total 200
4 drwxr-xr-x. 2 oracle oinstall 4096 Feb  9 09:52 books1
4 -rw-r--r--. 1 oracle oinstall 175 Dec  3 08:45 books1a.xq
4 -rw-r--r--. 1 oracle oinstall 179 Dec  3 08:44 books1b.xq
4 -rwxrwxrwx. 1 oracle oinstall 210 Dec  3 08:44 books1-local.sh
4 -rwxrwxrwx. 1 oracle oinstall 174 Dec  3 08:44 books1.sh
4 drwxr-xr-x. 2 oracle oinstall 4096 Feb  9 09:52 books2
4 -rwxrwxrwx. 1 oracle oinstall 95 Dec  3 08:44 books2.sh
4 -rw-r--r--. 1 oracle oinstall 203 Dec  3 08:44 books2.xq
4 -rw-r--r--. 1 oracle oinstall 197 Dec  3 08:44 books3oh.sql
4 -rwxrwxrwx. 1 oracle oinstall 222 Dec  3 08:44 books3.sh
4 -rw-r--r--. 1 oracle oinstall 606 Dec  3 08:44 books3.xq
4 -rw-r--r--. 1 oracle oinstall 708 Dec  3 08:44 books-create.hql
4 -rw-r--r--. 1 oracle oinstall 29 Dec  3 08:45 books-drop.hql
4 -rw-r--r--. 1 oracle oinstall 137 Dec  3 08:44 books-hive.hql
4 -rw-r--r--. 1 oracle oinstall 849 Dec  3 08:44 books.xml
4 -rwxrwxrwx. 1 oracle oinstall 1055 Dec  3 08:45 cheat.sh
4 -rwxrwxrwx. 1 oracle oinstall 372 Dec  3 08:45 cleanup.sh
92 -rw-r--r--. 1 oracle oinstall 92187 Dec  3 08:44 crm_data.csv
4 -rwxrwxrwx. 1 oracle oinstall 66 Dec  3 08:44 hive_oxh.sh
4 -rw-r--r--. 1 oracle oinstall 2350 Dec  3 08:44 logging.properties
4 -rwxrwxrwx. 1 oracle oinstall 176 Dec  3 08:45 nosql.custsums.sh
4 -rw-r--r--. 1 oracle oinstall 515 Dec  3 08:44 nosql-custsums.xq
4 -rwxrwxrwx. 1 oracle oinstall 294 Dec  3 08:44 nosql.join.sh
4 -rw-r--r--. 1 oracle oinstall 888 Dec  3 08:44 nosql-join.xq
4 -rwxrwxrwx. 1 oracle oinstall 170 Dec  3 08:44 nosql.sums.sh
4 -rw-r--r--. 1 oracle oinstall 336 Dec  3 08:44 nosql-sums.xq
4 -rw-r--r--. 1 oracle oinstall 1020 Dec  3 08:45 OXH_Functions.sql
4 -rwxrwxrwx. 1 oracle oinstall 76 Dec  3 08:45 Viewdata.sh
[oracle@bigdatalite OXH]$ cd books1
[oracle@bigdatalite books1]$ ls
part-m-00000  SUCCESS
[oracle@bigdatalite books1]$ cat part-m-00000
<title lang="en">Everyday Italian</title>
<title lang="en">Harry Potter</title>
<title lang="en">XQuery Kick Start</title>
<title lang="en">Learning XML</title>
[oracle@bigdatalite books1]$
```

```
[oracle@bigdatalite 0XH]$ pwd
/home/oracle/exercises/0XH
[oracle@bigdatalite 0XH]$ ls -l
total 200
drwxr-xr-x. 2 oracle oinstall 4096 Feb  9 09:52 books1
-rw-r--r--. 1 oracle oinstall 175 Dec  3 08:45 books1a.xq
-rw-r--r--. 1 oracle oinstall 179 Dec  3 08:44 books1b.xq
-rwxrwxrwx. 1 oracle oinstall 210 Dec  3 08:44 books1-local.sh
-rwxrwxrwx. 1 oracle oinstall 174 Dec  3 08:44 books1.sh
drwxr-xr-x. 2 oracle oinstall 4096 Feb  9 09:52 books2
-rwxrwxrwx. 1 oracle oinstall  95 Dec  3 08:44 books2.sh
-rw-r--r--. 1 oracle oinstall 203 Dec  3 08:44 books2.xq
-rw-r--r--. 1 oracle oinstall 197 Dec  3 08:44 books3oh.sql
-rwxrwxrwx. 1 oracle oinstall 222 Dec  3 08:44 books3.sh
-rw-r--r--. 1 oracle oinstall 606 Dec  3 08:44 books3.xq
-rw-r--r--. 1 oracle oinstall 708 Dec  3 08:44 books-create.hql
-rw-r--r--. 1 oracle oinstall  29 Dec  3 08:45 books-drop.hql
-rw-r--r--. 1 oracle oinstall 137 Dec  3 08:44 books-hive.hql
-rw-r--r--. 1 oracle oinstall 849 Dec  3 08:44 books.xml
-rwxrwxrwx. 1 oracle oinstall 1055 Dec  3 08:45 cheat.sh
-rwxrwxrwx. 1 oracle oinstall 372 Dec  3 08:45 cleanup.sh
-rw-r--r--. 1 oracle oinstall 92187 Dec  3 08:44 crm_data.csv
-rwxrwxrwx. 1 oracle oinstall  66 Dec  3 08:44 hive_oxh.sh
-rw-r--r--. 1 oracle oinstall 2350 Dec  3 08:44 logging.properties
-rwxrwxrwx. 1 oracle oinstall 176 Dec  3 08:45 nosql.custsums.sh
-rw-r--r--. 1 oracle oinstall 515 Dec  3 08:44 nosql-custsums.xq
-rwxrwxrwx. 1 oracle oinstall 294 Dec  3 08:44 nosql.join.sh
-rw-r--r--. 1 oracle oinstall 888 Dec  3 08:44 nosql-join.xq
-rwxrwxrwx. 1 oracle oinstall 170 Dec  3 08:44 nosql.sums.sh
-rw-r--r--. 1 oracle oinstall 336 Dec  3 08:44 nosql-sums.xq
-rw-r--r--. 1 oracle oinstall 1020 Dec  3 08:45 OXH_Functions.sql
-rwxrwxrwx. 1 oracle oinstall  76 Dec  3 08:45 Viewdata.sh
[oracle@bigdatalite 0XH]$ cd books2
[oracle@bigdatalite books2]$ ls -l
total 4
-rw-r--r--. 1 oracle oinstall 170 Feb  9 09:52 part-m-00000
-rw-r--r--. 1 oracle oinstall  0 Feb  9 09:52 _SUCCESS
[oracle@bigdatalite books2]$ cat part-m-00000
<book category="CHILDREN">&#xA; <title lang="en">Harry Potter</title>&#xA; <author>J K. Rowling</author>&#xA; <year>2005</year>&#xA; <price>29.99</price>&#xA;</book>
[oracle@bigdatalite books2]$
```

#### 8. View the contents of the books1.sh file.

```
cat books1.sh
```

```
[oracle@bigdatalite 0XH]$ cat books1.sh
hadoop fs -rm -r books1 books2

hadoop jar $OXH_HOME/lib/oxh.jar ./books1a.xq -print -output ./books1

hadoop jar $OXH_HOME/lib/oxh.jar ./books1b.xq -print -output ./books2

[oracle@bigdatalite 0XH]$
```

9. Copy the books.xml file to the Hadoop HDFS file system and confirm that it is copied to HDFS.

```
hadoop fs -put books.xml  
hadoop fs -ls books.xml
```

```
[oracle@bigdatalite 0XH]$ hadoop fs -put books.xml  
[oracle@bigdatalite 0XH]$ hadoop fs -ls books.xml  
Found 1 items  
-rw-r--r-- 1 oracle oracle 849 2015-02-09 07:32 books.xml  
[oracle@bigdatalite 0XH]$ █
```

10. Run the books1.sh script.

```
./books1.sh
```

The partial output is as follows:

```
[oracle@bigdatalite 0XH]$ ./book1.sh  
bash: ./book1.sh: No such file or directory  
[oracle@bigdatalite 0XH]$  
[oracle@bigdatalite 0XH]$  
[oracle@bigdatalite 0XH]$  
[oracle@bigdatalite 0XH]$ ./books1.sh  
rm: `books1': No such file or directory  
rm: `books2': No such file or directory  
15/02/09 10:00:43 INFO hadoop.xquery: 0XH: Oracle XQuery for Hadoop 4.0.1 (build  
4.0.1-cdh5.0.0-mr1 @mr2). Copyright (c) 2015, Oracle. All rights reserved.  
15/02/09 10:00:43 INFO hadoop.xquery: Executing query "./books1a.xq". Output pat  
h: "hdfs://bigdatalite.localdomain:8020/user/oracle/books1"  
15/02/09 10:00:45 INFO hadoop.xquery: Submitting map-reduce job "oxh:books1a.xq#  
0" id="9a1fd547-334b-4967-baf0-8aa861e89bce.0", inputs=[hdfs://bigdatalite.local  
domain:8020/user/oracle/books.xml], output=hdfs://bigdatalite.localdomain:8020/u  
ser/oracle/books1  
15/02/09 10:00:45 INFO client.RMProxy: Connecting to ResourceManager at localhos  
t/127.0.0.1:8032  
15/02/09 10:00:47 INFO input.FileInputFormat: Total input paths to process : 1  
15/02/09 10:00:47 INFO mapreduce.JobSubmitter: number of splits:1  
15/02/09 10:00:47 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_14  
23174990155_0001
```

You can click the following link in your terminal window, which will have a different job id number, to view details about the completed OXH job. Alternatively, you can use the Job History server to view the details about this completed job.

```
15/02/09 10:01:12 INFO impl.YarnClientImpl: Submitted application application_14  
23174990155_0002  
15/02/09 10:01:12 INFO mapreduce.Job: The url to track the job: http://bigdatali  
te.localdomain:8088/proxy/application_1423174990155_0002/  
15/02/09 10:01:12 INFO hadoop.xquery: Waiting for map-reduce job oxh:books1b.xq#  
0
```

The screenshot shows the Hadoop JobHistory interface. On the left, there's a sidebar with links for Application (About Jobs), Tools, and a search bar. The main area is titled "Retired Jobs" and displays a table of completed jobs. The table has columns for Submit Time, Start Time, Finish Time, Job ID, Name, User, Queue, and State. Two rows are visible:

Submit Time	Start Time	Finish Time	Job ID	Name	User	Queue	State
2015.02.09 10:01:12 EST	2015.02.09 10:01:18 EST	2015.02.09 10:01:26 EST	job_1423174990155_0002	oxh:books1b.xq#0	oracle	root.oracle	SUCCEEDED
2015.02.09 10:00:48 EST	2015.02.09 10:00:55 EST	2015.02.09 10:01:03 EST	job_1423174990155_0001	oxh:books1a.xq#0	oracle	root.oracle	SUCCEEDED

```
CPU time spent (ms)=3120
Physical memory (bytes) snapshot=190992384
Virtual memory (bytes) snapshot=688893952
Total committed heap usage (bytes)=186646528
File Input Format Counters
Bytes Read=0
File Output Format Counters
Bytes Written=0
15/02/09 10:01:05 INFO hadoop.xquery: Finished executing "./books1a.xq". Output path: "hdfs://bigdatalite.localdomain:8020/user/oracle/books1"
<title lang="en">Everyday Italian</title>
<title lang="en">Harry Potter</title>
<title lang="en">XQuery Kick Start</title>
<title lang="en">Learning XML</title>
15/02/09 10:01:07 INFO hadoop.xquery: OXH: Oracle XQuery for Hadoop 4.0.1 (build 4.0.1-cdh5.0.0-mr1 @mr2). Copyright (c) 2015, Oracle. All rights reserved.
15/02/09 10:01:08 INFO hadoop.xquery: Executing query "./books1b.xq". Output path: "hdfs://bigdatalite.localdomain:8020/user/oracle/books2"
15/02/09 10:01:10 INFO hadoop.xquery: Submitting map-reduce job "oxh:books1b.xq#0" id="67c976cd-932a-47f2-9403-e73b9f05alec.0", inputs=[hdfs://bigdatalite.localdomain:8020/user/oracle/books.xml], output=hdfs://bigdatalite.localdomain:8020/user/oracle/books2
15/02/09 10:01:10 INFO client.RMProxy: Connecting to ResourceManager at localhost/127.0.0.1:8032
15/02/09 10:01:12 INFO input.FileInputFormat: Total input paths to process : 1
15/02/09 10:01:12 INFO mapreduce.JobSubmitter: number of splits:1
15/02/09 10:01:12 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1423174990155_0002
15/02/09 10:01:12 INFO impl.YarnClientImpl: Submitted application application_1423174990155_0002
15/02/09 10:01:12 INFO mapreduce.Job: The url to track the job: http://bigdatalite.localdomain:8020/joinhistory?jobid=job_1423174990155_0002
```

```
HDFS: Number of write operations=2
Job Counters
    Launched map tasks=1
    Data-local map tasks=1
    Total time spent by all maps in occupied slots (ms)=5829
    Total time spent by all reduces in occupied slots (ms)=0
    Total time spent by all map tasks (ms)=5829
    Total vcore-seconds taken by all map tasks=5829
    Total megabyte-seconds taken by all map tasks=1492224
Map-Reduce Framework
    Map input records=1
    Map output records=0
    Input split bytes=122
    Spilled Records=0
    Failed Shuffles=0
    Merged Map outputs=0
    GC time elapsed (ms)=120
    CPU time spent (ms)=3230
    Physical memory (bytes) snapshot=191066112
    Virtual memory (bytes) snapshot=694857728
    Total committed heap usage (bytes)=189792256
File Input Format Counters
    Bytes Read=0
File Output Format Counters
    Bytes Written=0
15/02/09 10:01:28 INFO hadoop.xquery: Finished executing "./books1b.xq". Output
path: "hdfs://bigdatalite.localdomain:8020/user/oracle/books2"
<book category="CHILDREN">&#xA;    <title lang="en">Harry Potter</title>&#xA;    <au
thor>J K. Rowling</author>&#xA;    <year>2005</year>&#xA;    <price>29.99</price>&#x
A;</book>
[oracle@bigdatalite 0XH]$
```

11. Filter by price. Display only the books that cost less than \$30. Review the contents of the books2.xq script.
12. Run the books2.sh script.

```
./books2.sh
```

```
[oracle@bigdatalite OXH]$ ./books2.sh
rm: `books3': No such file or directory
15/02/09 10:19:45 INFO hadoop.xquery: OXH: Oracle XQuery for Hadoop 4.0.1 (build
4.0.1-cdh5.0.0-mr1 @mr2). Copyright (c) 2015, Oracle. All rights reserved.
15/02/09 10:19:45 INFO hadoop.xquery: Executing query "./books2.xq". Output path
: "hdfs://bigdatalite.localdomain:8020/user/oracle/books3"
15/02/09 10:19:47 INFO hadoop.xquery: Submitting map-reduce job "oxh:books2.xq#0"
" id="956449c6-8b71-4231-a142-ab5c070b5a78.0", inputs=[hdfs://bigdatalite.localdomain:8020/user/oracle/books.xml], output=hdfs://bigdatalite.localdomain:8020/user/oracle/books3
15/02/09 10:19:47 INFO client.RMProxy: Connecting to ResourceManager at localhos
t/127.0.0.1:8032
15/02/09 10:19:48 INFO input.FileInputFormat: Total input paths to process : 1
15/02/09 10:19:48 INFO mapreduce.JobSubmitter: number of splits:1
15/02/09 10:19:49 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_14
23174990155_0003
15/02/09 10:19:49 INFO impl.YarnClientImpl: Submitted application application_14
23174990155_0003
15/02/09 10:19:49 INFO mapreduce.Job: The url to track the job: http://bigdatali
te.localdomain:8088/proxy/application_1423174990155_0003/
15/02/09 10:19:49 INFO hadoop.xquery: Waiting for map-reduce job oxh:books2.xq#0
15/02/09 10:19:49 INFO mapreduce.Job: Running job: job_1423174990155_0003
15/02/09 10:19:56 INFO mapreduce.Job: Job job_1423174990155_0003 running in uber
mode : false
15/02/09 10:19:56 INFO mapreduce.Job: map 0% reduce 0%
```

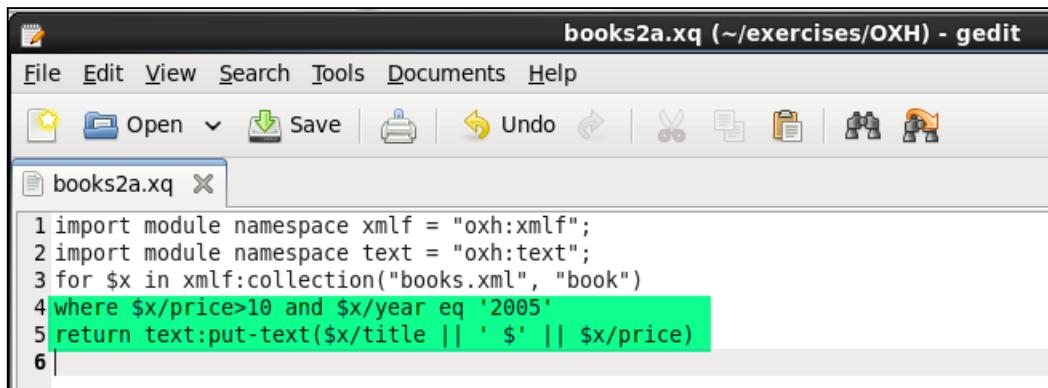
```
HDFS: Number of read operations=5
HDFS: Number of large read operations=0
HDFS: Number of write operations=2
Job Counters
    Launched map tasks=1
    Data-local map tasks=1
    Total time spent by all maps in occupied slots (ms)=5601
    Total time spent by all reduces in occupied slots (ms)=0
    Total time spent by all map tasks (ms)=5601
    Total vcore-seconds taken by all map tasks=5601
    Total megabyte-seconds taken by all map tasks=1433856
Map-Reduce Framework
    Map input records=4
    Map output records=0
    Input split bytes=122
    Spilled Records=0
    Failed Shuffles=0
    Merged Map outputs=0
    GC time elapsed (ms)=135
    CPU time spent (ms)=3200
    Physical memory (bytes) snapshot=190791680
    Virtual memory (bytes) snapshot=692408320
    Total committed heap usage (bytes)=189792256
File Input Format Counters
    Bytes Read=849
File Output Format Counters
    Bytes Written=0
15/02/09 10:20:04 INFO hadoop.xquery: Finished executing "./books2.xq". Output p
ath: "hdfs://bigdatalite.localdomain:8020/user/oracle/books3"
Harry Potter $29.99
[oracle@bigdatalite OXH]$
```

13. Add year to the filter. Display only the books that cost more than \$10 and that were published in 2005. Open the books2a.xq script and review the code.

- Edit the books2.xq script and review the code.
- Replace the where and return clauses as follows:

```
import module namespace xmlf = "oxh:xmlf";
import module namespace text = "oxh:text";
for $x in xmlf:collection("books.xml", "book")
where $x/price>10 and $x/year eq '2005'
return text:put-text($x/title || ' ' || $x/price)
```

- Save the file as books2a.xq in the OXH folder, and then exit gedit.

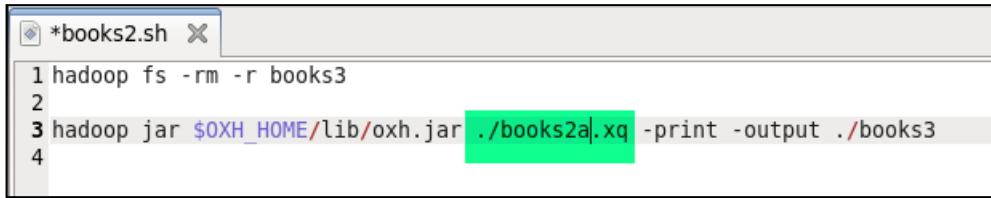


- View the new books2a.xq file.

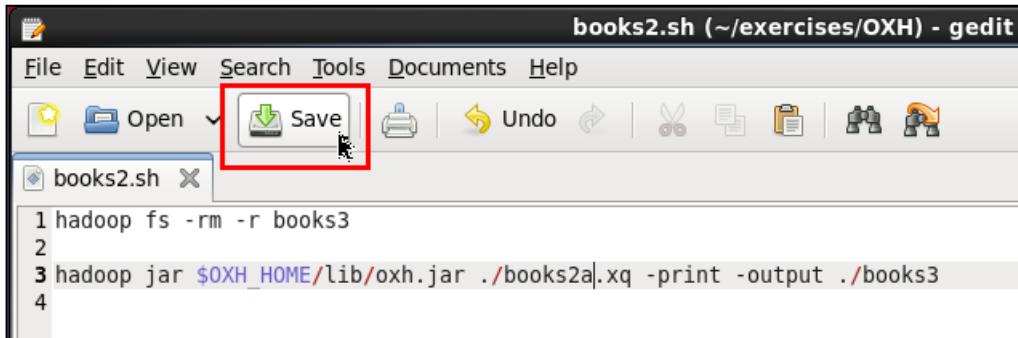
```
[oracle@bigdatalite OXH]$ ls -l
total 204
drwxr-xr-x. 2 oracle oinstall 4096 Feb  9 09:52 books1
-rw-r--r--. 1 oracle oinstall 175 Dec  3 08:45 books1a.xq
-rw-r--r--. 1 oracle oinstall 179 Dec  3 08:44 books1b.xq
-rwxrwxrwx. 1 oracle oinstall 210 Dec  3 08:44 books1-local.sh
-rwxrwxrwx. 1 oracle oinstall 174 Dec  3 08:44 books1.sh
drwxr-xr-x. 2 oracle oinstall 4096 Feb  9 09:52 books2
-rw-r--r--. 1 oracle oinstall 259 Feb  9 11:14 books2a.xq
-rwxrwxrwx. 1 oracle oinstall  95 Dec  3 08:44 books2.sh
-rw-r--r--. 1 oracle oinstall 203 Dec  3 08:44 books2.xq
-rw-r--r--. 1 oracle oinstall 197 Dec  3 08:44 books3oh.sql
```

```
[oracle@bigdatalite OXH]$ cat books2a.xq
import module namespace xmlf = "oxh:xmlf";
import module namespace text = "oxh:text";
for $x in xmlf:collection("books.xml", "book")
where $x/price>10 and $x/year eq '2005'
return text:put-text($x/title || ' ' || $x/price)
```

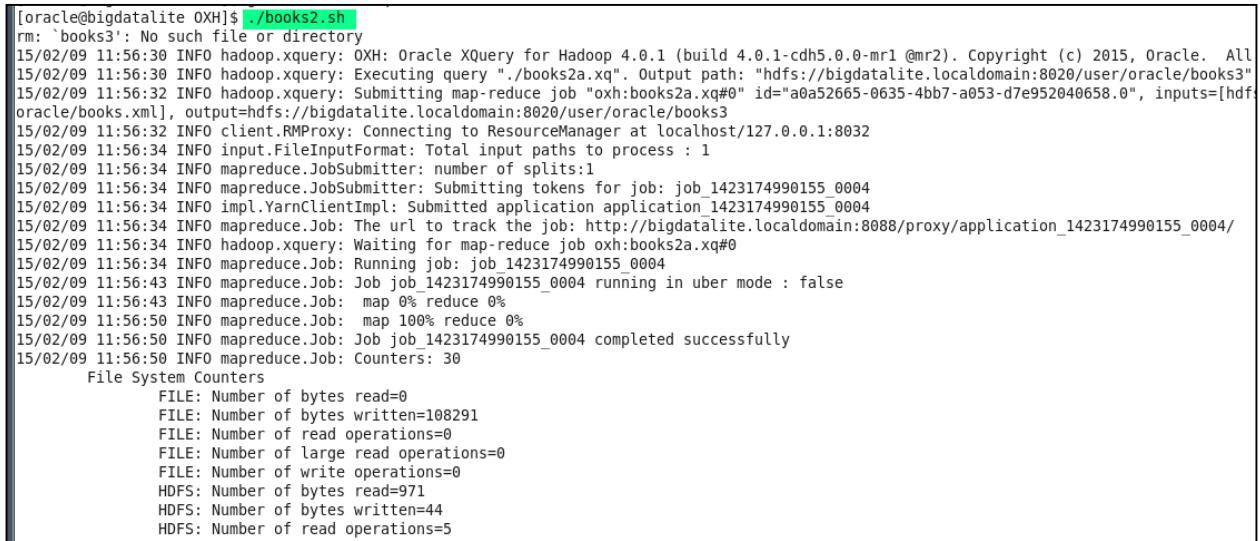
14. Open books2.sh by using gedit, replace books2.xq with books2a.xq, save your changes, exit gedit, and then run the script.



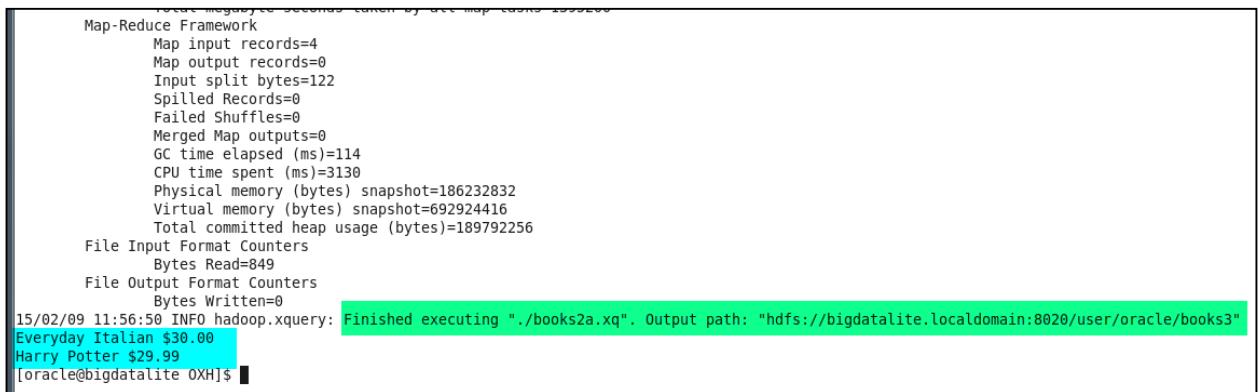
```
1 hadoop fs -rm -r books3
2
3 hadoop jar $OXH_HOME/lib/oxh.jar ./books2a.xq -print -output ./books3
4
```



```
./books2.sh
```



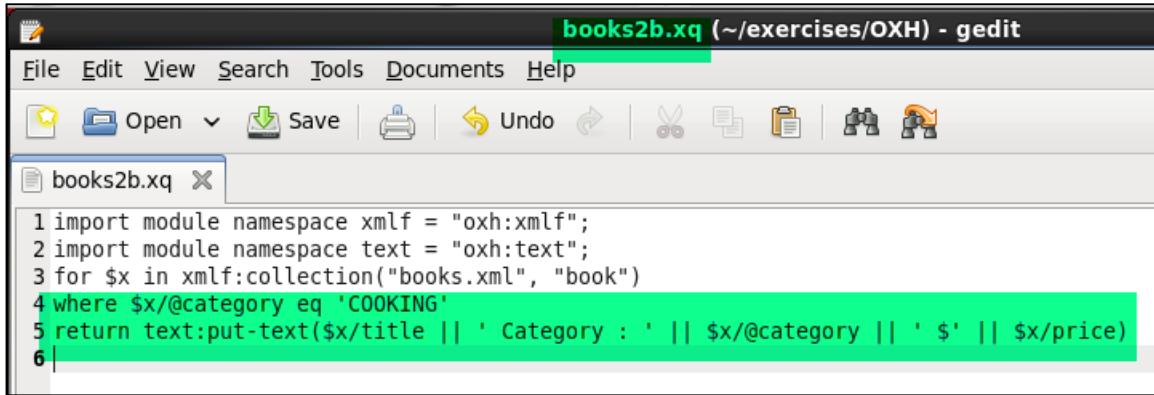
```
[oracle@bigdatalite OXH]$ ./books2.sh
rm: 'books3': No such file or directory
15/02/09 11:56:30 INFO hadoop.xquery: OXH: Oracle XQuery for Hadoop 4.0.1 (build 4.0.1-cdh5.0.0-mr1 @mr2). Copyright (c) 2015, Oracle. All rights reserved.
15/02/09 11:56:30 INFO hadoop.xquery: Executing query "./books2a.xq". Output path: "hdfs://bigdatalite.localdomain:8020/user/oracle/books3"
15/02/09 11:56:32 INFO hadoop.xquery: Submitting map-reduce job "oxh:books2a.xq#0" id="a0a52665-0635-4bb7-a053-d7e952040658.0", inputs=[hdfs://bigdatalite.localdomain:8020/user/oracle/books.xml], output=hdfs://bigdatalite.localdomain:8020/user/oracle/books3
15/02/09 11:56:32 INFO client.RMProxy: Connecting to ResourceManager at localhost/127.0.0.1:8032
15/02/09 11:56:34 INFO input.FileInputFormat: Total input paths to process : 1
15/02/09 11:56:34 INFO mapreduce.JobSubmitter: number of splits:1
15/02/09 11:56:34 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1423174990155_0004
15/02/09 11:56:34 INFO impl.YarnClientImpl: Submitted application application_1423174990155_0004
15/02/09 11:56:34 INFO mapreduce.Job: The url to track the job: http://bigdatalite.localdomain:8088/proxy/application_1423174990155_0004/
15/02/09 11:56:34 INFO hadoop.xquery: Waiting for map-reduce job oxh:books2a.xq#0
15/02/09 11:56:43 INFO mapreduce.Job: Running job: job_1423174990155_0004
15/02/09 11:56:43 INFO mapreduce.Job: Job job_1423174990155_0004 running in uber mode : false
15/02/09 11:56:43 INFO mapreduce.Job: map 0% reduce 0%
15/02/09 11:56:50 INFO mapreduce.Job: map 100% reduce 0%
15/02/09 11:56:50 INFO mapreduce.Job: Job job_1423174990155_0004 completed successfully
15/02/09 11:56:50 INFO mapreduce.Job: Counters: 30
File System Counters
FILE: Number of bytes read=0
FILE: Number of bytes written=108291
FILE: Number of read operations=0
FILE: Number of large read operations=0
FILE: Number of write operations=0
HDFS: Number of bytes read=971
HDFS: Number of bytes written=44
HDFS: Number of read operations=5
```



```
Map-Reduce Framework
Map input records=4
Map output records=0
Input split bytes=122
Spilled Records=0
Failed Shuffles=0
Merged Map outputs=0
GC time elapsed (ms)=114
CPU time spent (ms)=3130
Physical memory (bytes) snapshot=186232832
Virtual memory (bytes) snapshot=692924416
Total committed heap usage (bytes)=189792256
File Input Format Counters
Bytes Read=849
File Output Format Counters
Bytes Written=0
15/02/09 11:56:50 INFO hadoop.xquery: Finished executing "./books2a.xq". Output path: "hdfs://bigdatalite.localdomain:8020/user/oracle/books3"
Everyday Italian $30.00
Harry Potter $29.99
[oracle@bigdatalite OXH]$
```

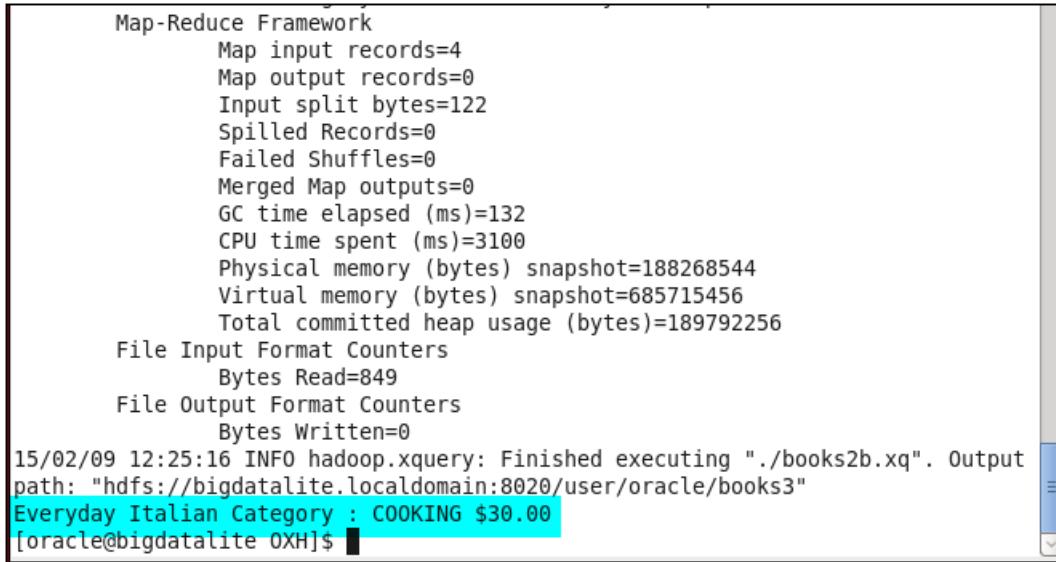
15. Filter by the category “COOKING”. Edit books2a.xq script, and change the code as follows. Save the file as books2b.xq in the OXH folder, and then exit gedit.

```
import module namespace xmlf = "oxh:xmlf";
import module namespace text = "oxh:text";
for $x in xmlf:collection("books.xml", "book")
where $x/@category eq 'COOKING'
return text:put-text($x/title || ' Category : ' || $x/@category
|| ' $' || $x/price)
```



16. Open books2.sh and replace books2a.xq with books2b.xq and run the script.

```
./books2.sh
```



## Practice 13-2: Working with Hive UDF and SerDe on XML Data

### Overview

In this practice, you work with Hive UDF and SerDe on XML data.

### Assumptions

**Note:** The scripts for this practice are in the /home/oracle/exercises/OXH folder.

### Tasks

1. Open a terminal window.
2. Enter the following command at the command prompt and then press **Enter**.

```
cd /home/oracle/exercises/OXH
```

3. View the contents of the `hive_oxh.sh` file.

```
cat hive_oxh.sh
```

4. Run the `hive_oxh.sh` file.

```
./hive_oxh.sh
```

**Note:** If you get issues while running the file, open and run the code one by one in the terminal window.

5. View the `CREATE TABLE` statement that will transform the `books.xml` file into a tabular structure that can be queried in Hive. You can use the `books-create.hql` script to create the table.

```
cd /home/oracle/exercises/OXH  
cat books-create.hql
```

6. Create the bookstore table and load the data. You can use the `books-create.hql` script.

```
cd /home/oracle/exercises/OXH  
./hive_oxh.sh -f books-create.hql
```

7. Query the XML file in Hive. Enter the following command at the Hive terminal and press Enter.

```
select title, category, numauthors, lang, year, price  
from bookstore;
```

8. Run the following SELECT statements.

```
select title, auth from bookstore
lateral view
explode(xml_query("AUTHORS/author", authors)) authtab as auth;

select title, auth from bookstore
lateral view
explode(xml_query("AUTHORS/author", authors)) authtab as auth
where auth like 'James%';
select title, authors from bookstore;
```

## Solution 13-2: Working with Hive UDF and SerDe on XML Data

### Overview

In this practice solution, you work with Hive UDF and SerDe on XML data.

### Assumptions

**Note:** The scripts for this practice are in the /home/oracle/exercises/OXH folder.

### Steps

1. Open a terminal window.



2. Enter the following command at the command prompt, and then press **Enter**.

```
cd /home/oracle/exercises/OXH
```

```
[oracle@bigdatalite ~]$ cd /home/oracle/exercises/OXH  
[oracle@bigdatalite OXH]$
```

3. View the contents of the `hive_oxh.sh` file.

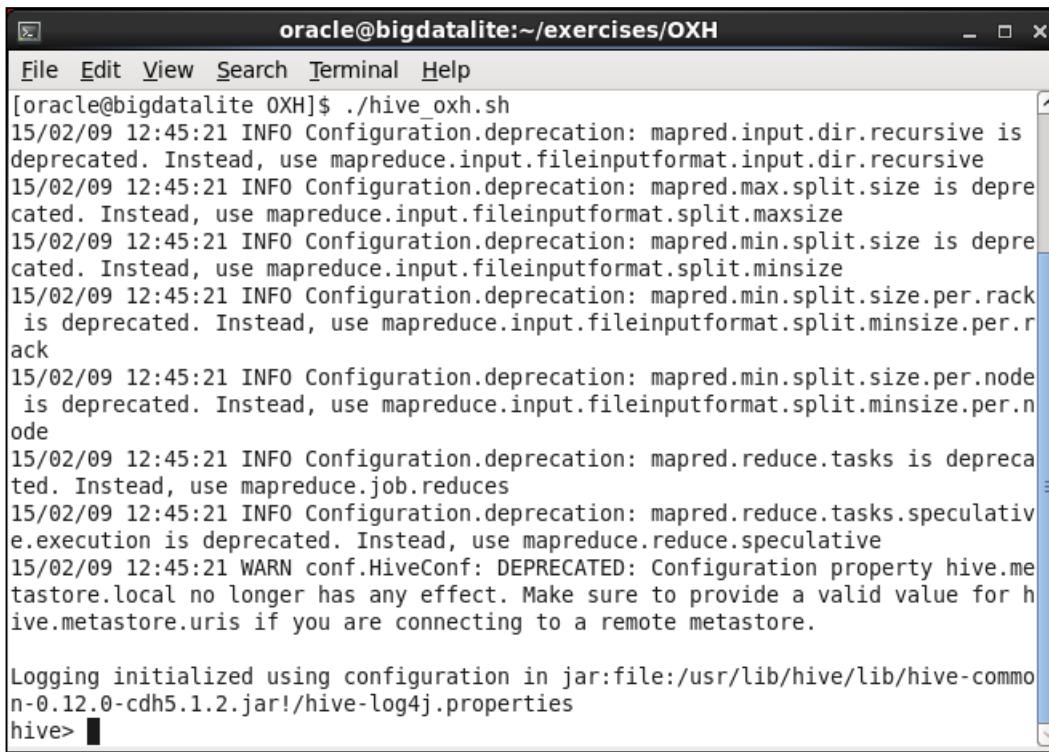
```
cat hive_oxh.sh
```

```
[oracle@bigdatalite OXH]$ cat hive_oxh.sh  
hive --auxpath $OXH_HOME/hive/lib -i $OXH_HOME/hive/init.sql $@  
[oracle@bigdatalite OXH]$
```

4. Run the `hive_oxh.sh` file.

```
./hive_oxh.sh
```

**Note:** If you get issues while running the file, open and run the code one by one in the terminal window.



A screenshot of a terminal window titled "oracle@bigdatalite:~/exercises/OXH". The window displays the output of the command "../hive\_oxh.sh". The log shows several deprecation warnings from Hive, such as "Configuration.deprecation: mapred.input.dir.recursive is deprecated. Instead, use mapreduce.input.fileinputformat.input.dir.recursive", and a warning about the "hive.metastore.local" property being deprecated. It also mentions the logging configuration file used.

```
[oracle@bigdatalite OXH]$ ../hive_oxh.sh
15/02/09 12:45:21 INFO Configuration.deprecation: mapred.input.dir.recursive is
deprecated. Instead, use mapreduce.input.fileinputformat.input.dir.recursive
15/02/09 12:45:21 INFO Configuration.deprecation: mapred.max.split.size is depre
cated. Instead, use mapreduce.input.fileinputformat.split.maxsize
15/02/09 12:45:21 INFO Configuration.deprecation: mapred.min.split.size is depre
cated. Instead, use mapreduce.input.fileinputformat.split.minsize
15/02/09 12:45:21 INFO Configuration.deprecation: mapred.min.split.size.per.rack
is deprecated. Instead, use mapreduce.input.fileinputformat.split.minsize.per.r
ack
15/02/09 12:45:21 INFO Configuration.deprecation: mapred.min.split.size.per.node
is deprecated. Instead, use mapreduce.input.fileinputformat.split.minsize.per.n
ode
15/02/09 12:45:21 INFO Configuration.deprecation: mapred.reduce.tasks is depreca
ted. Instead, use mapreduce.job.reduces
15/02/09 12:45:21 INFO Configuration.deprecation: mapred.reduce.tasks.speculativ
e.execution is deprecated. Instead, use mapreduce.reduce.speculative
15/02/09 12:45:21 WARN conf.HiveConf: DEPRECATED: Configuration property hive.me
tastore.local no longer has any effect. Make sure to provide a valid value for h
ive.metastore.uris if you are connecting to a remote metastore.

Logging initialized using configuration in jar:file:/usr/lib/hive/lib/hive-commo
n-0.12.0-cdh5.1.2.jar!/hive-log4j.properties
hive> ■
```

5. View the CREATE TABLE statement that will transform the books.xml file into a tabular structure that can be queried in Hive. You can use the books-create.hql script to create the table. Open a new terminal window and enter the following commands:

```
cd /home/oracle/exercises/OXH
cat books-create.hql
```

```
oracle@bigdatalite:~/exercises/OXH
File Edit View Search Terminal Help
[oracle@bigdatalite ~]$ cd /home/oracle/exercises/OXH
[oracle@bigdatalite OXH]$ cat books-create.hql
CREATE TABLE bookstore (title STRING, category STRING, lang STRING, year INT, au
thors STRING, numauthors INT, price FLOAT)
ROW FORMAT
SERDE 'oracle.hadoop.xquery.hive.OXMLSerDe'
STORED AS
INPUTFORMAT 'oracle.hadoop.xquery.hive.OXMLInputFormat'
OUTPUTFORMAT 'oracle.hadoop.xquery.hive.OXMLOutputFormat'
TBLPROPERTIES(
"oxh-elements" = "book",
"oxh-column.title" = "./title",
"oxh-column.category" = "./@category",
"oxh-column.lang" = "./title/@lang",
"oxh-column.year" = "./year",
"oxh-column.authors" = "fn:serialize(<AUTHORS>{./author}</AUTHORS>)",
"oxh-column.numauthors" = "fn:count(.//author)",
"oxh-column.price" = "./price"
);

LOAD DATA LOCAL INPATH 'books.xml' OVERWRITE INTO TABLE bookstore;

exit;
[oracle@bigdatalite OXH]$
```

**Note:** In books.xml, the array of authors is serialized into its own XML string.

The function fn:serialize(<AUTHORS>{ ./author }</AUTHORS>)

converts:

```
<book category="WEB">
<title lang="en">XQuery Kick Start</title>
<author>James McGovern</author>
<author>Per Bothner</author>
<author>Kurt Cagle</author>
<author>James Linn</author>
<author>Vaidyanathan Nagarajan</author>
<year>2003</year>
<price>49.99</price>
</book>
```

into:

```
<AUTHORS><author>James McGovern</author><author>Per
Bothner</author><author>Kurt
Cagle</author><author>James Linn</author><author>Vaidyanathan
Nagarajan</author></AUTHORS>
```

You can also count the number of author per book with fn:count (.//author).

6. Create the bookstore table and load the data. You can use the books-create.hql script.

```
cd /home/oracle/exercises/OXH  
./hive_oxh.sh -f books-create.hql
```

```
[oracle@bigdatalite OXH]$ cd /home/oracle/exercises/OXH  
[oracle@bigdatalite OXH]$ ./hive_oxh.sh -f books-create.hql  
15/02/09 14:13:10 INFO Configuration.deprecation: mapred.input.dir.recursive is  
deprecated. Instead, use mapreduce.input.fileinputformat.input.dir.recursive  
15/02/09 14:13:10 INFO Configuration.deprecation: mapred.max.split.size is depre-  
cated. Instead, use mapreduce.input.fileinputformat.split.maxsize  
15/02/09 14:13:10 INFO Configuration.deprecation: mapred.min.split.size is depre-  
cated. Instead, use mapreduce.input.fileinputformat.split.minsize  
15/02/09 14:13:10 INFO Configuration.deprecation: mapred.min.split.size.per.rack  
is deprecated. Instead, use mapreduce.input.fileinputformat.split.minsize.per.r  
ack  
15/02/09 14:13:10 INFO Configuration.deprecation: mapred.min.split.size.per.node  
is deprecated. Instead, use mapreduce.input.fileinputformat.split.minsize.per.n  
ode  
15/02/09 14:13:10 INFO Configuration.deprecation: mapred.reduce.tasks is depre-  
cated. Instead, use mapreduce.job.reduces  
15/02/09 14:13:10 INFO Configuration.deprecation: mapred.reduce.tasks.speculativ  
e.execution is deprecated. Instead, use mapreduce.reduce.speculative  
15/02/09 14:13:10 WARN conf.HiveConf: DEPRECATED: Configuration property hive.me  
tastore.local no longer has any effect. Make sure to provide a valid value for h  
ive.metastore.uris if you are connecting to a remote metastore.  
  
Logging initialized using configuration in jar:file:/usr/lib/hive/lib/hive-commo  
n-0.12.0-cdh5.1.2.jar!/hive-log4j.properties  
OK  
Time taken: 3.167 seconds  
Copying data from file:/home/oracle/exercises/OXH/books.xml  
Copying file: file:/home/oracle/exercises/OXH/books.xml  
Loading data to table default.bookstore  
chgrp: changing ownership of '/user/hive/warehouse/bookstore': User does not bel  
ong to hive  
OK  
Time taken: 1.529 seconds  
[oracle@bigdatalite OXH]$
```

7. Query the XML file in Hive. Enter the following command at the Hive terminal and press Enter.

```
select title, category, numauthors, lang, year, price  
from bookstore;
```

The output is displayed in the following partial screen capture.

```
at org.apache.hadoop.hive.cli.CliDriver.run(CliDriver.java:684)
at org.apache.hadoop.hive.cli.CliDriver.main(CliDriver.java:623)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.
java:57)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAcces-
sorImpl.java:43)
at java.lang.reflect.Method.invoke(Method.java:606)
at org.apache.hadoop.util.RunJar.main(RunJar.java:212)
FAILED: ParseException line 1:0 cannot recognize input near 'cd' '/' 'home'
hive> select title, category, numauthors, lang, year, price
   > from bookstore;
Total MapReduce jobs = 1
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1423174990155_0008, Tracking URL = http://bigdatalite.localdo-
main:8088/proxy/application_1423174990155_0008/
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1423174990155_0008
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2015-02-09 14:14:52,878 Stage-1 map = 0%,  reduce = 0%
2015-02-09 14:15:01,298 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 3.48 se-
c
2015-02-09 14:15:02,332 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 3.48 se-
c
MapReduce Total cumulative CPU time: 3 seconds 480 msec
Ended Job = job_1423174990155_0008
MapReduce Jobs Launched:
Job 0: Map: 1  Cumulative CPU: 3.48 sec  HDFS Read: 1079 HDFS Write: 149 SUCC-
SS
Total MapReduce CPU Time Spent: 3 seconds 480 msec
OK
Everyday Italian      COOKING 1      en      2005      30.0
Harry Potter CHILDREN 1      en      2005      29.99
XQuery Kick Start    WEB      5      en      2003      49.99
Learning XML          WEB      1      en      2003      39.95
Time taken: 22.1 seconds, Fetched: 4 row(s)
hive>
```

8. Run the following SELECT statements.

```
select title, auth from bookstore
lateral view
explode(xml_query("AUTHORS/author", authors)) authtab as auth;
```

```
hive> select title, auth from bookstore
    > lateral view
    > explode(xml_query("AUTHORS/author", authors)) authtab as auth;
Total MapReduce jobs = 1
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1423174990155_0009, Tracking URL = http://bigdatalite.localdo
main:8088/proxy/application_1423174990155_0009/
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1423174990155_0009
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2015-02-09 14:18:20,878 Stage-1 map = 0%, reduce = 0%
2015-02-09 14:18:30,286 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 4.32 se
c
2015-02-09 14:18:31,321 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 4.32 se
c
MapReduce Total cumulative CPU time: 4 seconds 320 msec
Ended Job = job_1423174990155_0009
MapReduce Jobs Launched:
Job 0: Map: 1   Cumulative CPU: 4.32 sec   HDFS Read: 1079 HDFS Write: 250 SUCCE
SS
Total MapReduce CPU Time Spent: 4 seconds 320 msec
OK
Everyday Italian      Giada De Laurentiis
Harry Potter      J K. Rowling
XQuery Kick Start      James McGovern
XQuery Kick Start      Per Bothner
XQuery Kick Start      Kurt Cagle
XQuery Kick Start      James Linn
XQuery Kick Start      Vaidyanathan Nagarajan
Learning XML      Erik T. Ray
Time taken: 18.653 seconds, Fetched: 8 row(s)
hive> ■
```

```
select title, auth from bookstore
lateral view
explode(xml_query("AUTHORS/author", authors)) authtab as auth
where auth like 'James%';
select title, authors from bookstore;
```

```
hive> select title, auth from bookstore
  > lateral view
  > explode(xml_query("AUTHORS/author", authors)) authtab as auth
  > where auth like 'James%';
Total MapReduce jobs = 1
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1423174990155_0010, Tracking URL = http://bigdatalite.localdomain:8088/proxy/application_1423174990155_0010/
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1423174990155_0010
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2015-02-09 14:19:52,304 Stage-1 map = 0%,  reduce = 0%
2015-02-09 14:20:01,693 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 4.16 sec
2015-02-09 14:20:02,726 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 4.16 sec
MapReduce Total cumulative CPU time: 4 seconds 160 msec
Ended Job = job_1423174990155_0010
MapReduce Jobs Launched:
Job 0: Map: 1  Cumulative CPU: 4.16 sec  HDFS Read: 1079 HDFS Write: 62 SUCCESS
OK
XQuery Kick Start      James McGovern
XQuery Kick Start      James Linn
Time taken: 18.474 seconds, Fetched: 2 row(s)
hive> select title, authors from bookstore;
Total MapReduce jobs = 1
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1423174990155_0011, Tracking URL = http://bigdatalite.localdomain:8088/proxy/application_1423174990155_0011/
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1423174990155_0011
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2015-02-09 14:20:11,452 Stage-1 map = 0%,  reduce = 0%
2015-02-09 14:20:20,078 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 3.76 sec
2015-02-09 14:20:21,114 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 3.76 sec
```

```
hive> select title, authors from bookstore;
Total MapReduce jobs = 1
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1423174990155_0011, Tracking URL = http://bigdatalite.localdo
main:8088/proxy/application_1423174990155_0011/
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1423174990155_0011
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2015-02-09 14:20:11,452 Stage-1 map = 0%,  reduce = 0%
2015-02-09 14:20:20,078 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 3.76 se
c
2015-02-09 14:20:21,114 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 3.76 se
c
MapReduce Total cumulative CPU time: 3 seconds 760 msec
Ended Job = job_1423174990155_0011
MapReduce Jobs Launched:
Job 0: Map: 1   Cumulative CPU: 3.76 sec   HDFS Read: 1079 HDFS Write: 386 SUCCE
SS
Total MapReduce CPU Time Spent: 3 seconds 760 msec
OK
Everyday Italian      <AUTHORS><author>Giada De Laurentiis</author></AUTHORS>
Harry Potter       <AUTHORS><author>J K. Rowling</author></AUTHORS>
XQuery Kick Start    <AUTHORS><author>James McGovern</author><author>Per Both
ner</author><author>Kurt Cagle</author><author>James Linn</author><author>Vaidya
nathan Nagarajan</author></AUTHORS>
Learning XML        <AUTHORS><author>Erik T. Ray</author></AUTHORS>
Time taken: 18.373 seconds, Fetched: 4 row(s)
hive>
```

## Practice 13-3: Loading Results from an XQuery into an Oracle Database

### Overview

In this practice, you load the results of a transformation of the `books.xml` file into a tabular format using Oracle OXH XQuery for Hadoop into Oracle Database using Oracle Loader for Hadoop (OLH).

**Note:** The scripts for this practice are in the `/home/oracle/exercises/OXH` folder.

### Assumptions

### Tasks

1. Open a terminal window.
2. Enter the following command at the command prompt, and then press **Enter**.

```
cd /home/oracle/exercises/setup
```

3. Log in to SQL\*Plus as `sys/welcome1@orcl` as `sysdba` and run the `setup.sql` script.

```
$ sqlplus sys/welcome1@orcl as sysdba  
SQL>@setup.sql
```

4. Enter the following command at the command prompt, and then press **Enter**.

```
cd /home/oracle/exercises/OXH
```

5. Connect as `bda/welcome1@orcl`.

```
$ sqlplus bda/welcome1@orcl
```

6. Create the target table `books`. Run the `books3olh.sql` script.

```
@books3olh
```

7. View the file `books3.xq`. OXH includes an Oracle database adapter that uses a custom `put` command to publish XQuery results to an Oracle table. The Oracle adapter will take the data types in the Oracle table and map them to data in the `put` command.

```
cat books3.xq
```

8. Run the `books3.sh` script.

```
./books3.sh
```

9. Run the `./Viewdata.sh` script to view the contents of the `books` table.

```
./Viewdata.sh
```

## Solution 13-3: Loading Results from an XQuery into an Oracle Database

### Overview

In this solution, you load the results of a transformation of the `books.xml` file into a tabular format using Oracle OXH XQuery for Hadoop into Oracle Database using Oracle Loader for Hadoop (OLH).

**Note:** The scripts for this practice are in the `/home/oracle/exercises/OXH` folder.

### Steps

1. Open a terminal window.



2. Enter the following command at the command prompt, and then press **Enter**.

```
cd /home/oracle/exercises/setup
```

```
[oracle@bigdatalite ~]$ cd /home/oracle/exercises/setup
[oracle@bigdatalite setup]$ pwd
/home/oracle/exercises/setup
[oracle@bigdatalite setup]$ ls -ls
total 568
 4 -rw-r--r--. 1 oracle oinstall      626 Dec  3 08:48 deinstall.sh
 4 -rw-r--r--. 1 oracle oinstall     221 Dec  3 08:49 deinstall.sql
 4 -rw-r--r--. 1 oracle oinstall     201 Dec  3 08:49 HOLStart.desktop
 8 -rw-r--r--. 1 oracle oinstall    6573 Dec  3 08:49 HOLStartHere.html
 4 drwxr-xr-x. 2 oracle oinstall    4096 Dec  3 08:49 img
 4 -rw-r--r--. 1 oracle oinstall   1736 Dec  3 08:49 ODI_HIVE.sql
176 -rw-r--r--. 1 oracle oinstall  177636 Dec  3 08:48 ODI_Imports.tar.gz
332 -rw-r--r--. 1 oracle oinstall  337016 Dec  3 08:49 ORA_CRM.sql
 4 -rw-r--r--. 1 oracle oinstall   1605 Dec  3 08:49 setup.sh
 4 -rw-r--r--. 1 oracle oinstall    738 Dec  3 08:49 setup.sql
 4 -rw-r--r--. 1 oracle oinstall   289 Dec  3 08:48 StartHOL.desktop
 4 -rw-r--r--. 1 oracle oinstall   434 Dec  3 08:48 start_oracle.sh
 4 -rw-r--r--. 1 oracle oinstall   286 Dec  3 08:49 StopHOL.desktop
 4 -rw-r--r--. 1 oracle oinstall   175 Dec  3 08:48 stop_oracle_noicon.sh
 4 -rw-r--r--. 1 oracle oinstall   348 Dec  3 08:49 stop_oracle.sh
 4 -rw-r--r--. 1 oracle oinstall   841 Dec  3 08:49 validate.sh
[oracle@bigdatalite setup]$
```

3. Log in to SQL\*Plus as `sys/welcome1@orcl` as `sysdba` and run the `setup.sql` script.

```
$ sqlplus sys/welcome1@orcl as sysdba
```

```
[oracle@bigdatalite setup]$ sqlplus sys/welcome1@orcl as sysdba
SQL*Plus: Release 12.1.0.2.0 Production on Tue Feb 10 11:22:09 2015
Copyright (c) 1982, 2014, Oracle. All rights reserved.

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing options

SQL> ■
```

```
SQL>@setup.sql
```

The following is the content of the `setup.sql` script:

```
setup.sql X
1 connect / as SYSDBA
2
3 drop user BDA cascade;
4
5 CREATE TABLESPACE BDA DATAFILE '/u01/app/oracle/oradata/orcl/BDA.dbf' SIZE 250M reuse AUTOEXTEND ON nologging;
6 CREATE USER BDA
7   IDENTIFIED BY welcome1 DEFAULT TABLESPACE BDA
8     QUOTA UNLIMITED ON BDA;
9
10 GRANT create procedure, create session, advisor, olap_user, unlimited tablespace to BDA;
11
12
13 grant execute on SYS.UTL_FILE to BDA;
14 create or replace directory OSCH_BIN_PATH as '/u01/connectors/osch/bin';
15
16 grant CREATE ANY DIRECTORY to BDA;
17 grant read, execute on directory OSCH_BIN_PATH to BDA;
18
19 GRANT CREATE MINING MODEL TO BDA;
20 GRANT RQADMIN to BDA;
21
22
23 connect BDA/welcome1
24
25 create or replace directory EXTERNAL_DIR as '/home/oracle/exercises/OSCH/etc';
26
27 @ORA_CRM
28 @ODI_HIVE
29
30 exit
31
```

The partial output is as follows:

```
1 row created.

1 row created.

1 row created.

1 row created.

Table created.

Index created.

Table altered.

Table altered.

Disconnected from Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64
bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing opt
ions
[oracle@bigdatalite setup]$
```

4. Enter the following command at the command prompt, and then press **Enter**.

```
cd /home/oracle/exercises/OXH
```

```
[oracle@bigdatalite setup]$ cd /home/oracle/exercises/0XH
[oracle@bigdatalite 0XH]$ pwd
/home/oracle/exercises/0XH
[oracle@bigdatalite 0XH]$ ls -l
total 208
drwxr-xr-x. 2 oracle oinstall 4096 Feb  9 09:52 books1
-rw-r--r--. 1 oracle oinstall   175 Dec  3 08:45 books1a.xq
-rw-r--r--. 1 oracle oinstall   179 Dec  3 08:44 books1b.xq
-rwxrwxrwx. 1 oracle oinstall  210 Dec  3 08:44 books1-local.sh
-rwxrwxrwx. 1 oracle oinstall  174 Dec  3 08:44 books1.sh
drwxr-xr-x. 2 oracle oinstall 4096 Feb  9 09:52 books2
-rw-r--r--. 1 oracle oinstall  225 Feb  9 12:07 books2a.xq
-rw-r--r--. 1 oracle oinstall  251 Feb  9 12:22 books2b.xq
-rwxrwxrwx. 1 oracle oinstall   96 Feb  9 12:24 books2.sh
-rw-r--r--. 1 oracle oinstall  203 Dec  3 08:44 books2.xq
-rw-r--r--. 1 oracle oinstall  197 Dec  3 08:44 books3olh.sql
-rwxrwxrwx. 1 oracle oinstall  222 Dec  3 08:44 books3.sh
-rw-r--r--. 1 oracle oinstall  606 Dec  3 08:44 books3.xq
-rw-r--r--. 1 oracle oinstall  708 Dec  3 08:44 books-create.hql
-rw-r--r--. 1 oracle oinstall   29 Dec  3 08:45 books-drop.hql
-rw-r--r--. 1 oracle oinstall  137 Dec  3 08:44 books-hive.hql
-rw-r--r--. 1 oracle oinstall  849 Dec  3 08:44 books.xml
-rwxrwxrwx. 1 oracle oinstall 1055 Dec  3 08:45 cheat.sh
-rwxrwxrwx. 1 oracle oinstall  372 Dec  3 08:45 cleanup.sh
-rw-r--r--. 1 oracle oinstall 92187 Dec  3 08:44 crm_data.csv
-rwxrwxrwx. 1 oracle oinstall   66 Dec  3 08:44 hive_oxh.sh
-rw-r--r--. 1 oracle oinstall  2350 Dec  3 08:44 logging.properties
-rwxrwxrwx. 1 oracle oinstall  176 Dec  3 08:45 nosql.custsums.sh
-rw-r--r--. 1 oracle oinstall  515 Dec  3 08:44 nosql-custsums.xq
-rwxrwxrwx. 1 oracle oinstall  294 Dec  3 08:44 nosql.join.sh
-rw-r--r--. 1 oracle oinstall  888 Dec  3 08:44 nosql-join.xq
-rwxrwxrwx. 1 oracle oinstall  170 Dec  3 08:44 nosql.sums.sh
-rw-r--r--. 1 oracle oinstall  336 Dec  3 08:44 nosql-sums.xq
-rw-r--r--. 1 oracle oinstall 1020 Dec  3 08:45 0XH_Functions.sql
-rwxrwxrwx. 1 oracle oinstall   76 Dec  3 08:45 Viewdata.sh
[oracle@bigdatalite 0XH]$
```

5. Connect as bda/welcome1@orcl.

```
$ sqlplus bda/welcome1@orcl
```

```
[oracle@bigdatalite 0XH]$ sqlplus bda/welcome1@orcl

SQL*Plus: Release 12.1.0.2.0 Production on Tue Feb 10 11:38:52 2015

Copyright (c) 1982, 2014, Oracle. All rights reserved.

Last Successful login time: Tue Feb 10 2015 11:23:41 -05:00

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing options

SQL> ■
```

6. Create the target table books. Run the books3olh.sql script.

```
@books3olh
```

The following is the content of the books3oh.sql script:

```
books3oh.sql X
1 connect BDA/welcome1
2
3 drop table BOOKS;
4
5 CREATE TABLE BOOKS
6 (
7   TITLE VARCHAR2(100)
8 , LANG VARCHAR2(10)
9 , CATEGORY VARCHAR2(40)
10 , YEAR INTEGER
11 , PRICE FLOAT
12 , AUTHOR VARCHAR2(100)
13 );
14
15 EXIT;
```

```
SQL> @books3oh
Connected.
drop table BOOKS
*
ERROR at line 1:
ORA-00942: table or view does not exist

Table created.

Disconnected from Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64
bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing opt
ions
[oracle@bigdatalite OXH]$
```

7. View the file books3.xq file. OXH includes an Oracle database adapter that uses a custom put command to publish XQuery results to an Oracle table. The Oracle adapter will take the data types in the Oracle table and map them to data in the put command.

```
cat books3.xq
```

```
[oracle@bigdatalite 0XH]$ cat books3.xq
import module namespace xmlf = "oxh:xmlf";
import module namespace text = "oxh:text";

declare
  %oracle:put
  %oracle-property:targetTable('books')
  %oracle:columns('title','lang','category','year','price','author')
  %oracle-property:connection.user('BDA')
  %oracle-property:connection.password('welcome1')
  %oracle-property:connection.url('jdbc:oracle:thin:@//localhost:1521/orcl')
function local:myPut($c1, $c2, $c3, $c4, $c5, $c6) external;

for $x in xmlf:collection("books.xml", "book")
for $y in $x/author
return local:myPut($x/title,$x/title/@lang,$x/@category,$x/year,$x/price,$y)

[oracle@bigdatalite 0XH]$
```

8. Run the books3.sh script.

```
./books3.sh
```

```
[oracle@bigdatalite 0XH]$ ./books3.sh
SQL*Plus: Release 12.1.0.2.0 Production on Tue Feb 10 11:49:20 2015
Copyright (c) 1982, 2014, Oracle. All rights reserved.

SQL> Connected.
SQL> SQL>
  COUNT(*)
-----
      0

SQL>
Table truncated.

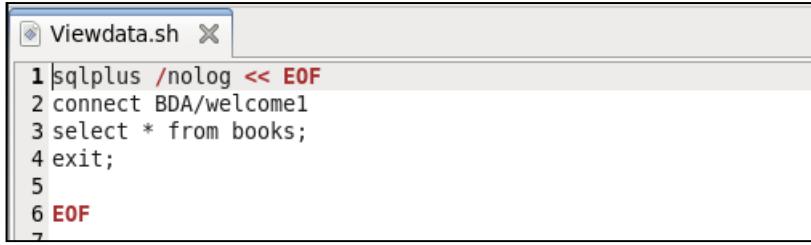
SQL>
  COUNT(*)
-----
      0

SQL> Disconnected from Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing options
rm: `books4': No such file or directory
15/02/10 11:49:24 INFO hadoop.xquery: 0XH: Oracle XQuery for Hadoop 4.0.1 (build 4.0.1-cdh5.)
```

```
Total megabyte seconds taken by all map tasks=1571040
Map-Reduce Framework
  Map input records=8
  Map output records=8
  Input split bytes=187
  Spilled Records=0
  Failed Shuffles=0
  Merged Map outputs=0
  GC time elapsed (ms)=140
  CPU time spent (ms)=2800
  Physical memory (bytes) snapshot=194068480
  Virtual memory (bytes) snapshot=689631232
  Total committed heap usage (bytes)=187695104
File Input Format Counters
  Bytes Read=1598
File Output Format Counters
  Bytes Written=1624
15/02/10 11:50:08 INFO hadoop.xquery: Finished executing "./books3.xq". Output path: "hdfs://bigdatalite.localdomain:8020/user/oracle/books4"
[oracle@bigdatalite OXH]$ █
```

- Run the `./Viewdata.sh` script to view the contents of the `books` table.

```
./Viewdata.sh
```



```
Viewdata.sh X
1sqlplus /nolog << EOF
2connect BDA/welcome1
3select * from books;
4exit;
5
6EOF
7
```

```
[oracle@bigdatalite 0XH]$ ./Viewdata.sh
SQL*Plus: Release 12.1.0.2.0 Production on Tue Feb 10 11:57:21 2015
Copyright (c) 1982, 2014, Oracle. All rights reserved.

SQL> Connected.
SQL>
TITLE
-----
LANG      CATEGORY          YEAR     PRICE
-----  
AUTHOR
-----
Everyday Italian
en          COOKING           2005      30
Giada De Laurentiis

Harry Potter
en          CHILDREN          2005      29.99
J K. Rowling

TITLE
-----
LANG      CATEGORY          YEAR     PRICE
-----  
AUTHOR
-----
XQuery Kick Start
en          WEB               2003      49.99
James McGovern

XQuery Kick Start
en          WEB               2003      49.99

TITLE
```

```

AUTHOR
-----
Per Bothner

XQuery Kick Start
en          WEB
Kurt Cagle                                         2003      49.99

XQuery Kick Start

TITLE
-----
LANG   CATEGORY           YEAR    PRICE
-----
AUTHOR
-----
en      WEB               2003    49.99
James Linn

XQuery Kick Start
en          WEB
Vaidyanathan Nagarajan                           2003      49.99

TITLE
-----
LANG   CATEGORY           YEAR    PRICE
-----
AUTHOR
-----
Learning XML
en      WEB               2003    39.95
Erik T. Ray

8 rows selected.

SQL> Disconnected from Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing options
[oracle@bigdatalite OXH]$
[oracle@bigdatalite OXH]$ 

```

You can optionally view the above output in Oracle SQL Developer as follows:

The screenshot shows the Oracle SQL Developer interface. On the left, the Connections tree shows a connection named 'bda' with a red box around the 'Tables (Filtered)' node, which contains a 'BOOKS' table. On the right, the main area displays the 'Data' tab for the 'BOOKS' table. The table has columns: TITLE, LANG, CATEGORY, YEAR, PRICE, and AUTHOR. The data consists of 8 rows, each representing a book entry. A red box highlights the entire data grid.

	TITLE	LANG	CATEGORY	YEAR	PRICE	AUTHOR
1	Everyday Italian	en	COOKING	2005	30	Giada De Laurentiis
2	Harry Potter	en	CHILDREN	2005	29.99	J. K. Rowling
3	XQuery Kick Start	en	WEB	2003	49.99	James McGovern
4	XQuery Kick Start	en	WEB	2003	49.99	Per Bothner
5	XQuery Kick Start	en	WEB	2003	49.99	Kurt Cagle
6	XQuery Kick Start	en	WEB	2003	49.99	James Linn
7	XQuery Kick Start	en	WEB	2003	49.99	Vaidyanathan Nagarajan
8	Learning XML	en	WEB	2003	39.95	Erik T. Ray

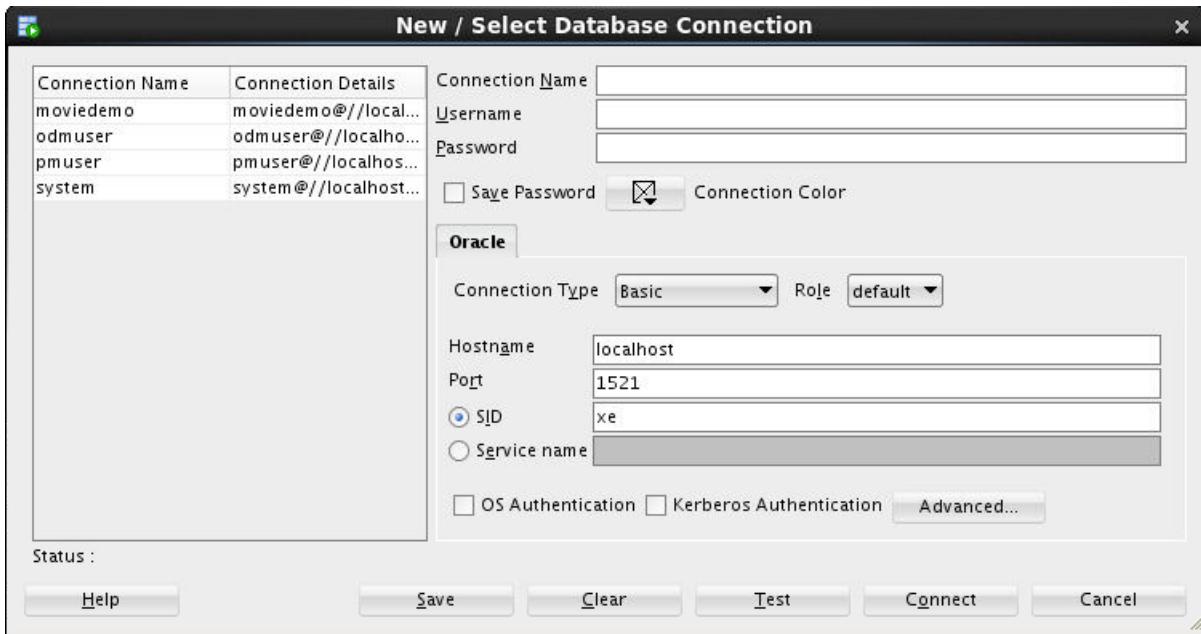
To use SQL Developer, you need to create a new Database Connection as follows:

1. Start SQL Developer. Click the **SQLDeveloper 4.0** icon on the toolbar in your Oracle Big Data Lite VM window. SQL Developer is displayed.

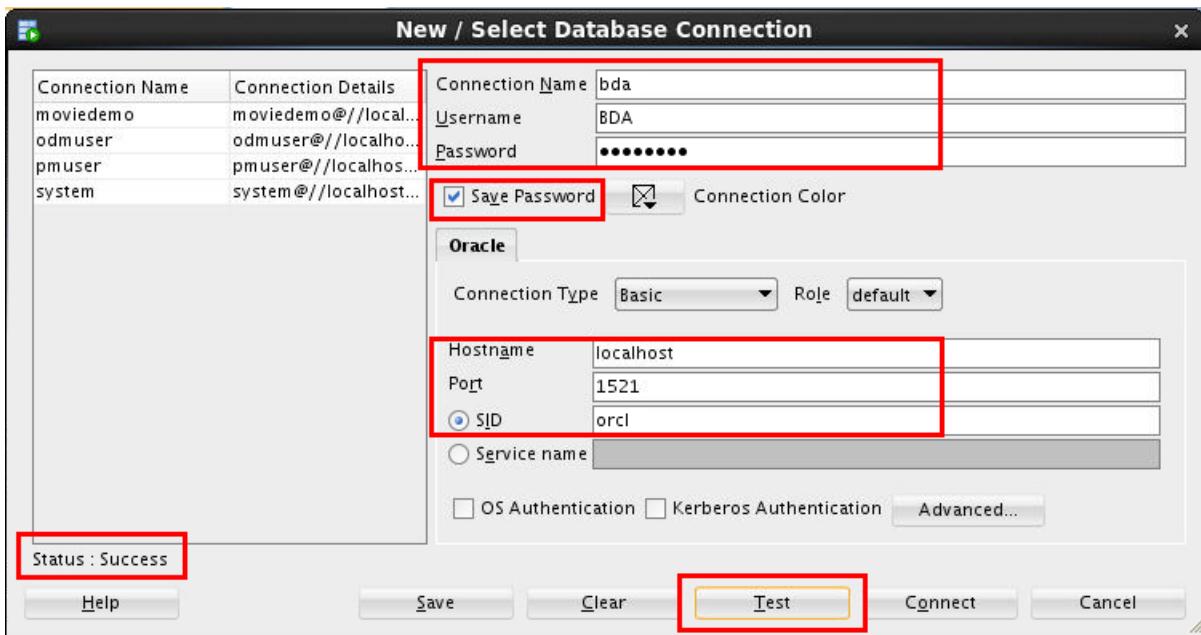


2. Click the **New Connection** icon (big green plus sign) in the **Connections** pane. The **New / Select Database Connection** window is displayed.

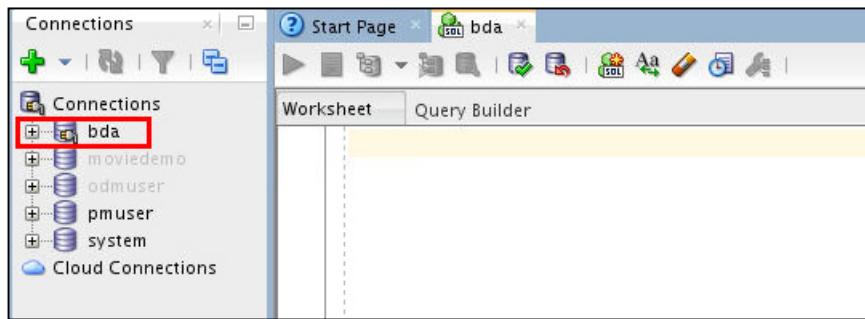




3. Enter the following information in the **New / Select Database Connection** window:
  - a. **Connection Name:** bda (can be any name you choose).
  - b. **Username:** BDA
  - c. **Password:** welcome1
  - d. **Save Password:** Select that option
  - e. **SID:** orcl
4. Click **Test** to test the new connection. The Status section at the bottom left-hand side of the window should display **Success**.



5. Click **Connect**. The new **bda** database connection is displayed in the list of available connections.



6. Drill-down on the + (plus sign) next to the **bda** connection, and then drill-down on **Tables**. On the **Books** tab in the view pane, select the **Data** sub-tab.

A screenshot of the Oracle SQL Developer interface. The 'Connections' panel on the left shows 'bda' expanded, with 'Tables (Filtered)' and 'BOOKS' (which is highlighted with a red box) listed under it. The 'Data' tab is selected in the 'Books' view pane on the right. A red box highlights the entire data grid, which contains the following data:

	TITLE	LANG	CATEGORY	YEAR	PRICE	AUTHOR
1	Everyday Italian	en	COOKING	2005	30	Giada De Laurentiis
2	Harry Potter	en	CHILDREN	2005	29.99	J. K. Rowling
3	XQuery Kick Start	en	WEB	2003	49.99	James McGovern
4	XQuery Kick Start	en	WEB	2003	49.99	Per Bothner
5	XQuery Kick Start	en	WEB	2003	49.99	Kurt Cagle
6	XQuery Kick Start	en	WEB	2003	49.99	James Linn
7	XQuery Kick Start	en	WEB	2003	49.99	Vaidyanathan Nagarajan
8	Learning XML	en	WEB	2003	39.95	Erik T. Ray

## **Practices for Lesson 14: Overview of Solr**

**Chapter 14**

## Practices for Lesson 14

---

### Practices Overview

In these practices, you will learn to index a file using Apache Solr.

## Guided Practice 14-1: Using Apache Solr

### Overview

In this practice, you will load “insur\_cust\_ltv\_sample” data into SOLR for indexing and make it searchable.

In order to start using Solr for indexing the data, you must configure a collection holding the index. A configuration for a collection requires a `solrconfig.xml` file, a `schema.xml` and any helper files may be referenced from the XML files. The `solrconfig.xml` file contains all of the Solr settings for a given collection, and the `schema.xml` file specifies the schema that Solr uses when indexing documents. For more details on how to configure it for your data set see <http://wiki.apache.org/solr/SchemaXml>

### Prerequisite

Open a terminal window and navigate to the following directory. Execute the `reset.sh` script to reset the machine for Solr.

```
cd /home/oracle/exercises/solr  
sh reset.sh
```

### Tasks

1. Open a new terminal and navigate to the below directory:

```
cd /home/oracle/exercises/solr  
ls -l  
  
[oracle@bigdatalite ~]$ cd /home/oracle/exercises/solr  
[oracle@bigdatalite solr]$ ls -l  
total 20  
drwxrwx---. 2 oracle oinstall 4096 Apr  8 12:40 insur_cust_ltv_sample  
-rwxrwx---. 1 oracle oinstall  670 Apr  8 12:56 oxh_solr.sh  
-rwxrwx---. 1 oracle oinstall 1474 Dec  2 22:15 oxh_solr.xml  
-rwxrwx---. 1 oracle oinstall 1099 Dec  2 22:15 oxh_solr.xq  
-rwxrwx---. 1 oracle oinstall   27 Apr  8 13:00 reset.sh  
[oracle@bigdatalite solr]$
```

2. Use the `solrctl` command-line tool and generate a skeleton of the instance directory.

```
solrctl instancedir --generate solr_configs
```

```
[oracle@bigdatalite solr]$ solrctl instancedir --generate solr_configs  
[oracle@bigdatalite solr]$
```

3. Analyze the contents of `solr_configs` directory and its `conf` subdirectory.

```
cd solr_configs  
ls -l  
cd conf  
ls -l
```

```
[oracle@bigdatalite solr]$ cd solr_configs  
[oracle@bigdatalite solr_configs]$ ls -l  
total 4  
drwxr-xr-x. 5 oracle oinstall 4096 Apr  8 05:26 conf  
[oracle@bigdatalite solr_configs]$ cd conf  
[oracle@bigdatalite conf]$ ls -l  
total 352  
-rw-r--r--. 1 oracle oinstall 1092 Apr  8 05:26 admin-extra.html  
-rw-r--r--. 1 oracle oinstall 953 Apr  8 05:26 admin-extra.menu-bottom.html  
-rw-r--r--. 1 oracle oinstall 951 Apr  8 05:26 admin-extra.menu-top.html  
-rw-r--r--. 1 oracle oinstall 4041 Apr  8 05:26 currency.xml  
-rw-r--r--. 1 oracle oinstall 1386 Apr  8 05:26 elevate.xml  
drwxr-xr-x. 2 oracle oinstall 4096 Apr  8 05:26 lang  
-rw-r--r--. 1 oracle oinstall 82327 Apr  8 05:26 mapping-FoldToASCII.txt  
-rw-r--r--. 1 oracle oinstall 3114 Apr  8 05:26 mapping-ISOLatin1Accent.txt  
-rw-r--r--. 1 oracle oinstall 894 Apr  8 05:26 protwords.txt  
-rw-r--r--. 1 oracle oinstall 59635 Apr  8 05:26 schema.xml  
-rw-r--r--. 1 oracle oinstall 921 Apr  8 05:26 scripts.conf  
-rw-r--r--. 1 oracle oinstall 72455 Apr  8 05:26 solrconfig.xml  
-rw-r--r--. 1 oracle oinstall 74771 Apr  8 05:26 solrconfig.xml.secure  
-rw-r--r--. 1 oracle oinstall 16 Apr  8 05:26 spellings.txt  
-rw-r--r--. 1 oracle oinstall 795 Apr  8 05:26 stopwords.txt  
-rw-r--r--. 1 oracle oinstall 1148 Apr  8 05:26 synonyms.txt  
-rw-r--r--. 1 oracle oinstall 1469 Apr  8 05:26 update-script.js  
drwxr-xr-x. 2 oracle oinstall 4096 Apr  8 05:26 velocity  
drwxr-xr-x. 2 oracle oinstall 4096 Apr  8 05:26 xslt  
[oracle@bigdatalite conf]$
```

4. Open and analyze the contents of schema.xml and solrconfig.xml files.

```
more schema.xml  
more solrconfig.xml
```

```
[oracle@bigdatalite conf]$ more schema.xml  
<?xml version="1.0" encoding="UTF-8" ?>  
<!--  
Licensed to the Apache Software Foundation (ASF) under one or more  
contributor license agreements. See the NOTICE file distributed with  
this work for additional information regarding copyright ownership.  
The ASF licenses this file to You under the Apache License, Version 2.0  
(the "License"); you may not use this file except in compliance with  
the License. You may obtain a copy of the License at  
  
http://www.apache.org/licenses/LICENSE-2.0  
  
Unless required by applicable law or agreed to in writing, software  
distributed under the License is distributed on an "AS IS" BASIS,  
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the specific language governing permissions and  
limitations under the License.  
-->  
  
<!--  
This is the Solr schema file. This file should be named "schema.xml" and  
should be in the conf directory under the solr home  
--More--(3%)
```

**Note:** Keep pressing Enter to view the full content of the file.

```
[oracle@bigdatalite conf]$ more solrconfig.xml
<?xml version="1.0" encoding="UTF-8" ?>
<!--
Licensed to the Apache Software Foundation (ASF) under one or more
contributor license agreements. See the NOTICE file distributed with
this work for additional information regarding copyright ownership.
The ASF licenses this file to You under the Apache License, Version 2.0
(the "License"); you may not use this file except in compliance with
the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
-->

<!--
For more details about configurations options that may appear in
this file, see http://wiki.apache.org/solr/SolrConfigXml.
-->
<config>
<!-- In all configuration below, a prefix of "solr." for class names
is an alias that causes solr to search appropriate packages,
--More--(3%)
```

5. Navigate to the below directory and list the contents:

```
cd /home/oracle/exercises/solr
ls -l
```

```
[oracle@bigdatalite solr]$ cd /home/oracle/exercises/solr
[oracle@bigdatalite solr]$ ls -l
total 24
drwxrwx---. 2 oracle oinstall 4096 Apr  8 12:40 insur_cust_ltv_sample
-rwxrwx---. 1 oracle oinstall  670 Apr  8 12:56 oxh_solr.sh
-rwxrwx---. 1 oracle oinstall 1474 Dec  2 22:15 oxh_solr.xml
-rwxrwx---. 1 oracle oinstall 1099 Dec  2 22:15 oxh_solr.xq
-rwxrwx---. 1 oracle oinstall  277 Apr  8 13:00 reset.sh
drwxr-xr-x. 3 oracle oinstall 4096 Apr  8 13:25 solr_configs
[oracle@bigdatalite solr]$
```

6. Open and analyze the contents of the `oxh_solr.xq` file. This file helps to parse the HDFS file and write the output to a Solr Index.

```
cat oxh_solr.xq
```

```
[oracle@bigdatalite solr]$ cat oxh_solr.xq
(: flags:disabled(include.solr=false;hd=1,2,3,4,100) :)
(: -Xhelper oracle.hadoop.xquery.adapter.solr.SolrTestHelper :)

(: PREREQUISITES

1. Configure Solr collection
solrctl instancedir --generate $HOME/solr_configs
solrctl instancedir --create oxh_collection_1 $HOME/solr_configs
solrctl collection --create oxh_collection_1 -s 1

2. Set the following Hadoop configuration properties
oracle.hadoop.xquery.solr.loader.zk-host = /solr
oracle.hadoop.xquery.solr.loader.collection = oxh_collection_1
oracle.hadoop.xquery.solr.loader.go-live = true

EXPECTED RESULTS

:)

import module "oxh:text";
import module "oxh:solr";

for $i in text:collection(
  oxh:property("oxh_tck.xq.in") || "/part*"
)

let $f := tokenize($i, ",")
let $cust_id := xs:string($f[1])
let $cust_n := concat(xs:string($f[15]), ' ', xs:string($f[16]))
let $age := xs:integer($f[22])
let $prof := xs:string($f[20])

return solr:put(
  <doc>
    <field name="id">{ $cust_id }</field>
    <field name="name">{ $cust_n }</field>
    <field name="title">{ $prof }</field>
  </doc>
)
[oracle@bigdatalite solr]$
```

- Open and analyze the contents of `oxh_solr.xml` file. This file contains the OLH connection information and OXH configuration information.

```
cat oxh_solr.xml
```

```
[oracle@bigdatalite solr]$ cat oxh_solr.xml
<?xml version="1.0"?>

<!--
Copyright (c) 2013, Oracle and/or its affiliates. All rights reserved.

Configuration for OXH TCK
-->
<configuration>

    <!-- OLH connection information -->

    <property>
        <name>oracle.hadoop.xquery.solr.loader.zk-host</name>
        <value>${zkhosts}</value>
    </property>

    <property>
        <name>oracle.hadoop.xquery.solr.loader.collection</name>
        <value>${collection}</value>
    </property>

    <property>
        <name>ha.zookeeper.quorum</name>
        <value>${zkquorum}</value>
    </property>

    <property>
        <name>oracle.hadoop.xquery.solr.loader.go-live</name>
        <value>true</value>
    </property>
```

```
<!-- OXH configuration -->

<property>
  <name>hdfs_tmp</name>
  <value>temp_out_oxh</value>
</property>

<property>
  <name>oxh_tck.xq.in</name>
  <value>${hdfs_data}</value>
</property>

<property>
  <name>oxh_tck.xq.out</name>
  <value>${hdfs_tmp}/oxh_tck.xq</value>
</property>

<property>
  <name>oxh_tck.xq.2.out</name>
  <value>${hdfs_tmp}/oxh_tck.xq.2</value>
</property>

<property>
  <name>oxh_tck.xq.3.out</name>
  <value>${hdfs_tmp}/oxh_tck.xq.3</value>
</property>

<property>
  <name>oxh_tck.xq.4.out</name>
  <value>${hdfs_tmp}/oxh_tck.xq.4</value>
</property>

<property>
  <name>oracle.hadoop.xquery.scratch</name>
  <value>${hdfs_tmp}/scratch</value>
</property>

</configuration>

[oracle@bigdatalite solr]$
```

8. Open and analyze the contents of the `oxh_solr.sh` file. This file uses the above `oxh_solr.xq` file and `oxh_solr.xml` file and indexes the HDFS file for Solr.

```
cat oxh_solr.sh
```

```
[oracle@bigdatalite solr]$ cat oxh_solr.sh
#!/bin/sh
#
# Copyright (c) 2013, Oracle and/or its affiliates. All rights reserved.
#
export ZKHOSTS=bigdatalite.localdomain:2181/solr
export ZKQUORUM=bigdatalite.localdomain:2181
export COLLECTION=collection1
export HDFS_DATA=/user/oracle/insur_cust_ltv_sample
export HDFS_TMP=/user/oracle/oxh_temp

rm -rf solr_configs/
solrctl instancedir --generate solr_configs
solrctl instancedir --create collection1 solr_configs
solrctl collection --create collection1 -s 1

hadoop jar $OXH_HOME/lib/oxh.jar \
-Dzkhosts=$ZKHOSTS -Dzkquorum=$ZKQUORUM -Dcollection=$COLLECTION \
-Dhdfs_data=$HDFS_DATA -Dhdfs_tmp=$HDFS_TMP \
-conf ./oxh_solr.xml oxh_solr.xq -clean -output $HDFS_TMP/oxh_solr

[oracle@bigdatalite solr]$
```

- Run the `oxh_solr.sh` file to index the HDFS file for Solr.

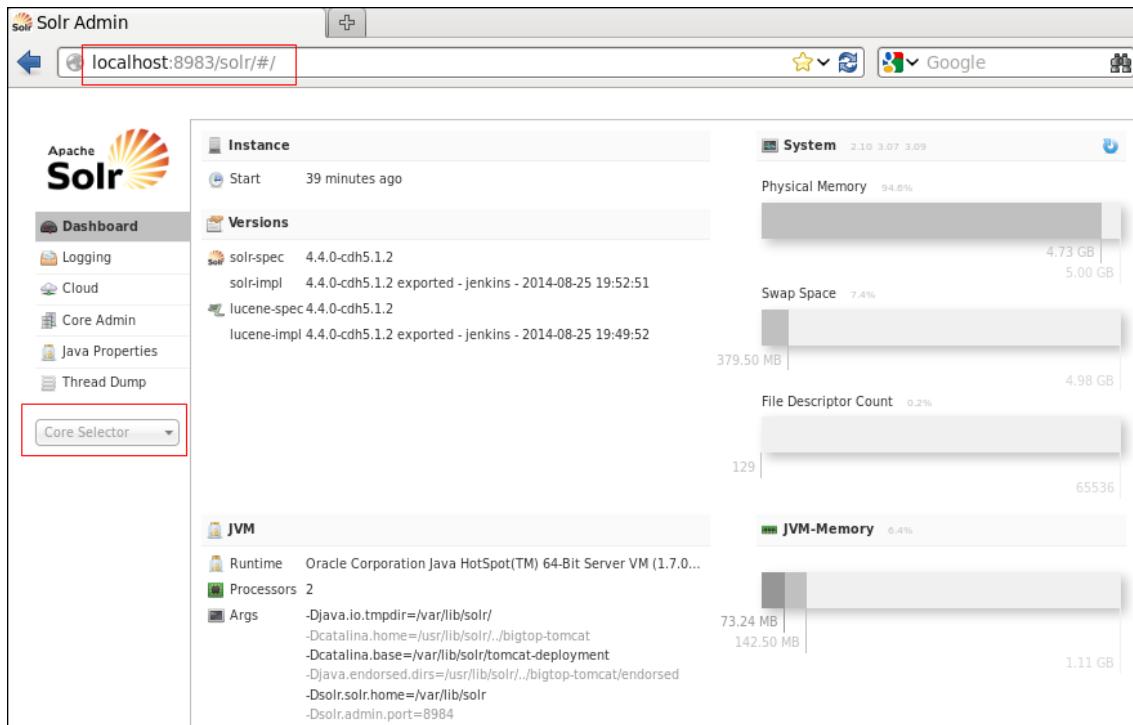
```
sh oxh_solr.sh
```

```
[oracle@bigdatalite solr]$ sh oxh_solr.sh
Uploading configs from solr_configs/conf to bigdatalite.localdomain:2181/solr. This may take up to a minute.
15/05/06 02:39:38 INFO hadoop.xquery: OXH: Oracle XQuery for Hadoop 4.0.1 (build 4.0.1-cdh5.0.0-mr1 @mr2). Copyright (c) 2015, Oracle. All rights reserved.
15/05/06 02:39:39 INFO hadoop.xquery: Executing query "oxh_solr.xq". Output path: "hdfs://bigdatalite.localdomain:8020/user/oracle/oxh_temp/oxh_solr"
15/05/06 02:39:41 INFO hadoop.xquery: Submitting map-reduce job "oxh:oxh_solr.xq#0" id="4652b25a-c042-4c68-9b59-1e313a0f5f03.0", inputs=[hdfs://bigdatalite.localdomain:8020/user/oracle/insur_cust_ltv_sample/part*], output=hdfs://bigdatalite.localdomain:8020/user/oracle/oxh_temp/scratch/4652b25a-c042-4c68-9b59-1e313a0f5f03.0
15/05/06 02:39:42 INFO client.RMProxy: Connecting to ResourceManager at localhost/127.0.0.1:8032
15/05/06 02:39:46 INFO input.FileInputFormat: Total input paths to process : 1
15/05/06 02:39:46 INFO mapreduce.JobSubmitter: number of splits:1
15/05/06 02:39:47 INFO mapreduce.JobSubmitter: tokens for job: job_1430236747904
15/05/06 02:40:57 INFO zookeeper.ZooKeeper: Session: 0x14d00c03510000b closed
15/05/06 02:40:57 INFO hadoop.GoLive: Done committing live merge
15/05/06 02:40:57 INFO hadoop.GoLive: Live merging of index shards into Solr cluster took 1.898 secs
15/05/06 02:40:57 INFO hadoop.GoLive: Live merging completed successfully
15/05/06 02:40:57 INFO hadoop.MapReduceIndexerTool: Succeeded with job: jobName: org.apache.solr.hadoop.MapReduceIndexerTool/OXHSolrMapper, jobId: job_1430236747904_0002
15/05/06 02:40:57 INFO hadoop.MapReduceIndexerTool: Success. Done. Program took 44.685 secs. Goodbye.
15/05/06 02:40:57 INFO zookeeper.ClientCnxn: EventThread shut down
15/05/06 02:40:57 INFO hadoop.xquery: Finished executing "oxh_solr.xq". Output path: "hdfs://bigdatalite.localdomain:8020/user/oracle/oxh_temp/oxh_solr"
[oracle@bigdatalite solr]$
```

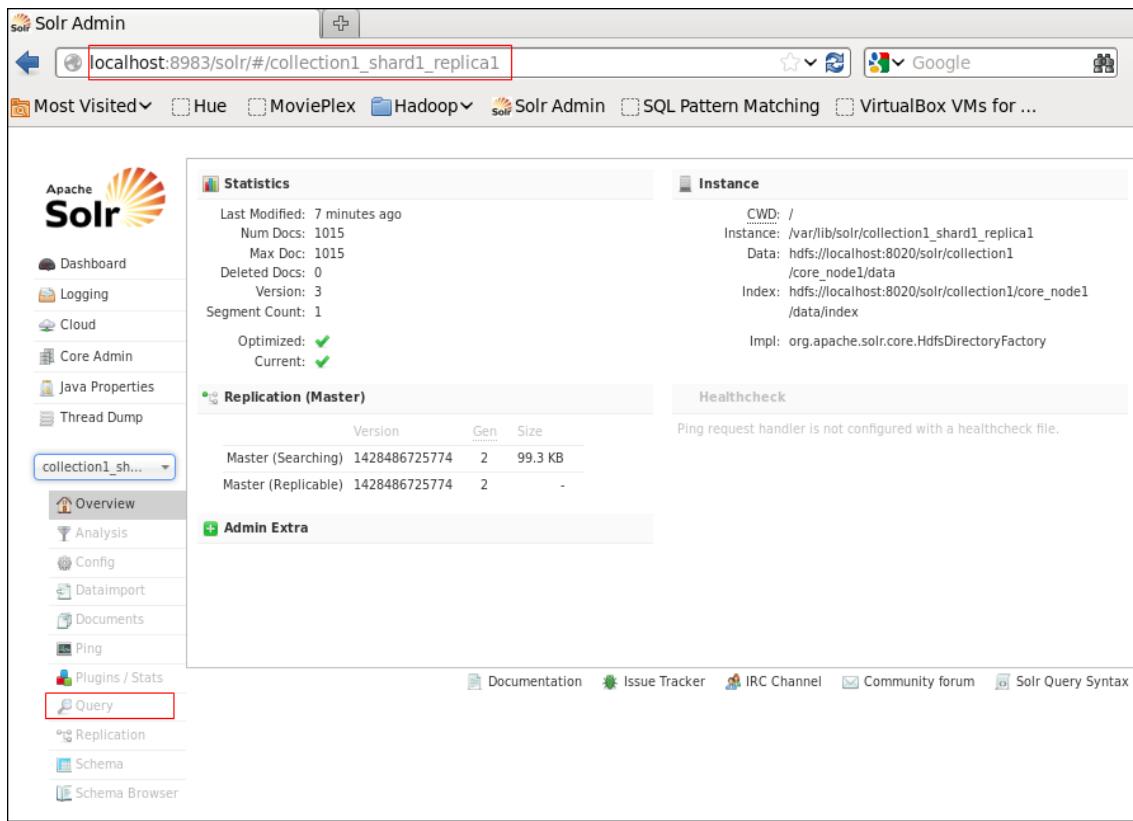
- You can open the SOLR webpage at <http://bigdatalite.localdomain:8983/solr> and browse the configs and collection you have created.

<http://bigdatalite.localdomain:8983/solr>

**Note:** Alternatively, you can use localhost instead of bigdatalite.localdomain.



- Click the **Core Selector** drop-down in the right pane. Select the **collection1\_shard1\_replica1** link.



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

12. Click the Query link on the right pane. You will get a search page. Click the Execute Query button.

The screenshot shows the Apache Solr admin interface for the 'collection1\_sh...' collection. On the left, there's a sidebar with various links like Dashboard, Logging, Cloud, Core Admin, Java Properties, Thread Dump, Overview, Analysis, Config, Dataimport, Documents, Ping, Plugins / Stats, Replication, Schema, and Schema Browser. The 'Query' link is highlighted with a red box. The main panel has a 'Request-Handler (qt)' section with a 'select' dropdown. Below it are fields for 'q' (containing '\*.\*'), 'fq', 'sort', 'start', 'rows' (set to 10), 'fl', 'df', and 'Raw Query Parameters' (key1=val1&key2=val2). There are several checkboxes for different query parsers: 'indent' (checked), 'debugQuery', 'dismax', 'edismax', 'hl', 'facet', 'spatial', and 'spellcheck'. At the bottom is a blue 'Execute Query' button. To the right, a browser window shows the JSON response from the Solr endpoint `http://localhost:8983/solr/collection1_shard1_replica1/select?q=%3A*&wt=json&indent=true`. The response includes the status, QTime, params, and a list of three documents with their IDs, names, titles, and version numbers.

```

{
  "responseHeader": {
    "status": 0,
    "QTime": 14,
    "params": {
      "indent": "true",
      "q": "*:*",
      "wt": "json"
    }
  },
  "response": {
    "numFound": 1015,
    "start": 0,
    "docs": [
      {
        "id": "CU1008",
        "name": "RANDELL POLLOCK",
        "title": [
          "Not specified"
        ],
        "_version_": 1497876889531842600
      },
      {
        "id": "CU10006",
        "name": "GREGORIO LAWRENCE",
        "title": [
          "Lab Technician"
        ],
        "_version_": 1497876889685983200
      },
      {
        "id": "CU10011",
        "name": "MOZELLA CAREY",
        "title": [
          "PROF-16"
        ],
        "_version_": 1497876889687031800
      }
    ]
  }
}

```

**Note:** In the box labeled “q” you have by default “\*.\*”. Click “Execute Query” at the bottom of the page to get the indexed output.

13. You can try giving various values to “q” to get the corresponding search output. Enter \*JOHN\* in the q field and click Execute Query. You can see that the search returns the data that has JOHN in it.

The screenshot shows the Apache Solr admin interface. On the left, there's a sidebar with various navigation links like Dashboard, Logging, Cloud, Core Admin, Java Properties, Thread Dump, and a dropdown for 'collection1\_sh...'. The main area has a 'Request-Handler (qt)' section with fields for 'q' (containing '\*JOHN\*'), 'fq', 'sort', 'start', 'rows' (set to 10), 'fl', 'df', and 'wt' (set to json). Below these are several checkboxes for query parameters: indent (checked), debugQuery, dismax, edismax, hl, facet, spatial, and spellcheck. At the bottom is a blue 'Execute Query' button. To the right is a browser window displaying the JSON response to the search query. The response includes a 'responseHeader' with status 0, QTime 9, and params (including indent: true, q: \*JOHN\*, \_id: 1428487567480, wt: json). The 'response' section shows 7 documents, each with an id, name, title, and version. The first document is 'JOHNNY FELICIANO' (id: CU13565), the second is 'JERAMY STJOHN' (id: CII14974), and the third is 'JOHNNIE KOHLER' (id: CU5262).

```
http://localhost:8983/solr/collection1_shard1_replica1/select?q=*JOHN*&wt=json&indent=true

{
  "responseHeader": {
    "status": 0,
    "QTime": 9,
    "params": {
      "indent": "true",
      "q": "*JOHN*",
      "_id": "1428487567480",
      "wt": "json"
    }
  },
  "response": {
    "numFound": 7,
    "start": 0,
    "docs": [
      {
        "id": "CU13565",
        "name": "JOHNNY FELICIANO",
        "title": [
          "Childcare Worker"
        ],
        "_version_": 1497876890008944600
      },
      {
        "id": "CII14974",
        "name": "JERAMY STJOHN",
        "title": [
          "PROF-51"
        ],
        "_version_": 1497876890197688300
      },
      {
        "id": "CU5262",
        "name": "JOHNNIE KOHLER",
        "title": [
          "Not specified"
        ],
        "_version_": 1497876890575175700
      }
    ]
  }
}
```

**Note:** The search will return more rows than that are displayed here.

## Guided Practice 14-2: Using Solr with Hue

### Overview

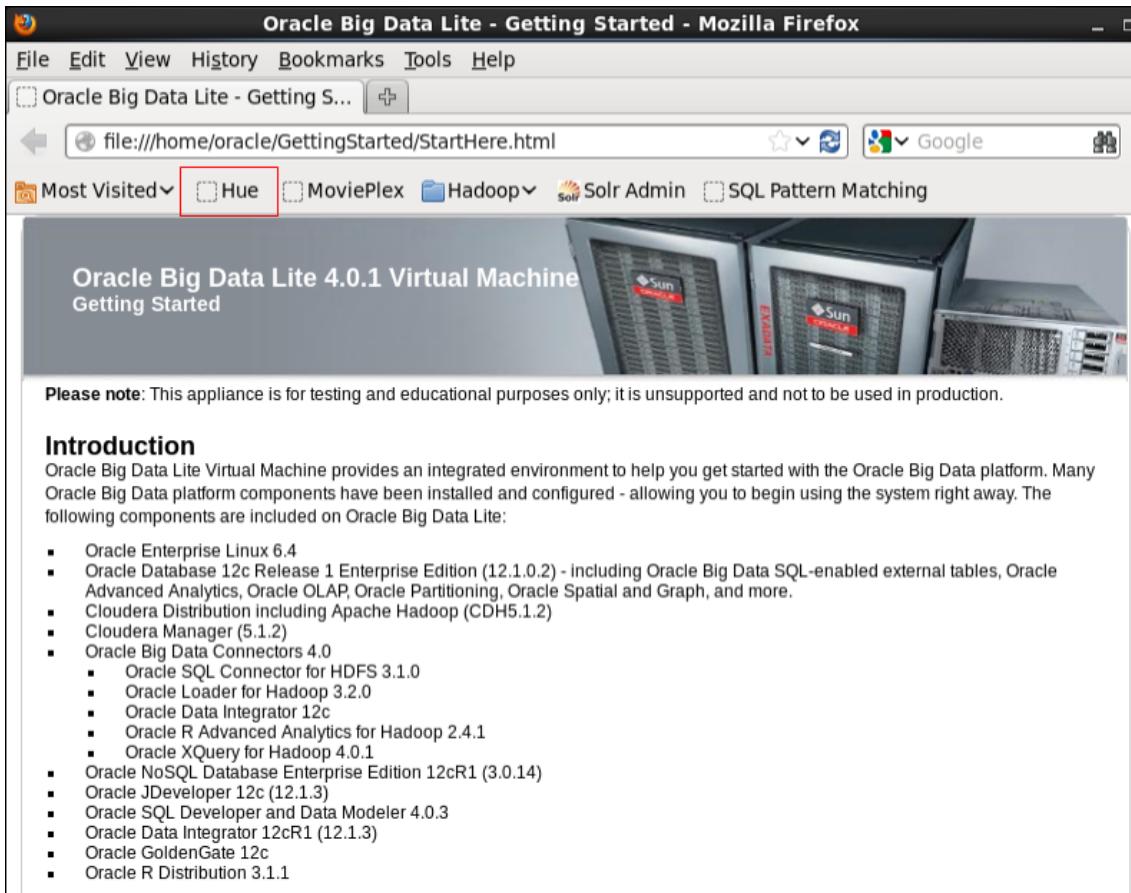
In this practice, you will learn to use Solr with Hue.

### Assumptions

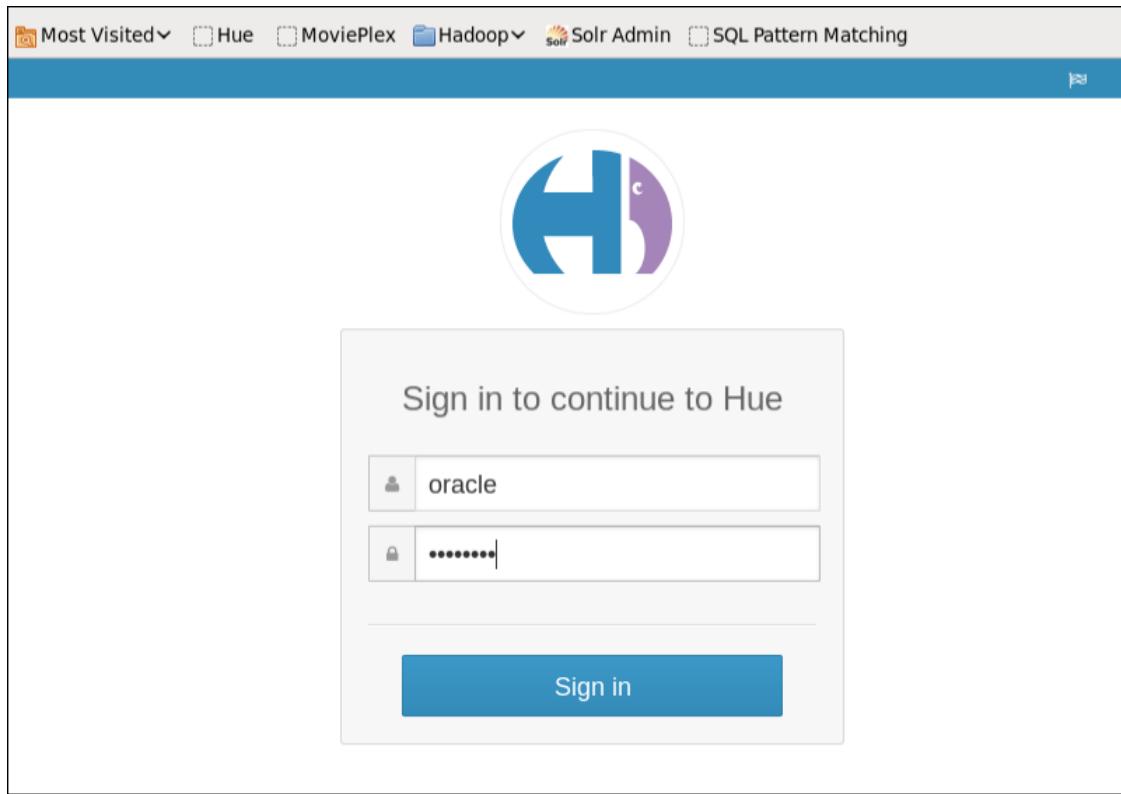
Practice 14-1 is completed.

### Tasks

1. Open Firefox and click Hue in the Bookmarks toolbar.



2. Enter the username and password in the login page of HUE and click Sign in.



**Note:** The username and password will be autofilled. Refer to the home page of BigDataLite <file:///home/oracle/GettingStarted/StartHere.html> for getting the username and password.

3. Click the Search drop-down and select Indexes on the HUE home page.

Time	Query	Result
12/10/14 23:10:12	select title, authors from bookstore;	See results...
12/10/14 23:08:22	select * from bookstore	
12/10/14 23:07:17	select title, authors from bookstore	

4. Select Collection1 on the page.

The screenshot shows the 'Collections' section of the Solr Indexer in Hue. At the top, there is a search bar labeled 'Filter collections...' and a 'Delete' button. Below the search bar is a checkbox labeled 'Name'. A list of collection names is displayed, with 'collection1' highlighted by a red box. Other visible collection names include 'moviedemo', 'collection1\_shard1\_replica1', 'moviedemo\_shard1\_replica1', and 'collection1' again.

5. Click the search link on the Solr Indexer page.

The screenshot shows the details for the 'collection1' selected in the previous step. On the left, there is an 'ACTIONS' sidebar with a 'Search' button highlighted by a red box. The main area shows the 'Collections > collection1' page. It displays a table of field configurations:

Name	Type	Unique key field	Required	Indexed	Stored
features	text_general		✓	✓	✓
links	string		✓	✓	✓
text	text_general		✓		✓
keywords	text_general		✓	✓	✓
id	string		✓	✓	✓

**Note:** There will be more rows than are displayed above.

6. Enter any text in the Collection1 search input box and click the search icon to get the corresponding results.

The screenshot shows a search interface with a search bar at the top labeled "Search collection1". A red box highlights the search bar. Below the search bar is a "Grid Results" section with a "Filter fields" button highlighted by a red box. To the right of the filter button, it says "Showing 1 to 10 of 1015 results". On the far right, there are three small icons. The main area displays a list of documents, each represented by a blue triangle icon followed by a JSON-like object. The objects contain fields such as "showDetails", "name", "title", "details", and "\_version\_". The list continues down the page, with a scroll bar visible on the right side.

## **Practices for Lesson 15: Apache Spark**

**Chapter 15**

## Practices for Lesson 15

---

### Practices Overview

In these practices, you will execute a word count exercise by using Scala.

## Practice 15-1: Using Apache Spark

## Overview

In this practice, you will execute a word count exercise using Scala.

You will use interactive Spark shell (spark-shell for Scala) to complete this practice.

## Prerequisite

Open a terminal window and navigate to the following directory. Execute the `reset.sh` script to reset the machine for Spark.

```
cd /home/oracle/exercises/spark  
sh reset.sh
```

## Tasks

1. Open a new terminal window and enter the below command to open an interactive Scala shell.

## spark-shell

**Note:** Spark provides high-level APIs in Java, Scala, and Python programming languages. The `spark-shell` command runs Spark interactively through a modified version of the Scala shell.

- At the Scala prompt, enter the following statement to open the files present in the spark input directory and get a handle to the file data.

```
val files =  
sc.textFile("hdfs://bigdatalite:8020/user/oracle/spark input")
```

```
scala> val files = sc.textFile("hdfs://bigdatalite:8020/user/oracle/spark_input")
15/04/08 07:25:05 INFO MemoryStore: ensureFreeSpace(159554) called with curMem=0, maxMem=309225062
15/04/08 07:25:05 INFO MemoryStore: Block broadcast_0 stored as values to memory (estimated size 155.8 KB, free 294.7 MB)
files: org.apache.spark.rdd.RDD[String] = MappedRDD[1] at textFile at <console>:12

scala>
```

3. Enter the following statement to set up the execution context for the file data.

```
val counts = files.flatMap(line => line.split(" ")).map(word =>
  (word, 1)).reduceByKey(_ + _)
```

```
scala> val counts = files.flatMap(line => line.split(" ")).map(word => (word, 1)).reduceByKey(_ + _)
15/04/08 07:26:27 WARN BlockReaderLocal: The short-circuit local reads feature cannot be used because libhadoop cannot be loaded.
15/04/08 07:26:27 INFO FileInputFormat: Total input paths to process : 2
counts: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[6] at reduceByKey at <console>:14
scala>
```

4. Execute the counts and save the output to a file.

```
counts.saveAsTextFile("hdfs://bigdatalite:8020/user/oracle/spark_output")
```

```
scala> counts.saveAsTextFile("hdfs://bigdatalite:8020/user/oracle/spark_output")
15/04/08 07:28:19 INFO deprecation: mapred.tip.id is deprecated. Instead, use mapreduce.task.id
15/04/08 07:28:19 INFO deprecation: mapred.task.id is deprecated. Instead, use mapreduce.task.attempt.id
15/04/08 07:28:19 INFO deprecation: mapred.task.is.map is deprecated. Instead, use mapreduce.task.ismap
15/04/08 07:28:19 INFO deprecation: mapred.task.partition is deprecated. Instead, use mapreduce.task.partition
15/04/08 07:28:19 INFO deprecation: mapred.job.id is deprecated. Instead, use mapreduce.job.id
15/04/08 07:28:19 INFO SparkContext: Starting job: saveAsTextFile at <console>:17
15/04/08 07:28:19 INFO DAGScheduler: Registering RDD 4 (reduceByKey at <console>:14)
15/04/08 07:28:19 INFO DAGScheduler: Got job 0 (saveAsTextFile at <console>:17) with 3 output partitions (allowLocal=false)
15/04/08 07:28:21 INFO TaskSetManager: Finished TID 3 in 835 ms on localhost (progress: 3/3)
15/04/08 07:28:21 INFO DAGScheduler: Completed ResultTask(0, 0)
15/04/08 07:28:21 INFO TaskSchedulerImpl: Removed TaskSet 0.0, whose tasks have all completed, from pool
15/04/08 07:28:21 INFO DAGScheduler: Stage 0 (saveAsTextFile at <console>:17) finished in 0.837 s
15/04/08 07:28:21 INFO SparkContext: Job finished: saveAsTextFile at <console>:17, took 1.587329033 s
scala>
```

5. Print out the counts to the console and check the generated counts.

```
counts.collect().foreach(println)
```

```
scala> counts.collect().foreach(println)
15/04/08 07:33:11 INFO SparkContext: Starting job: collect at <console>:17
15/04/08 07:33:11 INFO MapOutputTrackerMaster: Size of output statuses for shuffle 0 is 166 bytes
15/04/08 07:33:11 INFO DAGScheduler: Got job 1 (collect at <console>:17) with 3 output partitions (allowLocal=false)
15/04/08 07:33:11 INFO DAGScheduler: Final stage: Stage 2(collect at <console>:17)
15/04/08 07:33:11 INFO DAGScheduler: Parents of final stage: List(Stage 3)
15/04/08 07:33:11 INFO DAGScheduler: Missing parents: List()
15/04/08 07:33:11 INFO DAGScheduler: Submitting Stage 2 (MapPartitionsRDD[6] at reduceByKey at <console>:14), which has no
```

```
15/04/08 07:33:11 INFO DAGScheduler: Stage 2 (collect at <console>:17) finished in 0.096 s
15/04/08 07:33:11 INFO SparkContext: Job finished: collect at <console>:17, took 0.113178296 s
(unreliable,3)
(found,1)
(expensive,6)
(with,2)
(service,8)
(cover,1)
(company,2)
(efficient,1)
(awful,1)
(the,1)
(worthless,2)
(is,1)
(recommend,3)
(protocols,1)
(insurance,11)
(best,2)
(disappointed,3)
(staff,1)
(I,4)
(very,3)
(professional,1)
(and,7)
(customer,4)
(bank,1)
(professionals,1)
(it,3)
(terrible,2)
(will,1)
(good,2)
(worst,8)
scala>
```

6. Open a new terminal window and check the files that are created in HDFS.

```
hadoop fs -ls /user/oracle/spark-output
```

```
[oracle@bigdatalite ~]$ hadoop fs -ls /user/oracle/spark_output
Found 4 items
-rw-r--r-- 1 oracle oracle      0 2015-04-08 07:28 /user/oracle/spark_output/_SUCCESS
-rw-r--r-- 1 oracle oracle    114 2015-04-08 07:28 /user/oracle/spark_output/part-00000
-rw-r--r-- 1 oracle oracle    162 2015-04-08 07:28 /user/oracle/spark_output/part-00001
-rw-r--r-- 1 oracle oracle     66 2015-04-08 07:28 /user/oracle/spark_output/part-00002
[oracle@bigdatalite ~]$
```

7. You can also look at the execution by checking the local executor's webpage at [bigdatalite.localdomain:4040](http://bigdatalite.localdomain:4040). This webpage is generated from Spark shell. Click Stages to see the status.

```
http://bigdatalite.localdomain:4040
```

**Note:** Alternatively, you can also write localhost instead of bigdatalite.localdomain.

The screenshot shows the Spark Shell - Spark Stages interface. The browser address bar contains "bigdatalite.localdomain:4040/stages/". The main content area displays the following statistics:

- Total Duration: 12 min
- Scheduling Mode: FIFO
- Active Stages:** 0
- Completed Stages:** 3
- Failed Stages:** 0

Below these statistics, there are two sections: "Active Stages (0)" and "Completed Stages (3)".

**Active Stages (0)**

Stage Id	Description	Submitted	Duration	Tasks: Succeeded/Total	Shuffle Read	Shuffle Write
2	collect at <console>:17	2015/04/08 07:33:11	96 ms	3/3		
0	saveAsTextFile at <console>:17	2015/04/08 07:28:20	0.8 s	3/3		
1	reduceByKey at <console>:14	2015/04/08 07:28:19	0.4 s	3/3		2037.0 B

**Completed Stages (3)**

Stage Id	Description	Submitted	Duration	Tasks: Succeeded/Total	Shuffle Read	Shuffle Write
2	collect at <console>:17	2015/04/08 07:33:11	96 ms	3/3		
0	saveAsTextFile at <console>:17	2015/04/08 07:28:20	0.8 s	3/3		
1	reduceByKey at <console>:14	2015/04/08 07:28:19	0.4 s	3/3		2037.0 B

**Failed Stages (0)**

Stage Id	Description	Submitted	Duration	Tasks: Succeeded/Total	Shuffle Read	Shuffle Write	Failure Reason
----------	-------------	-----------	----------	------------------------	--------------	---------------	----------------

8. Exit the Scala prompt.

```
:quit
```

```
scala> :quit
Stopping spark context.
15/04/08 07:38:26 INFO SparkUI: Stopped Spark web UI at http://bigdatalite.localdomain:4040
15/04/08 07:38:26 INFO DAGScheduler: Stopping DAGScheduler
15/04/08 07:38:27 INFO MapOutputTrackerMasterActor: MapOutputTrackerActor stopped!
15/04/08 07:38:27 INFO ConnectionManager: Selector thread was interrupted!
15/04/08 07:38:27 INFO ConnectionManager: ConnectionManager stopped
15/04/08 07:38:27 INFO MemoryStore: MemoryStore cleared
15/04/08 07:38:27 INFO BlockManager: BlockManager stopped
15/04/08 07:38:27 INFO BlockManagerMasterActor: Stopping BlockManagerMaster
15/04/08 07:38:27 INFO BlockManagerMaster: BlockManagerMaster stopped
15/04/08 07:38:27 INFO SparkContext: Successfully stopped SparkContext
15/04/08 07:38:27 INFO RemoteActorRefProvider$RemotingTerminator: Shutting down remote daemon.
15/04/08 07:38:27 INFO RemoteActorRefProvider$RemotingTerminator: Remote daemon shut down; proceeding with flushing remote transports.
[oracle@bigdatalite spark]$
```

## **Practices for Lesson 16: Options for Integrating Your Big Data**

**Chapter 16**

## Practices for Lesson 16

---

There are no practices for this lesson.

## **Practices for Lesson 17: Overview of Apache Sqoop**

**Chapter 17**

## Practices for Lesson 17

---

There are no practices for this lesson.

## **Practices for Lesson 18: Using Oracle Loader for Hadoop (OLH)**

**Chapter 18**

## Practices for Lesson 18

---

### Practices Overview

In these practices, you will use Oracle Loader for Hadoop (OLH) to:

- Load data from HDFS files into Oracle Database
- Load data from Hive tables into Oracle Database

## **Guided Practice 18-1: Loading Data from HDFS Files into Oracle Database**

## Overview

In this practice, you learn how to use Oracle Loader for Hadoop (OLH) to load data into a table in Oracle Database. You load data in online mode with the direct path load option.

## Prerequisite

Open a terminal window and navigate to the following directory. Execute the `reset.sh` script to reset the machine for OLH.

```
cd /home/oracle/exercises/olh  
sh reset.sh
```

## Tasks

1. Open a terminal window and navigate to the below directory.

```
cd /home/oracle/exercises/olh  
ls -l
```

```
[oracle@bigdatalite olh]$ cd /home/oracle/exercises/olh
[oracle@bigdatalite olh]$ ls -l
total 36
-rwxrwx--- 1 oracle oinstall 608 Apr  8 08:46 create_movieapp_log_stage_part.q
-rwxrwx--- 1 oracle oinstall 832 Jan 29 2013 loaderMap_moviesession.xml
-rwxrwx--- 1 oracle oinstall 720 Jan 29 2013 moviesession.sql
-rwxrwx--- 1 oracle oinstall 2354 Feb 19 17:48 moviesession.xml
-rwxrwx--- 1 oracle oinstall 437 Feb 19 18:03 movie_tab.sql
-rwxrwx--- 1 oracle oinstall 2134 Feb 19 18:05 olh_hive_part.xml
-rwxrwx--- 1 oracle oinstall 41 Apr  8 13:47 reset.sh
-rwxrwx--- 1 oracle oinstall 529 Apr  8 14:19 runolh_hive_part.sh
-rwxrwx--- 1 oracle oinstall 148 Apr  8 13:54 runolh_session.sh
[oracle@bigdatalite olh]$
```

2. Examine the data present in the HDFS. Execute the below commands to examine the data.

```
hadoop fs -ls moviedemo/session  
hadoop fs -cat moviedemo/session/*00000 | more
```

```
[oracle@bigdatalite ~]$ hadoop fs -ls moviedemo/session
Found 3 items
-rw-r--r-- 1 oracle oracle 0 2013-10-23 14:52 moviedemo/session/_SUCCESS
drwxr-xr-x - oracle oracle 0 2013-10-23 14:52 moviedemo/session/_logs
-rw-r--r-- 1 oracle oracle 328026 2014-04-25 17:53 moviedemo/session/part-r-00000
[oracle@bigdatalite ~]$ hadoop fs -cat moviedemo/session/*00000 | more
471034351265778000 2010-02-10:02:45:16 1000050 305 0 0 0 0 0 0 0 0
152801481293166800 2010-12-24:18:29:59 1000050 483 0 0 0 0 0 0 0 0 0
164123601346472000 2012-09-01:13:15:03 1000083 507 0 0 0 0 0 0 0 0 0
53212591346472000 2012-09-01:08:49:22 1000083 543 0 0 0 0 0 0 0 0 0
255571981291957200 2010-12-10:28:04:11 1000083 7434 1 112 1 5556 1256 1 1 361 0 0
42636791286510400 2010-10-08:09:13:18 1000104 253 0 0 0 0 0 0 0 0 0
126472321348891200 2012-09-29:22:38:43 1000186 1962 0 0 0 0 0 0 0 1 569 0
158628731287806400 2010-10-23:18:03:16 1000498 1030 0 0 0 0 0 0 0 1 391 0
1579811344139200 2012-08-05:22:58:09 1000498 1192 0 0 0 0 0 0 0 4 891 0
246849441312603200 2011-08-06:13:14:59 1000672 137 0 0 0 0 0 0 0 0 0
67480361343448000 2012-07-28:00:41:11 1000679 1129 0 0 0 0 0 0 0 1 1129 0
236599611348286400 2012-09-22:16:48:00 1000817 426 0 0 0 0 0 0 0 0 0
350506161282104000 2010-08-18:17:46:01 1000817 443 1 443 0 0 0 0 0 0 0
--More--
```

Keep pressing the Enter key to load more data. Use :q to stop loading more data.

3. Examine the moviesession.xml file.

```
cat moviesession.xml
```

```
[oracle@bigdatalite olh]$ cat moviesession.xml
<?xml version="1.0" encoding="UTF-8" ?>
<configuration>

<!-- Input settings -->

<property>
  <name>mapreduce.inputformat.class</name>
  <value>oracle.hadoop.loader.lib.input.DelimitedTextInputFormat</value>
</property>

<property>
  <name>mapred.input.dir</name>
  <value>/user/oracle/moviedemo/session/*00000</value>
</property>

<property>
  <name>oracle.hadoop.loader.input.fieldTerminator</name>
  <value>\u0009</value>
</property>
```

```
<!-- Output settings -->

<property>
  <name>mapreduce.outputformat.class</name>
  <value>oracle.hadoop.loader.lib.output.OCIOutputFormat</value>
</property>

<property>
  <name>mapred.output.dir</name>
  <value>temp_out_session</value>
</property>

<!-- Table information -->

<property>
  <name>oracle.hadoop.loader.loaderMap.targetTable</name>
  <value>MOVIE_SESSIONS_TAB</value>
</property>

<property>
  <name>oracle.hadoop.loader.input.fieldNames</name>
  <value>SESSION_ID,TIME_ID,CUST_ID,DURATION_SESSION,NUM_RATED,DURATION_RATED,NUM_COMPLETED,DURATION_COMPLETED,TIME_TO_FIRST_START,NUM_STARTED,NUM_BROWSERED,DURATION_BROWSERED,NUM_LISTED,DURATION_LISTED,NUM_INCOMPLETE,NUM_SEARCHED</value>
</property>

<property>
  <name>oracle.hadoop.loader.defaultDateFormat</name>
  <value>yyyy-MM-dd:HH:mm:ss</value>
</property>
```

```
<!-- Connection information -->

<property>
    <name>oracle.hadoop.loader.connection.url</name>
    <value>jdbc:oracle:thin:@${HOST}:${TCPPORT}/${SERVICE_NAME}</value>
</property>

<property>
    <name>TCPPORT</name>
    <value>1521</value>
</property>

<property>
    <name>HOST</name>
    <value>bigdatalite.localdomain</value>
</property>

<property>
    <name>SERVICE_NAME</name>
    <value>orcl</value>
</property>

<property>
    <name>oracle.hadoop.loader.connection.user</name>
    <value>MOVIEDEMO</value>
</property>

<property>
    <name>oracle.hadoop.loader.connection.password</name>
    <value>welcome1</value>
    <description> Having password in cleartext is NOT RECOMMENDED - use Oracle Wallet instead </description>
</property>

<property>
    <name>oracle.hadoop.loader.logBadRecords</name>
    <value>true</value>
</property>

</configuration>
```

**Note:** This file contains the configuration parameters for the execution of OLH. You can review the file to see the parameters, including:

- `mapreduce.inputformat.class`: Specifies the input format of the input data file. In this example, the input data is delimited text, so the value for this parameter is the class name  
`oracle.hadoop.loader.lib.input.DelimitedTextInputFormat`.
- `oracle.hadoop.loader.input.fieldTerminator`: Specifies the character that is used as a field terminator in the input data file. In this example, the field terminator is Tab (represented by its hex value).
- `mapreduce.outputformat.class`: Specifies the type of load. We specify here the value `OCIOOutputFormat` to use the direct path online load option.
- `mapred.input.dir`: Location of the input data file on HDFS
- `mapred.output.dir`: Specifies the HDFS directory where output files should be written, such as the `_SUCCESS` and `_log` files
- `oracle.hadoop.loader.loaderMap.targetTable`: Specifies the name of the target table
- `oracle.hadoop.loader.input.fieldNames`: Specifies column names of the target table

4. Examine `moviesession.sql` file. This is the script to drop any existing `movie_sessions_tab` table and create a new `movie_sessions_tab` table.

```
cat moviesession.sql
```

```
[oracle@bigdatalite olh]$ cat moviesession.sql
SET ECHO ON

-- drop table if leftover from previous labs
DROP TABLE MOVIE_SESSIONS_TAB;

CREATE TABLE MOVIE_SESSIONS_TAB
(
    "SESSION_ID" NUMBER,
    "TIME_ID" DATE,
    "CUST_ID"      NUMBER,
    "DURATION_SESSION" NUMBER,
    "NUM_RATED"    NUMBER,
    "DURATION_RATED" NUMBER,
    "NUM_COMPLETED" NUMBER,
    "DURATION_COMPLETED" NUMBER,
    "TIME_TO_FIRST_START" NUMBER,
    "NUM_STARTED"   NUMBER,
    "NUM_BROWSERED" NUMBER,
    "DURATION_BROWSERED" NUMBER,
    "NUM_LISTED"    NUMBER,
    "DURATION_LISTED" NUMBER,
    "NUM_INCOMPLETE" NUMBER,
    "NUM_SEARCHED"   NUMBER
)
PARTITION BY HASH(CUST_ID);

SET ECHO OFF
[oracle@bigdatalite olh]$
```

5. Create the target table `movie_sessions_tab` in the database, where the data needs to be loaded. This can be achieved by executing the `moviesession.sql` file.

```
sqlplus moviedemo/welcome1
@moviesession.sql
exit
```

```
[oracle@bigdatalite olh]$ sqlplus moviedemo/welcome1

SQL*Plus: Release 12.1.0.2.0 Production on Tue Feb 24 13:22:45 2015

Copyright (c) 1982, 2014, Oracle. All rights reserved.

Last Successful login time: Tue Feb 24 2015 11:14:01 -05:00

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing options

SQL>
```

```

SQL> @moviesession.sql
SQL>
SQL> -- drop table if leftover from previous labs
SQL> DROP TABLE MOVIE_SESSIONS_TAB;

DROP TABLE MOVIE_SESSIONS_TAB
*
ERROR at line 1:
ORA-00942: table or view does not exist

SQL>
SQL> CREATE TABLE MOVIE_SESSIONS_TAB
  2  (
  3      "SESSION_ID" NUMBER,
  4      "TIME_ID" DATE,
  5      "CUST_ID"      NUMBER,
  6      "DURATION_SESSION" NUMBER,
  7      "NUM_RATED"    NUMBER,
  8      "DURATION_RATED" NUMBER,
  9      "NUM_COMPLETED" NUMBER,
 10     "DURATION_COMPLETED" NUMBER,
 11     "TIME_TO_FIRST_START" NUMBER,
 12     "NUM_STARTED"   NUMBER,
 13     "NUM_BROWSED"   NUMBER,
 14     "DURATION_BROWSER" NUMBER,
 15     "NUM_LISTED"    NUMBER,
 16     "DURATION_LISTED" NUMBER,
 17     "NUM_INCOMPLETE" NUMBER,
 18     "NUM_SEARCHED"   NUMBER
 19  )
20 PARTITION BY HASH(CUST_ID);

Table created.

SQL>
SQL> SET ECHO OFF
SQL> SQL> exit
Disconnected from Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing options
[oracle@bigdatalite olh]$ █

```

**Note:** This table is hash-partitioned on the `cust_id` column.

- Examine the `runolh_session.sh` script. This is the script to invoke Oracle Loader for Hadoop, which runs as a MapReduce job on the Hadoop cluster. It uses `moviesession.xml`, the file containing the configuration parameters.

```
cat runolh_session.sh
```

```
[oracle@bigdatalite olh]$ cat runolh_session.sh
hadoop jar ${OLH_HOME}/jlib/oraloader.jar \
  oracle.hadoop.loader.OraLoader \
  -conf moviesession.xml \
  -D mapred.reduce.tasks=2
[oracle@bigdatalite olh]$ █
```

- Run the `runolh_session.sh` script to invoke the OLH job to load the session data. This starts the MapReduce job, which loads the data into the target table.

```
sh runolh_session.sh
```

```
[oracle@bigdatalite olh]$ sh runolh_session.sh
Oracle Loader for Hadoop Release 3.2.0 - Production

Copyright (c) 2011, 2014, Oracle and/or its affiliates. All rights reserved.

15/02/24 13:37:49 INFO loader.OraLoader: Oracle Loader for Hadoop Release 3.2.0 - Production

Copyright (c) 2011, 2014, Oracle and/or its affiliates. All rights reserved.

15/02/24 13:37:49 INFO loader.OraLoader: Built-Against: hadoop-2.2.0 hive-0.13.0 avro-1.7.3 ja
15/02/24 13:37:50 INFO Configuration.deprecation: mapreduce.outputformat.class is deprecated.
tformat.class
15/02/24 13:37:50 INFO Configuration.deprecation: mapred.output.dir is deprecated. Instead, us
mat.outputdir
15/02/24 13:37:58 INFO Configuration.deprecation: mapred.submit.replication is deprecated. Ins
t.file.replication
15/02/24 13:37:59 INFO loader.OraLoader: oracle.hadoop.loader.enableSorting disabled, no sorti
15/02/24 13:37:59 INFO Configuration.deprecation: mapreduce.outputformat.class is deprecated.
tformat.class
```

```
Spilled Records=9052
Shuffled Maps =2
Failed Shuffles=0
Merged Map outputs=2
GC time elapsed (ms)=571
CPU time spent (ms)=11730
Physical memory (bytes) snapshot=691511296
Virtual memory (bytes) snapshot=3070218240
Total committed heap usage (bytes)=438304768
Rows skipped by input error
Parse Error=1
Total rows skipped by input error=1
Shuffle Errors
BAD_ID=0
CONNECTION=0
IO_ERROR=0
WRONG_LENGTH=0
WRONG_MAP=0
WRONG_REDUCE=0
File Input Format Counters
Bytes Read=328026
File Output Format Counters
Bytes Written=3414
15/02/24 13:39:00 INFO Configuration.deprecation: mapred.output.dir is deprecated. Instead, use mapred
mat.outputdir
[oracle@bigdatalite olh]$ █
```

**Note:** One row had a parse error. The .bad file containing the row and the error are logged in the \_olh directory under the directory specified in mapred.output.dir.

- After the MapReduce job is completed, check to see if the rows are loaded.

```
sqlplus moviedemo/welcome1
select count(*) from movie_sessions_tab;
```

```
[oracle@bigdatalite olh]$ sqlplus moviedemo/welcome1
SQL*Plus: Release 12.1.0.2.0 Production on Tue Feb 24 13:46:34 2015
Copyright (c) 1982, 2014, Oracle. All rights reserved.

Last Successful login time: Tue Feb 24 2015 13:38:52 -05:00

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing options

SQL> select count(*) from movie_sessions_tab;
  COUNT(*)
  -----
        4526
SQL>
```

9. The table is available for querying. You can execute the following queries, or you can try your own queries.

```
select cust_id from movie_sessions_tab where rownum < 10;

SQL> select cust_id from movie_sessions_tab where rownum < 10;
  CUST_ID
  -----
1446693
1446693
1446522
1446522
1446522
1446522
1446440
1446414
1446414

9 rows selected.

SQL>
```

## Guided Practice 18-2: Loading Data from Hive Tables into Oracle Database

### Overview

In this practice, you will load data from a Hive table into a table in Oracle Database. Because the Hive table is partitioned, an additional property can be specified to selectively load partitions.

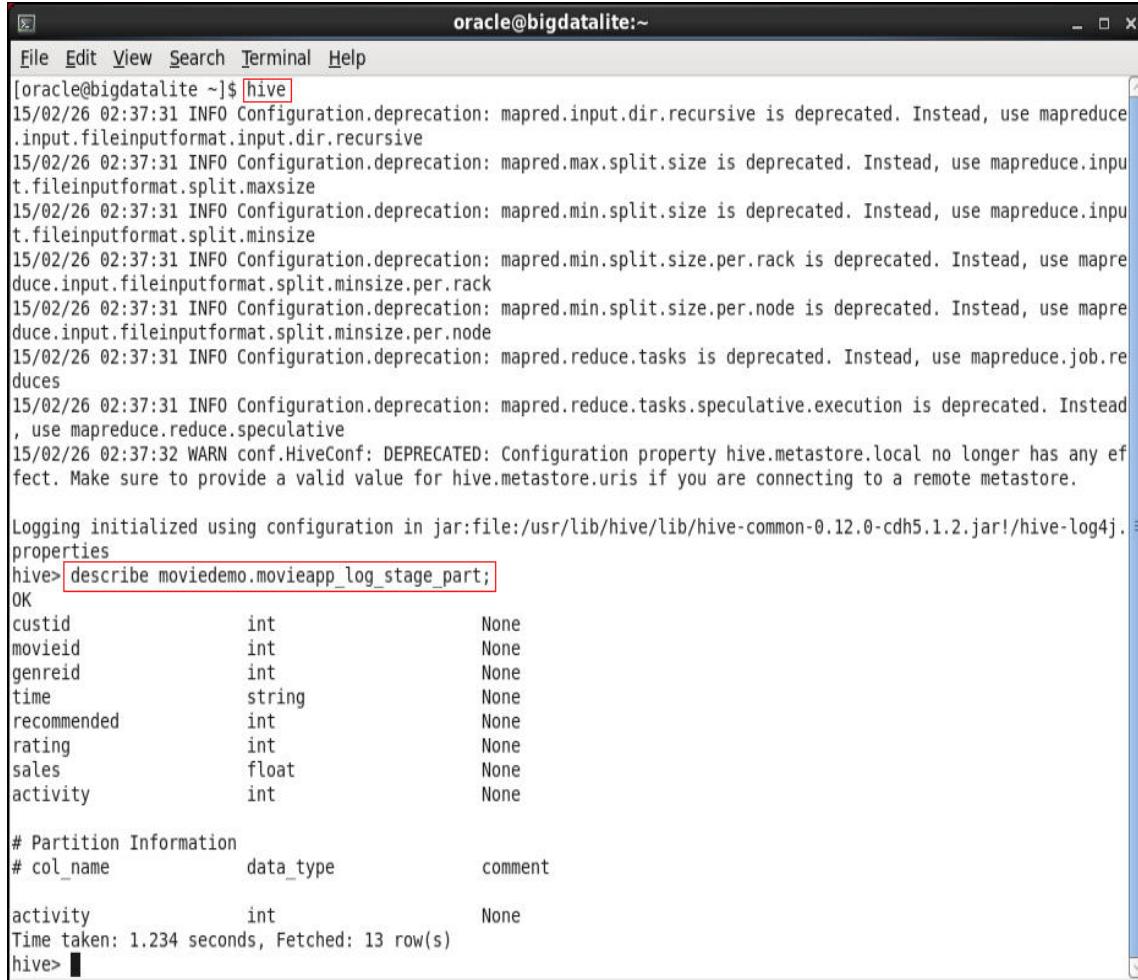
### Assumptions

Practice 18-1 is completed.

### Tasks

1. Open a new terminal. Execute the below commands to examine the data present in the Hive table.

```
hive
describe moviedemo.movieapp_log_stage_part;
select count(*) from moviedemo.movieapp_log_stage_part where
activity = 2;
quit;
```



The screenshot shows a terminal window titled "oracle@bigdatalite:~". The user has run the following Hive commands:

```
[oracle@bigdatalite ~]$ hive
15/02/26 02:37:31 INFO Configuration.deprecation: mapred.input.dir.recursive is deprecated. Instead, use mapreduce.input.fileinputformat.input.dir.recursive
15/02/26 02:37:31 INFO Configuration.deprecation: mapred.max.split.size is deprecated. Instead, use mapreduce.input.fileinputformat.split.maxsize
15/02/26 02:37:31 INFO Configuration.deprecation: mapred.min.split.size is deprecated. Instead, use mapreduce.input.fileinputformat.split.minsize
15/02/26 02:37:31 INFO Configuration.deprecation: mapred.min.split.size.per.rack is deprecated. Instead, use mapreduce.input.fileinputformat.split.minsize.per.rack
15/02/26 02:37:31 INFO Configuration.deprecation: mapred.min.split.size.per.node is deprecated. Instead, use mapreduce.input.fileinputformat.split.minsize.per.node
15/02/26 02:37:31 INFO Configuration.deprecation: mapred.reduce.tasks is deprecated. Instead, use mapreduce.job.reduces
15/02/26 02:37:31 INFO Configuration.deprecation: mapred.reduce.tasks.speculative.execution is deprecated. Instead, use mapreduce.reduce.speculative
15/02/26 02:37:32 WARN conf.HiveConf: DEPRECATED: Configuration property hive.metastore.local no longer has any effect. Make sure to provide a valid value for hive.metastore.uris if you are connecting to a remote metastore.

Logging initialized using configuration in jar:file:/usr/lib/hive/lib/hive-common-0.12.0-cdh5.1.2.jar!/hive-log4j.properties
hive> describe moviedemo.movieapp_log_stage_part;
OK
custid          int           None
movieid         int           None
genreid         int           None
time            string        None
recommended    int           None
rating          int           None
sales           float         None
activity        int           None

# Partition Information
# col_name      data_type   comment
activity        int           None
Time taken: 1.234 seconds, Fetched: 13 row(s)
hive>
```

```

hive> select count(*) from moviedemo.movieapp_log_stage_part where activity = 2;
Total MapReduce jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapred.reduce.tasks=<number>
Starting Job = job_1424933626501_0003, Tracking URL = http://bigdatalite.localdomain:8088/proxy/application_1424933626501_0003/
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1424933626501_0003
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2015-02-26 02:40:55,347 Stage-1 map = 0%, reduce = 0%
2015-02-26 02:41:04,164 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 1.57 sec
2015-02-26 02:41:05,211 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 1.57 sec
2015-02-26 02:41:06,282 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 1.57 sec
2015-02-26 02:41:07,359 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 1.57 sec
2015-02-26 02:41:08,425 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 1.57 sec
2015-02-26 02:41:09,562 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 1.57 sec
2015-02-26 02:41:10,670 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 1.57 sec
2015-02-26 02:41:11,768 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 1.57 sec
2015-02-26 02:41:12,814 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 3.62 sec
2015-02-26 02:41:13,856 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 3.62 sec
MapReduce Total cumulative CPU time: 3 seconds 620 msec
Ended Job = job_1424933626501_0003
MapReduce Jobs Launched:
Job 0: Map: 1 Reduce: 1 Cumulative CPU: 3.62 sec HDFS Read: 191089 HDFS Write: 5 SUCCESS
Total MapReduce CPU Time Spent: 3 seconds 620 msec
OK
4390
Time taken: 34.898 seconds, Fetched: 1 row(s)
hive> 
```

2. Open a terminal and navigate to the below directory.

```

cd /home/oracle/exercises/olh
ls -l
[oracle@bigdatalite olh]$ cd /home/oracle/exercises/olh
[oracle@bigdatalite olh]$ ls -l
total 36
-rwxrwx---. 1 oracle oinstall 608 Apr  8 08:46 create_movieapp_log_stage_part.q
-rwxrwx---. 1 oracle oinstall 832 Jan 29 2013 loaderMap_moviesession.xml
-rwxrwx---. 1 oracle oinstall 720 Jan 29 2013 moviesession.sql
-rwxrwx---. 1 oracle oinstall 2354 Feb 19 17:48 moviesession.xml
-rwxrwx---. 1 oracle oinstall 437 Feb 19 18:03 movie_tab.sql
-rwxrwx---. 1 oracle oinstall 2134 Feb 19 18:05 olh_hive_part.xml
-rwxrwx---. 1 oracle oinstall 41 Apr  8 13:47 reset.sh
-rwxrwx---. 1 oracle oinstall 529 Apr  8 14:19 runolh_hive_part.sh
-rwxrwx---. 1 oracle oinstall 148 Apr  8 13:54 runolh_session.sh
[oracle@bigdatalite olh]$ 
```

3. Examine the olh\_hive\_part.xml file.

```

cat olh_hive_part.xml 
```

```
[oracle@bigdatalite olh]$ cat olh_hive_part.xml
<?xml version="1.0" encoding="UTF-8" ?>
<configuration>

<!-- Input settings -->

<property>
  <name>mapreduce.inputformat.class</name>
  <value>oracle.hadoop.loader.lib.input.HiveToAvroInputFormat</value>
</property>

<property>
  <name>oracle.hadoop.loader.input.hive.databaseName</name>
  <value>moviedemo</value>
</property>

<property>
  <name>oracle.hadoop.loader.input.hive.tableName</name>
  <value>movieapp_log_stage_part</value>
</property>

<property>
  <name>oracle.hadoop.loader.input.fieldTerminator</name>
  <value>\u0009</value>
</property>

<!-- Output settings -->

<property>
  <name>mapreduce.outputformat.class</name>
  <value>oracle.hadoop.loader.lib.output.OCIOutputFormat</value>
</property>

<property>
  <name>mapreduce.output.fileoutputformat.outputdir</name>
  <value>temp_out_session_p</value>
</property>

<!-- Table information -->

<property>
  <name>oracle.hadoop.loader.loaderMap.targetTable</name>
  <value>MOVIE_LOCAL_TAB</value>
</property>

<property>
  <name>oracle.hadoop.loader.input.fieldNames</name>
  <value>CUSTID,MOVIEID,GENREID,TIME,RECOMMENDED,RATING,ACTIVITY,SALES</value>
</property>

<property>
  <name>oracle.hadoop.loader.defaultDateFormat</name>
  <value>yyyy-MM-dd:HH:mm:ss</value>
</property>
```

```
<!-- Connection information -->

<property>
    <name>oracle.hadoop.loader.connection.url</name>
    <value>jdbc:oracle:thin:@bigdatalite.localdomain:1521/orcl</value>
</property>

<property>
    <name>oracle.hadoop.loader.connection.user</name>
    <value>MOVIEDEMO</value>
</property>

<property>
    <name>oracle.hadoop.loader.connection.password</name>
    <value>welcome1</value>
<description> Having password in cleartext is NOT RECOMMENDED - use Oracle Wallet instead </description>
</property>

<property>
    <name>oracle.hadoop.loader.logBadRecords</name>
    <value>true</value>
</property>

</configuration>

[oracle@bigdatalite olh]$
```

**Note:** This file contains the configuration parameters for the execution of OLH. You can review the file to see the parameters, including:

- `mapreduce.inputformat.class`: Specifies the input format of the input data as Hive
  - `oracle.hadoop.loader.input.hive.tableName`: Name of the Hive table
  - `oracle.hadoop.loader.input.hive.databaseName`: Name of the Hive database
4. Examine the `movie_tab.sql` file. This is the script to drop any existing `movie_local_tab` table and create a new `movie_local_tab` table.

```
cat movie_tab.sql
```

```
[oracle@bigdatalite olh]$ cat movie_tab.sql
SET ECHO ON

-- drop table if leftover from previous labs
DROP TABLE MOVIE_LOCAL_TAB;

CREATE TABLE MOVIE_LOCAL_TAB
(
    "CUSTID"          NUMBER,
    "MOVIEID"         NUMBER,
    "GENREID"         NUMBER,
    "TIME"            DATE,
    "RECOMMENDED"    NUMBER,
    "RATING"          NUMBER,
    "ACTIVITY"        NUMBER,
    "SALES"           NUMBER
)
PARTITION BY HASH(ACTIVITY);

SET ECHO OFF

[oracle@bigdatalite olh]$
```

5. Create the target table movie\_local\_tab in the database, where the data needs to be loaded. This can be achieved by executing the movie\_tab.sql file.

```
sqlplus moviedemo/welcome1
@movie_tab.sql
exit
```

```
[oracle@bigdatalite olh]$ sqlplus moviedemo/welcome1
SQL*Plus: Release 12.1.0.2.0 Production on Tue Feb 24 14:30:33 2015
Copyright (c) 1982, 2014, Oracle. All rights reserved.

Last Successful login time: Tue Feb 24 2015 14:01:14 -05:00

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing options

SQL> @movie_tab.sql
SQL>
SQL> -- drop table if leftover from previous labs
SQL> DROP TABLE MOVIE_LOCAL_TAB;
DROP TABLE MOVIE_LOCAL_TAB
*
ERROR at line 1:
ORA-00942: table or view does not exist

SQL>
SQL> CREATE TABLE MOVIE_LOCAL_TAB
  2  (
  3      "CUSTID"          NUMBER,
  4      "MOVIEID"          NUMBER,
  5      "GENREID"          NUMBER,
  6      "TIME"              DATE,
  7      "RECOMMENDED"      NUMBER,
  8      "RATING"            NUMBER,
  9      "ACTIVITY"          NUMBER,
 10      "SALES"             NUMBER
 11  )
 12 PARTITION BY HASH(ACTIVITY);

Table created.

SQL>
SQL> SET ECHO OFF
SQL> exit
Disconnected from Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production
```

6. Examine the `runolh_hive_part.sh` script. This is the script to invoke Oracle Loader for Hadoop, which runs as a MapReduce job. It uses `olh_hive_part.xml`, the file containing the configuration parameters.

```
cat runolh_hive_part.sh
```

```
[oracle@bigdatalite olh]$ cat runolh_hive_part.sh
# Add HIVE_HOME/lib* to HADOOP_CLASSPATH.  This is not preset
# in the VM since this breaks Pig in the previous lab.

export HADOOP_CLASSPATH=$HADOOP_CLASSPATH:$HIVE_HOME/lib/*

hadoop jar ${OLH_HOME}/jlib/oraloader.jar \
  oracle.hadoop.loader.OraLoader \
  -conf olh_hive_part.xml \
  -libjars /usr/lib/hive/lib/hive-exec.jar,/usr/lib/hive/lib/hive-metastore.jar,/usr/lib/hive/lib/libfb303-0.9.0.jar \
  -D mapred.reduce.tasks=4 \
  -D oracle.hadoop.loader.input.hive.partitionFilter='activity < 4'

[oracle@bigdatalite olh]$
```

**Note:** `oracle.hadoop.loader.input.hive.partitionFilter` is a partition filter clause specified using HiveQL syntax to identify the partitions to load. If this property is

not specified, all partitions will be loaded. Here, you specify this property as a `-D` parameter. It has the value `activity < 4`, so only partitions that satisfy this condition will be loaded. Note that the partition filter should not include any non-partitioned columns. It should include only partitioned columns. Including non-partitioned columns will not raise an error but can give inconsistent results.

- Run the `runolh_hive_part.sh` script to invoke the OLH job to load the session data. This starts the MapReduce job, which loads the data into the target table.

```
sh runolh_hive_part.sh
```

```
[oracle@bigdatalite olh]$ sh runolh_hive_part.sh
Oracle Loader for Hadoop Release 3.2.0 - Production

Copyright (c) 2011, 2014, Oracle and/or its affiliates. All rights reserved.

15/02/24 15:37:42 INFO Configuration.deprecation: mapred.reduce.tasks is deprecated. Instead, use mapreduce.job.reduces
15/02/24 15:37:42 INFO loader.OraLoader: Oracle Loader for Hadoop Release 3.2.0 - Production

Copyright (c) 2011, 2014, Oracle and/or its affiliates. All rights reserved.

15/02/24 15:37:42 INFO loader.OraLoader: Built-Against: hadoop-2.2.0 hive-0.13.0 avro-1.7.3 jackson-1.8.8
15/02/24 15:37:43 INFO Configuration.deprecation: mapreduce.outputformat.class is deprecated. Instead, use mapreduce.job.outp
utformat.class
15/02/24 15:37:43 INFO Configuration.deprecation: mapred.output.dir is deprecated. Instead, use mapreduce.output.fileoutputfo
rmat.outputdir
15/02/24 15:37:50 INFO Configuration.deprecation: mapred.submit.replication is deprecated. Instead, use mapreduce.client.subm
it.file.replication
15/02/24 15:37:50 INFO loader.OraLoader: oracle.hadoop.loader.enableSorting disabled, no sorting key provided
```

```
Map-Reduce Framework
  Map input records=10453
  Map output records=10453
  Map output bytes=722605
  Map output materialized bytes=743559
  Input split bytes=2496
  Combine input records=0
  Combine output records=0
  Reduce input groups=1
  Reduce shuffle bytes=743559
  Reduce input records=10453
  Reduce output records=10453
  Spilled Records=20906
  Shuffled Maps =8
  Failed Shuffles=0
  Merged Map outputs=8
  GC time elapsed (ms)=1809
  CPU time spent (ms)=22300
  Physical memory (bytes) snapshot=1319796736
  Virtual memory (bytes) snapshot=6117388288
  Total committed heap usage (bytes)=843579392

Shuffle Errors
  BAD_ID=0
  CONNECTION=0
  IO_ERROR=0
  WRONG_LENGTH=0
  WRONG_MAP=0
  WRONG_REDUCE=0

File Input Format Counters
  Bytes Read=0
File Output Format Counters
  Bytes Written=6458
[oracle@bigdatalite olh]$
```

8. After the MapReduce job is completed, check to see if the rows are loaded.

```
sqlplus moviedemo/welcome1
select count(*) from movie_local_tab where activity = 2;
```

```
[oracle@bigdatalite olh]$ sqlplus moviedemo/welcome1

SQL*Plus: Release 12.1.0.2.0 Production on Tue Feb 24 15:40:21 2015

Copyright (c) 1982, 2014, Oracle. All rights reserved.

Last Successful login time: Tue Feb 24 2015 15:39:02 -05:00

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing options

SQL> select count(*) from movie_local_tab where activity = 2;

  COUNT(*)
  -----
        4390

SQL> █
```

**Note:** Notice that the number of rows fetched by the query in the Hive table is same as the number of rows fetched by a similar query in this step.



## **Practices for Lesson 19: Using Copy to BDA**

**Chapter 19**

## Practices for Lesson 19

---

### Practices Overview

Copy to BDA is a feature of Oracle Big Data SQL. It enables you to copy Oracle Database tables to Oracle Big Data Appliance for query within Hive.

As you learned in the lesson, common usage includes these primary steps:

- Identify the target directory using an Oracle Directory object.
- Create an external table with the ORACLE\_DATAPUMP access driver. This driver creates an external table and populates the external Data Pump format files with Oracle Database table data.
- Copy the data pump files to Hadoop.
- Create a Hive table over the data pump files. The data pump files can then be queried with Hive and any Hadoop application that can access a Hive table.

## Practice 19-1: View Source Data and Identify Target Directory

### Overview

In this guided practice, you use SQL Plus to view the Oracle table that is to be copied (`movie_fact`). Then, you create an Oracle Directory object to identify the target directory in Hadoop.

### Assumptions

To ensure accuracy of command line statements, copy/paste commands from **lab\_19\_01.txt**

### Tasks

1. Open a Terminal window, and execute the following at the command prompt:

```
export HADOOP_CLASSPATH=/u01/c2bda/orahivedp-1.0/jlib/*:$HADOOP_CLASSPATH
```

#### Note:

- This export command is not a normal requirement for Copy To BDA. However, it is necessary for the current classroom environment.
- Use this same terminal window for the remainder of the practices for lesson 19. Do not use a separate terminal window for these three practices.

2. View the Oracle table to be copied by launching SQL Plus in the Terminal window.

```
$ sqlplus moviedemo/welcome1
SQL> desc movie_fact
```

The screenshot shows a terminal window titled "oracle@bigdatalite:~". The window displays the following SQL\*Plus session:

```
File Edit View Search Terminal Help
SQL*Plus: Release 12.1.0.2.0 Production on Wed Feb 11 12:20:48 2015
Copyright (c) 1982, 2014, Oracle. All rights reserved.

Last Successful login time: Fri Nov 14 2014 07:15:39 -05:00

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing
ns

SQL> desc movie_fact
Name          Null?    Type
-----        -----
CUST_ID           NUMBER
MOVIE_ID          NUMBER
GENRE_ID          NUMBER
TIME_ID           DATE
RECOMMENDED       NUMBER
ACTIVITY_ID       NUMBER
RATING            NUMBER
SALES             NUMBER
```

3. Execute the following two SQL commands to view information about the table:

```
select count(*) from movie_fact;
select cust_id, time_id from movie_fact where cust_id = 1082895 and
movie_id = 1166958;
```

```
oracle@bigdatalite:~ 
File Edit View Search Terminal Help
SQL> select count(*) from movie_fact;

COUNT(*)
-----
3259348

SQL> select cust_id, time_id from movie_fact where cust_id = 1082895 and
= 1166958;

CUST_ID TIME_ID
-----
1082895 06-JAN-10
1082895 29-APR-10
1082895 29-APR-10
1082895 09-JAN-11
1082895 04-MAR-11
1082895 04-MAR-11
1082895 04-MAR-11
1082895 04-MAR-11
1082895 20-MAR-11
1082895 15-JUL-11
1082895 07-AUG-11

CUST_ID TIME_ID
-----
1082895 13-AUG-11
1082895 07-AUG-11
1082895 06-NOV-11

14 rows selected.

SQL> ■
```

4. Create an Oracle Directory object, using the following command:

```
CREATE DIRECTORY dp_data_dir as
'/home/oracle/movie/moviework/c2bda';
```

```
SQL> CREATE DIRECTORY dp_data_dir as '/home/oracle/movie/moviework/c2bda';

Directory created.

SQL> ■
```

5. Leave the SQL Plus session open.

## Practice 19-2: Create External Tables and Copy the Data

### Overview

In this guided practice, you create two external tables from the `movie_fact` table.

### Assumptions

Practice 19-1 has been successfully completed.

### Tasks

1. In the same SQL Plus session as the previous practice, create an external table using the following syntax: (Note: To ensure accuracy, copy/paste this statement from **lab\_19\_02.txt**)

```
CREATE TABLE export_movie_fact
    ORGANIZATION EXTERNAL (
        TYPE oracle_datapump
        DEFAULT DIRECTORY dp_data_dir
        LOCATION('movie_fact1.dmp', 'movie_fact2.dmp',
                 'movie_fact3.dmp', 'movie_fact4.dmp')
    ) PARALLEL 4
    AS SELECT * FROM movie_fact;
```

```
SQL> CREATE TABLE export_movie_fact ORGANIZATION EXTERNAL ( TYPE oracle_datapump
  DEFAULT DIRECTORY dp_data_dir LOCATION ( 'movie_fact1.dmp', 'movie_fact2.dmp',
'movie_fact3.dmp', 'movie_fact4.dmp' ) ) PARALLEL 4 AS SELECT * from movie_fact;
Table created.
```

**Note:** There are 4 `LOCATION` files, and there is a `PARALLEL 4` clause. Therefore, data will be added to the four .dmp files in parallel.

2. Now, create a second external table, corresponding to the second query in step 2 of the last practice. (Note: To ensure accuracy, copy/paste this statement from **lab\_19\_02.txt**)

```
CREATE TABLE export_movie_fact_query
    ORGANIZATION EXTERNAL (
        TYPE oracle_datapump
        DEFAULT DIRECTORY dp_data_dir
        LOCATION('movie_fact_1082895_1166958.dmp')
    ) PARALLEL
    AS SELECT cust_id, time_id
    FROM movie_fact
    WHERE cust_id = 1082895 AND movie_id = 1166958;
```

```
SQL> CREATE TABLE export_movie_fact_query ORGANIZATION EXTERNAL (TYPE oracle_datapump
  DEFAULT DIRECTORY dp_data_dir LOCATION ('movie_fact_1082895_1166958.dmp'))
  PARALLEL AS SELECT cust_id, time_id FROM movie_fact WHERE cust_id = 1082895 AND
  movie_id = 1166958;
Table created.
```

3. Verify the contents of these external tables.

```
SELECT * FROM export_movie_fact_query;  
SELECT COUNT(*) FROM export_movie_fact;
```

```
SQL> SELECT * FROM export_movie_fact_query;
```

```
CUST_ID TIME_ID
```

```
-----  
1082895 06-JAN-10  
1082895 29-APR-10  
1082895 29-APR-10  
1082895 06-NOV-11  
1082895 09-JAN-11  
1082895 04-MAR-11  
1082895 04-MAR-11  
1082895 04-MAR-11  
1082895 04-MAR-11  
1082895 20-MAR-11  
1082895 15-JUL-11
```

```
CUST_ID TIME_ID
```

```
-----  
1082895 07-AUG-11  
1082895 13-AUG-11  
1082895 07-AUG-11
```

```
14 rows selected.
```

```
SQL> SELECT COUNT(*) FROM export_movie_fact;
```

```
COUNT(*)
```

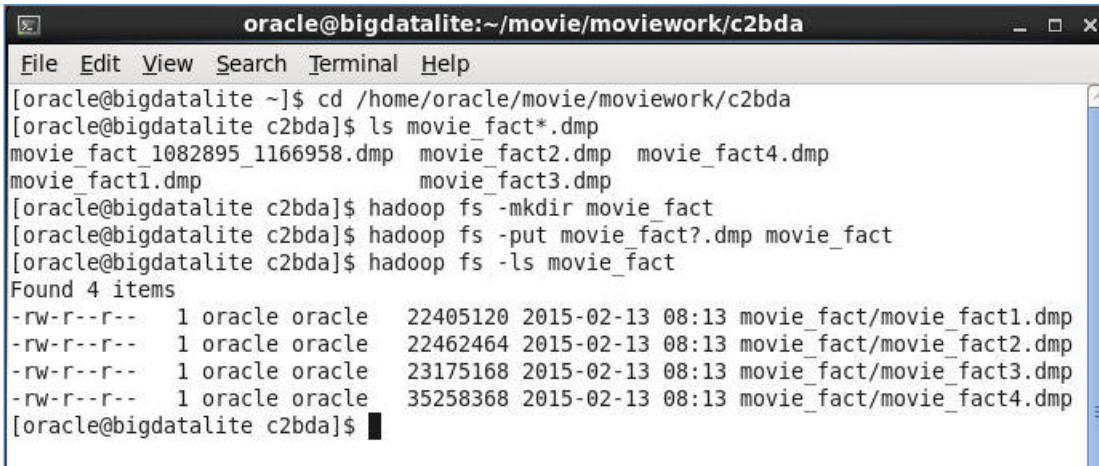
```
-----  
3259348
```

```
SQL> █
```

**Note:** These queries point to the external tables, which reference the data pump files in /home/oracle/movie/moviework/c2bda

4. Exit the SQL Plus session, but leave the terminal window open.
5. In the same terminal window, copy the data pump files from the local file system to HDFS for the export\_movie\_fact table, and then view the files in their new location, by executing the following commands at the \$ prompt:

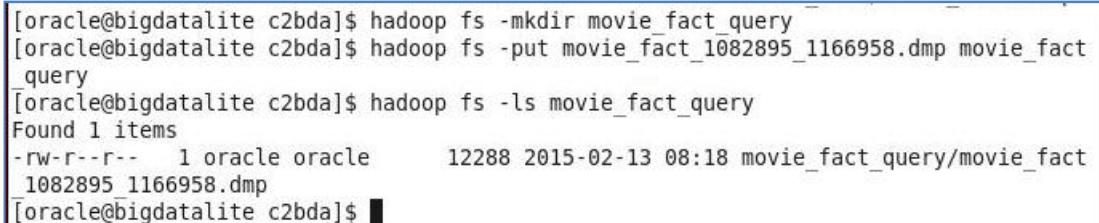
```
cd /home/oracle/movie/moviework/c2bda  
ls movie_fact*.dmp  
hadoop fs -mkdir movie_fact  
hadoop fs -put movie_fact?.dmp movie_fact  
hadoop fs -ls movie_fact
```



```
oracle@bigdatalite:~/movie/moviework/c2bda
File Edit View Search Terminal Help
[oracle@bigdatalite ~]$ cd /home/oracle/movie/moviework/c2bda
[oracle@bigdatalite c2bda]$ ls movie_fact*.dmp
movie_fact_1082895_1166958.dmp movie_fact2.dmp movie_fact4.dmp
movie_fact1.dmp movie_fact3.dmp
[oracle@bigdatalite c2bda]$ hadoop fs -mkdir movie_fact
[oracle@bigdatalite c2bda]$ hadoop fs -put movie_fact?.dmp movie_fact
[oracle@bigdatalite c2bda]$ hadoop fs -ls movie_fact
Found 4 items
-rw-r--r-- 1 oracle oracle 22405120 2015-02-13 08:13 movie_fact/movie_fact1.dmp
-rw-r--r-- 1 oracle oracle 22462464 2015-02-13 08:13 movie_fact/movie_fact2.dmp
-rw-r--r-- 1 oracle oracle 23175168 2015-02-13 08:13 movie_fact/movie_fact3.dmp
-rw-r--r-- 1 oracle oracle 35258368 2015-02-13 08:13 movie_fact/movie_fact4.dmp
[oracle@bigdatalite c2bda]$
```

6. Then, copy the data pump files from the local file system to HDFS for the export\_movie\_fact\_query table, and then view the copied files, by executing the following commands at the \$ prompt:

```
hadoop fs -mkdir movie_fact_query
hadoop fs -put movie_fact_1082895_1166958.dmp movie_fact_query
hadoop fs -ls movie_fact_query
```



```
[oracle@bigdatalite c2bda]$ hadoop fs -mkdir movie_fact_query
[oracle@bigdatalite c2bda]$ hadoop fs -put movie_fact_1082895_1166958.dmp movie_fact_query
[oracle@bigdatalite c2bda]$ hadoop fs -ls movie_fact_query
Found 1 items
-rw-r--r-- 1 oracle oracle 12288 2015-02-13 08:18 movie_fact_query/movie_fact_1082895_1166958.dmp
[oracle@bigdatalite c2bda]$
```

7. Leave the terminal window open.

## Practice 19-3: Create Hive External Tables and Query the Data

### Overview

In this guided practice, you create Hive tables in Hadoop and query the data using HQuery.

### Assumptions

Practice 19-2 has been successfully completed. (Note: To ensure accuracy, copy/paste commands from **lab\_19\_03.txt**)

### Tasks

1. Using the same terminal window session from the previous practice, add the following three jars to Hive.

```
$ hive
hive> add jar /u01/c2bda/orahivedp-1.0/jlib/oraloader.jar;
hive> add jar /u01/c2bda/orahivedp-1.0/jlib/ojdbc6.jar;
hive> add jar /u01/c2bda/orahivedp-1.0/jlib/orahivedp.jar;
```

**Note:** This is not a normal requirement for Copy To BDA. It is only used for the current classroom environment.

```
Logging initialized using configuration in jar:file:/usr/lib/hive/lib/hive-common-0.12.0-cd
h5.1.2.jar!/hive-log4j.properties
hive> add jar /u01/c2bda/orahivedp-1.0/jlib/oraloader.jar;
Added /u01/c2bda/orahivedp-1.0/jlib/oraloader.jar to class path
Added resource: /u01/c2bda/orahivedp-1.0/jlib/oraloader.jar
hive> add jar /u01/c2bda/orahivedp-1.0/jlib/ojdbc6.jar;
Added /u01/c2bda/orahivedp-1.0/jlib/ojdbc6.jar to class path
Added resource: /u01/c2bda/orahivedp-1.0/jlib/ojdbc6.jar
hive> add jar /u01/c2bda/orahivedp-1.0/jlib/orahivedp.jar;
Added /u01/c2bda/orahivedp-1.0/jlib/orahivedp.jar to class path
Added resource: /u01/c2bda/orahivedp-1.0/jlib/orahivedp.jar
hive> ■
```

2. Create the first Hive external table. (Copy/paste from **lab\_19\_03.txt**)

```
hive> CREATE EXTERNAL TABLE movie_fact
> ROW FORMAT SERDE 'oracle.hadoop.hive.datapump.DPSerde'
> STORED AS
> INPUTFORMAT 'oracle.hadoop.hive.datapump.DPInputFormat'
> OUTPUTFORMAT
> 'org.apache.hadoop.hive.io.HiveIgnoreKeyTextOutputFormat'
> LOCATION '/user/oracle/movie_fact';
```

```
hive> CREATE EXTERNAL TABLE movie_fact ROW FORMAT SERDE 'oracle.hadoop.hive.datapump.DPSerde' STORED AS INPUTFORMAT 'oracle.hadoop.hive.datapump.DPInputFormat' OUTPUTFORMAT 'org.apache.hadoop.hive.io.HiveIgnoreKeyTextOutputFormat' LOCATION '/user/oracle/movie_fact';
OK
Time taken: 1.874 seconds
```

### 3. Query the Hive table.

```
DESCRIBE movie_fact;  
SELECT COUNT(*) FROM movie_fact;
```

```
hive> DESCRIBE movie_fact;  
OK  
cust_id          decimal(38,18)      from deserializer  
movie_id         decimal(38,18)      from deserializer  
genre_id         decimal(38,18)      from deserializer  
time_id          date             from deserializer  
recommended      decimal(38,18)      from deserializer  
activity_id      decimal(38,18)      from deserializer  
rating           decimal(38,18)      from deserializer  
sales            decimal(38,18)      from deserializer  
Time taken: 0.397 seconds. Fetched: 8 row(s)  
hive> SELECT COUNT(*) FROM movie_fact;  
Total MapReduce jobs = 1  
Launching Job 1 out of 1  
Number of reduce tasks determined at compile time: 1  
In order to change the average load for a reducer (in bytes):  
  set hive.exec.reducers.bytes.per.reducer=<number>  
In order to limit the maximum number of reducers:  
  set hive.exec.reducers.max=<number>  
In order to set a constant number of reducers:  
  set mapred.reduce.tasks=<number>  
Starting Job = job_1422550335537_0001, Tracking URL = http://bigdatalite.localdomain:8088/proxy/application_1422550335537_0001/  
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1422550335537_0001  
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1  
2015-02-13 14:47:31,554 Stage-1 map = 0%,  reduce = 0%  
2015-02-13 14:47:42,416 Stage-1 map = 100%,  reduce = 0%,  Cumulative CPU 5.7 sec  
2015-02-13 14:47:43,464 Stage-1 map = 100%,  reduce = 0%,  Cumulative CPU 5.7 sec  
2015-02-13 14:47:44,502 Stage-1 map = 100%,  reduce = 0%,  Cumulative CPU 5.7 sec  
2015-02-13 14:47:45,562 Stage-1 map = 100%,  reduce = 0%,  Cumulative CPU 5.7 sec  
2015-02-13 14:47:46,608 Stage-1 map = 100%,  reduce = 0%,  Cumulative CPU 5.7 sec  
2015-02-13 14:47:47,648 Stage-1 map = 100%,  reduce = 0%,  Cumulative CPU 5.7 sec  
2015-02-13 14:47:48,710 Stage-1 map = 100%,  reduce = 0%,  Cumulative CPU 5.7 sec  
2015-02-13 14:47:49,753 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 7.15 sec  
2015-02-13 14:47:50,786 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 7.15 sec  
MapReduce Total cumulative CPU time: 7 seconds 150 msec  
Ended Job = job_1422550335537_0001  
MapReduce Jobs Launched:  
Job 0: Map: 1  Reduce: 1  Cumulative CPU: 7.15 sec  HDFS Read: 103270475 HDFS Write: 8 SUCCESS  
Total MapReduce CPU Time Spent: 7 seconds 150 msec  
OK  
3259348  
Time taken: 31.6 seconds, Fetched: 1 row(s)  
hive> ■
```

4. Create the second Hive external table. (Copy/paste from **lab\_19\_03.txt**)

```
hive> CREATE EXTERNAL TABLE movie_fact_query  
> ROW FORMAT SERDE 'oracle.hadoop.hive.datapump.DPSerde'  
> STORED AS  
> INPUTFORMAT 'oracle.hadoop.hive.datapump.DPInputFormat'  
> OUTPUTFORMAT  
    'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'  
> LOCATION '/user/oracle/movie_fact_query';
```

```
hive> CREATE EXTERNAL TABLE movie_fact_query ROW FORMAT SERDE 'oracle.hadoop.hive.  
.datapump.DPSerde' STORED AS INPUTFORMAT 'oracle.hadoop.hive.datapump.DPInputForm  
at' OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat' LOC  
ATION '/user/oracle/movie_fact_query';  
OK  
Time taken: 0.058 seconds  
hive> █
```

5. Query the Hive table.

```
DESCRIBE movie_fact_query;  
SELECT * FROM movie_fact_query;
```

```
hive> DESCRIBE movie_fact_query;  
OK  
cust_id          decimal(38,18)      from deserializer  
time_id          date              from deserializer  
Time taken: 0.111 seconds, Fetched: 2 row(s)  
hive> SELECT * FROM movie_fact_query;  
OK  
1082895 2010-01-06  
1082895 2010-04-29  
1082895 2010-04-29  
1082895 2011-11-06  
1082895 2011-01-09  
1082895 2011-03-04  
1082895 2011-03-04  
1082895 2011-03-04  
1082895 2011-03-20  
1082895 2011-07-15  
1082895 2011-08-07  
1082895 2011-08-13  
1082895 2011-08-07  
Time taken: 0.155 seconds, Fetched: 14 row(s)  
hive> █
```

6. Exit from Hive using the `exit;` command.
7. Then, exit from the Terminal window using the `exit` command.

## **Practices for Lesson 20: Using Oracle SQL Connector for HDFS**

**Chapter 20**

## Practices for Lesson 20

---

### Practices Overview

In these practices, you will use Oracle SQL Connector for Hadoop Distributed File System (HDFS) to:

- Access data from HDFS files
- Access data from Hive tables
- Access data from partitioned Hive tables

## Guided Practice 20-1: Accessing HDFS Files by Using OSCH

### Overview

In this practice, you will use Oracle SQL Connector for Hadoop Distributed File System (HDFS) to access data in HDFS files.

### Prerequisite

Open a terminal window and navigate to the following directory. Execute the `reset.sh` script to reset the machine for OSCH.

```
cd /home/oracle/exercises/osch  
sh reset.sh
```

### Tasks

1. Open a terminal window and navigate to the below directory.

```
cd /home/oracle/exercises/osch  
ls -l
```

```
[oracle@bigdatalite osch]$ cd /home/oracle/exercises/osch  
[oracle@bigdatalite osch]$ ls -l  
total 32  
-rwxrwx---. 1 oracle oinstall 608 Apr  8 08:46 create_movieapp_log_stage_part.q  
-rwxrwx---. 1 oracle oinstall 391 Apr  8 15:01 genloc_moviefact_hivepart.sh  
-rwxrwx---. 1 oracle oinstall 329 Apr  8 14:56 genloc_moviefact_hive.sh  
-rwxrwx---. 1 oracle oinstall 139 Apr  8 14:39 genloc_moviefact_text.sh  
-rwxrwx---. 1 oracle oinstall 952 Aug 15 2014 moviefact_hivepart.xml  
-rwxrwx---. 1 oracle oinstall 952 Jan 30 2013 moviefact_hive.xml  
-rwxrwx---. 1 oracle oinstall 1256 Jan 29 2013 moviefact_text.xml  
-rwxrwx---. 1 oracle oinstall 520 Apr  8 14:30 reset.sh  
[oracle@bigdatalite osch]$
```

2. Review the data in the HDFS file system.

```
hadoop fs -cat /user/oracle/moviework/data/part* | head
```

```
[oracle@bigdatalite osch]$ hadoop fs -cat /user/oracle/moviework/data/part* | head  
1000679 205 46 2010-12-03:03:14:54 1 1 1  
1000693 77 20 2011-08-14:10:46:55 1 1 3  
1000693 88 48 2012-04-20:19:14:50 1 1 3  
1000693 116 48 2011-11-24:05:43:00 1 1 5  
1000693 141 45 2011-01-01:05:17:57 1 1 4  
1000693 176 24 2012-04-20:04:22:06 1 1 2  
1000693 180 9 2010-12-31:22:26:18 0 1 4  
1000693 240 3 2010-05-21:15:51:01 1 1 3  
1000693 278 8 2011-12-02:03:25:36 1 1 1  
1000693 279 46 2010-02-06:16:42:34 1 1 4  
[oracle@bigdatalite osch]$
```

**Note:** Huge data may be present in the `part*` files, so only the top files are being displayed using the `head` pipe.

3. Open and review the `moviefact_text.xml` file.

```
cat moviefact_text.xml
```

```
[oracle@bigdatalite osch]$ cat moviefact_text.xml
<?xml version="1.0"?>
<configuration>

<property>
  <name>oracle.hadoop.extbl.tableName</name>
  <value>MOVIE_FACT_EXT_TAB_FILE</value>
</property>

<property>
  <name>oracle.hadoop.extbl.sourceType</name>
  <value>text</value>
</property>

<property>
  <name>oracle.hadoop.extbl.dataPaths</name>
  <value>/user/oracle/moviework/data/part*</value>
</property>

<property>
  <name>oracle.hadoop.connection.url</name>
  <value>jdbc:oracle:thin:@localhost:1521:orcl</value>
</property>

<property>
  <name>oracle.hadoop.connection.user</name>
  <value>MOVIEDEMO</value>
</property>

<property>
  <name>oracle.hadoop.extbl.locationFileCount</name>
  <value>2</value>
</property>

<property>
  <name>oracle.hadoop.extbl.fieldTerminator</name>
  <value>\u0009</value>
</property>

<property>
  <name>oracle.hadoop.extbl.columnNames</name>
  <value>CUST_ID,MOVIE_ID,GENRE_ID,TIME_ID,RECOMMENDED,ACTIVITY_ID,RATING,SALES</value>
</property>

<property>
  <name>oracle.hadoop.extbl.defaultDirectory</name>
  <value>MOVIEWORKSHOP_DIR</value>
</property>

</configuration>
[oracle@bigdatalite osch]$
```

4. Open and review the genloc\_moviefact\_text.sh file.

```
cat genloc_moviefact_text.sh
```

```
[oracle@bigdatalite osch]$ cat genloc_moviefact_text.sh
hadoop jar $OSCH_HOME/jlib/orahdfs.jar \
  oracle.hadoop.extbl.ExternalTable \
  -conf moviefact_text.xml \
  -createTable
[oracle@bigdatalite osch]$
```

- Run the `genloc_moviefact_text.sh` file.

```
sh genloc_moviefact_text.sh
```

```
[oracle@bigdatalite osch]$ sh genloc_moviefact_text.sh  
Oracle SQL Connector for HDFS Release 3.1.0 - Production
```

```
Copyright (c) 2011, 2014, Oracle and/or its affiliates. All rights reserved.
```

```
[Enter Database Password:]
```

**Note:** Refer to <file:///home/oracle/GettingStarted/StartHere.html> for the password for Oracle Database 12c.

```
[Enter Database Password:]  
The create table command succeeded.
```

```
CREATE TABLE "MOVIEDEMO"."MOVIE_FACT_EXT_TAB_FILE"  
(  
    "CUST_ID"                VARCHAR2(4000),  
    "MOVIE_ID"                VARCHAR2(4000),  
    "GENRE_ID"                VARCHAR2(4000),  
    "TIME_ID"                 VARCHAR2(4000),  
    "RECOMMENDED"             VARCHAR2(4000),  
    "ACTIVITY_ID"              VARCHAR2(4000),  
    "RATING"                  VARCHAR2(4000),  
    "SALES"                   VARCHAR2(4000)  
)  
ORGANIZATION EXTERNAL  
(  
    TYPE ORACLE_LOADER  
    DEFAULT DIRECTORY "MOVIEWORKSHOP_DIR"  
    ACCESS PARAMETERS  
    (  
        RECORDS DELIMITED BY 0X'0A'  
        CHARACTERSET AL32UTF8  
        PREPROCESSOR "OSCH_BIN_PATH":'hdfs_stream'  
        FIELDS TERMINATED BY 0X'09'  
        MISSING FIELD VALUES ARE NULL  
        (  
            "CUST_ID" CHAR(4000),  
            "MOVIE_ID" CHAR(4000),  
            "GENRE_ID" CHAR(4000),  
            "TIME_ID" CHAR(4000),  
            "RECOMMENDED" CHAR(4000),  
            "ACTIVITY_ID" CHAR(4000),  
            "RATING" CHAR(4000),  
            "SALES" CHAR(4000)  
        )  
    )  
    LOCATION  
    (  
        'osch-20150306021515-5979-1',  
        'osch-20150306021515-5979-2'  
    )  
) PARALLEL REJECT LIMIT UNLIMITED;
```

```
The following location files were created.  
osch-20150408024130-1438-1 contains 1 URI, 12754882 bytes  
    12754882 hdfs://bigdatalite.localdomain:8020/user/oracle/moviework/data/part-00001  
osch-20150408024130-1438-2 contains 3 URIs, 1072 bytes  
    438 hdfs://bigdatalite.localdomain:8020/user/oracle/moviework/data/part-00002  
    432 hdfs://bigdatalite.localdomain:8020/user/oracle/moviework/data/part-00003  
    202 hdfs://bigdatalite.localdomain:8020/user/oracle/moviework/data/part-00004  
[oracle@bigdatalite osch]$
```

6. Log in to sqlplus and query the newly created movie\_fact\_ext\_tab\_file table.

```
sqlplus moviedemo/welcome1  
describe movie_fact_ext_tab_file;
```

```
[oracle@bigdatalite osch]$ sqlplus moviedemo/welcome1  
  
SQL*Plus: Release 12.1.0.2.0 Production on Fri Mar 6 03:04:35 2015  
  
Copyright (c) 1982, 2014, Oracle. All rights reserved.  
  
Last Successful login time: Fri Mar 06 2015 02:15:13 -05:00  
  
Connected to:  
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production  
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing options  
  
SQL> describe movie_fact_ext_tab_file;  
Name          Null?    Type  
-----  
CUST_ID           VARCHAR2(4000)  
MOVIE_ID          VARCHAR2(4000)  
GENRE_ID          VARCHAR2(4000)  
TIME_ID           VARCHAR2(4000)  
RECOMMENDED       VARCHAR2(4000)  
ACTIVITY_ID       VARCHAR2(4000)  
RATING            VARCHAR2(4000)  
SALES             VARCHAR2(4000)  
  
SQL> ■
```

7. Fetch the cust\_id for a few rows in the table.

```
select cust_id from movie_fact_ext_tab_file where rownum <10;
```

```
SQL> select cust_id from movie_fact_ext_tab_file where rownum <10;  
  
CUST_ID  
-----  
1000679  
1000693  
1000693  
1000693  
1000693  
1000693  
1000693  
1000693  
1000693  
  
9 rows selected.  
  
SQL>
```

8. Fetch the count of rows of the movie\_fact\_ext\_tab\_file table.

```
select count(*) from movie_fact_ext_tab_file;  
  
SQL> select count(*) from movie_fact_ext_tab_file;  
  
COUNT(*)  
-----  
300025  
  
SQL>
```

## Guided Practice 20-2: Accessing Hive Tables by Using OSCH

### Overview

In this practice, you will use Oracle SQL Connector for Hadoop Distributed File System (HDFS) to access data in Hive tables.

### Assumptions

Practice 20-1 is completed.

### Tasks

1. Open a terminal window and navigate to the below directory.

```
cd /home/oracle/exercises/osch  
ls -l
```

```
[oracle@bigdatalite osch]$ cd /home/oracle/exercises/osch  
[oracle@bigdatalite osch]$ ls -l  
total 32  
-rwxrwx---. 1 oracle oinstall 608 Apr  8 08:46 create_movieapp_log_stage_part.q  
-rwxrwx---. 1 oracle oinstall 391 Apr  8 15:01 genloc_moviefact_hivepart.sh  
-rwxrwx---. 1 oracle oinstall 329 Apr  8 14:56 genloc_moviefact_hive.sh  
-rwxrwx---. 1 oracle oinstall 139 Apr  8 14:39 genloc_moviefact_text.sh  
-rwxrwx---. 1 oracle oinstall 952 Aug 15  2014 moviefact_hivepart.xml  
-rwxrwx---. 1 oracle oinstall 952 Jan 30  2013 moviefact_hive.xml  
-rwxrwx---. 1 oracle oinstall 1256 Jan 29  2013 moviefact_text.xml  
-rwxrwx---. 1 oracle oinstall 520 Apr  8 14:30 reset.sh  
[oracle@bigdatalite osch]$
```

2. Review the data in the Hive table.

```
hive  
use moviedemo;  
show tables;  
exit;
```

```
[oracle@bigdatalite osch]$ hive
15/03/06 04:38:30 INFO Configuration.deprecation: mapred.input.dir.recursive is deprecated. Instead, use mapreduce.input.fileinputformat.input.dir.recursive
15/03/06 04:38:30 INFO Configuration.deprecation: mapred.max.split.size is deprecated. Instead, use mapreduce.input.fileinputformat.split.maxsize
15/03/06 04:38:30 INFO Configuration.deprecation: mapred.min.split.size is deprecated. Instead, use mapreduce.input.fileinputformat.split.minsize
15/03/06 04:38:30 INFO Configuration.deprecation: mapred.min.split.size.per.rack is deprecated. Instead, use mapreduce.input.fileinputformat.split.minsize.per.rack
15/03/06 04:38:30 INFO Configuration.deprecation: mapred.min.split.size.per.node is deprecated. Instead, use mapreduce.input.fileinputformat.split.minsize.per.node
15/03/06 04:38:30 INFO Configuration.deprecation: mapred.reduce.tasks is deprecated. Instead, use mapreduce.job.reduces
15/03/06 04:38:30 INFO Configuration.deprecation: mapred.reduce.tasks.speculative.execution is deprecated. Instead, use mapreduce.reduce.speculative
15/03/06 04:38:30 WARN conf.HiveConf: DEPRECATED: Configuration property hive.metastore.local no longer has any effect. Make sure to provide a valid value for hive.metastore.uris if you are connecting to a remote metastore.

Logging initialized using configuration in jar:file:/usr/lib/hive/lib/hive-common-0.12.0-cdh5.1.2.jar!/hive-log4j.properties
hive> use moviedemo;
OK
Time taken: 0.866 seconds
hive> show tables;
OK
movieapp_log_stage
movieapp_log_stage_1
movieapp_log_stage_part
Time taken: 0.461 seconds, Fetched: 3 row(s)
hive> exit;
[oracle@bigdatalite osch]$
```

3. Open and review the moviefact\_hive.xml file.

```
cat moviefact_hive.xml
```

```
[oracle@bigdatalite osch]$ cat moviefact_hive.xml
<?xml version="1.0"?>
<configuration>

<property>
  <name>oracle.hadoop.extbl.tableName</name>
  <value>MOVIE_FACT_EXT_TAB_HIVE</value>
</property>

<property>
  <name>oracle.hadoop.extbl.sourceType</name>
  <value>hive</value>
</property>

<property>
  <name>oracle.hadoop.extbl.hive.tableName</name>
  <value>movieapp_log_stage_1</value>
</property>

<property>
  <name>oracle.hadoop.extbl.hive.databaseName</name>
  <value>moviedemo</value>
</property>

<property>
  <name>oracle.hadoop.connection.url</name>
  <value>jdbc:oracle:thin:@localhost:1521:orcl</value>
</property>

<property>
  <name>oracle.hadoop.connection.user</name>
  <value>MOVIEDEMO</value>
</property>

<property>
  <name>oracle.hadoop.extbl.defaultDirectory</name>
  <value>MOVIEWORKSHOP_DIR</value>
</property>

</configuration>
[oracle@bigdatalite osch]$
```

4. Open and review the genloc\_moviefact\_hive.sh file.

```
cat genloc_moviefact_hive.sh
```

```
[oracle@bigdatalite osch]$ cat genloc_moviefact_hive.sh
# Add HIVE_HOME/lib* to HADOOP_CLASSPATH. This cannot be done
# in the login profiles since this breaks Pig in the previous lab.
export HADOOP_CLASSPATH=$HADOOP_CLASSPATH:$HIVE_HOME/lib/*

hadoop jar $OSCH_HOME/jlib/orahdfs.jar \
  oracle.hadoop.extbl.ExternalTable \
  -conf moviefact_hive.xml \
  -createTable
[oracle@bigdatalite osch]$
```

5. Run the genloc\_moviefact\_hive.sh file.

```
sh genloc_moviefact_hive.sh
```

```
[oracle@bigdatalite osch]$ sh genloc moviefact hive.sh
Oracle SQL Connector for HDFS Release 3.1.0 - Production

Copyright (c) 2011, 2014, Oracle and/or its affiliates. All rights reserved.

[Enter Database Password:]
```

**Note:** Refer to <file:///home/oracle/GettingStarted/StartHere.html> for the password.

```
[Enter Database Password:]
15/03/06 03:49:11 INFO Configuration.deprecation: mapred.input.dir.recursive is deprecated.
Instead, use mapreduce.input.fileinputformat.input.dir.recursive
15/03/06 03:49:11 INFO Configuration.deprecation: mapred.max.split.size is deprecated. Instead,
use mapreduce.input.fileinputformat.split.maxsize
15/03/06 03:49:11 INFO Configuration.deprecation: mapred.min.split.size is deprecated. Instead,
use mapreduce.input.fileinputformat.split.minsize
15/03/06 03:49:11 INFO Configuration.deprecation: mapred.min.split.size.per.rack is deprecated.
Instead, use mapreduce.input.fileinputformat.split.minsize.per.rack
15/03/06 03:49:11 INFO Configuration.deprecation: mapred.min.split.size.per.node is deprecated.
Instead, use mapreduce.input.fileinputformat.split.minsize.per.node
15/03/06 03:49:11 INFO Configuration.deprecation: mapred.reduce.tasks is deprecated. Instead,
use mapreduce.job.reduces
15/03/06 03:49:11 INFO Configuration.deprecation: mapred.reduce.tasks.speculative.execution
is deprecated. Instead, use mapreduce.reduce.speculative
15/03/06 03:49:12 WARN conf.HiveConf: DEPRECATED: Configuration property hive.metastore.local
no longer has any effect. Make sure to provide a valid value for hive.metastore.uris if you
are connecting to a remote metastore.
15/03/06 03:49:12 INFO hive.metastore: Trying to connect to metastore with URI thrift://bigd
atalite.localdomain:9083
15/03/06 03:49:12 INFO hive.metastore: Connected to metastore.
The create table command succeeded.

CREATE TABLE "MOVIEDEMO"."MOVIE_FACT_EXT_TAB_HIVE"
(
    "CUSTID"                      INTEGER,
    "MOVIEID"                      INTEGER,
    "GENREID"                      INTEGER,
    "TIME"                          VARCHAR2(4000),
    "RECOMMENDED"                  INTEGER,
    "ACTIVITY"                     INTEGER,
    "RATING"                        INTEGER,
    "SALES"                         NUMBER
)
```

```
ORGANIZATION EXTERNAL
(
  TYPE ORACLE_LOADER
  DEFAULT DIRECTORY "MOVIEWORKSHOP_DIR"
  ACCESS PARAMETERS
  (
    RECORDS DELIMITED BY 0X'0A'
    CHARACTERSET AL32UTF8
    PREPROCESSOR "OSCH_BIN_PATH":'hdfs_stream'
    FIELDS TERMINATED BY 0X'01'
    MISSING FIELD VALUES ARE NULL
    (
      "CUSTID" CHAR NULLIF "CUSTID"=0X'5C4E',
      "MOVIEID" CHAR NULLIF "MOVIEID"=0X'5C4E',
      "GENREID" CHAR NULLIF "GENREID"=0X'5C4E',
      "TIME" CHAR(4000) NULLIF "TIME"=0X'5C4E',
      "RECOMMENDED" CHAR NULLIF "RECOMMENDED"=0X'5C4E',
      "ACTIVITY" CHAR NULLIF "ACTIVITY"=0X'5C4E',
      "RATING" CHAR NULLIF "RATING"=0X'5C4E',
      "SALES" CHAR NULLIF "SALES"=0X'5C4E'
    )
  )
  LOCATION
  (
    'osch-20150306034913-770-1'
  )
) PARALLEL REJECT LIMIT UNLIMITED;

The following location files were created.

osch-20150306034913-770-1 contains 1 URI, 269627 bytes

  269627 hdfs://bigdatalite.localdomain:8020/user/hive/warehouse/moviedemo.db/movieapp_log_stage_1/000000_0

[oracle@bigdatalite osch]$ █
```

6. Log in to sqlplus and query the newly created movie\_fact\_ext\_tab\_hive table.

```
sqlplus moviedemo/welcome1
describe movie_fact_ext_tab_hive;
```

```
[oracle@bigdatalite osch]$ sqlplus moviedemo/welcome1

SQL*Plus: Release 12.1.0.2.0 Production on Fri Mar 6 03:52:41 2015

Copyright (c) 1982, 2014, Oracle. All rights reserved.

Last Successful login time: Fri Mar 06 2015 03:52:00 -05:00

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing options

SQL> describe movie_fact_ext_tab_hive;
Name          Null?    Type
-----        -----   -----
CUSTID        NUMBER(38)
MOVIEID       NUMBER(38)
GENREID       NUMBER(38)
TIME          VARCHAR2(4000)
RECOMMENDED   NUMBER(38)
ACTIVITY      NUMBER(38)
RATING         NUMBER(38)
SALES          NUMBER

SQL>
```

7. Fetch the custid for a few rows in the table.

```
select custid from movie_fact_ext_tab_hive where rownum <10;
```

```
SQL> select custid from movie_fact_ext_tab_hive where rownum <10;
```

```
CUSTID
-----
1000083
1000672
1000693
1000693
1000693
1000693
1000693
1000693
1000693
```

```
9 rows selected.
```

```
SQL>
```

8. Fetch the count of rows of the movie\_fact\_ext\_tab\_hive table.

```
select count(*) from movie_fact_ext_tab_hive;
```

```
SQL> select count(*) from movie_fact_ext_tab_hive;
```

```
COUNT(*)
-----
6063
```

```
SQL>
```

## Guided Practice 20-3: Accessing Partitioned Hive Tables by Using OSCH

### Overview

In this practice, you will use Oracle SQL Connector for Hadoop Distributed File System (HDFS) to access data in partitioned Hive tables.

### Assumptions

Practice 20-1 is completed.

### Tasks

1. Open a terminal window and navigate to the below directory.

```
cd /home/oracle/exercises/osch  
ls -l
```

```
[oracle@bigdatalite osch]$ cd /home/oracle/exercises/osch  
[oracle@bigdatalite osch]$ ls -l  
total 32  
-rwxrwx---. 1 oracle oinstall 608 Apr  8 08:46 create_movieapp_log_stage_part.q  
-rwxrwx---. 1 oracle oinstall 391 Apr  8 15:01 genloc_moviefact_hivepart.sh  
-rwxrwx---. 1 oracle oinstall 329 Apr  8 14:56 genloc_moviefact_hive.sh  
-rwxrwx---. 1 oracle oinstall 139 Apr  8 14:39 genloc_moviefact_text.sh  
-rwxrwx---. 1 oracle oinstall 952 Aug 15  2014 moviefact_hivepart.xml  
-rwxrwx---. 1 oracle oinstall 952 Jan 30  2013 moviefact_hive.xml  
-rwxrwx---. 1 oracle oinstall 1256 Jan 29  2013 moviefact_text.xml  
-rwxrwx---. 1 oracle oinstall 520 Apr  8 14:30 reset.sh  
[oracle@bigdatalite osch]$
```

2. Review the data in the Hive table.

```
hive  
use moviedemo;  
show tables;  
exit;
```

```
[oracle@bigdatalite osch]$ hive
15/03/06 04:38:30 INFO Configuration.deprecation: mapred.input.dir.recursive is deprecated. Instead, use mapreduce.input.fileinputformat.input.dir.recursive
15/03/06 04:38:30 INFO Configuration.deprecation: mapred.max.split.size is deprecated. Instead, use mapreduce.input.fileinputformat.split.maxsize
15/03/06 04:38:30 INFO Configuration.deprecation: mapred.min.split.size is deprecated. Instead, use mapreduce.input.fileinputformat.split.minsize
15/03/06 04:38:30 INFO Configuration.deprecation: mapred.min.split.size.per.rack is deprecated. Instead, use mapreduce.input.fileinputformat.split.minsize.per.rack
15/03/06 04:38:30 INFO Configuration.deprecation: mapred.min.split.size.per.node is deprecated. Instead, use mapreduce.input.fileinputformat.split.minsize.per.node
15/03/06 04:38:30 INFO Configuration.deprecation: mapred.reduce.tasks is deprecated. Instead, use mapreduce.job.reduces
15/03/06 04:38:30 INFO Configuration.deprecation: mapred.reduce.tasks.speculative.execution is deprecated. Instead, use mapreduce.reduce.speculative
15/03/06 04:38:30 WARN conf.HiveConf: DEPRECATED: Configuration property hive.metastore.local no longer has any effect. Make sure to provide a valid value for hive.metastore.uris if you are connecting to a remote metastore.

Logging initialized using configuration in jar:file:/usr/lib/hive/lib/hive-common-0.12.0-cdh5.1.2.jar!/hive-log4j.properties
hive> use moviedemo;
OK
Time taken: 0.866 seconds
hive> show tables;
OK
movieapp_log_stage
movieapp_log_stage_1
movieapp_log_stage_part
Time taken: 0.461 seconds, Fetched: 3 row(s)
hive> exit;
[oracle@bigdatalite osch]$
```

3. Open and review the moviefact\_hive.xml file.

```
cat moviefact_hivepart.xml
```

```
[oracle@bigdatalite osch]$ cat moviefact_hivepart.xml
<?xml version="1.0"?>
<configuration>

<property>
  <name>oracle.hadoop.extbl.tableName</name>
  <value>MOVIE_FACT_META_PART</value>
</property>

<property>
  <name>oracle.hadoop.extbl.sourceType</name>
  <value>hive</value>
</property>

<property>
  <name>oracle.hadoop.extbl.hive.tableName</name>
  <value>movieapp_log_stage_part</value>
</property>

<property>
  <name>oracle.hadoop.extbl.hive.databaseName</name>
  <value>moviedemo</value>
</property>

<property>
  <name>oracle.hadoop.connection.url</name>
  <value>jdbc:oracle:thin:@localhost:1521:orcl</value>
</property>

<property>
  <name>oracle.hadoop.connection.user</name>
  <value>MOVIEDEMO</value>
</property>

<property>
  <name>oracle.hadoop.extbl.defaultDirectory</name>
  <value>MOVIEWORKSHOP_DIR</value>
</property>

</configuration>
[oracle@bigdatalite osch]$
```

- Open and review the `genloc_moviefact_hivepart.sh` file.

```
cat genloc_moviefact_hivepart.sh

[oracle@bigdatalite osch]$ cat genloc_moviefact_hivepart.sh
# Add HIVE_HOME/lib* to HADOOP_CLASSPATH. This is not preset
# in the VM since this breaks Pig in the previous lab.

export HADOOP_CLASSPATH=$HADOOP_CLASSPATH:$HIVE_HOME/lib/*

hadoop jar $OSCH_HOME/jlib/orahdfs.jar \
  oracle.hadoop.extbl.ExternalTable \
  -conf moviefact_hivepart.xml \
  -D oracle.hadoop.extbl.hive.partitionFilter='activity < 4' \
  -createTable
[oracle@bigdatalite osch]$
```

- Run the `genloc_moviefact_hivepart.sh` file.

```
sh genloc_moviefact_hivepart.sh
```

```
[oracle@bigdatalite osch]$ sh genloc moviefact_hivepart.sh  
Oracle SQL Connector for HDFS Release 3.1.0 - Production  
  
Copyright (c) 2011, 2014, Oracle and/or its affiliates. All rights reserved.  
  
[Enter Database Password:]
```

**Note:** Refer <file:///home/oracle/GettingStarted/StartHere.html> for the password.

```
CREATE VIEW "MOVIEDEMO"."MOVIE_FACT_META_PART_1"  
(  
    "ACTIVITY",  
    "CUSTID",  
    "MOVIEID",  
    "GENREID",  
    "TIME",  
    "RECOMMENDED",  
    "RATING",  
    "SALES"  
)  
AS  
(  
    select  
        CAST (2 AS INTEGER),  
        "CUSTID",  
        "MOVIEID",  
        "GENREID",  
        "TIME",  
        "RECOMMENDED",  
        "RATING",  
        "SALES"  
    from OSCHMOVIE_FACT_META_PART_1  
)
```

See Oracle OSCH metadata table MOVIE\_FACT\_META\_PART for partition attributes and mappings to Oracle OSCH views.

Directly query views MOVIE\_FACT\_META\_PART\_1...MOVIE\_FACT\_META\_PART\_2 to access specific Hive partitions.

```
[oracle@bigdatalite osch]$
```

6. Log in to sqlplus and query the newly created movie\_fact\_meta\_part table.

```
sqlplus moviedemo/welcome1  
describe movie_fact_meta_part;
```

```
[oracle@bigdatalite osch]$ sqlplus moviedemo/welcome1

SQL*Plus: Release 12.1.0.2.0 Production on Fri Mar 6 04:49:29 2015

Copyright (c) 1982, 2014, Oracle. All rights reserved.

Last Successful login time: Fri Mar 06 2015 04:45:48 -05:00

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing options

SQL> describe movie_fact_meta_part;
Name          Null?    Type
-----        -----
VIEW_NAME      NOT NULL VARCHAR2(30)
EXT_TABLE_NAME NOT NULL VARCHAR2(30)
HIVE_TABLE_NAME NOT NULL VARCHAR2(4000)
HIVE_DB_NAME   NOT NULL VARCHAR2(4000)
HIVE_PART_FILTER VARCHAR2(4000)
ACTIVITY       NUMBER(38)

SQL>
```

7. Select the views of the table by using the `view_name` field.

```
select view_name from movie_fact_meta_part;

SQL> select view_name from movie_fact_meta_part;

VIEW_NAME
-----
MOVIE_FACT_META_PART_1
MOVIE_FACT_META_PART_2

SQL>
```

8. Query the tables to find the number of rows present in them.

```
select count(*) from movie_fact_meta_part_1;
select count(*) from movie_fact_meta_part_2;

SQL> select count(*) from movie_fact_meta_part_1;

COUNT(*)
-----
4390

SQL> select count(*) from movie_fact_meta_part_2;

COUNT(*)
-----
6063

SQL>
```

# **Practices for Lesson 21: Using Oracle Data Integrator and Oracle GoldenGate with Hadoop**

**Chapter 21**

## Practices for Lesson 21

---

### Practices Overview

In these guided practices, you use Oracle Data Integrator Studio to create mappings that load data from Oracle Database into a Hive table by using the SQL to Hive-HBASE-File (Sqoop) Integration Knowledge Module (IKM).

## Practice 21-1: Review Topology and Model Setup

### Overview

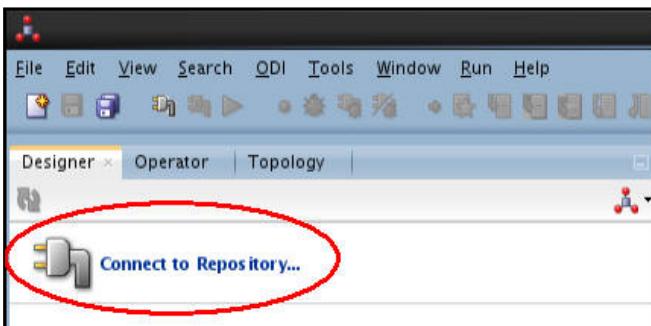
In this guided practice, you open Oracle Data Integrator Studio and use the Topology manager to review the model setup for a data loading process.

### Tasks

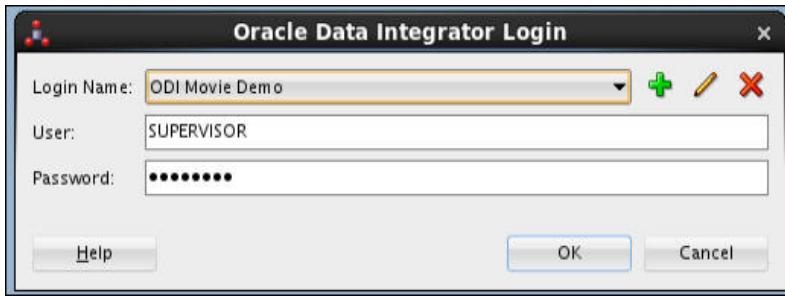
1. Click the ODI Studio icon on the toolbar menu, as shown here:



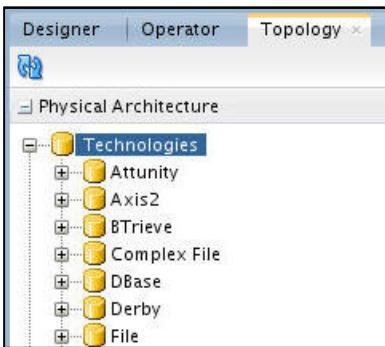
2. Close the Start Page Tab. Then, in the Designer tab, click **Connect to Repository**. The Data Integrator Login dialog appears.



3. In the ODI Login dialog, accept the default user and password values and click **OK**.

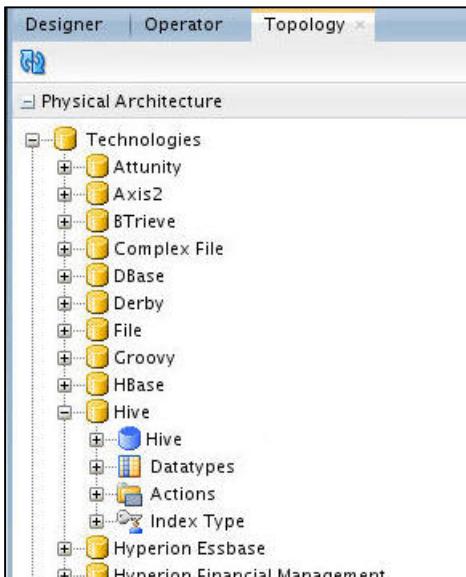


4. Select the **Topology** tab and open the **Physical Architecture** navigator. Then, drill on the **Technologies** node to view a list of supported ODI technologies.

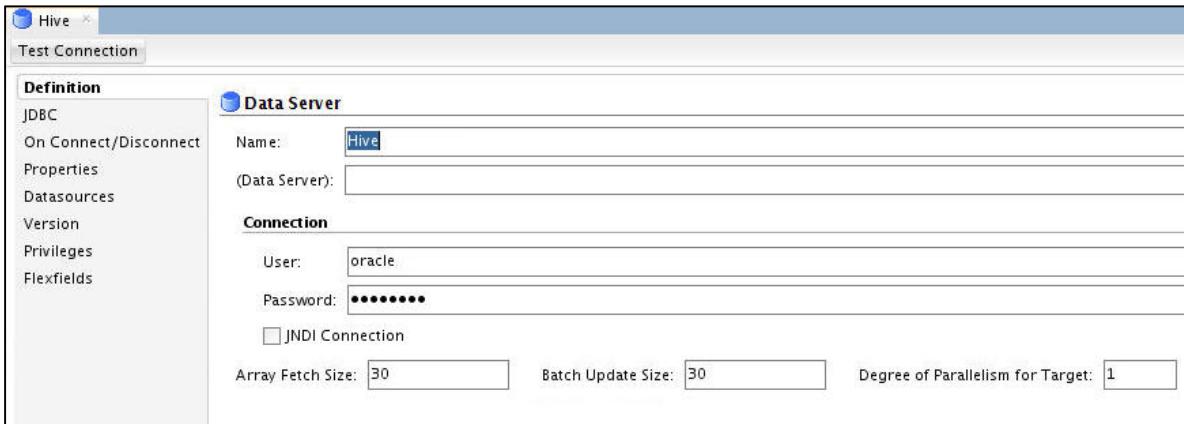


**Note:** Connectivity information has already been set up for Hive, MySQL, and Oracle Database sources and targets.

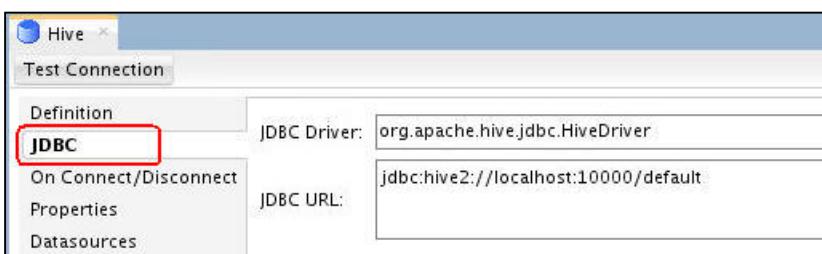
5. Drill on the **Hive** node to see the configured data server.



6. Double-click the **Hive** data server ( ) to review its settings.



7. Click on the **JDBC** tab to view Hive connection information.



8. Close the Hive tabbed pane.

## Practice 21-2: Map and Load Data Into Hive Tables

### Overview

In this guided practice you load data from an Oracle Database into Hive tables by using Apache Sqoop.

As you learned in the lesson, ODI's Knowledge Modules (KMs) are designed to leverage the processing capabilities of the native environment. In this case, ODI leverages Sqoop's ability to start parallel Map-Reduce processes in Hadoop in order to load chunks of data from the database in parallel.

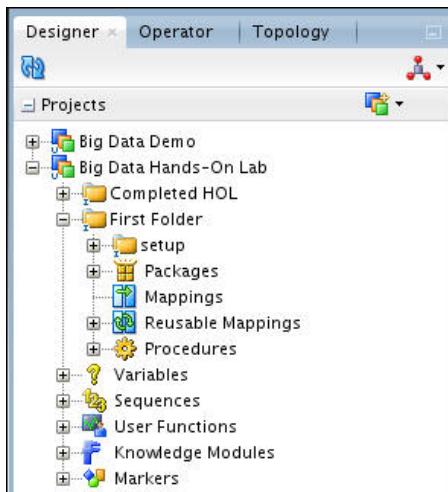
Here, you create a mapping that will load data from the Oracle Database `MOVIE` table into the Hive `movie` table. When you create the mapping, ODI enables you to select the appropriate Knowledge Module, and then it generates the Sqoop code transparently.

### Tasks

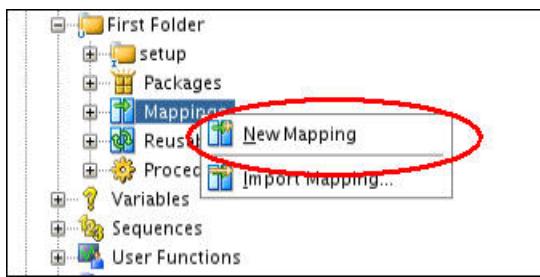
1. First, select the **Designer** navigator. By default the Projects category is displayed.



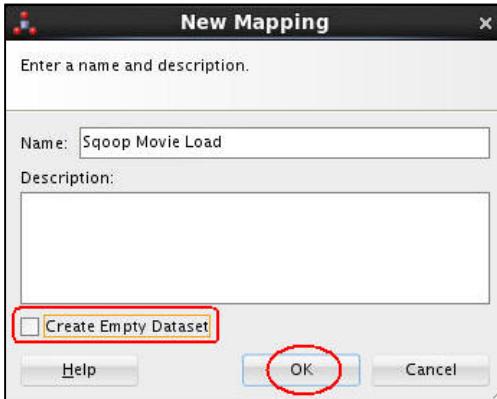
2. Drill on **Big Data Hands-On Lab > First Folder**, as shown here:



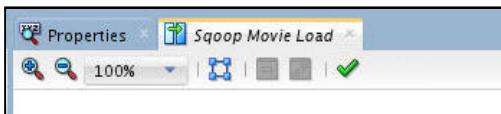
3. Right-click on **Mappings** and select **New Mapping** from the menu.



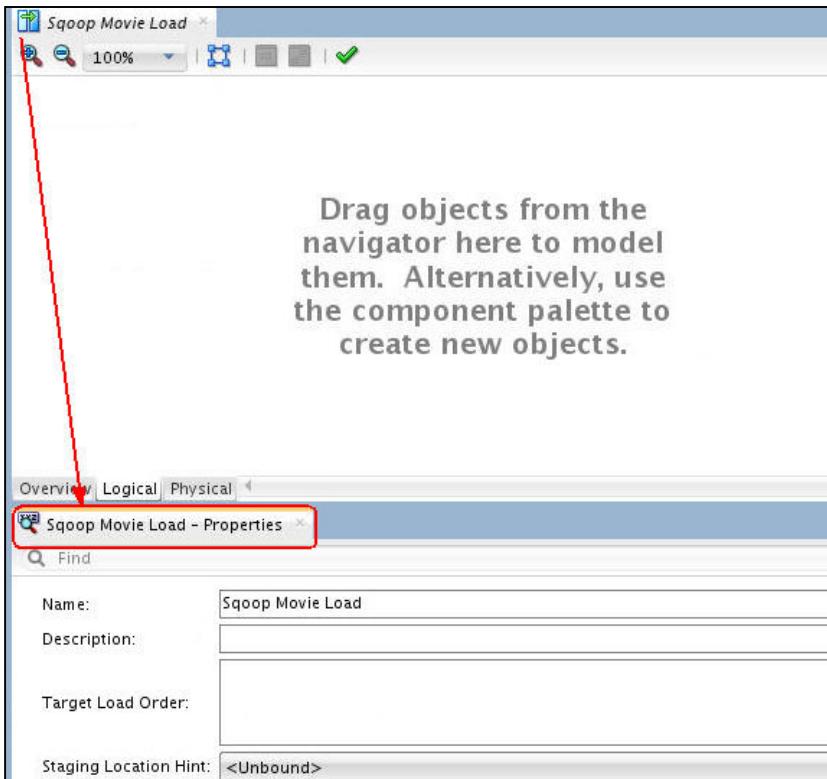
- In the New Mapping dialog, change the name to **Sqoop Movie Load** and ensure that the **Create Empty Dataset** option is deselected. Then click **OK**.



**Result:** A blank tabbed pane with the mapping name appears. In addition, a Properties tab appears next to it.

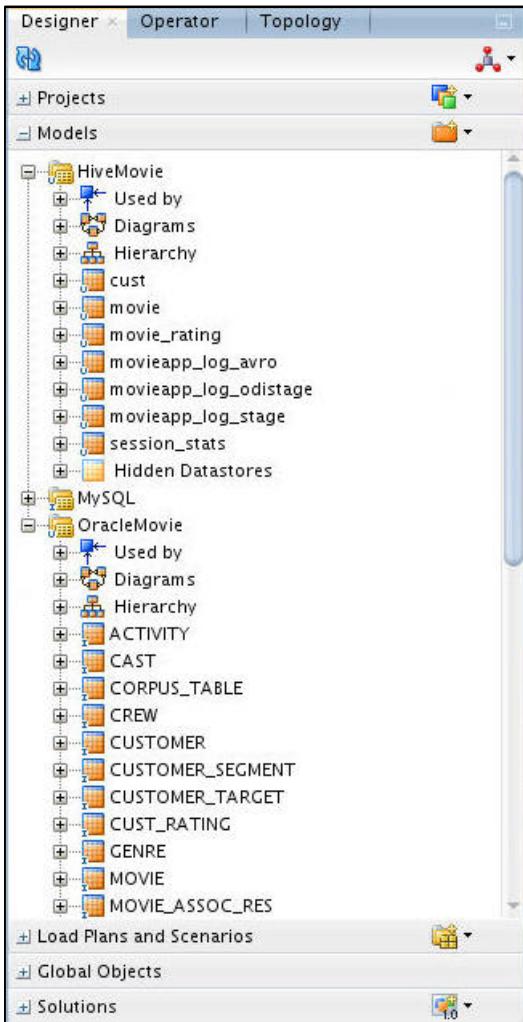


- Drag the Properties pane to the bottom of the mapping tabbed pane, like this:



**Note:** The Properties pane may also be resized.

6. Next, in the Designer tab collapse the Projects category and open the **Models** category. Then, drill on the **HiveMovie** and **OracleMovie** nodes as shown here:

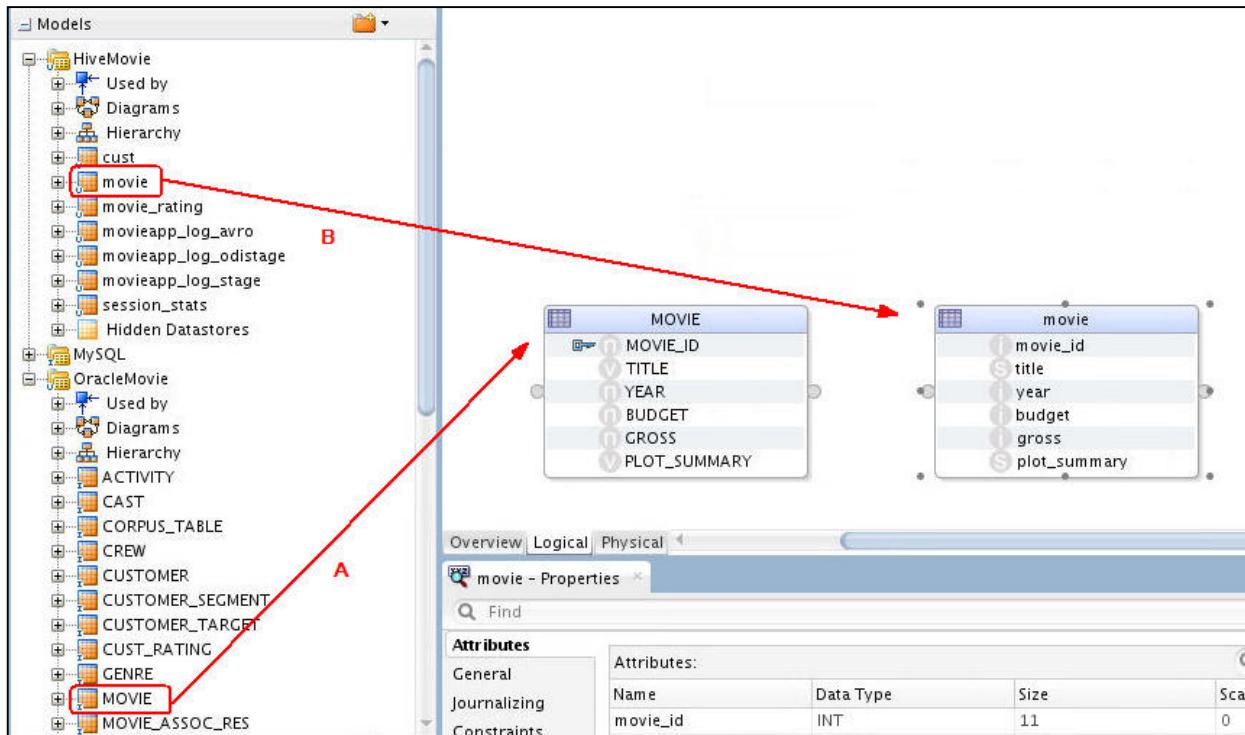


**Note:** For this mapping, you will specify the **MOVIE** table from OracleMovie model as the source, and the **movie** table from the HiveMovie model as the target.

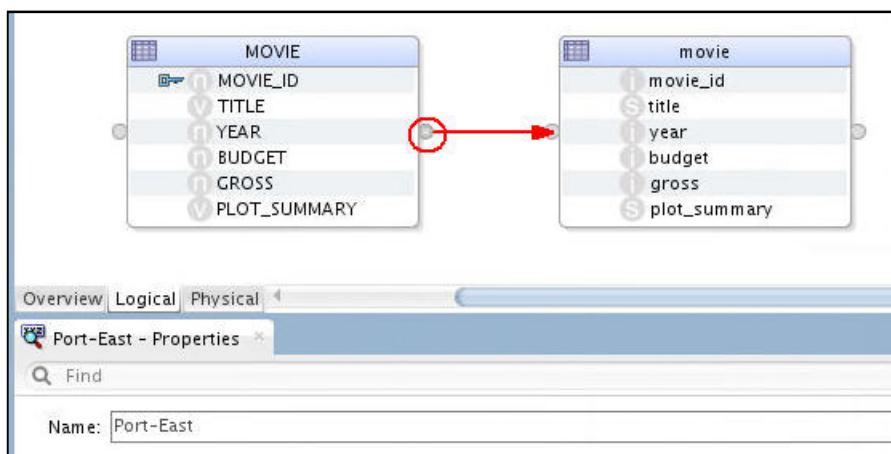
7. To perform the mapping:

- First: From the OracleMovie model, drag and drop **MOVIE** from the navigator to the left side of the Sqoop Movie Load pane. Result: A datastore object named **MOVIE** appears.
- Second: From the HiveMovie model, drag and drop **movie** from navigator to the right of the **MOVIE** object in the Sqoop Movie Load panel.

Result: Both datastore objects appear as shown below.

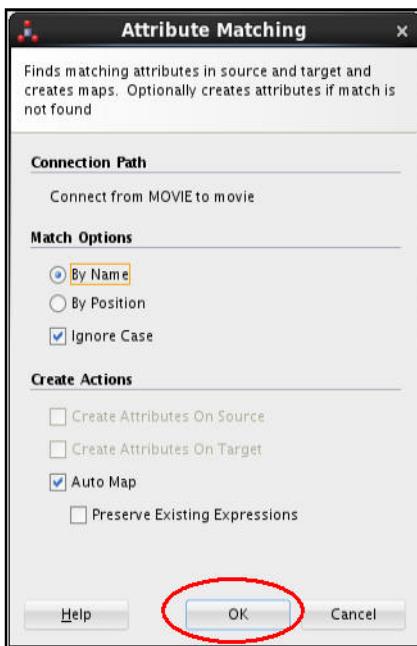


- Next, click the “output port” of the **MOVIE** table object (small circle on the right of the object). The Properties table updates with the value “Port-East”.



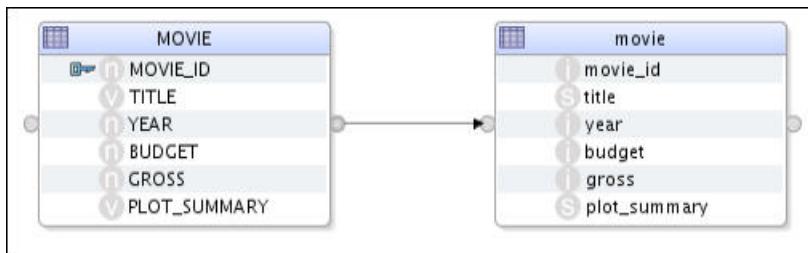
- Drag the Port-East object onto the “input port” of the **movie** table object as shown above, and then release the mouse button.

**Result:** The Attribute Matching dialog appears automatically.

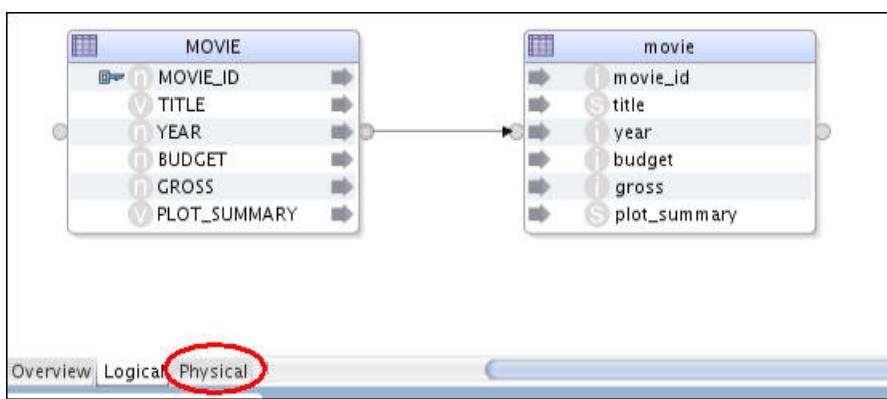


- E. In the Attribute Matching dialog, accept the default settings and click **OK**. In this case, ODI maps all same-name fields from source to target.

**Result:** The two table objects should be mapped, as shown here:

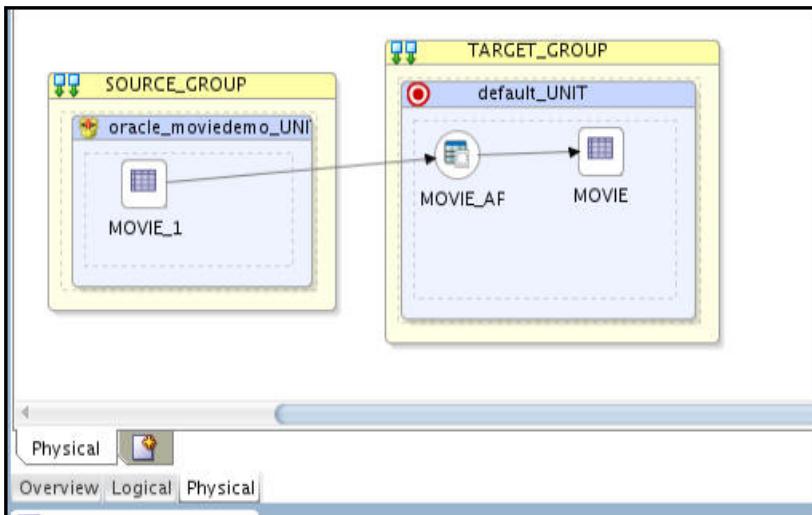


8. Now that the logical flow has now been defined, you set up the physical implementation. First, click on the **Physical** tab of the editor, as shown here:

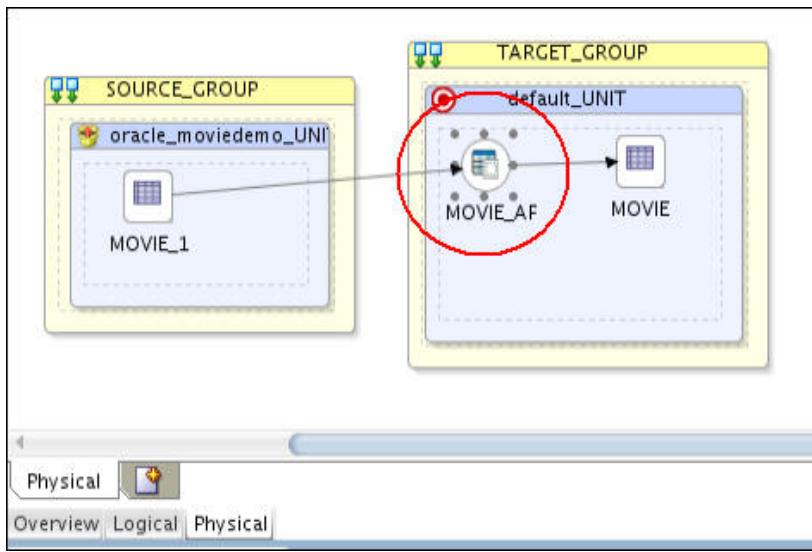


**Note:**

- The physical tab shows the actual systems involved in the transformation, in this case the Oracle Database source and the Hive target.
- The physical tab enables you to choose the Load Knowledge Module (LKM) that controls data movement between systems as well as the Integration Knowledge Module (IKM) that controls transformation of data.

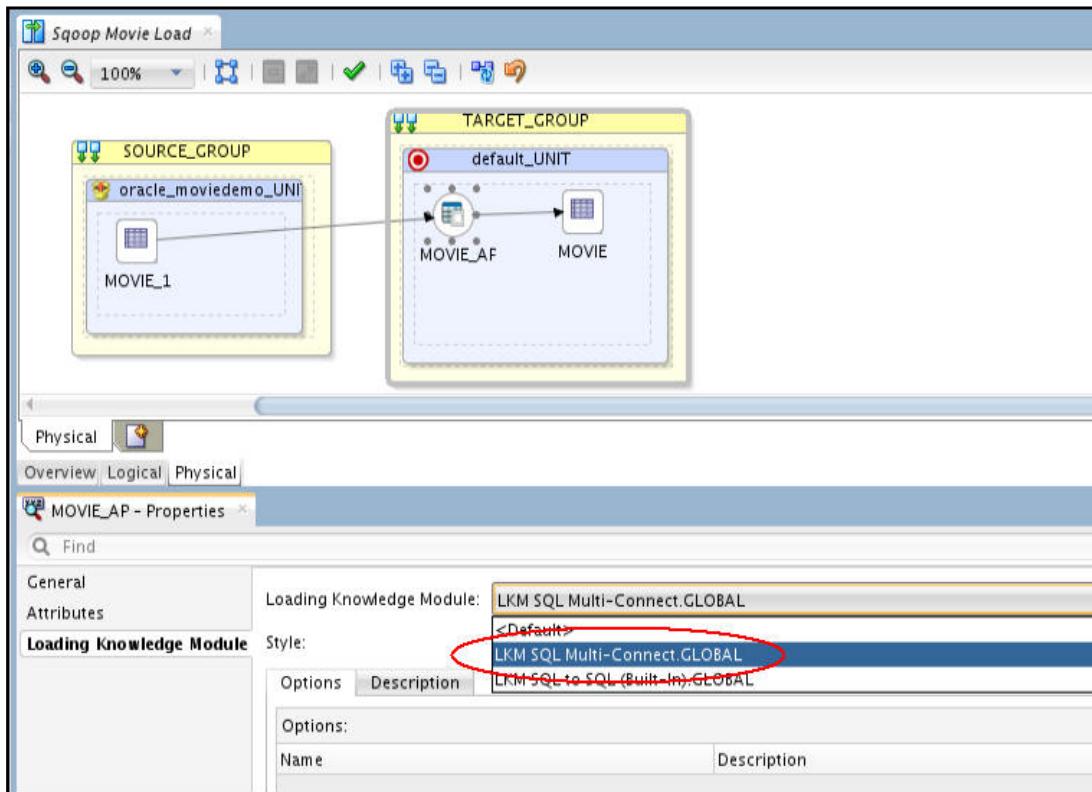


9. Select the access point **MOVIE\_AP** as shown below. (Although the object name looks like **MOVIE\_AF** in the graphic, it's actually **MOVIE\_AP**, as shown in the Properties inspector.)

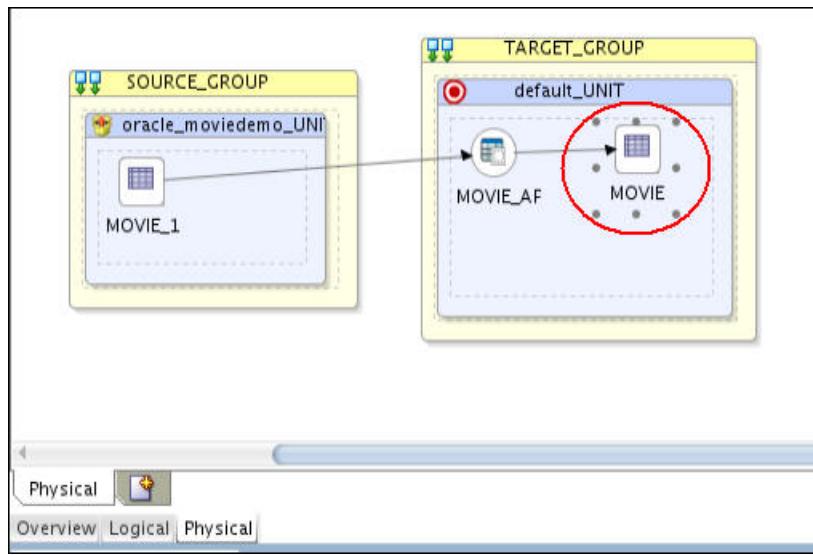


10. In the Properties Editor underneath the Mapping editor, select the **Loading Knowledge Module** tab. Then, select the **LKM SQL Multi-Connect.GLOBAL** option from the pulldown.

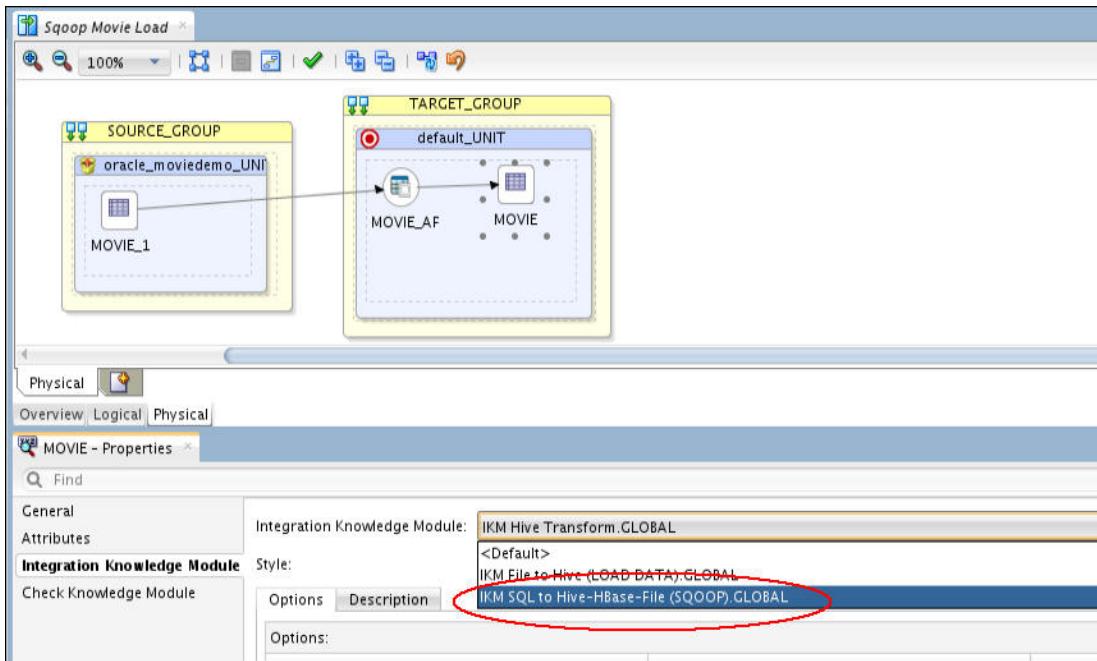
**Note:** This LKM enables the IKM to perform loading activities.



11. Next, select the **MOVIE** object in the **TARGET\_GROUP**, like this:



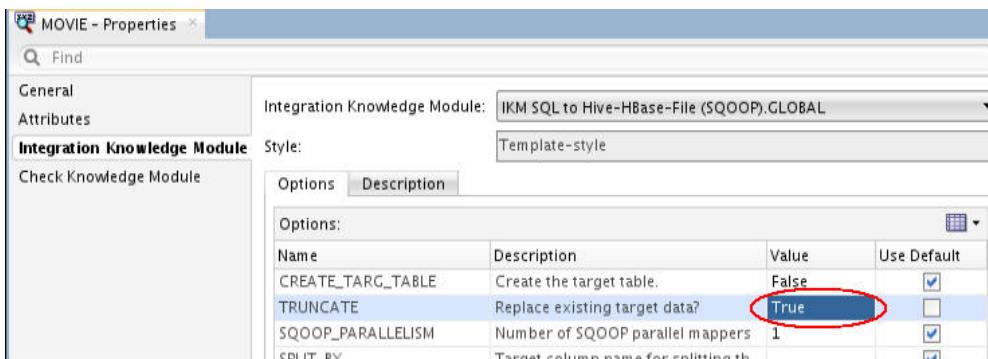
12. In the Properties editor, select the **Integration Knowledge Module** tab and choose the **IKM SQL to Hive-HBase-File (SQOOP).GLOBAL** from the pulldown.



**Note:** If this IKM is not visible in the list, make sure that you performed the previous step and chose the LKM SQL Multi-Connect.

13. Review the list of KM options for this IKM, which are used to configure and tune the Sqoop process to load data.
14. Then, change the TRUNCATE option to True by double-clicking on the False value and selecting **True** from the dropdown menu.

**Note:** The Use Default checkbox for this is automatically deselected.

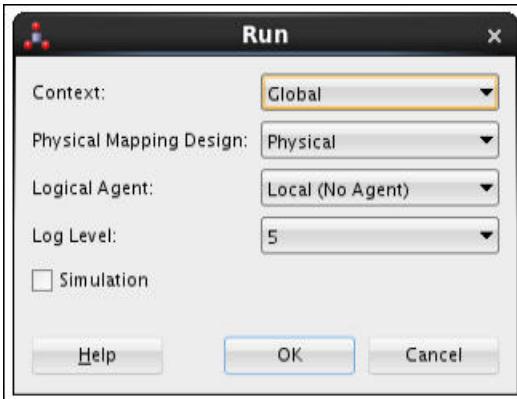


**Result:** The mapping is now complete.

15. In the ODI Taskbar above the mapping editor, click the **Run** button. (Then, select **Yes** in the Confirmation box.)



16. The Run dialog automatically appears. Click **OK** for the run dialog.



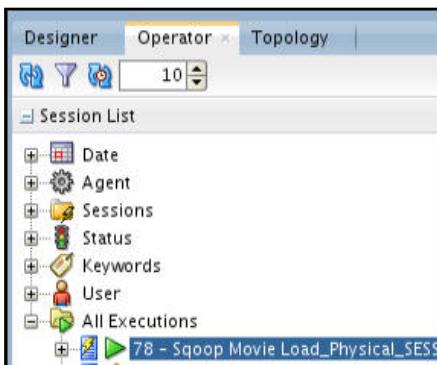
**Note:**

- The defaults use the local agent that is embedded in the ODI Studio UI.
- After a moment a Session Information dialog will appear.

17. Click **OK** in the Session Information dialog.

18. To review execution, select the **Operator** navigator and expand the **All Executions** node to see the current execution.

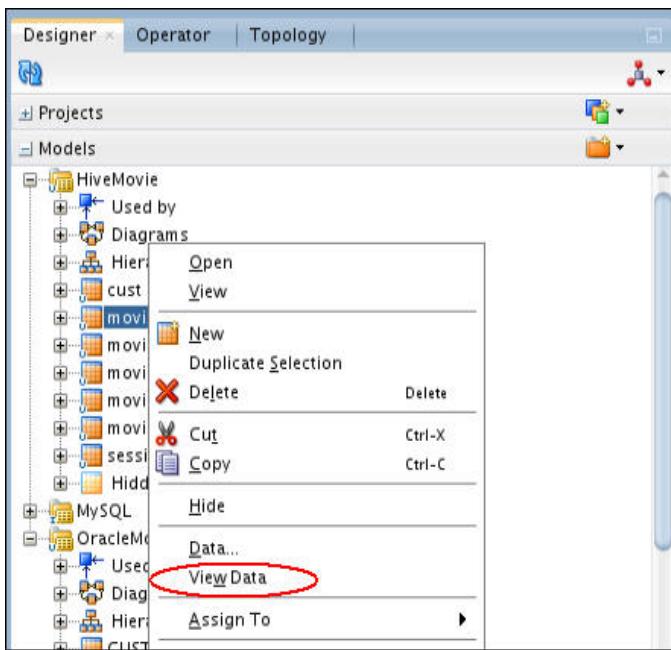
**Note:** If the execution has not have finished yet, it will show the icon for an ongoing task (green run icon). You can refresh the view by pressing the Refresh tool.



**Note:**

- Once the load is complete, a warning icon will appear. This warning icon is expected for this particular load and still means the load was successful.
- You can expand the Execution tree to see the individual tasks of the execution.

19. To view the loaded data, select the Designer navigator. Then, in the HiveMovie model, right-click on **movie** and select **View Data** from the menu.



**Result:** A Data editor appears with all rows of the movie table in Hive.

The screenshot shows the 'Sqoop Movie Load' pane with a 'Data: movie' tab selected. A data editor window is open, displaying the contents of the 'movie' table. The table has columns: movie\_id, title, year, budget, and gross. The data consists of 16 rows of movie information, such as 'Crank: High Voltage' from 2009 with a budget of 20000000 and a gross of 34560577.

	movie_id	title	year	budget	gross
1	1054798	Crank: High Voltage	2009	20000000	34560577
2	25196	Crazy Heart	2009	7000000	47405566
3	10973	Creature from the Black Lagoon	1954	0	1300000
4	36228	Genesis	2004	0	0
5	10603	George of the Jungle	1997	55000000	174463257
6	35588	Geronimo: An American Legend	1993	35000000	18635620
7	10575	Heaven	1987	0	0
8	13403	Hedwig and the Angry Inch	2001	6000000	3644200
9	50927	Helpmates	1932	0	0
10	38167	Eat Pray Love	2010	60000000	204594016
11	37058	I'm Here	2010	0	0
12	425	Ice Age	2002	59000000	383257136
13	11887	High School Musical 3: Senior Year	2008	11000000	252909177
14	8488	Hitch	2005	70000000	368100420
15	61594	Il grande duello	1972	0	0
16	2312	In the Name of the King: A Dungeon Siege II Prequel	2007	60000000	13097915

19. When done viewing the loaded data, close the Data editor.
20. Then, close the Properties inspector and the Sqoop Movie Load pane.
21. Finally, select **File > Exit** to quit Oracle Data Integrator Studio.

## **Practices for Lesson 22: Using Oracle Big Data SQL**

**Chapter 22**

## Practices for Lesson 22

---

### Practices Overview

In this case study, the MoviePlex online movie streaming company wants you to help them:

- Determine the effectiveness of their product offerings
- Enrich their understanding of its customer behavior

The company's web site collects every customer interaction in massive JSON formatted log file. By using Big Data SQL, you will unlock the information contained in log file activity data – and then by combine it with enterprise data in the company data warehouse, in order to provide the analysis that the company needs to improve its business model and make more intelligent business decisions.

In these guided practices, you perform the following tasks as part of a case study on using Oracle Big Data SQL:

- Complete the configuration of Oracle Big Data SQL on the Big Data Lite VM
- Review the HDFS data that you want to access
- Create Oracle External Tables for access to the Hadoop data
- Apply Oracle Database security over data in Hadoop data
- Execute Analytic SQL queries against joined Hadoop and RDBMS data to provide the answers to questions posed by the company

## Practice 22-1: Complete the Configuration of Big Data SQL

### Overview

In this guided practice, you use SQL Developer to complete the configuration of Big Data SQL on the Big Data Lite VM. As you learned in the lesson, the primary configuration steps include:

- Create the Common and Cluster directories on the Exadata server
- Deploy configuration files to the directories
- Create Oracle Directory objects that reference the configuration directories
- Install the required software.

All aspects of the Big Data SQL configuration are complete on the Big Data Lite VM, except for the creation of Oracle Directory objects that reference the configuration directories. Therefore, in this practice you will:

- A. Review the existing configuration directories (Common and Cluster), and the associated files
- B. Create the required Oracle Directory objects over these directories

### Assumptions

None

### Tasks

1. First, open a Terminal window, change to the Common directory location, and then view the contents of the `bigdata.properties` file:

```
cd /u01/bigdatasql_config/  
ls  
cat bigdata.properties
```

The screenshot shows a terminal window titled "oracle@bigdatalite:/u01/bigdatasql\_config". The window displays the contents of the `bigdata.properties` file. The file contains several properties, including Java settings like `java.libjvm.file` and `java.classpath.oracle`, and Hadoop-related paths like `hadoop-common.jar` and `java.classpath.hadoop`. A red box highlights the line `bigdata.properties`, which is the last line of the file. The entire file content is as follows:

```
File Edit View Search Terminal Help  
oracle@bigdatalite ~]$ cd /u01/bigdatasql_config/  
oracle@bigdatalite bigdatasql_config]$ ls  
bigdatalite bigdata-log4j.properties bigdata-log4j.properties.bak bigdata.properties hive_aux_1  
oracle@bigdatalite bigdatasql_config]$ cat bigdata.properties  
java.libjvm.file=/usr/java/latest/jre/lib/amd64/server/libjvm.so  
java.classpath.oracle=/u01/app/oracle/product/12.1.0.2/dbhome_1/jlib/oracle-hadoop-sql.jar:/u01/app/oracle/product/12.1.0.2/dbhome_1/jlib/oracle-hadoop-mapreduce.jar  
hadoop-common.jar:/u01/app/oracle/product/12.1.0.2/dbhome_1/jlib/oracle-hadoop-mapreduce.jar  
u01/app/oracle/product/12.1.0.2/dbhome_1/jlib/oracle-hadoop-mapreduce.jar  
java.classpath.hadoop=/usr/lib/hadoop/client-0.20/*:/usr/lib/hadoop-0.20-mapreduce/*:/usr/lib/hadoop-mapreduce/*  
java.classpath.hive=/usr/lib/hive/lib/*:/u01/bigdatasql_config/hive_aux_jars/hive-hcatalog-core.jar  
LD_LIBRARY_PATH=/usr/java/latest/jre/lib/amd64/server/  
bigdata.cluster.default=bigdatalite
```

### Note:

- The properties, which are not specific to a hadoop cluster, include items such as the location of the Java VM, classpaths, and the `LD_LIBRARY_PATH`.
- In addition, the last line of the file specifies the default cluster property - in this case `bigdatalite`.

- Change to the Cluster directory location and view the configuration files.

```
cd /u01/bigdatasql_config/bigdatalite
ls
```

```
[oracle@bigdatalite bigdatasql_config]$ cd /u01/bigdatasql_config/bigdatalite
[oracle@bigdatalite bigdatalite]$ ls
core-site.xml  hive-env.sh      hive-site.xml      mapred-site.xml
hdfs-site.xml  hive-env.sh-bak  hive-site.xml-bak
[oracle@bigdatalite bigdatalite]$ █
```

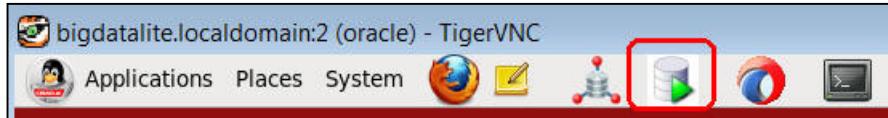
**Note:** These are the files required to connect Oracle Database to HDFS and to Hive.

- Now, define the corresponding Oracle Directory objects for these configuration directories.

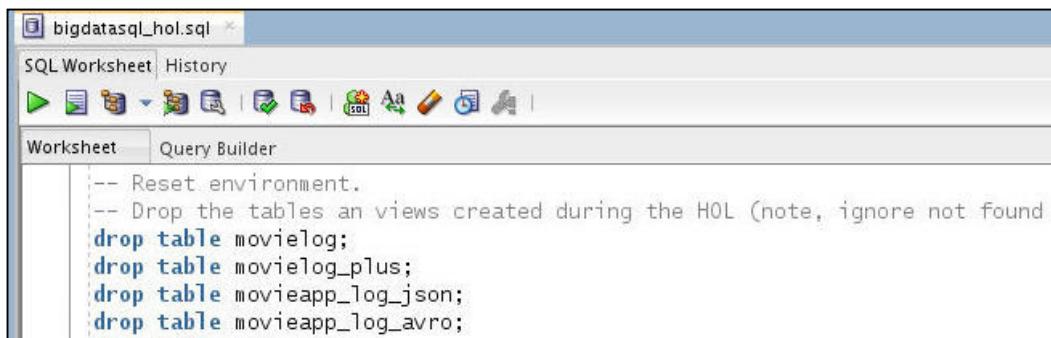
**Note:** As you learned in the lesson, the Oracle Directory objects for Big Data SQL have a specific naming convention:

- ORACLE\_BIGDATA\_CONFIG references the Common Directory
- In this case, ORACLE\_BIGDATA\_CL\_bigdatalite references the Cluster Directory. The last part of the name must match the physical directory name in the file system, and is case sensitive (therefore the quotes when creating this object).

- Launch SQL Developer by clicking on the program icon as shown here:



- Select **File > Open** and open the `bigdatasql_hol.sql` file from the `/home/oracle/bigdatasql-hol/` directory. The file is displayed as shown here.



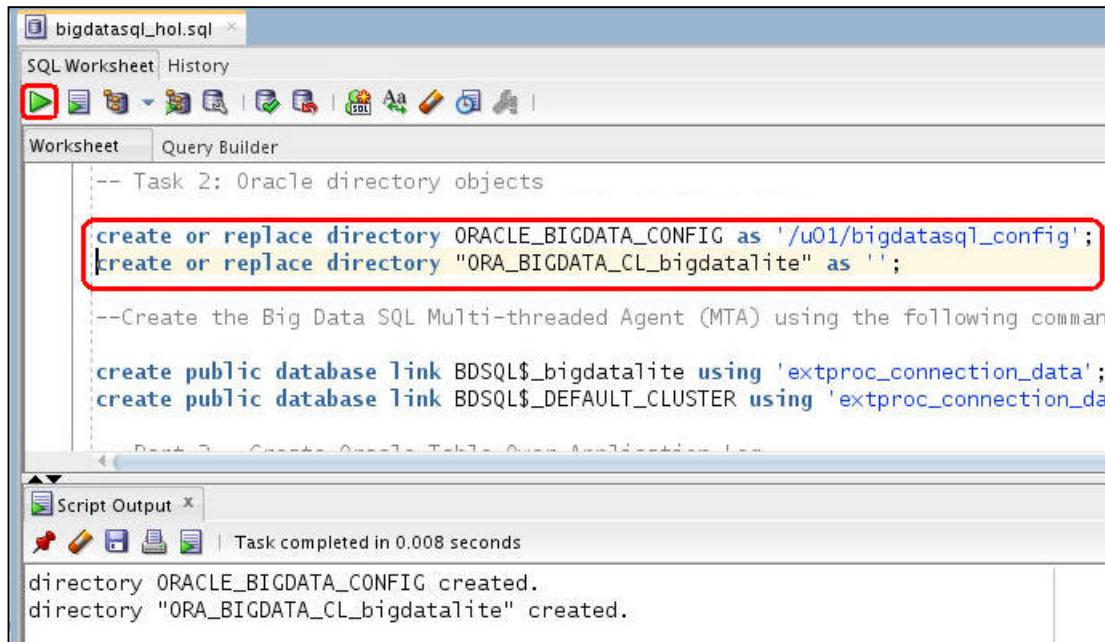
**Note:**

- This .sql file contains the commands that you will need to run from SQL Developer during the remainder of these practices.
- However, it also contains additional commands that are not used in this practice. Therefore, run only those commands that are specified in the practice steps.

C. Scroll down to the “Task 2” section, and execute the two `create or replace directory` statements that are highlighted in the screenshot below.

**Note:**

- To run any statement in a SQL Worksheet: click on the first line in the statement, and then press the **Run Statement** tool (green right-pointing arrowhead).
- If prompted for a connection, select **moviedemo** from the list and click **OK**.



The screenshot shows the Oracle SQL Developer interface. A SQL Worksheet window titled "bigdatasql\_hol.sql" is open. The worksheet contains the following code:

```
-- Task 2: Oracle directory objects
create or replace directory ORACLE_BIGDATA_CONFIG as '/u01/bigdatasql_config';
create or replace directory "ORA_BIGDATA_CL_bigdatalite" as '';

--Create the Big Data SQL Multi-threaded Agent (MTA) using the following command

create public database link BDSQL$_bigdatalite using 'extproc_connection_data';
create public database link BDSQL$_DEFAULT_CLUSTER using 'extproc_connection_da
```

The first two lines of the code are highlighted with a red rectangle. Below the worksheet is a "Script Output" pane showing the results of the execution:

```
directory ORACLE_BIGDATA_CONFIG created.
directory "ORA_BIGDATA_CL_bigdatalite" created.
```

**Result:** The two Oracle directory objects are created, as shown in the Script Output pane.

**Note:** Notice that there is no location specified for the Cluster Directory. It is expected that the directory will:

- Be a subdirectory of `ORACLE_BIGDATA_CONFIG`
- Use the cluster name as identified by the Oracle directory object

D. Leave SQL Developer open.

## Practice 22-2: Review HDFS Data That You Want to Access

### Overview

In this guided practice, you:

- Review the MoviePlex application log stored as JSON format in HDFS
- Create a simple Oracle external table over the log file
- Review the Hive tables that have been defined over the JSON data

### Assumptions

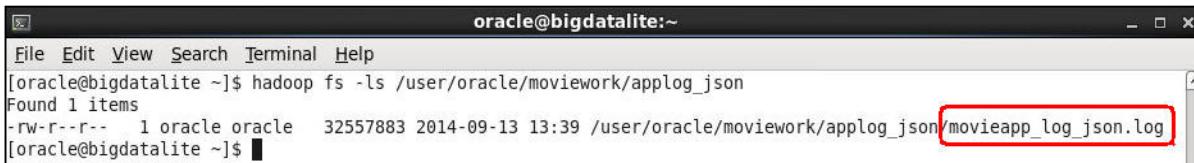
Practice 22-1 has been successfully completed.

### Tasks

1. First, review the JSON format application log data.

- A. Switch back to the terminal window and execute the following command at the \$ prompt to view the contents of the applog\_json directory:

```
hadoop fs -ls /user/oracle/moviework/applog_json
```



```
[oracle@bigdatalite ~]$ hadoop fs -ls /user/oracle/moviework/applog_json
Found 1 items
-rw-r--r-- 1 oracle oracle 32557883 2014-09-13 13:39 /user/oracle/moviework/applog_json/movieapp_log_json.log
[oracle@bigdatalite ~]$
```

- B. Now, to view the contents of the log file, execute the following command:

```
hadoop fs -tail /user/oracle/moviework/applog_json/movieapp_log_json.log
```



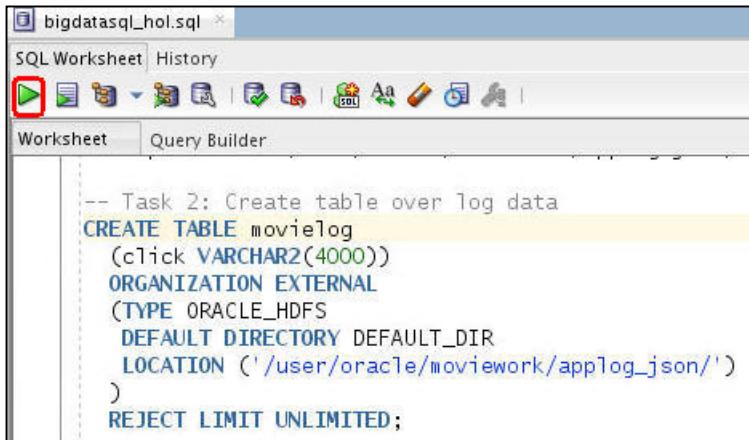
```
[oracle@bigdatalite ~]$ hadoop fs -tail /user/oracle/moviework/applog_json/movieapp_log_json.log
,"recommended":"Y","activity":2}
{"custid":1135508,"movieid":240,"genreid":8,"time":"2012-10-01:02:04:15","recommended":"Y","activity":5}
{"custid":1135508,"movieid":1092,"genreid":20,"time":"2012-10-01:02:10:23","recommended":"N","activity":5}
{"custid":1135508,"movieid":4638,"genreid":8,"time":"2012-10-01:02:10:54","recommended":"N","activity":7}
 {"custid":1135508,"movieid":4638,"genreid":8,"time":"2012-10-01:02:16:49","recommended":"N","activity":7}
 {"custid":1135508,"movieid":null,"genreid":null,"time":"2012-10-01:02:24:00","recommended":null,"activity":9}
 {"custid":1135508,"movieid":240,"genreid":8,"time":"2012-10-01:02:31:12","recommended":"Y","activity":11,"price":2.}
```

**Note:** The file contains every click that has taken place on the web site. The JSON log captures the following information about each interaction:

- custid: The customer accessing the site
- movieid: The movie that the user clicked on
- genreid: The genre that the movie belongs to
- time: When the activity occurred
- recommended: Did the customer click on a recommended movie?
- activity: A code for the various activities that can take place, including log in/out, view a movie, purchase a movie, show movie listings, etc.
- price: The price of a purchased movie

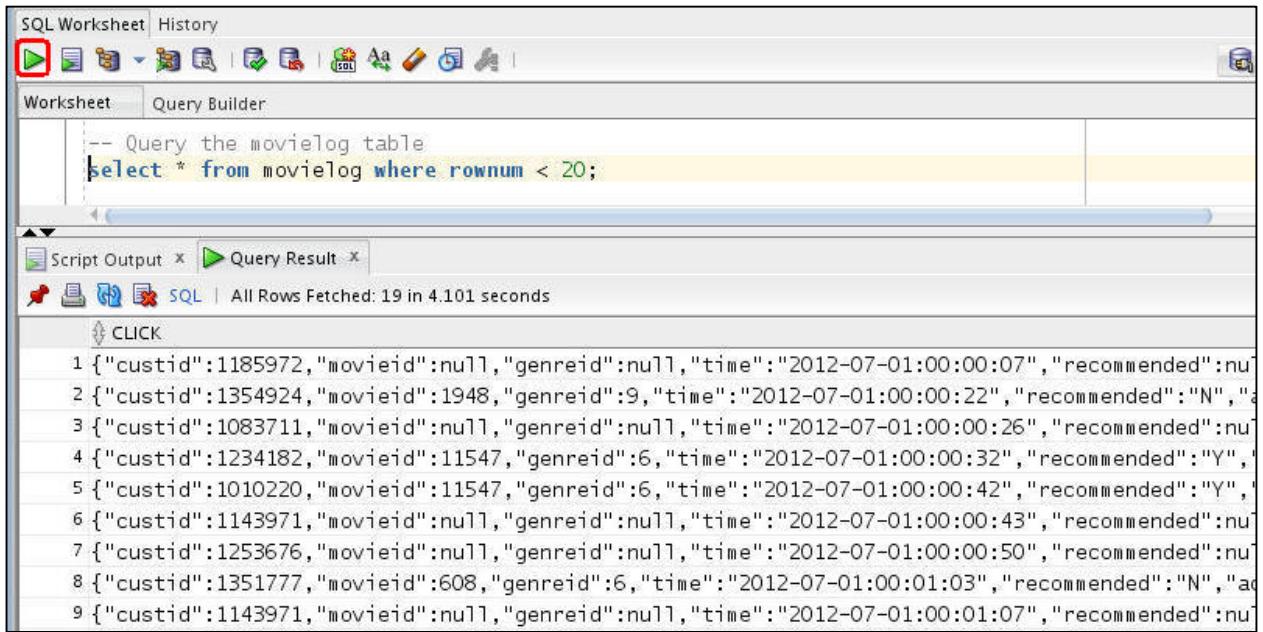
2. Create a simple Oracle external table over the file. This table will contain a single column where each record contains a JSON document.

- A. Switch back to the `bigdatasql_hol.sql` worksheet in SQL Developer, and execute the `CREATE TABLE` statement shown below:



```
-- Task 2: Create table over log data
CREATE TABLE movielog
  (click VARCHAR2(4000))
  ORGANIZATION EXTERNAL
  (TYPE ORACLE_HDFS
  DEFAULT DIRECTORY DEFAULT_DIR
  LOCATION ('/user/oracle/moviework/applog_json/')
)
REJECT LIMIT UNLIMITED;
```

- B. Scroll down just after the previous statement, and execute the following Select statement to review the data in the table `movielog`:



```
-- Query the movielog table
select * from movielog where rownum < 20;
```

Script Output x | Query Result x

SQL | All Rows Fetched: 19 in 4.101 seconds

```

1 {"custid":1185972,"movieid":null,"genreid":null,"time":"2012-07-01:00:00:07","recommended":null}
2 {"custid":1354924,"movieid":1948,"genreid":9,"time":"2012-07-01:00:00:22","recommended":"N","a...
3 {"custid":1083711,"movieid":null,"genreid":null,"time":"2012-07-01:00:00:26","recommended":null}
4 {"custid":1234182,"movieid":11547,"genreid":6,"time":"2012-07-01:00:00:32","recommended":"Y",...
5 {"custid":1010220,"movieid":11547,"genreid":6,"time":"2012-07-01:00:00:42","recommended":"Y",...
6 {"custid":1143971,"movieid":null,"genreid":null,"time":"2012-07-01:00:00:43","recommended":null}
7 {"custid":1253676,"movieid":null,"genreid":null,"time":"2012-07-01:00:00:50","recommended":null}
8 {"custid":1351777,"movieid":608,"genreid":6,"time":"2012-07-01:00:01:03","recommended":"N",...
9 {"custid":1143971,"movieid":null,"genreid":null,"time":"2012-07-01:00:01:07","recommended":null}
```

#### Note:

- The Query Results tab displays the output.
- The output looks similar to the previous `tail` statement – a record is returned for each JSON document.

- C. Scroll down again and execute the CREATE OR REPLACE VIEW statement shown below. This creates a view which will simplify queries against the JSON data. This view will also be useful in subsequent exercises when applying security policies to the table.

The screenshot shows the Oracle SQL Worksheet interface. The title bar says "SQL Worksheet History". Below it is a toolbar with various icons. The main area has tabs "Worksheet" and "Query Builder", with "Worksheet" selected. The code in the worksheet pane is:

```
-- Define a view to make it easier to query click data
CREATE OR REPLACE VIEW movielog_v AS
SELECT
    CAST(m.click.custid AS NUMBER) custid,
    CAST(m.click.movieid AS NUMBER) movieid,
    CAST(m.click.activity AS NUMBER) activity,
    CAST(m.click.genreid AS NUMBER) genreid,
    CAST(m.click.recommended AS VARCHAR2(1)) recommended,
    CAST(m.click.time AS VARCHAR2(20)) time,
    CAST(m.click.rating AS NUMBER) rating,
    CAST(m.click.price AS NUMBER) price
FROM movielog m;
```

3. Next, you review the Hive tables that have been defined over the JSON log data.

**Note:**

- Hive enables SQL access to data stored in Hadoop and NoSQL stores. There are two parts to Hive: the Hive execution engine and the Hive Metastore.
- The Hive Metastore has become the standard metadata repository for data stored in Hadoop. It contains the definitions of tables, the location of data files, and the routines required parse that data (e.g. StorageHandlers, InputFormats and SerDes). There are many query execution engines that use the Hive Metastore while bypassing the Hive execution engine. Oracle Big Data SQL is an example of such an engine.
- After reviewing these hive definitions, you will create tables in the Oracle Database that will query the underlying Hive data stored in HDFS.

- A. Switch back to the terminal window and execute the following command at the \$ prompt to launch the Hive CLI. Note: Ignore the warning messages generated by the CLI.

hive

```
.fileinputformat.split.minsize
15/02/16 10:52:42 INFO Configuration.deprecation: mapred.min.split.size.per.rack is deprecated. I
uce.input.fileinputformat.split.minsize.per.rack
15/02/16 10:52:42 INFO Configuration.deprecation: mapred.min.split.size.per.node is deprecated. I
uce.input.fileinputformat.split.minsize.per.node
15/02/16 10:52:42 INFO Configuration.deprecation: mapred.reduce.tasks is deprecated. Instead, use
uces
15/02/16 10:52:42 INFO Configuration.deprecation: mapred.reduce.tasks.speculative.execution is de
use mapreduce.reduce.speculative
15/02/16 10:52:42 WARN conf.HiveConf: DEPRECATED: Configuration property hive.metastore.local no
ect. Make sure to provide a valid value for hive.metastore.uris if you are connecting to a remote
Logging initialized using configuration in jar:file:/usr/lib/hive/lib/hive-common-0.12.0-cdh5.1.2
roperties
hive> ■
```

- B. At the `hive>` prompt, enter the following command to display the list of tables in the default hive database:

```
show tables;
```

```
hive> show tables;
OK
cust
movie
movie_fact
movie_fact_query
movie_rating
movieapp_log_avro
movieapp_log_json
movieapp_log_odistage
movieapp_log_stage
movielog
session_stats
Time taken: 0.086 seconds, Fetched: 11 row(s)
```

**Note:**

- There are several tables have been defined in the hive database. There are tables defined over Avro data, JSON data, and tab delimited text files.
- Next you will review two tables that have been defined over the JSON data.

- C. The first table is very simple in structure. Review the definition of the table by executing the following command:

```
show create table movielog;
```

```
hive> show create table movielog;
OK
CREATE EXTERNAL TABLE `movielog`(
  `click` string)
ROW FORMAT DELIMITED
  LINES TERMINATED BY '\n'
STORED AS INPUTFORMAT
  'org.apache.hadoop.mapred.TextInputFormat'
OUTPUTFORMAT
  'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'
LOCATION
  'hdfs://bigdatalite.localdomain:8020/user/oracle/moviework/applog_json'
TBLPROPERTIES (
  'transient_lastDdlTime'='1410630461')
Time taken: 0.271 seconds, Fetched: 12 row(s)
hive> ■
```

**Note:**

- There is a single string column called `click`, and the table is referring to data stored in the `/user/oracle/moviework/applog_json` folder.
- There is no special processing of the data; the table simply displays the JSON as a line of text.

D. Next, query the data in the `movielog` table by executing the following command:

```
select * from movielog limit 10;
```

```
hive> select * from movielog limit 10;
OK
{"custid":1185972,"movieid":null,"genreid":null,"time":"2012-07-01:00:00:07","recommended":null,"activity":8}
{"custid":1354924,"movieid":1948,"genreid":9,"time":"2012-07-01:00:00:22","recommended":"N","activity":7}
{"custid":1083711,"movieid":null,"genreid":null,"time":"2012-07-01:00:00:26","recommended":null,"activity":9}
{"custid":1234182,"movieid":11547,"genreid":6,"time":"2012-07-01:00:00:32","recommended":"Y","activity":7}
{"custid":1010220,"movieid":11547,"genreid":6,"time":"2012-07-01:00:00:42","recommended":"Y","activity":6}
 {"custid":1143971,"movieid":null,"genreid":null,"time":"2012-07-01:00:00:43","recommended":null,"activity":8}
 {"custid":1253676,"movieid":null,"genreid":null,"time":"2012-07-01:00:00:50","recommended":null,"activity":9}
 {"custid":1351777,"movieid":608,"genreid":6,"time":"2012-07-01:00:01:03","recommended":"N","activity":7}
 {"custid":1143971,"movieid":null,"genreid":null,"time":"2012-07-01:00:01:07","recommended":null,"activity":9}
 {"custid":1363545,"movieid":27205,"genreid":9,"time":"2012-07-01:00:01:18","recommended":"Y","activity":7}
Time taken: 0.941 seconds, Fetched: 10 row(s)
hive> ■
```

**Note:**

- Because there are no columns in the select list and no filters applied, the query simply scans the file and returning the results.
- No MapReduce job is executed.

There are more useful ways to query the JSON data. The next steps will show how Hive can parse the JSON data using a serializer/deserializer - or SerDe.

E. The second table queries that same file - however this time it is using a SerDe that will translate the attributes into columns. Review the definition of the table by executing the following command:

```
show create table movieapp_log_json;
```

```
hive> show create table movieapp_log_json;
OK
CREATE EXTERNAL TABLE `movieapp_log_json`(
`custid` int COMMENT 'from deserializer',
`movieid` int COMMENT 'from deserializer',
`genreid` int COMMENT 'from deserializer',
`time` string COMMENT 'from deserializer',
`recommended` string COMMENT 'from deserializer',
`activity` int COMMENT 'from deserializer',
`rating` int COMMENT 'from deserializer',
`price` float COMMENT 'from deserializer',
`position` int COMMENT 'from deserializer')
ROW FORMAT SERDE
  'org.apache.hive.hcatalog.data.JsonSerDe'
STORED AS INPUTFORMAT
  'org.apache.hadoop.mapred.TextInputFormat'
OUTPUTFORMAT
  'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'
LOCATION
  'hdfs://bigdatalite.localdomain:8020/user/oracle/moviework/applog_json'
TBLPROPERTIES (
  'transient_lastDdlTime'='1410635962')
Time taken: 0.092 seconds, Fetched: 20 row(s)
hive> ■
```

**Note:**

- There are columns defined for each field in the JSON document - making it much easier to understand and query the data.
- A java class `org.apache.hive.hcatalog.data.JsonSerDe` is used to deserialize the JSON file.

This is also an illustration of Hadoop's schema on read paradigm; a file is stored in HDFS, but there is no schema associated with it until that file is read. These two examples use two different schemas to read that same data; these schemas are encapsulated by the Hive tables `movieilog` and `movieapp_log_json`.

- F. Next, query the data in the `movieapp_log_json` table by executing the following command (the query may take a moment to return these results):

```
select * from movieapp_log_json where rating > 4;
```

```
1237866 8367 15 2012-09-30:08:09:51 N 1 5 NULL NULL  
1126945 9008 46 2012-09-30:11:27:38 Y 1 5 NULL NULL  
1420278 862 2 2012-09-30:14:00:27 Y 1 5 NULL NULL  
1444055 675 12 2012-09-30:17:25:01 N 1 5 NULL NULL  
1431502 94730 30 2012-09-30:18:14:41 Y 1 5 NULL NULL  
1040916 1135249 14 2012-09-30:18:43:58 Y 1 5 NULL NULL  
1015245 48988 6 2012-09-30:18:55:11 N 1 5 NULL NULL  
1201929 9913 9 2012-09-30:19:12:19 N 1 5 NULL NULL  
1171159 116 8 2012-09-30:21:18:37 Y 1 5 NULL NULL  
1094886 217 11 2012-09-30:22:44:21 N 1 5 NULL NULL  
1178337 19908 6 2012-09-30:23:27:47 N 1 5 NULL NULL  
1084372 544 6 2012-10-01:01:11:35 N 1 5 NULL NULL  
Time taken: 18.662 seconds, Fetched: 704 row(s)  
hive> ■
```

**Note:** This is a much better way to query and view the data than in the previous table.

- The Hive query execution engine converted this query into a MapReduce job.
- The author of the query does not need to worry about the underlying implementation - Hive handles this automatically.

- G. At the `hive>` prompt, execute the `exit;` command to close the Hive CLI.

- H. Then, exit the Terminal window and return to the SQL Developer window.

## Practice 22-3: Leverage Hive Metadata for the Oracle External Tables - Then Query the Hadoop Data

### Overview

Oracle Big Data SQL is able to leverage the Hive metadata when creating and querying external tables.

In this guided practice, you:

- Create Oracle external tables over two Hive tables: `movieapp_log_json` and `movieapp_log_avro`. Oracle Big Data SQL will utilize the existing InputFormats and SerDes required to process this data.
- Query the log data using SQL

### Assumptions

Practice 22-2 has been successfully completed.

### Tasks

1. Back in SQL Developer, scroll down to the “Task 2: Leverage Hive Metatdata” section and run the `CREATE TABLE` statement shown below to create an external table over the Hive `movieapp_log_json` table:

The screenshot shows the Oracle SQL Developer interface with a SQL Worksheet window open. The worksheet contains the following SQL code:

```
-- Task 2: Leverage Hive Metadata when Creating Oracle Tables

-- Create table over JSON
CREATE TABLE movieapp_log_json (
    custid      INTEGER ,
    movieid     INTEGER ,
    genreid     INTEGER ,
    time        VARCHAR2(20) ,
    recommended VARCHAR2(4) ,
    activity    NUMBER,
    rating      INTEGER,
    price       NUMBER
)
ORGANIZATION EXTERNAL
(
    TYPE ORACLE_HIVE
    DEFAULT DIRECTORY DEFAULT_DIR
)
REJECT LIMIT UNLIMITED;
```

The code is highlighted with syntax coloring. A red box highlights the play button icon in the toolbar above the worksheet, indicating the script is ready to be executed. Below the worksheet, the Script Output tab shows the execution results:

```
directory ORACLE_BIGDATA_CONFIG created.
directory "ORA_BIGDATA_CL_bigdatalite" created.
table MOVIELOG created.
view MOVIELOG_V created.
table MOVIEAPP_LOG_JSON created.
```

**Note:** The ORACLE\_HIVE access driver invokes Big Data SQL at query compilation time to retrieve the metadata details from the Hive Metastore. By default, it will query the metastore for a table name that matches the name of the external table: movieapp\_log\_json. As you will see later, this default can be overridden using ACCESS PARAMETERS.

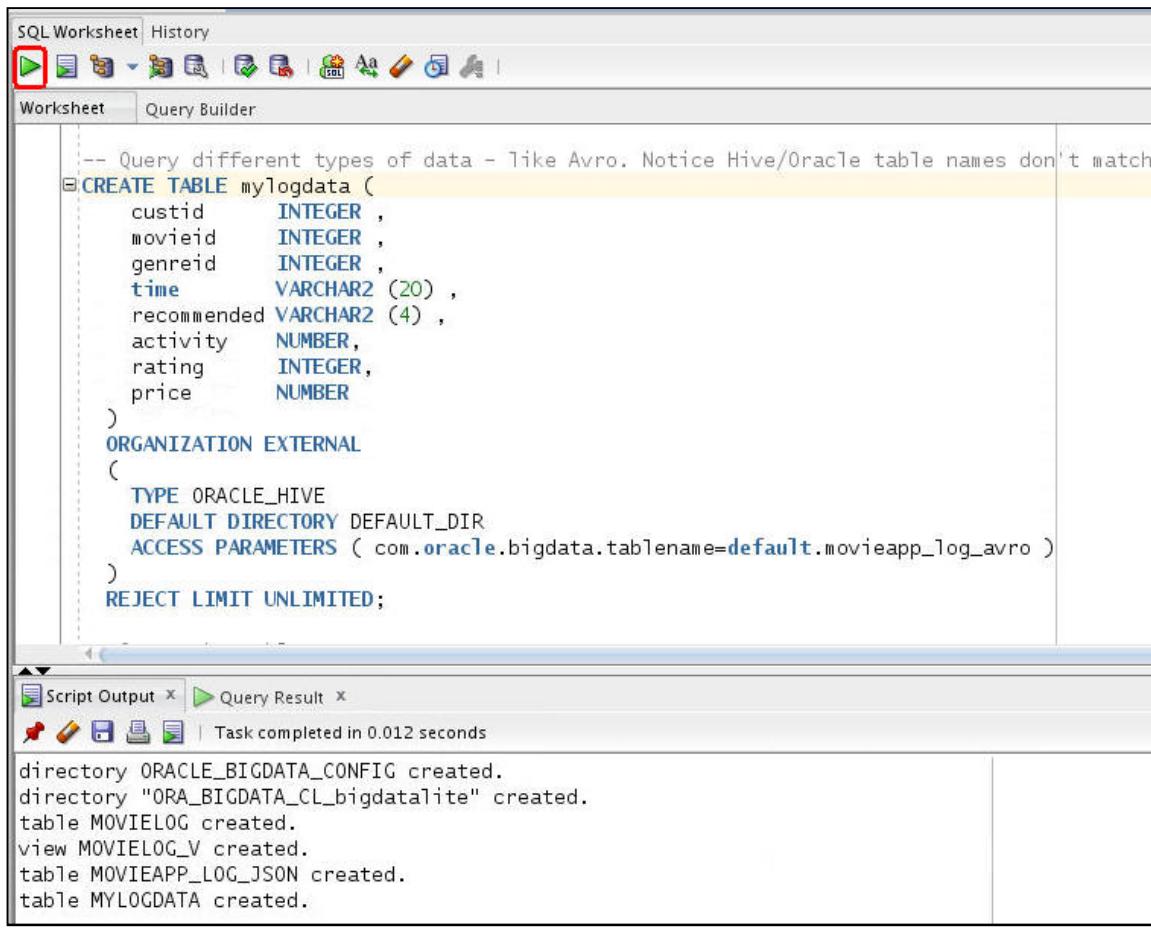
2. Query the table using the SELECT statement shown below:

```
-- Query the table - find clicks where movies were highly rated movies.
SELECT * FROM movieapp_log_json WHERE rating > 4;
```

CUSTID	MOVIEID	GENREID	TIME	RECOMMENDED	ACTIVITY	RATING	PRICE
1	1126174	1647	9 2012-07-01:00:20:11 N		1	5 (null)	
2	1161010	15121	25 2012-07-01:01:35:04 Y		1	5 (null)	
3	1161861	752	45 2012-07-01:01:55:40 Y		1	5 (null)	
4	1303830	242	8 2012-07-01:03:11:37 N		1	5 (null)	
5	1446850	278	24 2012-07-01:03:54:20 Y		1	5 (null)	
6	1180818	585	6 2012-07-01:04:38:28 Y		1	5 (null)	
7	1134664	8587	25 2012-07-01:04:49:17 Y		1	5 (null)	
8	1180818	855	3 2012-07-01:07:48:37 Y		1	5 (null)	
9	1368963	862	6 2012-07-01:09:51:46 Y		1	5 (null)	
10	1020256	1128682	8 2012-07-01:09:57:30 N		1	5 (null)	

**Note:**

- As mentioned previously, at query compilation time, Oracle Big Data SQL queries the Hive Metastore for all the information required to select data. This metadata includes the location of the data and the classes required to process the data (e.g. StorageHandlers, InputFormats and SerDes).
  - In this example, Big Data SQL scanned the files found in the /user/oracle/movie/moviework/applog\_json directory and then used the Hive SerDe to parse each JSON document.
  - In a true Oracle Big Data Appliance environment, the input splits would be processed in parallel across the nodes of the cluster by the Big Data SQL Server, the data would then be filtered locally using Smart Scan, and only the filtered results (rows and columns) would be returned to Oracle Database.
- There is a second Hive table over the same movie log content - except the data is in Avro format - not JSON text format. Create an Oracle table over that Avro-based Hive table using the following CREATE TABLE statement:



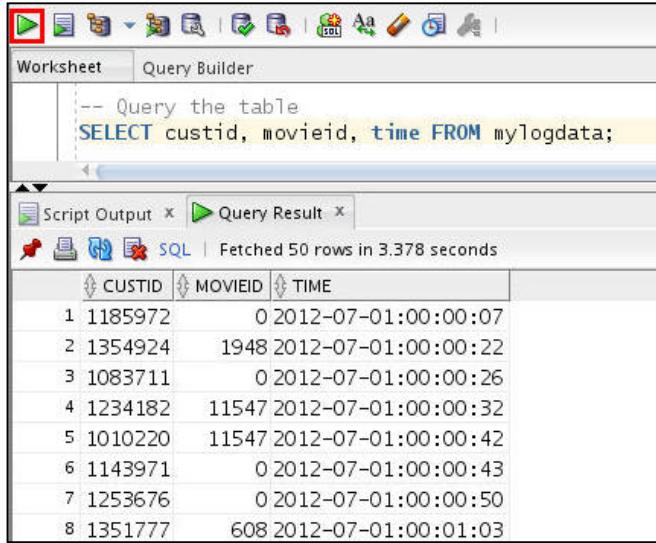
```
-- Query different types of data - like Avro. Notice Hive/Oracle table names don't match
CREATE TABLE mylogdata (
    custid      INTEGER ,
    movieid     INTEGER ,
    genreid     INTEGER ,
    time        VARCHAR2 (20) ,
    recommended VARCHAR2 (4) ,
    activity    NUMBER ,
    rating      INTEGER ,
    price       NUMBER
)
ORGANIZATION EXTERNAL
(
    TYPE ORACLE_HIVE
    DEFAULT DIRECTORY DEFAULT_DIR
    ACCESS PARAMETERS ( com.oracle.bigdata.tablename=default.movieapp_log_avro )
)
REJECT LIMIT UNLIMITED;
```

Script Output | Task completed in 0.012 seconds

```
directory ORACLE_BIGDATA_CONFIG created.
directory "ORA_BIGDATA_CL_bigdatalite" created.
table MOVIELOG created.
view MOVIELOG_V created.
table MOVIEAPP_LOG_JSON created.
table MYLOGDATA created.
```

**Note:** In this instance, the Oracle table name does not match the Hive table name. Therefore, an ACCESS PARAMETER was specified that references the Hive table.

4. Query the mylogdata table using the following SELECT statement:



```
-- Query the table
SELECT custid, movieid, time FROM mylogdata;
```

Script Output | SQL | Fetched 50 rows in 3.378 seconds

CUSTID	MOVIEID	TIME
1	1185972	0 2012-07-01:00:00:07
2	1354924	1948 2012-07-01:00:00:22
3	1083711	0 2012-07-01:00:00:26
4	1234182	11547 2012-07-01:00:00:32
5	1010220	11547 2012-07-01:00:00:42
6	1143971	0 2012-07-01:00:00:43
7	1253676	0 2012-07-01:00:00:50
8	1351777	608 2012-07-01:00:01:03

**Note:** Oracle Big Data SQL utilized the Avro InputFormat to query the data.

## Practice 22-4: Apply Oracle Security Policies Over Hadoop Data

### Overview

In our case study example, we need to protect personally identifiable information, including the customer last name and customer id. To accomplish this task, an Oracle Data Redaction policy that obscures these two fields has already been set up on the `customer` table in the sample schema.

Note: Do not run this code. This has already been executed by using the `DBMS_REDACT` PL/SQL package. The code is shown here for your information only.

```
DBMS_REDACT.ADD_POLICY(
    object_schema => 'MOVIEDEMO',
    object_name => 'CUSTOMER',
    column_name => 'CUST_ID',
    policy_name => 'customer_redaction',
    function_type => DBMS_REDACT.PARTIAL,
    function_parameters => '9,1,7',
    expression => '1=1'
);

DBMS_REDACT.ALTER_POLICY(
    object_schema => 'MOVIEDEMO',
    object_name => 'CUSTOMER',
    action => DBMS_REDACT.ADD_COLUMN,
    column_name => 'LAST_NAME',
    policy_name => 'customer_redaction',
    function_type => DBMS_REDACT.PARTIAL,
    function_parameters =>
'VVVVVVVVVVVVVVVVVVVVVVVVV, VVVVVVVVVVVVVVVVVVVVVVVVVV, *, 3, 25',
    expression => '1=1'
);
```

Notes about the redaction policies:

The first PL/SQL call creates a policy called `customer_redaction`:

- It is applied to the `cust_id` column in the `moviedemo.customer` table
- It performs a partial redaction, replacing the first 7 characters with the number "9"
- The `expression => "1=1"` setting specifies that the redaction policy will always apply

The second API call updates the `customer_redaction` policy, redacting a second column in that same table. It will replace the characters 3 to 25 in the `LAST_NAME` column with an '\*'.

The application of redaction policies does not change underlying data. Oracle Database performs the redaction at execution time, as the data is displayed to the application user.

In this guided practice, you apply an equivalent redaction policy to two of our Oracle Big Data SQL tables, which will have the following effects:

- The first procedure redacts data sourced from JSON in HDFS
- The second procedure redacts Avro data sourced from Hive
- Both policies will redact the `custid` attribute.

### Assumptions

Practice 22-3 has been successfully completed.

## Tasks

- In SQL Developer, apply the equivalent redaction policy on the data sourced from both the HDFS and Hive sources, by executing the following PL/SQL Package:

Note: Click on the first line of the package (BEGIN statement), and click **Run Statement**.

```
-- Part 4: Apply Oracle Security Policies over Data in Hadoop
-- Task 1 - Redact data sourced from both HDFS and Hive sources

BEGIN
    -- JSON file in HDFS
    DBMS_REDACT.ADD_POLICY(
        object_schema => 'MOVIEDEMO',
        object_name => 'MOVIELOG_V',
        column_name => 'CUSTID',
        policy_name => 'movielog_v_redaction',
        function_type => DBMS_REDACT.PARTIAL,
        function_parameters => '9,1,7',
        expression => '1=1'
    );

    -- Avro data from Hive
    DBMS_REDACT.ADD_POLICY(
        object_schema => 'MOVIEDEMO',
        object_name => 'MYLOGDATA',
        column_name => 'CUSTID',
        policy_name => 'mylogdata_redaction',
        function_type => DBMS_REDACT.PARTIAL,
        function_parameters => '9,1,7',
        expression => '1=1'
    );
END;
/
```

**Note:** The Script Output tab shows anonymous block completed.

- Query the Avro data with a SELECT statement. Notice the redacted CUSTID column.

```
-- Review the redacted data
SELECT * FROM mylogdata WHERE rounum < 20;
```

CUSTID	MOVIEID	CENREID	TIME	RECOMMENDED	ACTIVITY	RATING	PRICE
1 9999999	0	0	2012-07-01:00:00:07 null		8 (null) (null)		
2 9999999	1948	9	2012-07-01:00:00:22 N		7 (null) (null)		
3 9999999	0	0	2012-07-01:00:00:26 null		9 (null) (null)		
4 9999999	11547	6	2012-07-01:00:00:32 Y		7 (null) (null)		
5 9999999	11547	6	2012-07-01:00:00:42 Y		6 (null) (null)		
6 9999999	0	0	2012-07-01:00:00:43 null		8 (null) (null)		

3. Now, join the redacted HDFS data to the Oracle Database `customer` table by executing the following `SELECT` statement:

```
-- Review the redacted data
SELECT * FROM mylogdata WHERE rownum < 20;

-- Join the redacted HDFS table with the redacted customer table
SELECT f.custid, c.last_name, f.movieid, f.time
FROM customer c, movielog_v f
WHERE c.cust_id = f.custid;
```

CUSTID	LAST_NAME	MOVIEID	TIME
1 9999999 La****	(null)	2012-07-01:00:00:07	
2 9999999 Bu****	1948	2012-07-01:00:00:22	
3 9999999 Cu*****	(null)	2012-07-01:00:00:26	
4 9999999 Ha****	11547	2012-07-01:00:00:32	
5 9999999 Re****	11547	2012-07-01:00:00:42	
6 9999999 Go*****	(null)	2012-07-01:00:00:43	
7 9999999 On*****	(null)	2012-07-01:00:00:50	
8 9999999 Re*****	608	2012-07-01:00:01:03	
9 9999999 Go*****	(null)	2012-07-01:00:01:07	
10 9999999 E1****	27205	2012-07-01:00:01:18	
11 9999999 Be*****	1124	2012-07-01:00:01:26	

**Note:** As you can see, the redacted data sourced from Hadoop works seamlessly with the rest of the data in your Oracle Database.

## Practice 22-5: Using Analytic SQL on Joined Data

### Overview

In this guided practice, you will enrich MoviePlex's understanding of customers by utilizing an RFM analysis. This query will identify:

- Recency: When was the last time the customer accessed the site?
- Frequency: What is the level of activity for that customer on the site?
- Monetary: How much money has the customer spent?

To answer these questions, SQL Analytic Functions are applied to data residing in both the application logs on Hadoop, and in sales data in Oracle Database tables. Customer's behavior will be ranked on a scale of 1-5 in each area (1=lowest, 5=highest). For example, an RFM combined score of 551 indicates the customers behavior in terms of:

- Recent visits (R=5)
- Activity on the site (F=5)
- Spending (M=1)

It looks like this customer performs a lot of research on the site, but then buys elsewhere!

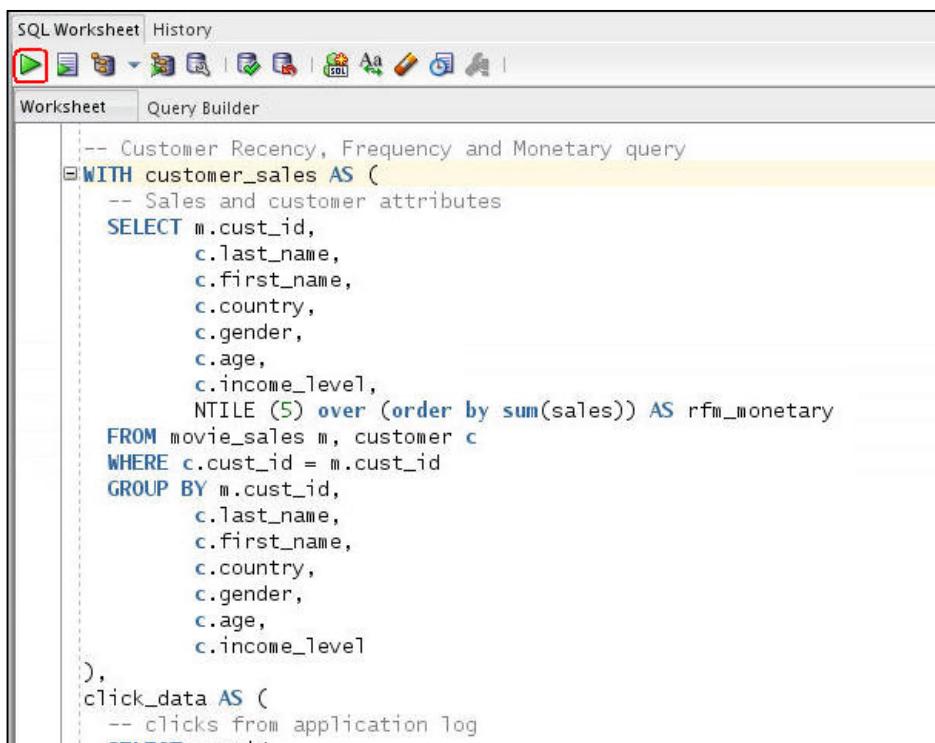
### Assumptions

Practice 22-4 has been successfully completed.

### Tasks

1. Scroll down in the script and run the following Customer RFM query, which applies Oracle NTILE functions across all of the data.

**Note:** This long query is shown in two parts below. Run it by using the same techniques as before: click on the first line of the package (WITH statement), and click **Run Statement**.



The screenshot shows an Oracle SQL Worksheet window. The title bar says "SQL Worksheet | History". Below the title bar is a toolbar with various icons. The main area is divided into two tabs: "Worksheet" (selected) and "Query Builder". The "Worksheet" tab contains the following SQL code:

```
-- Customer Recency, Frequency and Monetary query
WITH customer_sales AS (
    -- Sales and customer attributes
    SELECT m.cust_id,
        c.last_name,
        c.first_name,
        c.country,
        c.gender,
        c.age,
        c.income_level,
        NTILE (5) over (order by sum(sales)) AS rfm_monetary
    FROM movie_sales m, customer c
    WHERE c.cust_id = m.cust_id
    GROUP BY m.cust_id,
        c.last_name,
        c.first_name,
        c.country,
        c.gender,
        c.age,
        c.income_level
),
click_data AS (
    -- clicks from application log
)
```

```
-- clicks from application log
SELECT custid,
       NTILE (5) over (order by max(time)) AS rfm_recency,
       NTILE (5) over (order by count(1))    AS rfm_frequency
  FROM movie_log_v
 GROUP BY custid
)
SELECT c.cust_id,
       c.last_name,
       c.first_name,
       cd.rfm_recency,
       cd.rfm_frequency,
       c.rfm_monetary,
       cd.rfm_recency*100 + cd.rfm_frequency*10 + c.rfm_monetary AS rfm_combined,
       c.country,
       c.gender,
       c.age,
       c.income_level
  FROM customer_sales c, click_data cd
 WHERE c.cust_id = cd.custid
;
```

**Result:** The output looks like this:

	CUST_ID	LAST_NAME	FIRST_NAME	RFM_RECENCY	RFM_FREQUENCY	RFM_MONETARY	RFM_COMBINED	COUNTRY	GENDER	AGE
1	1355633	Li*****	Allie	1	2	1	121	United States	Female	31
2	1149089	Is*****	Aleron	4	1	1	411	United Kingdom	Male	21
3	1403696	No*****	Kalakanya	1	1	1	111	Mexico	Female	48
4	1105457	An***	Nanon	1	1	1	111	Portugal	Female	29
5	1001430	At*****	Mel	2	2	1	221	United States	Male	42
6	1345248	Tu***	Roesia	1	1	1	111	Italy	Female	34
7	1032193	Sa*****	Kanna	2	2	1	221	Brazil	Female	47
8	1072542	Bi*****	Pia	2	3	1	231	Germany	Female	46
9	1263220	Tr*****	Francois	2	1	1	211	Thailand	Male	16
10	1310816	Ta*****	Paki	2	1	1	211	Egypt	Male	72
11	1380526	Ks*****	Edmond	1	1	1	111	Canada	Male	29
12	1343065	La*	Xing-Jiang	3	1	1	311	China	Female	74
13	1064674	So***	Brigitte	5	3	1	531	Argentina	Female	40
14	1220753	Gu*****	Barbara	5	2	1	521	United States	Female	22
15	1042019	Sa*****	Ksa	5	1	1	511	India	Female	79
16	1163928	Ag*****	Tatiana	3	1	1	311	United Kingdom	Female	75

#### Note:

- The `customer_sales` subquery selects from the Oracle Database fact table `movie_sales` to categorize customers based on sales.
- The `click_data` subquery performs a similar task for web site activity stored in the application logs - categorizing customers based on their activity and recent visits.
- These two subqueries are then joined to produce the complete RFM score.

2. Now, target customers who the company may be losing to competition. To do this, execute the following amended the query -- which finds important customers (high monetary score) that have not visited the site recently (low recency score).

Run this next query (shown here in two parts):

```

SQL Worksheet History
Worksheet Query Builder

-- Find important customers who haven't visited in a while
WITH customer_sales AS (
    -- Sales and customer attributes
    SELECT m.cust_id,
        c.last_name,
        c.first_name,
        c.country,
        c.gender,
        c.age,
        c.income_level,
        NTILE(5) over (order by sum(sales)) AS rfm_monetary
    FROM movie_sales m, customer c
    WHERE c.cust_id = m.cust_id
    GROUP BY m.cust_id,
        c.last_name,
        c.first_name,
        c.country,
        c.gender,
        c.age,
        c.income_level
),
click_data AS (
    -- clicks from application log
    SELECT custid,
        NTILE(5) over (order by max(time)) AS rfm_recency,
        NTILE(5) over (order by count(1)) AS rfm_frequency
    FROM movielog_v
    GROUP BY custid
)
SELECT c.cust_id,
    c.last_name,
    c.first_name,
    cd.rfm_recency,
    cd.rfm_frequency,
    c.rfm_monetary,
    cd.rfm_recency*100 + cd.rfm_frequency*10 + c.rfm_monetary AS rfm_combined,
    c.country,
    c.gender,
    c.age,
    c.income_level
FROM customer_sales c, click_data cd
WHERE c.cust_id = cd.custid
AND c.rfm_monetary >= 4
AND cd.rfm_recency <= 2
ORDER BY c.rfm_monetary desc, cd.rfm_recency desc
;

```

**Result:** The output looks like the following:

Script Output | Query Result | Fetched 50 rows in 2.031 seconds

CUST_ID	LAST_NAME	FIRST_NAME	RFM_RECEI...	RFM_FREQU...	RFM_MONET...	RFM_COMBI...	COUNTRY	GENDER	AGE	INCOMELEV
1	1186226 Ra***	Azeggagh	2	4	5	245	Mexico	Male	35	D: 70,000
2	1250279 Na*****	Mantel	2	4	5	245	France	Male	36	A: Below 3
3	1168239 Ya****	Hayato	2	4	5	245	Japan	Male	34	D: 70,000
4	1154570 Ba***	Karthikkeyan	2	5	5	255	United Kingdom	Male	29	B: 30,000
5	1071072 So**	Harjinder	2	5	5	255	Argentina	Male	23	C: 50,000
6	1170897 In***	Miyu	2	3	5	235	Japan	Female	28	B: 30,000
7	1138496 Ya*****	Kanakasundari	2	4	5	245	Japan	Female	37	D: 70,000
8	1001640 Mc***	MeL	2	4	5	245	United States	Female	19	A: Below 3
9	1347515 Ha*	Ju-Long	2	4	5	245	China	Male	63	C: 50,000
10	1140198 Ab*****	Margo	2	3	5	235	Mexico	Female	55	A: Below 3
11	1000928 Bo***	Feidhlim	2	4	5	245	Mexico	Male	66	A: Below 3
12	1063155 Pa***	Cornelius	2	4	5	245	United States	Male	67	D: 70,000
13	1237086 Go**	Nanami	2	3	5	235	Japan	Female	67	C: 50,000
14	1034809 Ma*****	Dillon	2	3	5	235	United States	Male	66	A: Below 3
15	1031895 Mc*****	Marci	2	3	5	235	United States	Female	67	E: 90,000
16	1145161 Fi***	Cakramardika	2	4	5	245	Hungary	Female	57	C: 50,000
17	1015416 Ca***	Aurora	2	3	5	235	United States	Female	69	A: Below 3
18	1070014 Ma***	Condaco	2	3	5	225	United States	Female	66	A: Below 3

**Note:** These are the at-risk customers for the MoviePlex company. They were at one time active, big spenders on the site. But, they are not active visitors recently.

- Close the `bigdatasql_hol.sql` file and exit SQL Developer.

**Note:**

After the class is over, you may access additional hands-on practice instructions for using Big Data SQL with the Big Data Lite VM environment.

- Go to [Oracle Learning Library](https://apexapps.oracle.com/pls/apex/f?p=44785:1:0) at <https://apexapps.oracle.com/pls/apex/f?p=44785:1:0>
- Search for “**Analyze All Your Data with Oracle Big Data SQL**” at the home page.
- Launch the tutorial with the same name.

This tutorial provides additional detail on all of the practices found in this activity document. It also contains an additional practice: “Part 6: Using SQL Pattern Matching”.



## **Practices for Lesson 23: Using Oracle Advanced Analytics: Oracle Data Mining and Oracle R Enterprise**

**Chapter 23**

## Practices for Lesson 23

---

### Overview

In these guided practices, you use:

- The Oracle Data Miner GUI to create a data mining workflow and perform data mining activities on big data
- Oracle R Enterprise (ORE) to perform statistical analysis on big data

## Practice 23-1: Using Oracle Data Miner 4.0 with Big Data

### Overview

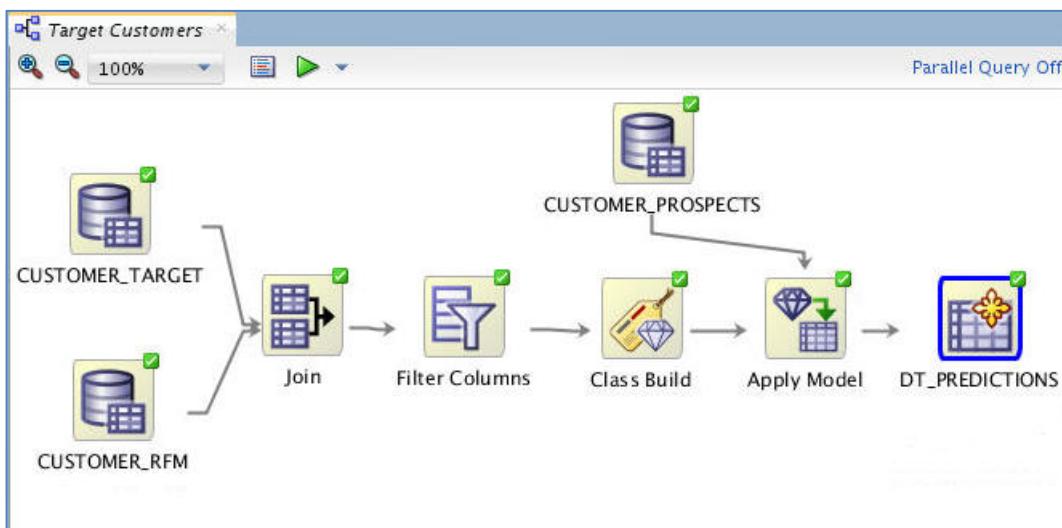
In this guided practice, you use the Oracle Data Miner 4.0 GUI to perform predictive analysis against data on the Big Data Lite sample data.

In this case study, the “Electronics R Us” company wants you to identify customers who are most likely to be the most valuable customers in terms of sales.

To predict the company's likely best customers, you create a Data Miner workflow and add visual elements (workflow nodes) that enable you to:

- Select and join source data from two tables
- Build and compare several Classification models
- Select and run the models that produce the most actionable results

The completed workflow looks like this:



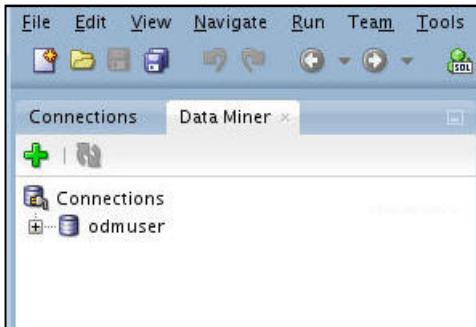
Follow the instructions in this guided practice to build the workflow, run the classification models, select the best model, and examine the predictive results.

### Tasks

1. To begin, click on the program icon shown here launch SQL Developer:



2. Select **View > Data Miner > Data Miner Connections** from the SQL Developer menu bar. Result: The Data Miner tab opens, like this:



3. Close the SQL Developer Start Page and the Connections tab, leaving only the Data Miner tab visible, like this:

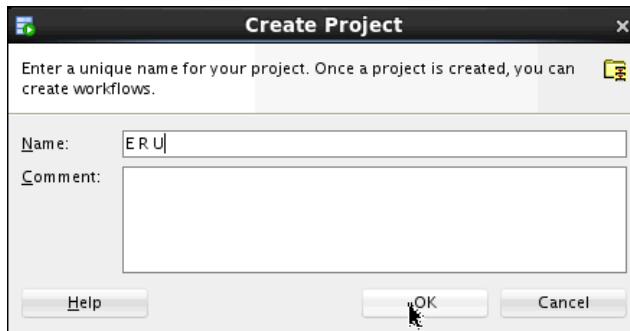


4. Next, create a Data Miner Project by performing the following steps:

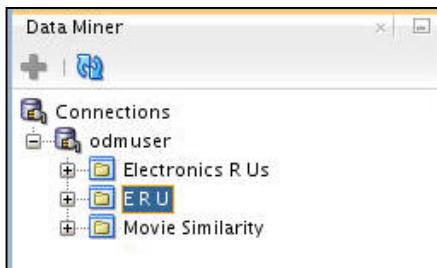
- A. In the Data Miner tab, right-click **odmuser** and select **New Project**, as shown here:



- B. In the Create Project window, enter the name **E R U** and then click **OK**.

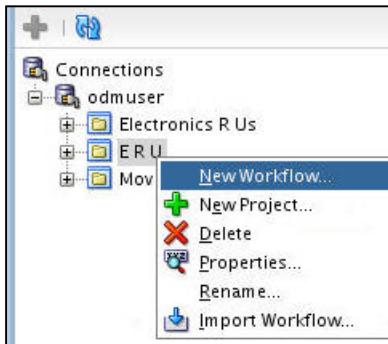


**Result:** The new project appears below odmuser connection node.

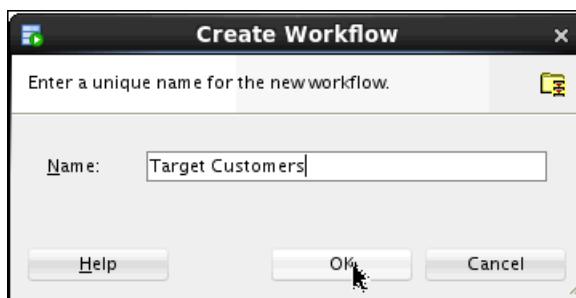


5. Next, you create the Workflow and add two Data Sources.

A. Right-click your project (E R U) and select **New Workflow** from the menu.

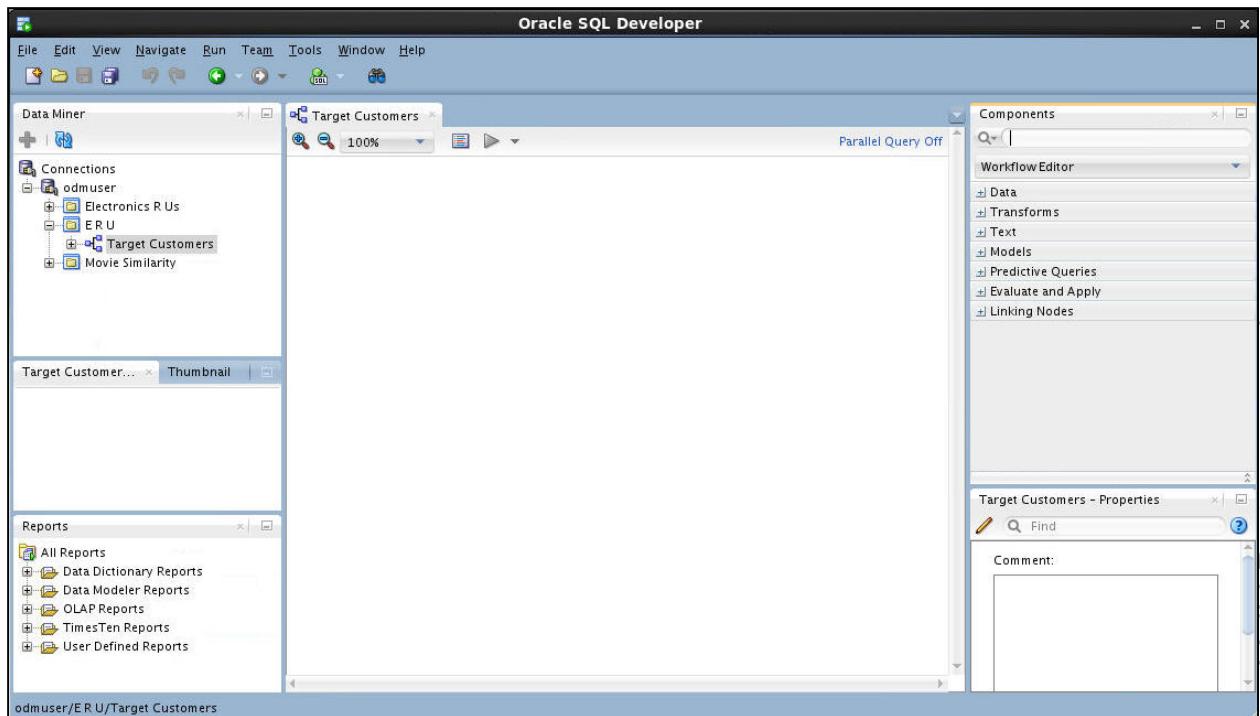


B. In the Create Workflow window, enter **Target Customers** as the name and click **OK**.



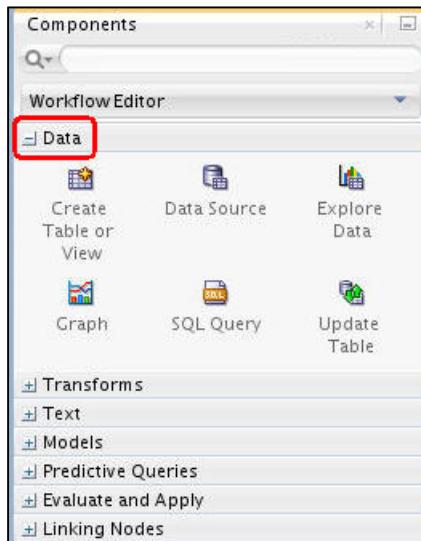
**Result:**

- In the middle of the SQL Developer window, an empty workflow tabbed window opens with the name that you specified.
- On the upper right-hand side of the interface, the Components tab of the Workflow Editor appears.
- On the lower right-hand side of the interface, the Properties tab appears.
- In addition, three other Oracle Data Miner interface elements may be opened on the lower left-hand side of the interface. These will be examined later



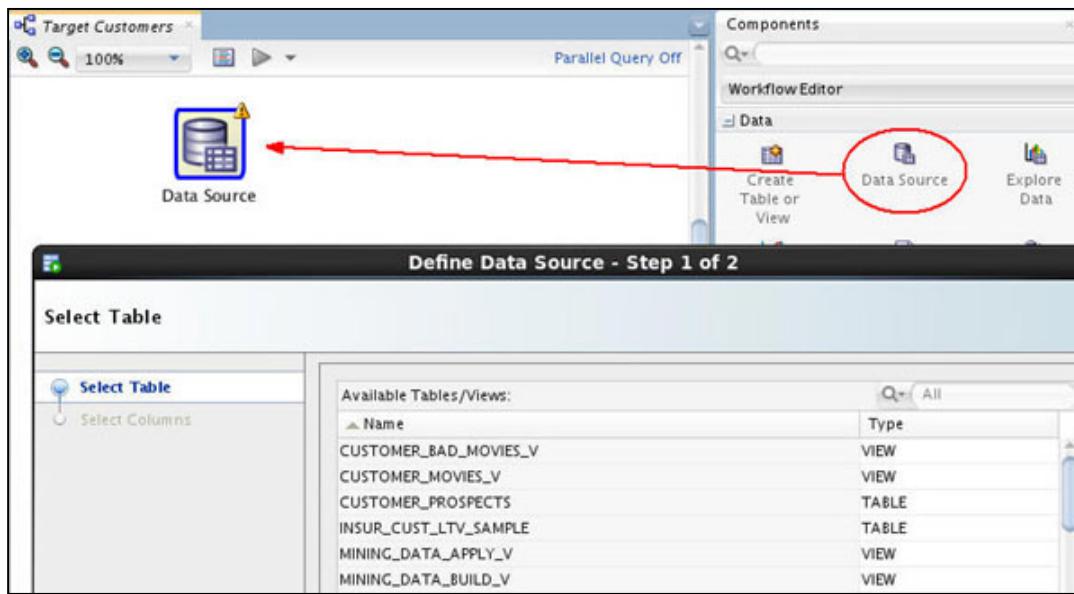
**Note:** As you will learn, you can open, close, resize, and move Data Miner tabbed panes around the SQL Developer window to suit your needs.

- C. The first element of any workflow is the source data. Here, you add two Data Source nodes to the workflow that identify customer data. In the Components tab, drill on the **Data** category. A group of six data nodes appear, as shown here:



- D. Drag and drop the Data Source node onto the Workflow pane.

**Result:** A Data Source node appears in the Workflow pane and the Define Data Source wizard opens, like this:

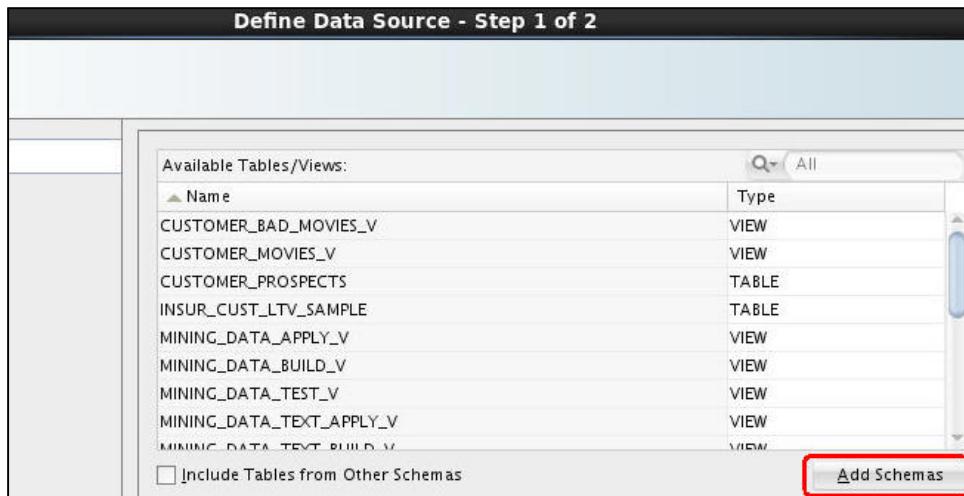


**Note:**

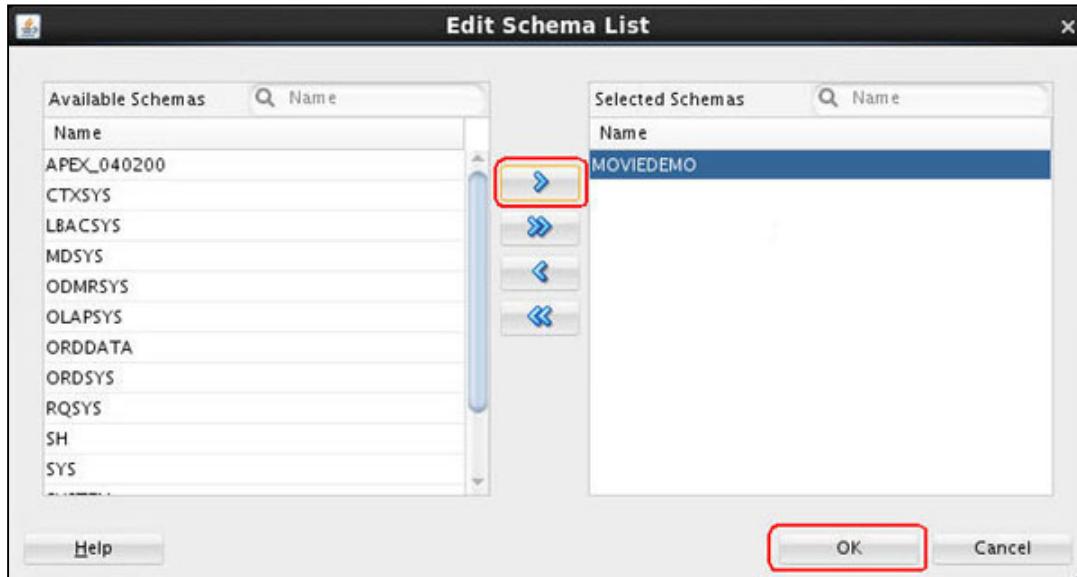
- Workspace node names and model names are generated automatically by Oracle Data Miner. In this example, the name "Data Source" is generated. You can change the name of any node or model using the Property Inspector.
- Only those tables and views that are in the user's schema are displayed by default. Next, you add objects to the list from other schemas to which odmuser has been given access.

- E. In Step 1 of the wizard, perform the following:

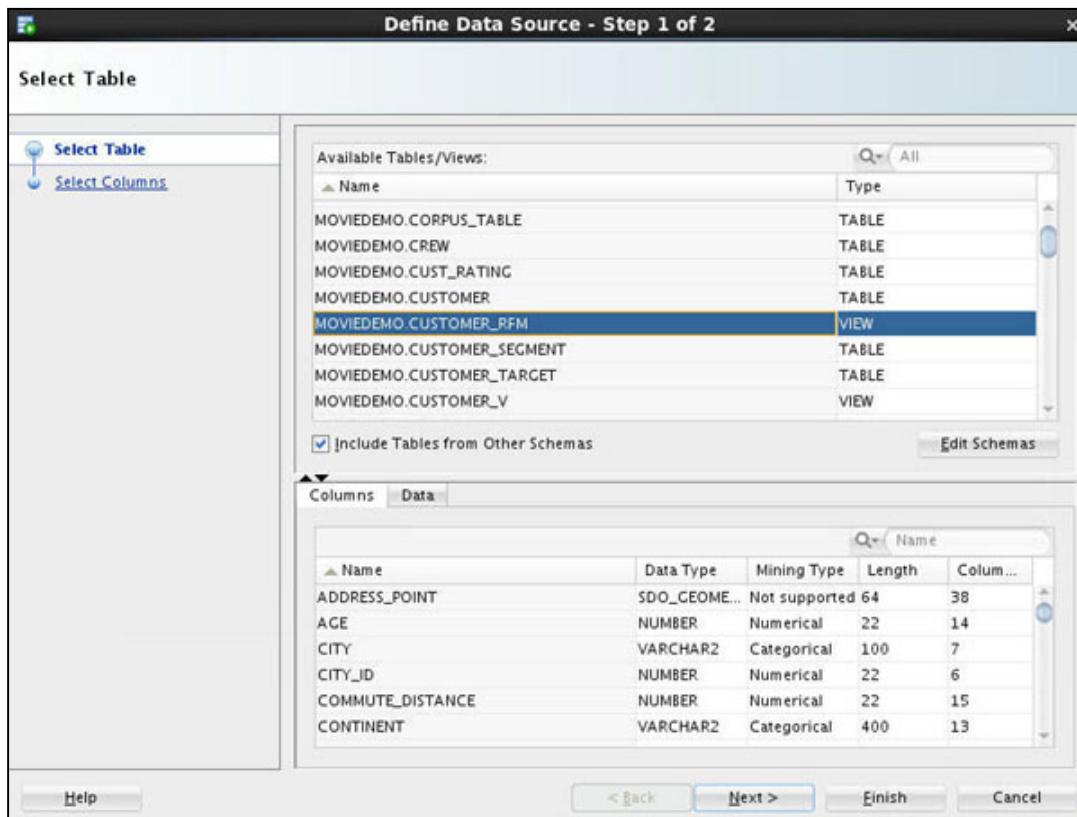
- Click the **Add Schemas** button below the Available Tables/Views box.



- In the Edit Schema List dialog, select **MOVIEDEMO** in the Available list and move it to the Selected list, as shown here below. Then click **OK**.

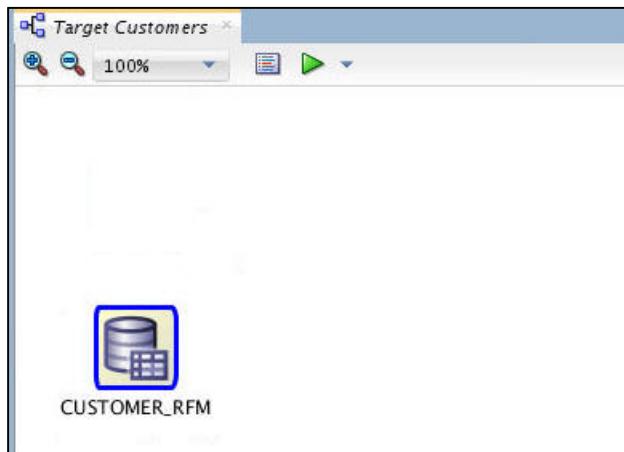


- Now, select **MOVIEDEMO.CUSTOMER\_RFM** from the Available Tables/Views list, as shown below, and then click **Next**.



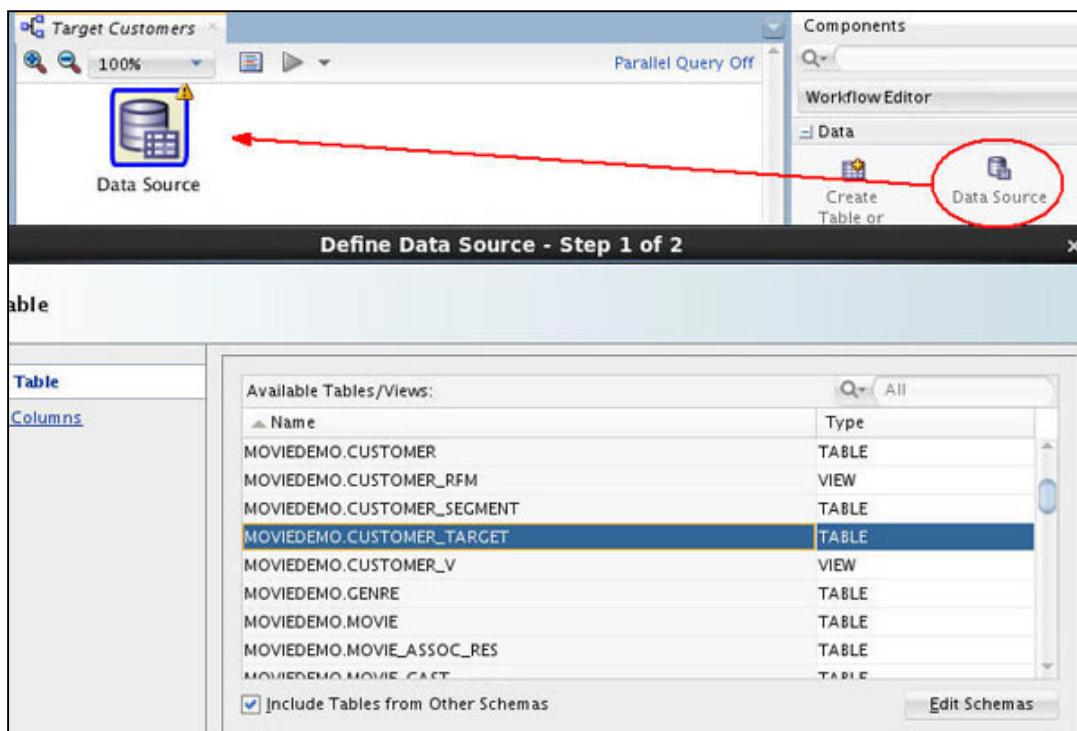
- F. In Step 2 of the wizard, click **Finish**.

**Result:** The data source node takes the name of the selected table, and the properties associated with the node are displayed in the Properties tab.



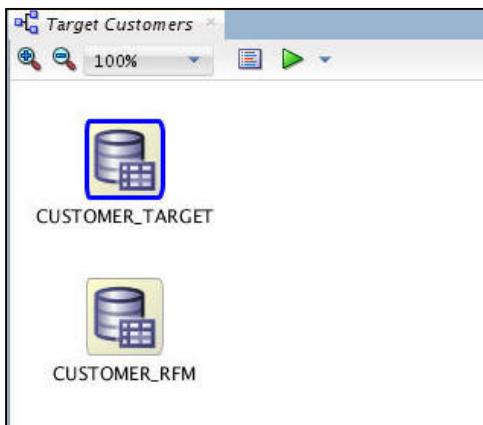
- G. Next, add a second data source to the workflow using the same techniques:

- Drag and drop a Data Source node onto the workflow pane, just above the CUSTOMER.RFM node.
- In Step 1 of the wizard, select **MOVIEDEMO.CUSTOMER\_TARGET** from the list.



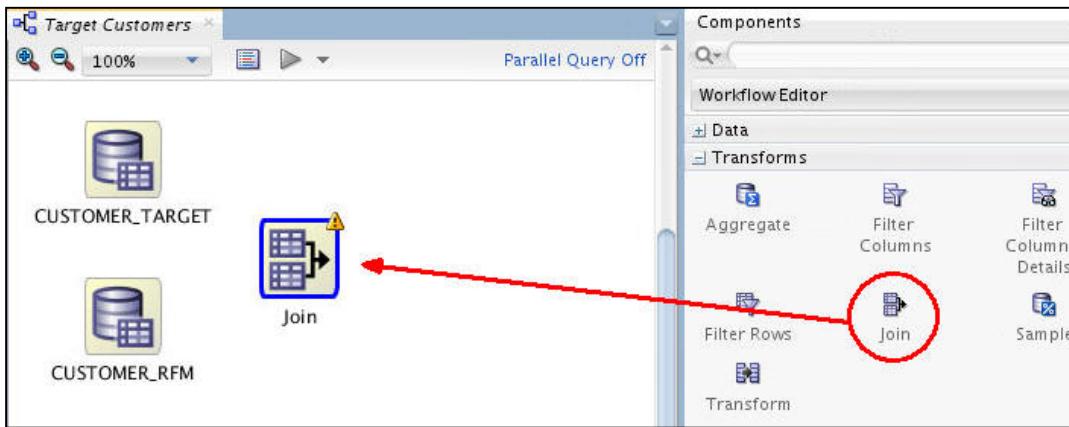
- Then click **Finish**.

**Result:** Both data source nodes are displayed in the workflow pane.



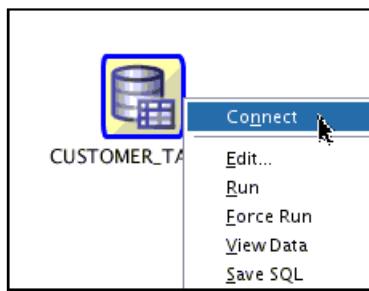
6. Now, perform the following steps to join the tables:

- In the Components tab, collapse the Data category and drill on the **Transforms** category.
- Drag and drop a Join node onto the workflow pane, as shown here:



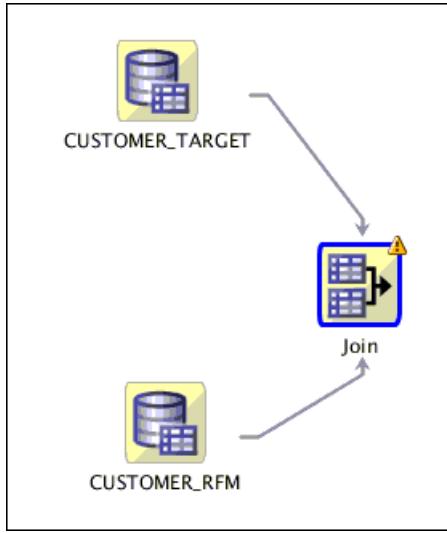
- Next, connect the two data source nodes to the Join node:

- Right-click the CUSTOMER\_TARGET node and select **Connect** from the menu.



- Then drag the pointer to the Join node and click again to connect the two nodes.
- Connect the CUSTOMER\_RFIM node to the Join node in the same way.

**Result:** the workflow should look like this:

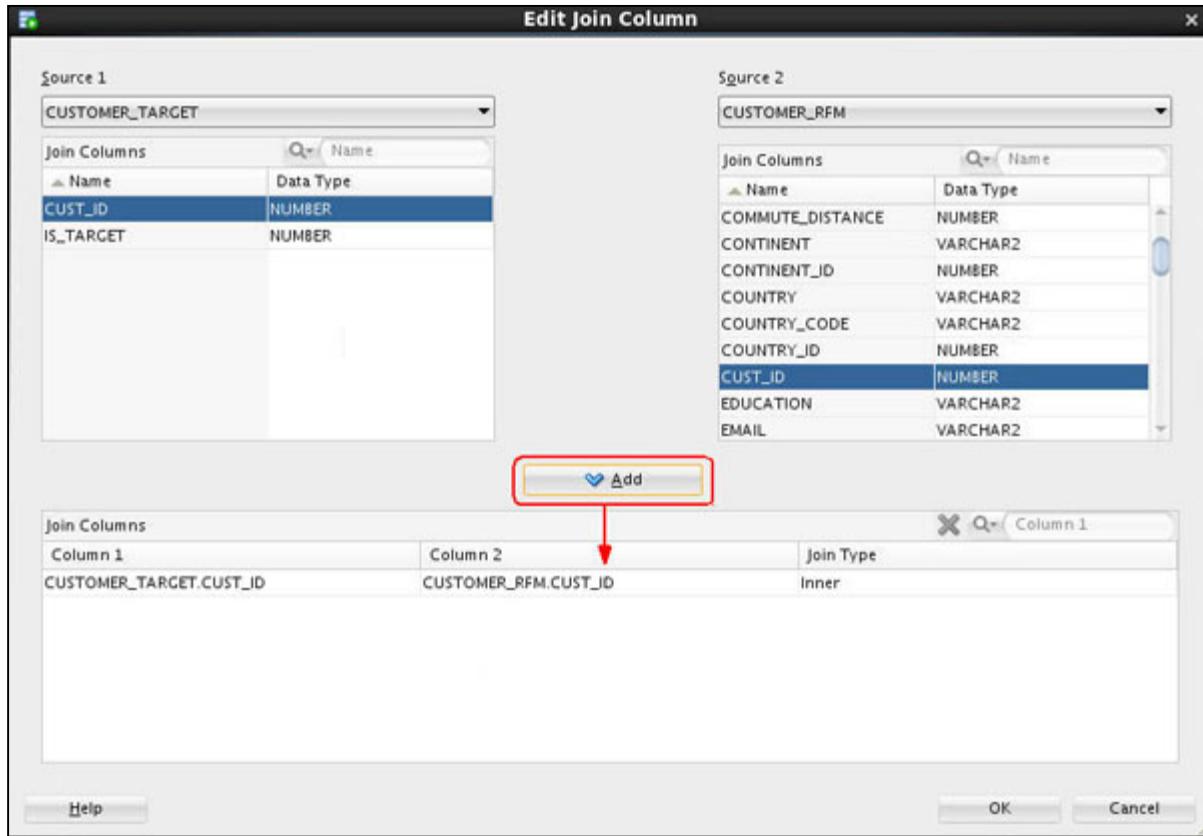


- D. Finally, define the join definition by performing the following:
1. Double-click the Join node to display the Edit Join Node window. The Join tab is displayed by default.
  2. Click the Add tool (green "+" icon) to open the Edit Join Column window.

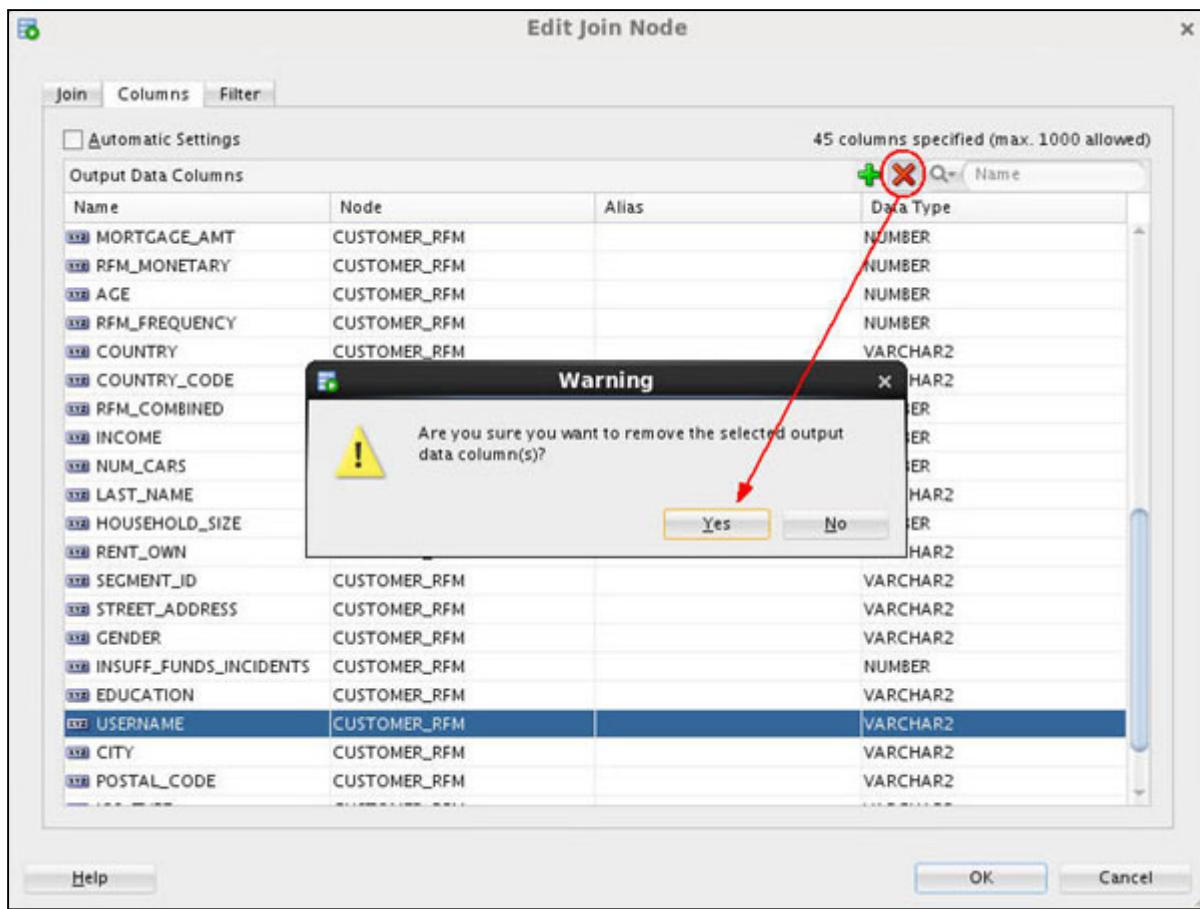


3. In the Edit Join Column window:
  - Select **CUSTOMER\_TARGET** as Source 1 and **CUSTOMER\_RFM** as Source 2.
  - Select the **CUST\_ID** column from both sources.
  - Then, click the **Add** button to define the Join Columns.

**Result:** The Edit Join Column window should now look like this:



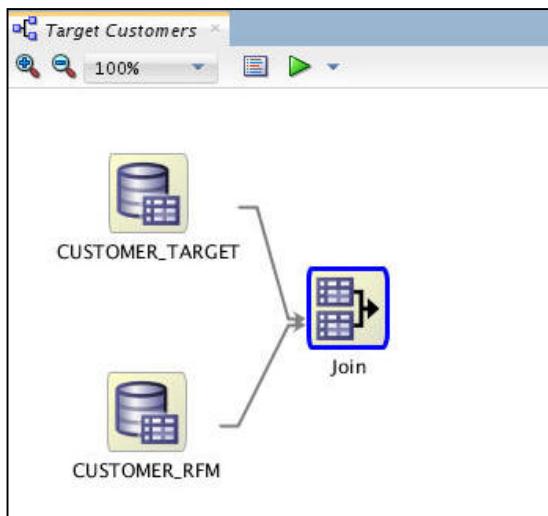
- Click **OK** in the Edit Join Column window to save the join definition.
4. Next, select the **Columns** tab of the Edit Join Node window and perform the following (as shown in the screenshot below):
  - Deselect the **Automatic Settings** option (top left side of Columns tab).
  - Scroll down to the bottom of the Output Data Columns list, and select the **USERNAME** column from the **CUSTOMER\_RFM** node.
  - Click the **Remove** tool (red "x" icon).
  - Click **Yes** in the Warning window.



**Note:** Now, the total number of specified columns should be 44.

- Finally, click **OK** in the Edit Join Node window to display the workflow pane.

**Result:** The workflow should now look like something like this. Notice that the warning indicator is no longer displayed on the Join node.



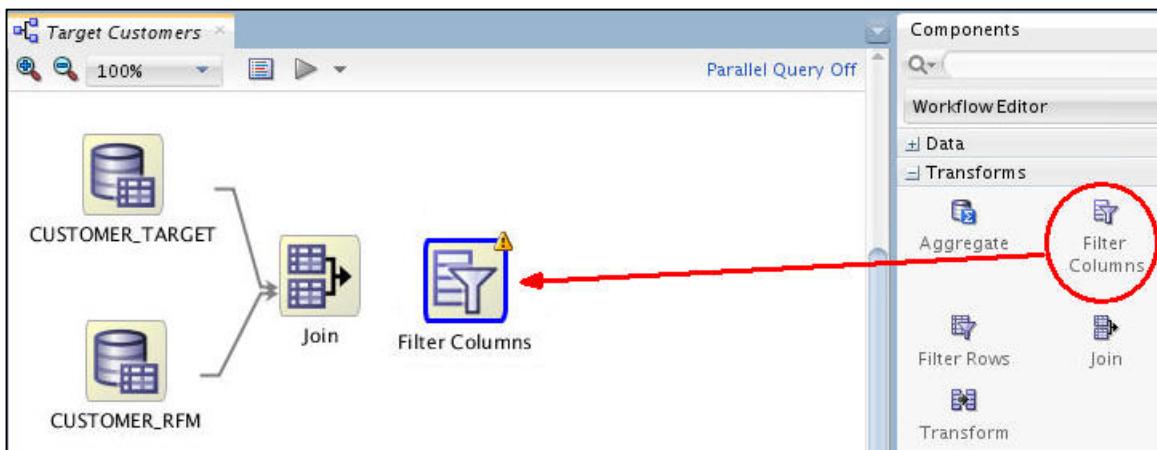
Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

**Notes for the creation of Classification models in the workflow:**

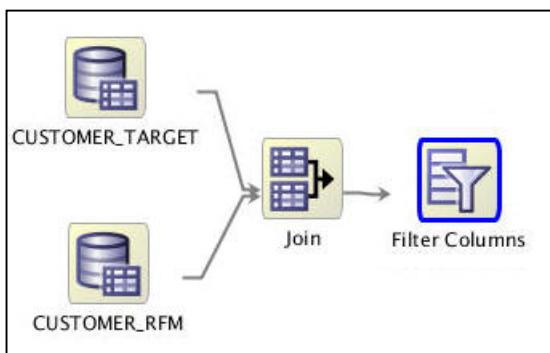
- As stated previously, you want to predict which customers are most likely to be the best customers in terms of sales. Since you want to predict individual behavior, a classification model is selected.
- When using Oracle Data Miner, a classification model node creates four models using different algorithms. This default behavior makes it easier to figure out which algorithm gives the best predictions.
- In combination with Model node, a Filter Columns node may be used to specify additional instructions for attribute importance for the Target variable. The filter node can provide suggestions on which variables to include/exclude based on data quality filters, and the attribute importance of each input variable. This extra, optional step theoretically reduces “noisy” less relevant input variables.

7. First add and define a Filter Columns node.

- A. Using the Transforms category in the Components tab drag and drop a Filter Columns node to the Workflow pane, like this:



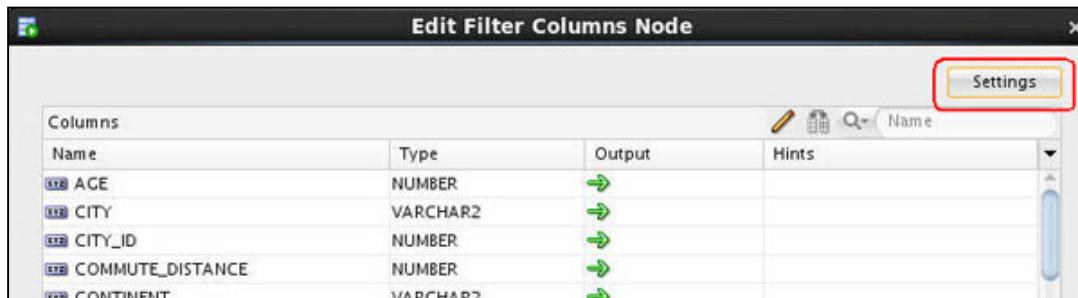
- B. Connect the Join node to the Filter node using the same technique you learned previously. The result should look like this:



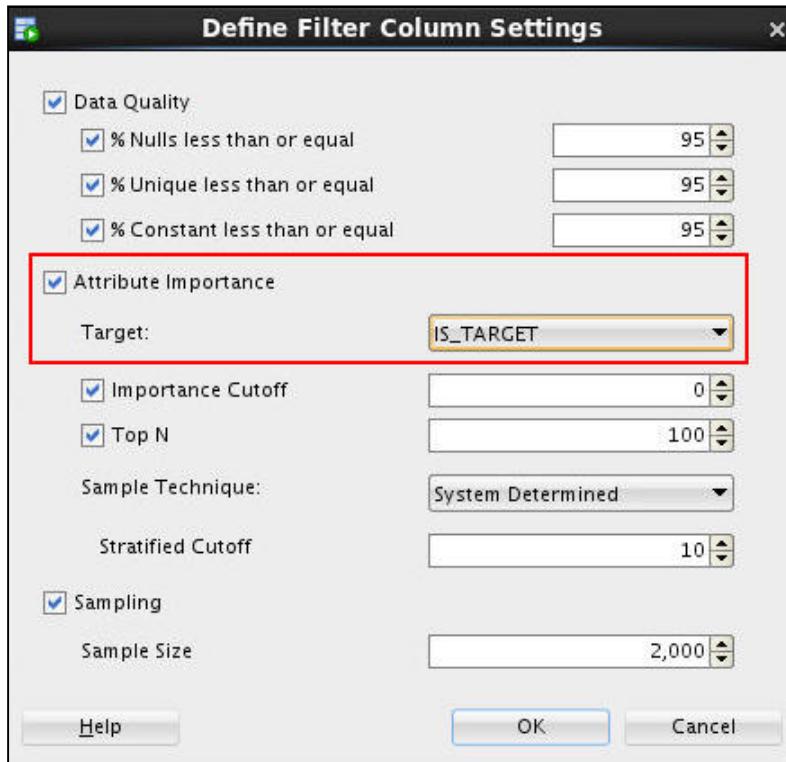
C.

Next, specify attribute importance for the Target field in the Filter Columns node.

1. Double-click the Filter Columns node, and then click the **Settings** button in the top-right corner of the Edit Filter Columns Node window, as shown here:

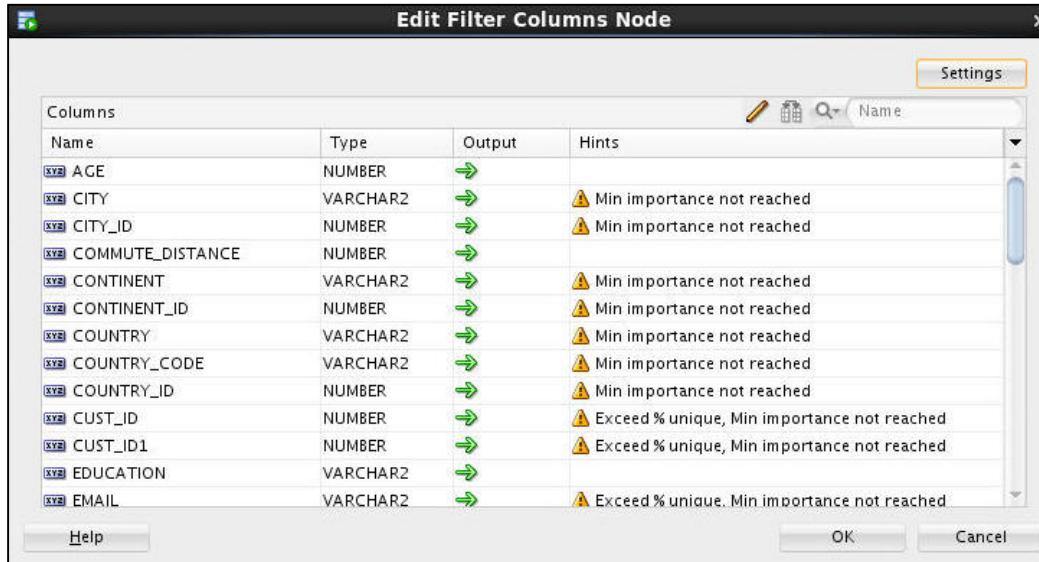


2. In the Define Filter Column Settings window, enable the **Attribute Importance** option, and then select **IS\_TARGET** from the Target drop-down list, as shown here (accept all of the other option defaults):



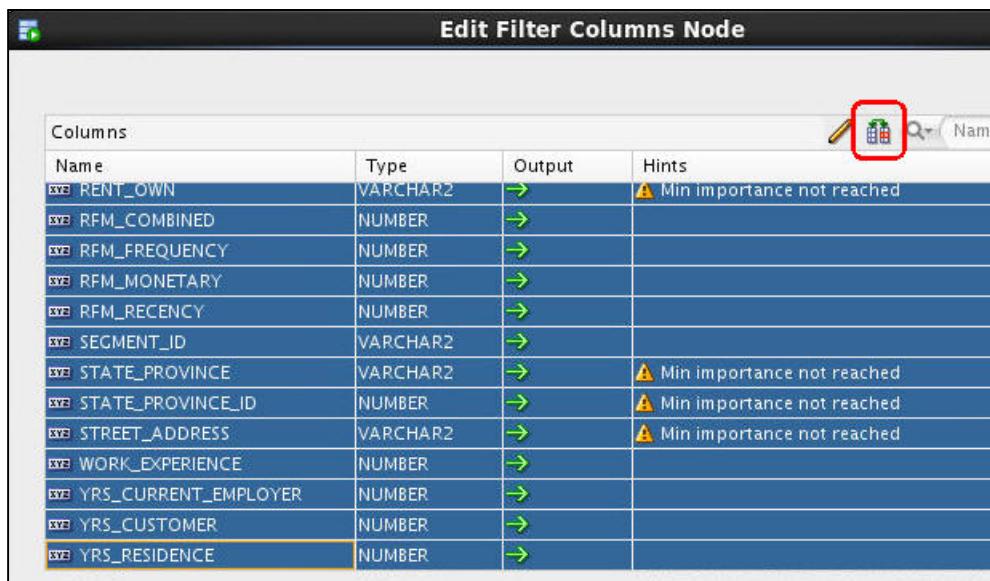
3. Click **OK**.
4. Finally, click **OK** in the Edit Filter Columns Node window to save the Attribute Importance specification.

- D. Run the Filter Columns node to see the results of the Attribute Importance specification.
1. Right-click the Filter Columns node and select **Run** from the menu.
  2. When the Run process is complete (green check mark in node border), double-click the Filter Columns node to view the Attribute Importance recommendations.



**Note:**

- You can ignore the recommendations, implement the recommendations for all columns, or selectively choose to implement some of the recommendations.
  - In this case, you will implement all recommendations except for the CUST\_ID column.
3. Select all of the rows in the list. Then, click the **Apply Recommended Output Settings** tool, as shown here:



**Result:** All of the recommended columns are filtered out (a red "x" is added to the Output designation for each column), as shown here:

Columns			
Name	Type	Output	Hints
xyz_RENT_OWNER	VARCHAR2	→x	⚠ Min importance not reached
xyz_RF_M_Combined	NUMBER	→	
xyz_RF_M_Frequency	NUMBER	→	
xyz_RF_M_Monetary	NUMBER	→	
xyz_RF_M_Recency	NUMBER	→	
xyz_SEGMENT_ID	VARCHAR2	→	
xyz_STATE_Province	VARCHAR2	→x	⚠ Min importance not reached
xyz_STATE_Province_ID	NUMBER	→x	⚠ Min importance not reached
xyz_STREET_Address	VARCHAR2	→x	⚠ Min importance not reached
xyz_WORK_Experience	NUMBER	→	
xyz_YRS_Current_Employer	NUMBER	→	
xyz_YRS_Customer	NUMBER	→	
xyz_YRS_Residence	NUMBER	→	

4. Now, click the Output designation for the **CUST\_ID** column, so that it will be passed forward as a column in the modeling process. (The red "x" disappears.) The Filter Columns list should now look like this:

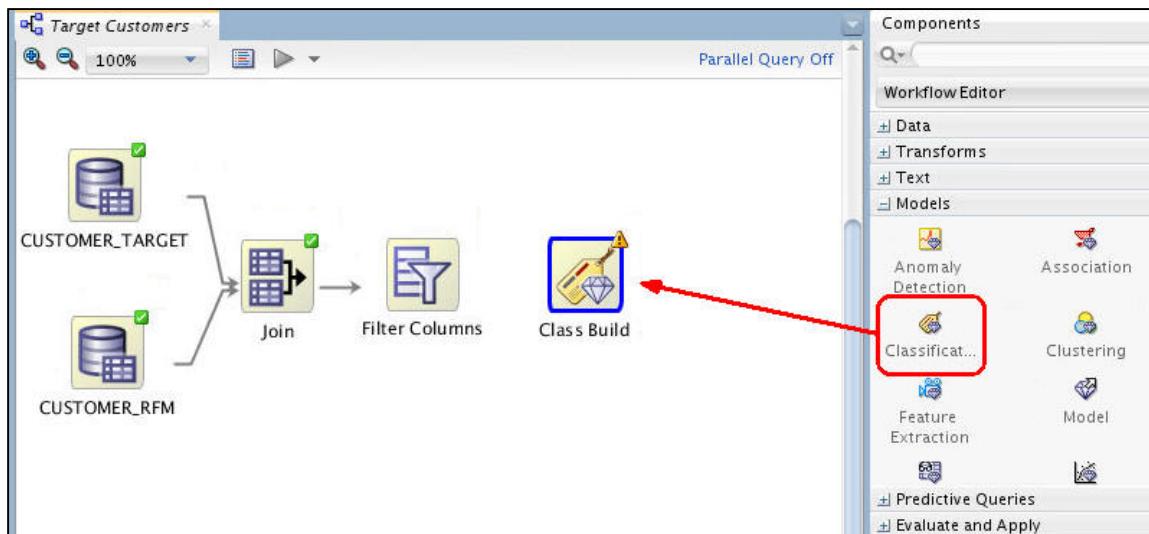
Edit Filter Columns Node			
Columns			
Name	Type	Output	Hints
xyz_CONTINENT_ID	NUMBER	→x	⚠ Min importance not reached
xyz_COUNTRY	VARCHAR2	→x	⚠ Min importance not reached
xyz_COUNTRY_CODE	VARCHAR2	→x	⚠ Min importance not reached
xyz_COUNTRY_ID	NUMBER	→x	⚠ Min importance not reached
xyz_CUST_ID	NUMBER	→	⚠ Exceed % unique, Min importance not reached
xyz_CUST_ID1	NUMBER	→x	⚠ Exceed % unique, Min importance not reached
xyz_EDUCATION	VARCHAR2	→	
xyz_EMAIL	VARCHAR2	→x	⚠ Exceed % unique, Min importance not reached
xyz_FIRST_NAME	VARCHAR2	→x	⚠ Min importance not reached
xyz_FULL_TIME	VARCHAR2	→x	⚠ Min importance not reached
xyz_GENDER	VARCHAR2	→x	⚠ Min importance not reached
xyz_HOUSEHOLD_SIZE	NUMBER	→	
xyz_INCOME	NUMBER	→x	⚠ Exceed % unique, Min importance not reached

5. Click **OK** to close the Edit Filter Columns Node window.

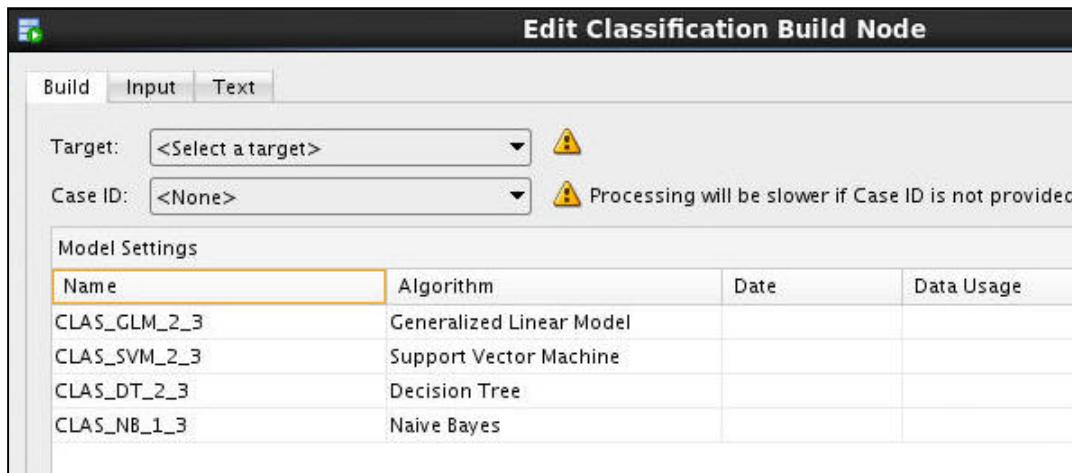
8. Next, add and define the Classification node.

A. First, collapse the Transforms category, and expand the **Models** category in the Components tab.

B. Then, drag and drop the **Classification** node to the Workflow pane like this:



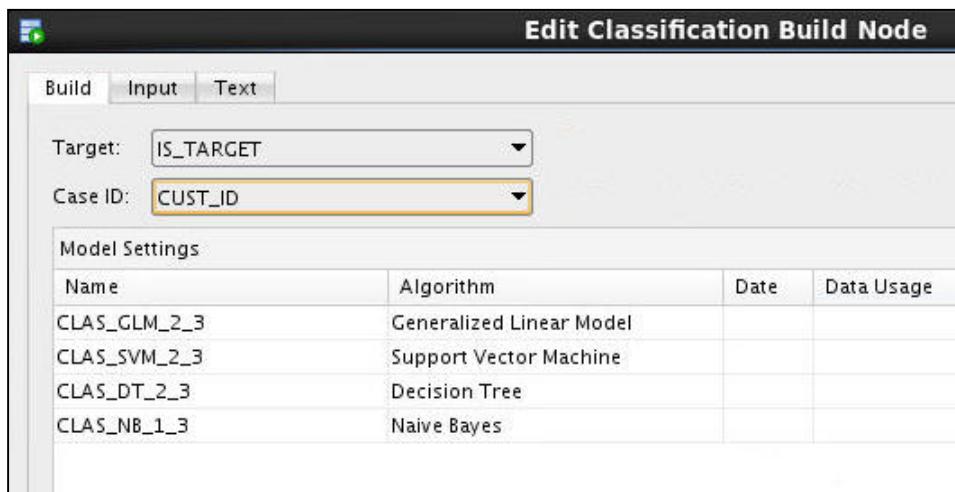
C. First, connect the Filter Columns node to the Class Build node using the same technique described previously. Result: the Edit Classification Build Node window automatically appears, as shown here.



#### Note:

- A yellow "!" indicator is displayed next to the Target and Case ID fields. This means that an attribute should be selected for these items.
- The names for each model are automatically generated, and yours may differ slightly from those in this example.

- D. In the Build tab, select **IS\_TARGET** as the Target attribute, and **CUST\_ID** as the Case ID attribute.



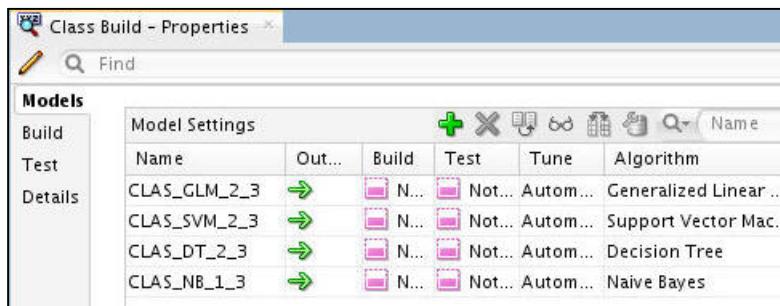
**Note:**

- A Case ID is used to uniquely define each record. This helps with model repeatability and is consistent with good data mining practices.
- As stated previously, all four algorithms for Classification modeling are selected by default. They will be automatically run unless you specify otherwise.

- E. Click **OK** in the Edit Classification Build Node window to save your changes.

Result: The classification build node is ready to run.

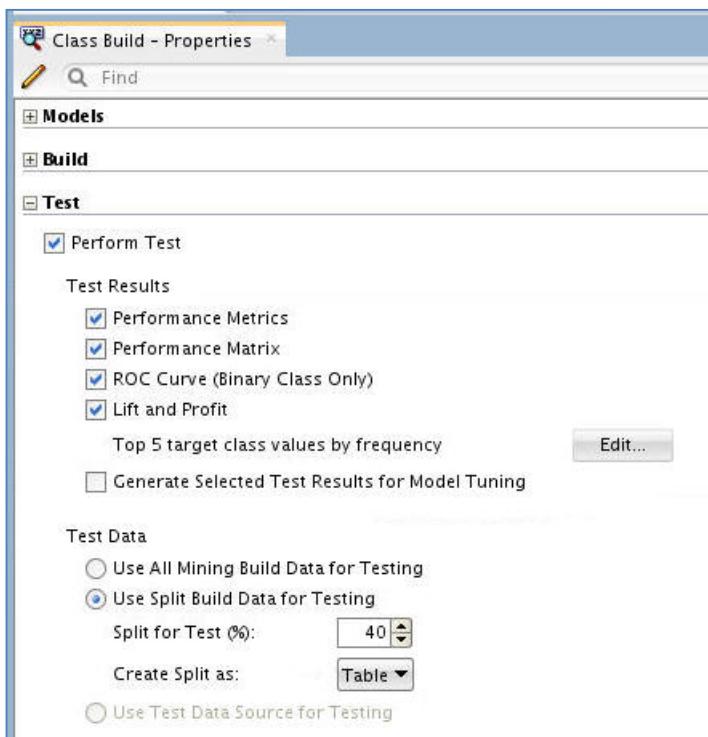
**Note:** In the Models section of the Properties tab, you can see the current status for each of the selected default algorithms, as shown below



## Notes for building the models in the workflow:

- Next, you build the selected models against the source data. With classification models, this operation is also called “training”, and the model is said to “learn” from the training data.
- A common data mining practice is to build (or train) your model against part of the source data, and then to test the model against the remaining portion of your data. By default, Oracle Data Miner uses this approach, at a 40/60 split.
- Before building the models, select Class Build node and expand the Test section of the Properties tab. In the Test section, you can specify:
  - Whether or not to perform a test during the build process
  - Which test results to generate
  - How you want the test data managed

The default settings for the test phase are shown here:

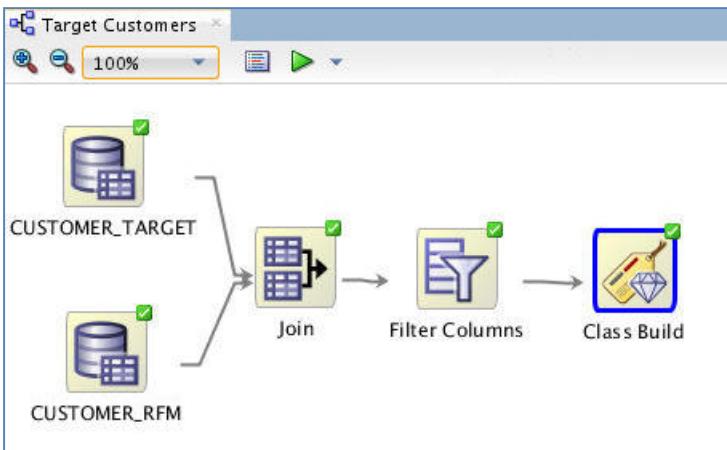


9. Next, you build the four classification models.

- A. Right-click the Class Build node and select **Run** from the pop-up menu.

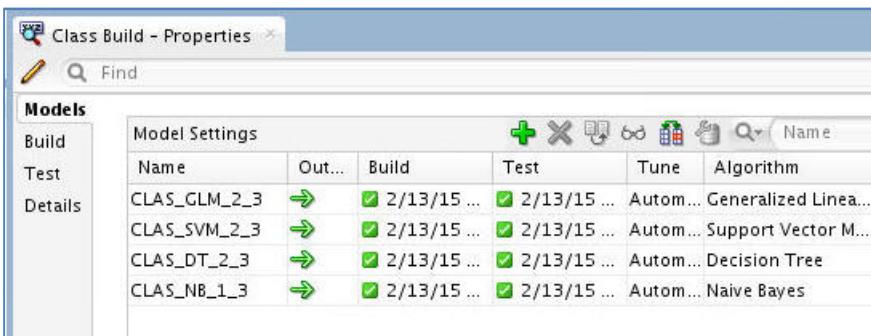
**Note:**

- When the node runs it builds and tests all of the models that are defined in the node.
- As before, a green gear icon appears on the node borders to indicate a server process is running, and the status is shown at the top of the workflow window.
- When the build is complete, all nodes contain a green check mark in the node border, as shown below.



**Note:** In addition, you can view several pieces of information about the build using the Properties Inspector.

- B. Select the Class Build node in the workflow, and then select the Models section in the Properties tab. (Expand the size of the Properties tab and move it if desired.)



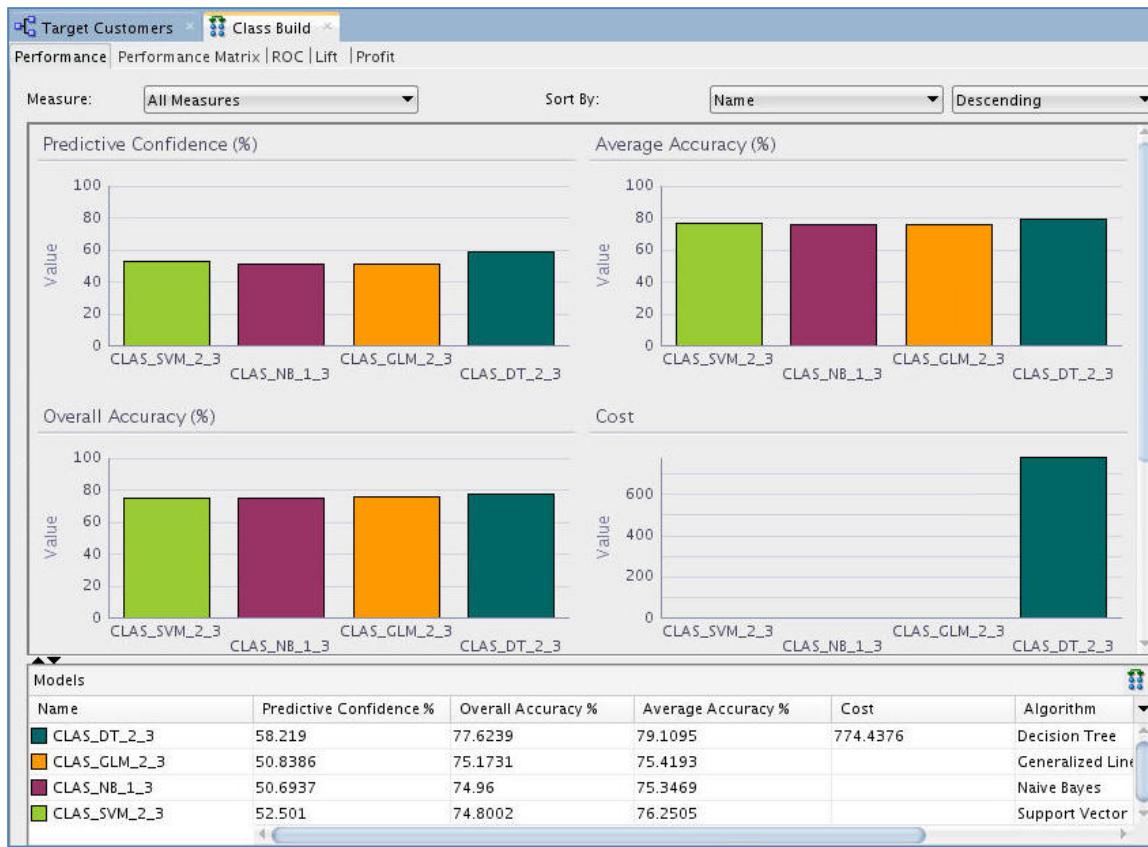
### Note:

- All four models have been successfully built.
  - The models all have the same target (IS\_TARGET) but use different algorithms.
  - The source data is automatically divided into test data and build data.

#### 10. Compare the Models.

**Note:** After you build/train the selected models, you can view and evaluate the results for all of the models in a comparative format. Here, you compare the relative results of all four classification models. Follow these steps:

- A. Right-click the Class Build node and select **Compare Test Results** from the menu.  
**Note:** As shown below, a Class Build display tab opens with a graphical comparison of the four models in the Performance tab.

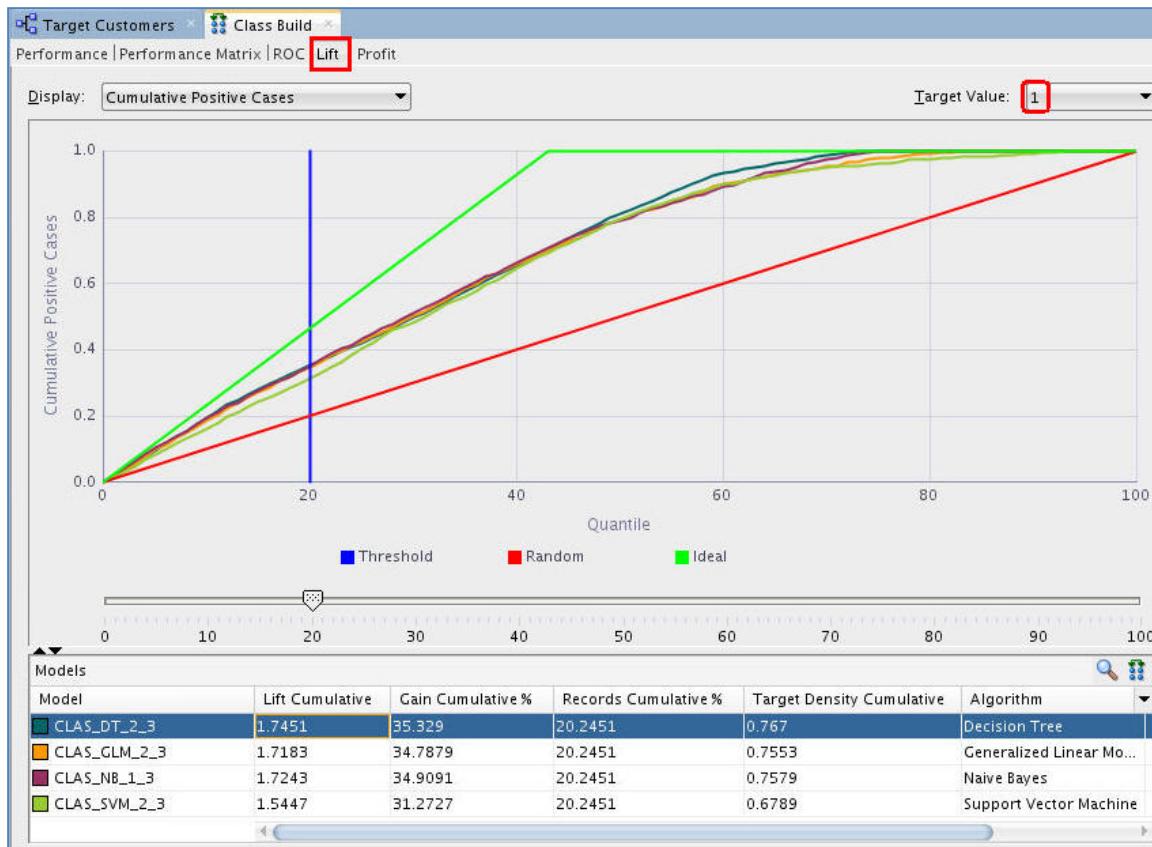


### Note:

Since the sample data set is very small, the numbers you get may differ slightly from those shown in this example. In addition, the histogram colors that you see may be different than those shown in this example.

- The comparison results include five tabs: Performance, Performance Matrix, ROC, Lift, and Profit.
- The Performance tab provides numeric and graphical information for each model on Predictive Confidence, Average Accuracy, and Overall Accuracy.
- In the example, the Performance tab seems to indicate that the Decision Tree (DT) model is providing the highest predictive confidence, overall accuracy %, and average accuracy %. The other models show mixed results.

- B. Now, select the **Lift** tab. Then, select a Target Value of **1 (yes)**, in the upper-right side of the graph.



### Note:

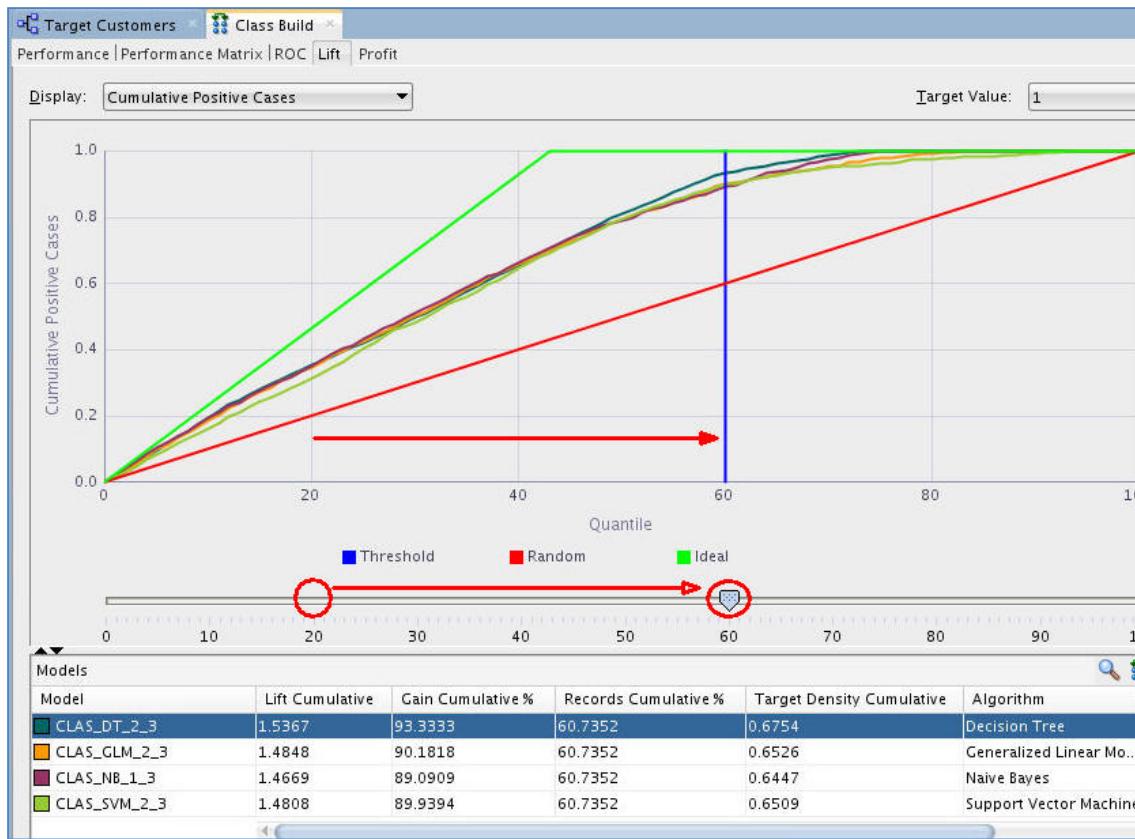
- The Lift tab provides a graphical presentation showing lift for each model, a red line for the random model, and a vertical blue line for threshold.
- Lift is a different type of model test. It is a measure of how “fast” the model finds the actual positive target values.
- The Lift viewer compares lift results for the given target value in each model.
- The Lift viewer displays Cumulative Positive Cases and Cumulative Lift.

Using the example shown above at the 20th quantile, the DT model has the highest Lift Cumulative and Gain Cumulative %, although all of the models are very close in terms of these measurements.

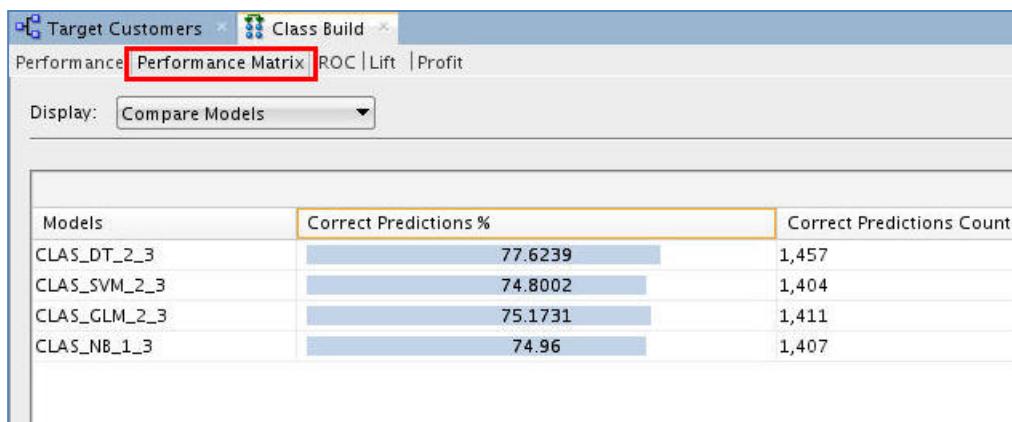
- C. In the Lift tab, you can move the Quantile measure point line along the X axis of the graph by using the slider tool, as shown below. The data in the Models pane at the bottom updates automatically as you move the slider left or right.

Perform the following, as shown in the image below:

- As you move up the quantile range, the Lift Cumulative and Gain Cumulative % of the DT model separates a bit from the other models.
- As you move to the 60th quantile, the DT model shows the largest separation in terms of increased Lift and Gain.



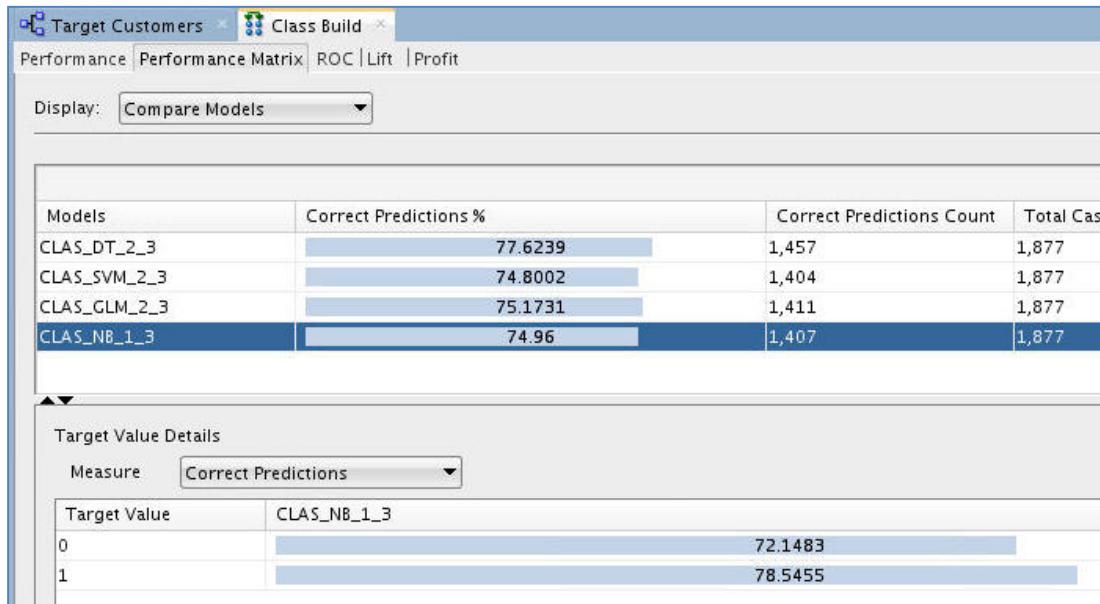
D. Next, select the **Performance Matrix** tab.



**Note:** The Performance Matrix shows that the DT model has the highest Correct Predictions percentage, at 77.6%. The NB model is next at 74.8%.

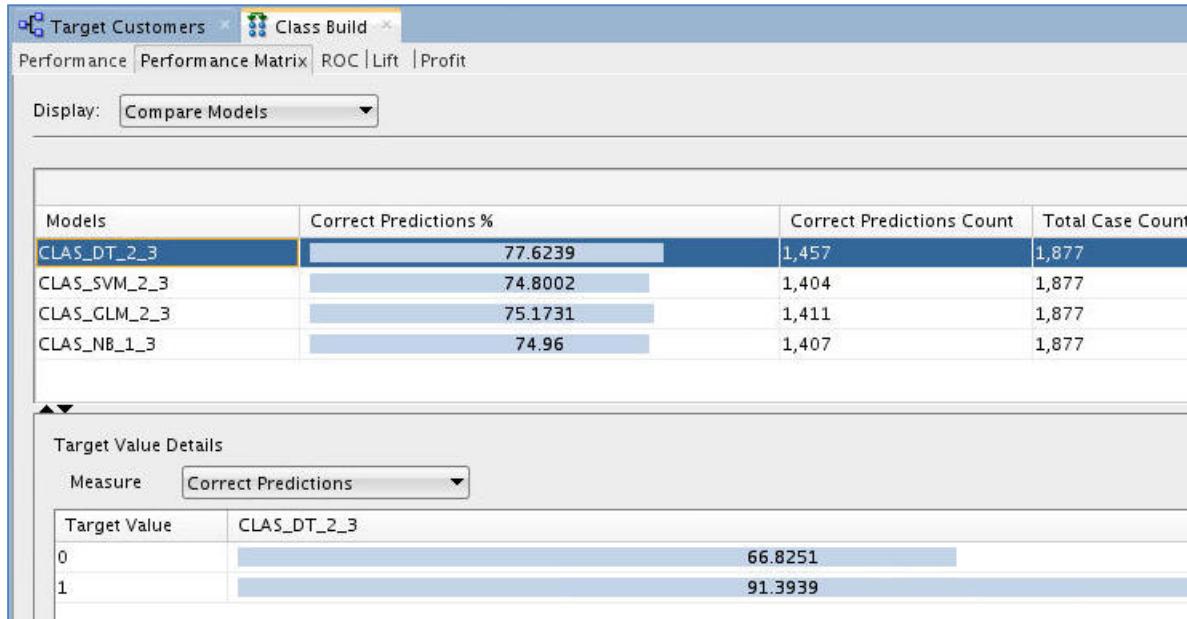
E. You can also compare the details for models in this tab.

- First, select the **NB** model to view its Target Value Details in the lower pane. Recall that the "Target Value" for each of the models is the IS\_TARGET attribute.



**Note:** The NB model indicates a 72% correct prediction outcome for customers that aren't considered targets and a 78.5% correct prediction outcome for customers that are considered targets.

- Next, select the DT model.



**Note:** The DT model indicates a 66.8% correct prediction outcome for non-target customers and a 91.4% correct prediction outcome for target customers. After considering the initial analysis, you decide to investigate the DT model more closely.

F. Dismiss the **Class Build** tabbed window.

## 11. Select and Examine a Specific Model.

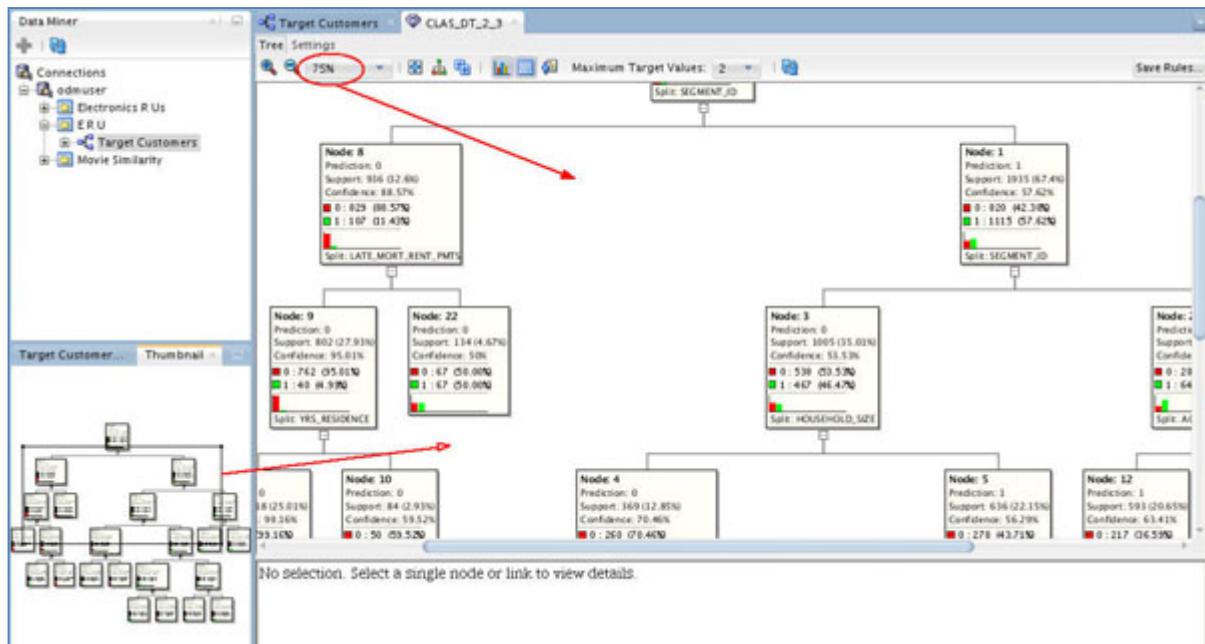
Using the comparative analysis performed in the previous step, the Decision Tree model is selected for further analysis .Follow these steps to examine the Decision Tree model:

- A. Back in the workflow pane, right-click the Class Build node again, and select **View Models > CLAS\_DT\_2\_3** (Note: The name of your DT model may be different).

Result: A window opens that displays a graphical presentation of the Decision Tree. The interface provides several methods of viewing navigation:

- The Thumbnail tab (the lower-left) provides a high level view of the entire tree. The Thumbnail tab shows that this tree contains seven levels, although you view fewer of the nodes in the primary display window. By showing the entire tree, the Thumbnail tab also illustrates the particular branches that lead to the terminal node.
- You can move the viewer box around within the Thumbnail tab to dynamically locate your view in the primary window. You can also use the scroll bars in the primary display window to select a different location within the decision tree display.
- Finally, you can change the viewer percentage zoom in the primary viewer window to increase or decrease the size of viewable content.

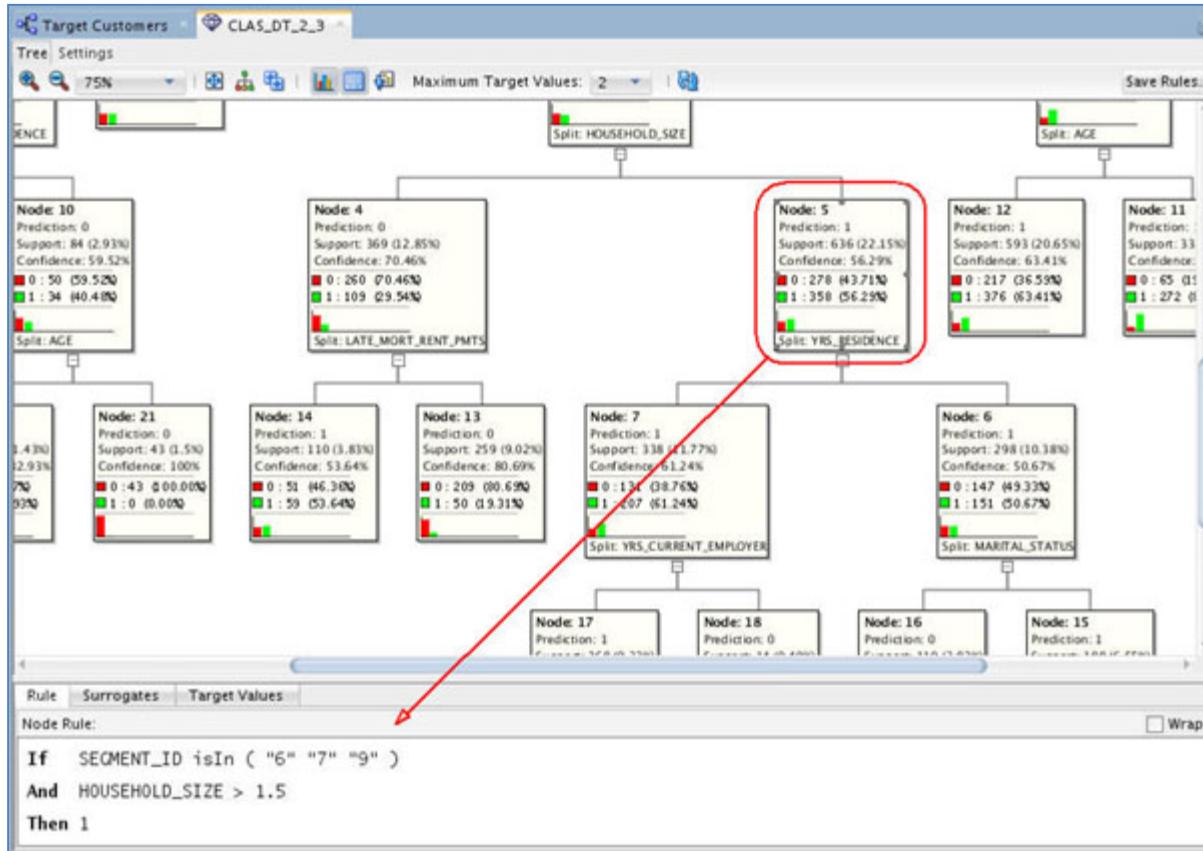
For example, set the primary viewer window for the decision tree to 75% zoom.



B. First, navigate to and select **Node 5**.

**Note:**

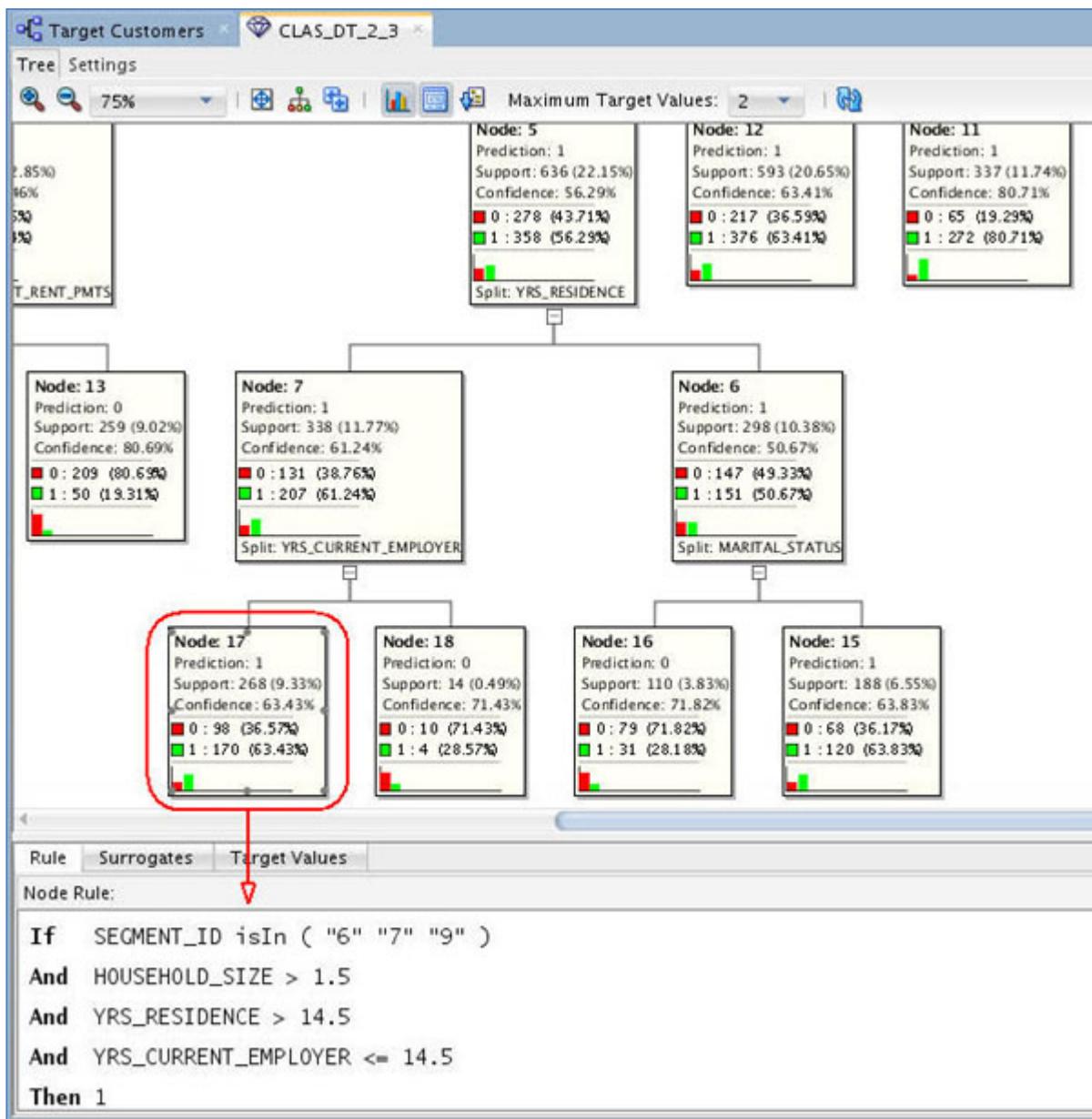
- At each level within the decision tree, an IF/THEN statement that describes a rule is displayed. As each additional level is added to the tree, another condition is added to the IF/THEN statement.
- For each node in the tree, summary information about the particular node is shown in the box.
- In addition, the IF/THEN statement rule appears in the Rule tab, as shown below, when you select a particular node.



**Note:**

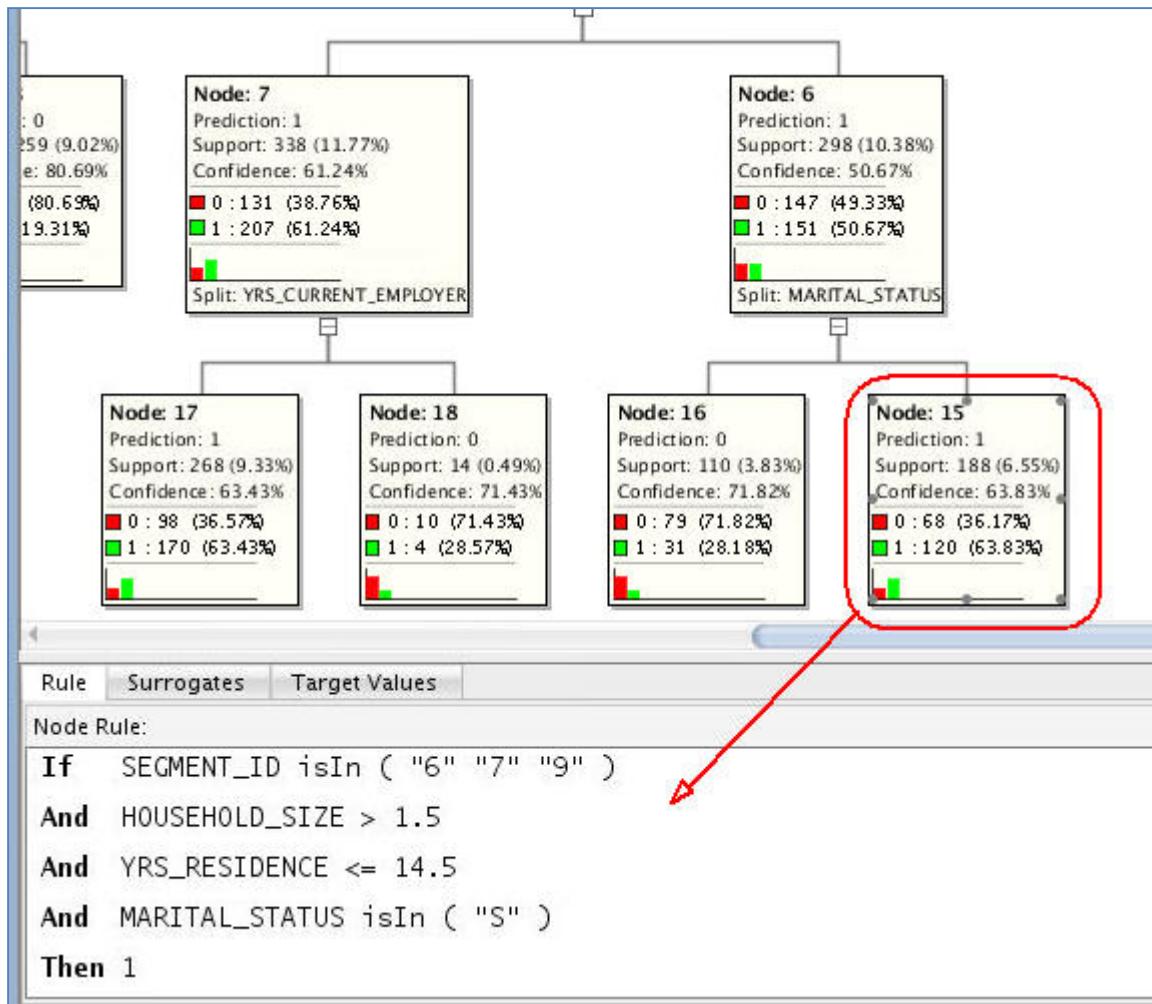
- At this level, we see that the first split is based on the SEGMENT\_ID attribute, and the second split is based on the HOUSEHOLD\_SIZE attribute.
- Node 5 indicates that if SEGMENT\_ID IS 6, 7, OR 9, and HOUSEHOLD\_SIZE is greater than 1.5, then there is a 56% chance that the customer is part of the target group.

C. Next, scroll down to the bottom of the tree. There are two leaf nodes that indicate a prediction of 1 (part of the target group). Select **Node 17**, as shown here:

**Note:**

- At this node, the final splits are added for the YRS\_RESIDENCE and YRS\_CURRENT\_EMPLOYER attributes.
- This node indicates that if YRS\_RESIDENCE is greater than 14.5 and YRS\_CURRENT\_EMPLOYER is less than or equal to 14.5 years, then there is a 63.4% chance that the customer will be in the target group of best customers.

D. Now, select Node 15, as shown here:

**Note:**

- At this node, the final splits are added for the YRS\_RESIDENCE and MARITAL\_STATUS attributes.
- This node indicates that if YRS\_RESIDENCE is less than or equal to 14.5 and MARITAL\_STATUS is "S" (Single), then there is a 63.8% chance that the customer will be in the target group of best customers.

E. Dismiss the Decision Tree display tab (CLAS\_DT\_2\_3).

## 12. Apply the model and generate predictive results.

In this next step, you apply the Decision Tree model and then create a table to display the results. When using Oracle Data Miner, you "apply" a model in order to make predictions - in this case to predict which customers are more likely to buy.

To apply a model, you perform the following steps:

- First, specify the desired model (or models) in the Class Build node.
- Second, add a new Data Source node to the workflow. (This node will serve as the "Apply" data.)
- Third, an Apply node to the workflow.
- Next, connect both the Class Build node and the new Data Source node to the Apply node.
- Finally, you run the Apply node to create predictive results from the model.

Follow these steps to apply the model and display the results:

- A. In the workflow, select the Class Build node. Then, using the Models section of the Properties tab, deselect all of the models except for the DT model.

To deselect a model, click the large green arrow in the model's **Output** column. This action adds a small red "x" to the column, indicating that the model will not be used in the next build.

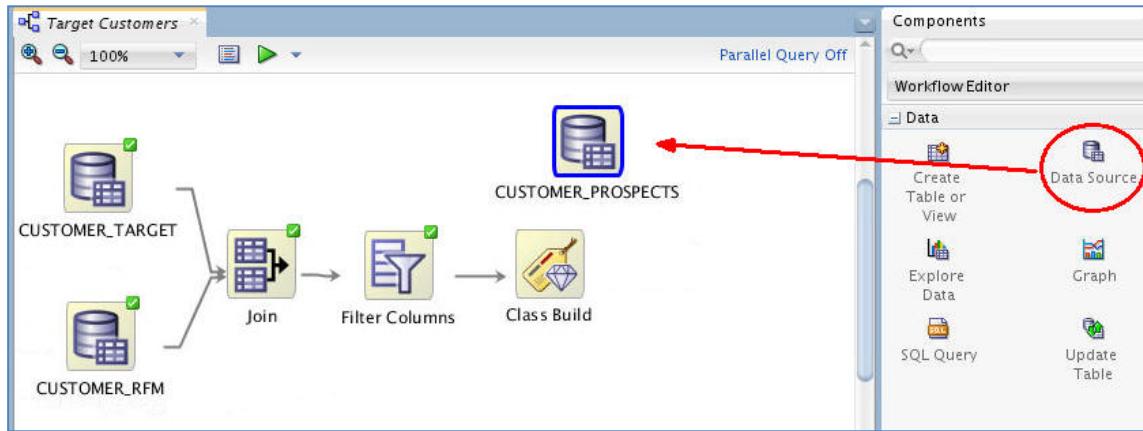
When you finish, the Models tab of the Property Inspector should look like this:

Model Settings						
	Name	Out...	Build	Test	Tune	Algorithm
Build	CLAS_GLM_2_3	✖️	✓	✓	Autom...	Generalized Linea...
Test	CLAS_SVM_2_3	✖️	✓	✓	Autom...	Support Vector M...
Details	CLAS_DT_2_3	➡️	✓	✓	Autom...	Decision Tree
	CLAS_NB_1_3	✖️	✓	✓	Autom...	Naive Bayes

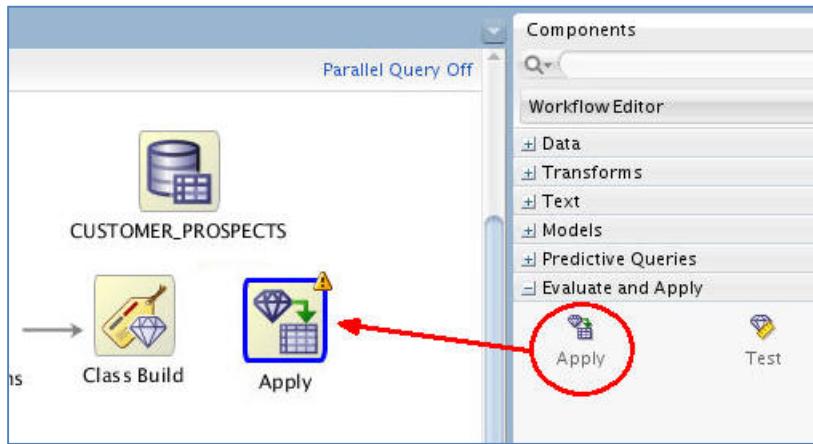
**Note:** Now, only the DT model will be passed to subsequent nodes.

B. Next, add a new Data Source node in the workflow by performing the following:

- From the Data category in the Components tab, drag and drop a Data Source node to the workflow canvas, as shown below. The Define Data Source wizard opens automatically.
- In Step 1 of the wizard, select **ODMUSER.CUSTOMER\_PROSPECTS** from the list and click **Finish**. The workflow should now look like this:

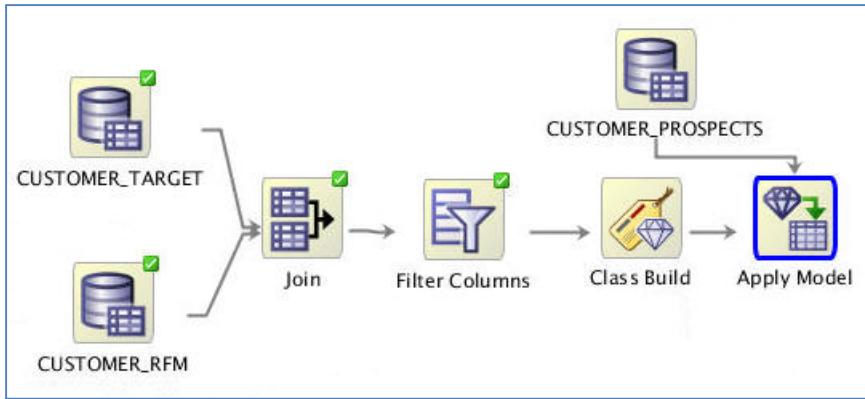


C. Next, expand the **Evaluate and Apply** category in the Components tab, and drag/drop the Apply node on the workflow canvas, like this:



D. Click the name of the Apply node and rename it to **Apply Model**.

E. Using the techniques described previously, connect the Class Build node to the Apply Model node, and then the CUSTOMER\_PROSPECTS node to the Apply Model node, like this:

**Note:**

Before you run the Apply Model node, consider the resulting output. By default, an apply node creates two columns of information for each customer:

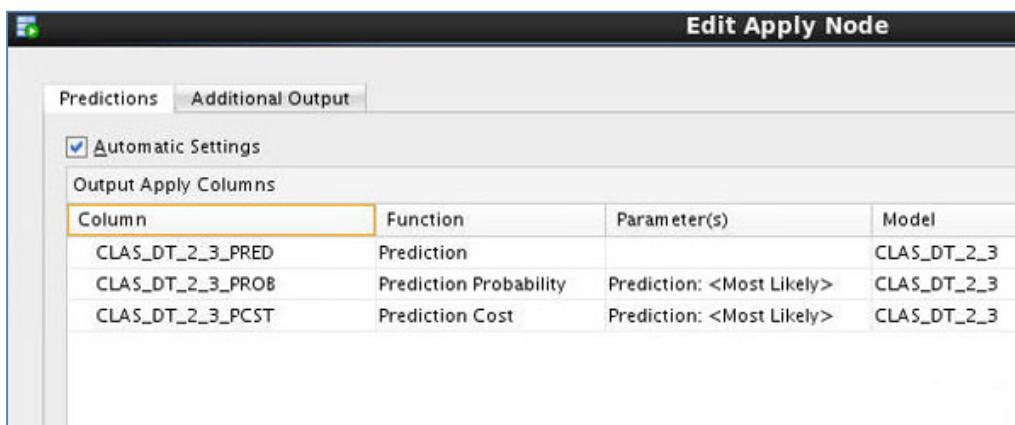
- The prediction: 1 or 0 (Yes or No)
- The probability of the prediction

However, you really want to know this information for each customer, so that you can readily associate the predictive information with a given customer. To get this information, you need to add a third column to the apply output: CUST\_ID.

F. Follow these instructions to add the customer identifier to the output:

- Right-click the Apply Model node and select **Edit**.

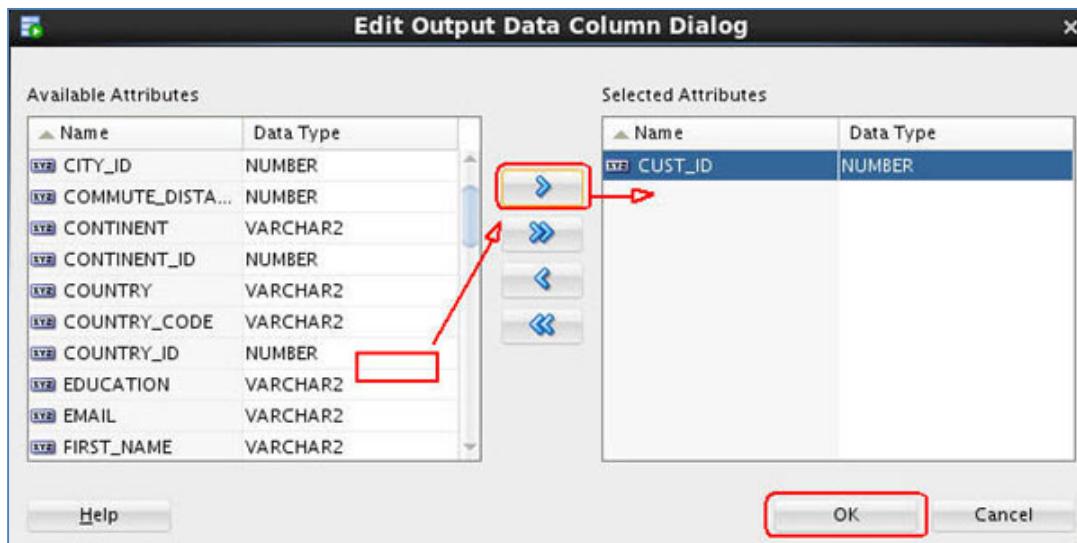
**Result:** The Edit Apply Node window appears. The Prediction, Prediction Probability, and Prediction Cost columns are defined automatically in the Predictions tab.



- Select the **Additional Output** tab, and then click the green "+" sign, like this:



- In the Edit Output Data Column Dialog:
  - Select **CUST\_ID** in the Available Attributes list.
  - Move it to the Selected Attributes list by using the shuttle control.
  - Then, click **OK**.



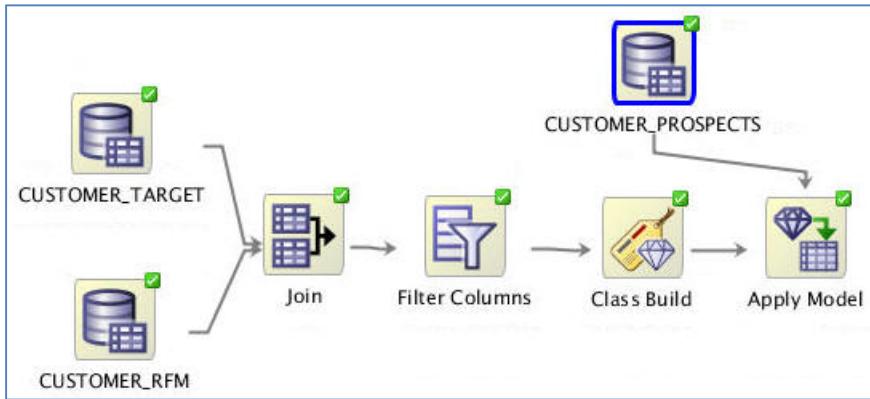
**Result:** the CUST\_ID column is added to the Additional Output tab, as shown here:



- Finally, click **OK** in the Edit Apply Node window to save your changes. Now you are ready to apply the model.

- G. Right-click the Apply Model node and select **Run** from the menu.

**Result:** As before, the workflow document is automatically saved, and small green gear icons appear in each of the nodes that are being processed. When the process is complete, green check mark icons are displayed in the border of all workflow nodes to indicate that the server process completed successfully.

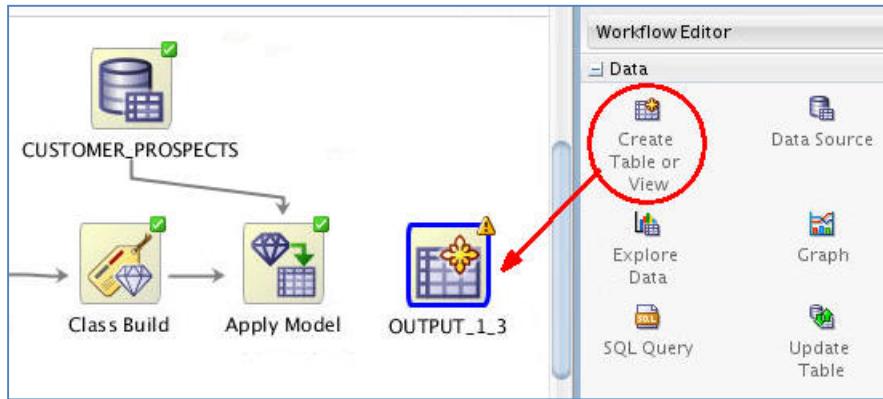


13. Optionally, you can create a database table to store the model prediction results that are defined in the "Apply Model" node.

The table may be used for any number of reasons. For example, an application could read the predictions from that table, and suggest an appropriate response, like sending the customer a letter, offering the customer a discount, or some other appropriate action.

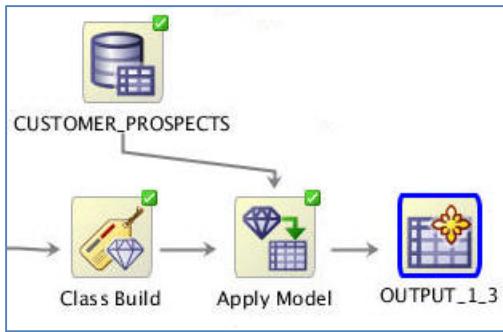
- A. To create a table of model prediction results, perform the following:

- Using the Data category in the Components pane, drag the Create Table or View node to the workflow window, like this:



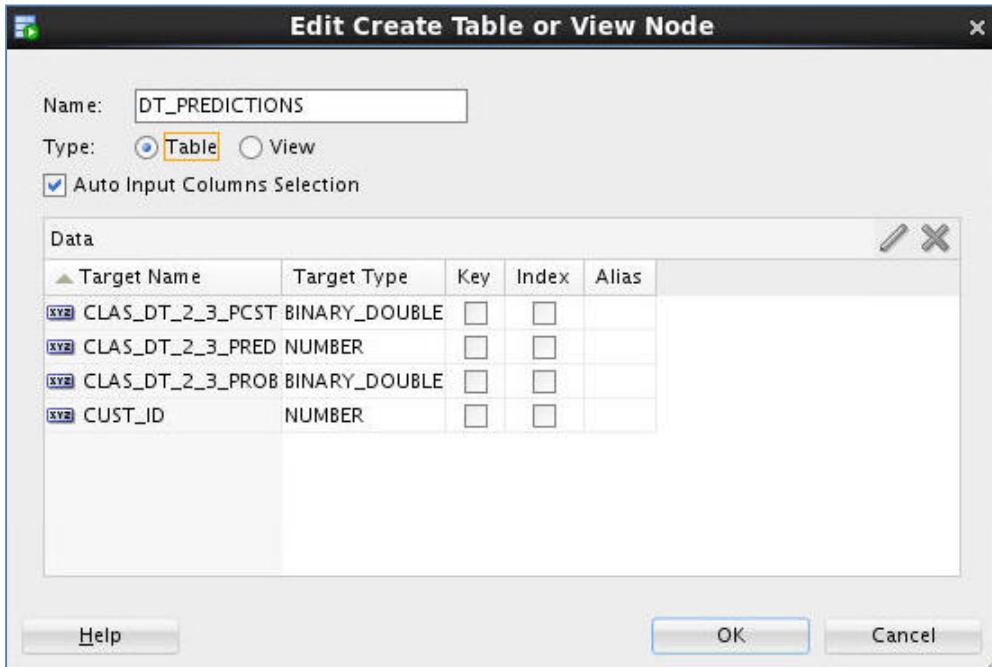
**Result:** An OUTPUT node is created (the name of your OUTPUT node may be different than shown in the example).

- B. Connect the Apply Model node to the OUTPUT node.



- C. To specify a name for the table that will be created (otherwise, Data Miner will create a default name), do the following:

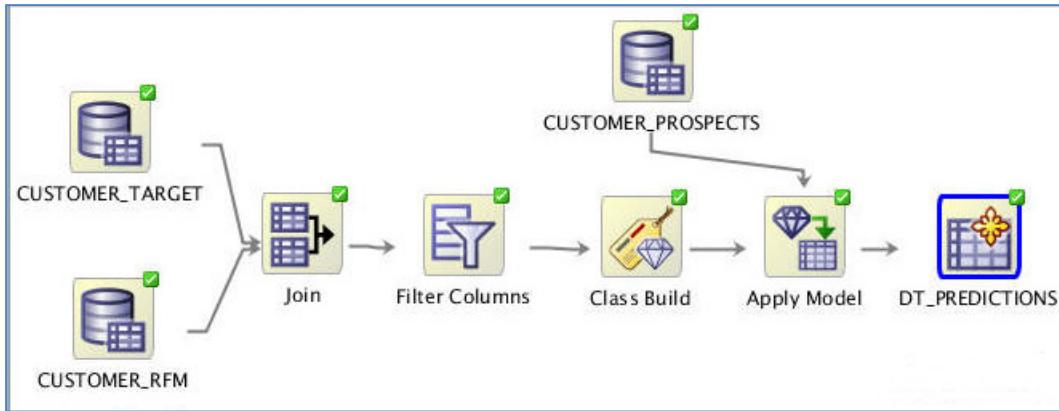
- Right-click the OUTPUT node and select **Edit** from the menu.
- In the Edit Create Table or View Node window, change the default table name to **DT\_PREDICTIONS**, as shown here:



- Then, click **OK**.

- D. Lastly, right-click the DT\_PREDICTIONS node and select **Run** from the menu. When the run process is complete, all nodes contain a green check mark in the border, as shown below.

**Note:** After you run the DT\_PREDICTIONS node, the table is created in your schema.



14. To view the predictive results, right-click the DT\_PREDICTIONS node and select **View Data** from the Menu.

- A. Then, sort the table results by specify the following criteria (as shown below):
- First - the predicted outcome (CLAS\_DT\_2\_3\_PRED), in Descending order. The prediction of "1" (Yes) appears at the top of the table display.
  - Second - prediction probability (CLAS\_DT\_2\_3\_PROB), in Descending order -- meaning that the highest prediction probabilities are at the top as well.

CLAS_DT_2_3_PRED	CLAS_DT_2_3_PROB	CLAS_DT_2_3_PCST	CUST_ID
1	1.08071216617210...	0.3358118309271...	1,161,115
2	1.08071216617210...	0.3358118309271...	1,168,350

**Select columns to sort by**

Available Columns: CLAS\_DT\_2\_3\_PCST (Asc), CUST\_ID (Asc)

Selected Columns: CLAS\_DT\_2\_3\_PRED (Des), CLAS\_DT\_2\_3\_PROB (Des)

Sort...  Ascending  Descending  Nulls First

Apply Sort Cancel

The sorted results are shown here:

The screenshot shows the Oracle Data Miner interface with the 'DT\_PREDICTIONS' tab selected. The table has four columns: CLAS\_DT\_2\_3\_PRED, CLAS\_DT\_2\_3\_PROB, CLAS\_DT\_2\_3\_PCST, and CUST\_ID. The data consists of 17 rows, each containing a value of 1 for CLAS\_DT\_2\_3\_PRED, a probability value for CLAS\_DT\_2\_3\_PROB, a PCST value, and a CUST\_ID value.

	CLAS_DT_2_3_PRED	CLAS_DT_2_3_PROB	CLAS_DT_2_3_PCST	CUST_ID
1	1	0.8292682926829...	0.2972533242615...	1,027,523
2	1	0.8292682926829...	0.2972533242615...	1,250,357
3	1	0.8292682926829...	0.2972533242615...	1,165,487
4	1	0.8292682926829...	0.2972533242615...	1,141,540
5	1	0.8292682926829...	0.2972533242615...	1,145,654
6	1	0.8292682926829...	0.2972533242615...	1,040,225
7	1	0.8292682926829...	0.2972533242615...	1,139,791
8	1	0.8292682926829...	0.2972533242615...	1,006,503
9	1	0.8292682926829...	0.2972533242615...	1,143,173
10	1	0.8292682926829...	0.2972533242615...	1,008,814
11	1	0.8292682926829...	0.2972533242615...	1,041,269
12	1	0.8292682926829...	0.2972533242615...	1,120,215
13	1	0.8292682926829...	0.2972533242615...	1,097,157
14	1	0.8292682926829...	0.2972533242615...	1,061,354
15	1	0.8292682926829...	0.2972533242615...	1,157,946
16	1	0.8292682926829...	0.2972533242615...	1,307,911
17	1	0.8292682926829...	0.2972533242615...	1,072,086

**Note:**

- Each time you run an Apply node, Oracle Data Miner takes a different sample of the data to display. With each Apply, both the data and the order in which it is displayed may change. Therefore, the sample in your table may be different from the sample shown here. This is particularly evident when only a small pool of data is available, which is the case in the schema for this lesson.
  - You can also filter the table by entering a Where clause in the Filter box.
  - The table contents can be displayed using any Oracle application or tools, such as Oracle Application Express, Oracle BI Answers, Oracle BI Dashboards, and so on.
- B. When you are done viewing the results, dismiss the tab for the DT\_PREDICTIONS table, and click **Save All**.

## Practice 23-2: Using Oracle R Enterprise with Big Data

### Overview

In this guided practice, you use Oracle R Enterprise (ORE) to run R scripts and interact with big data. You also learn how to apply some of the statistical analysis techniques that are available in ORE.

### Tasks

1. In terminal window, start the R Console at the \$ prompt.

```
R
```

```
[oracle@bigdatalite ~]$ R

Oracle Distribution of R version 3.1.1  (--) -- "Sock it to Me"
Copyright (C)  The R Foundation for Statistical Computing
Platform: x86_64-unknown-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

You are using Oracle's distribution of R. Please contact
Oracle Support for any problems you encounter with this
distribution.

> ■
```

2. Load the ORE packages and then connect to the moviedemo schema in the database using the following two commands:

**Note:** You may receive one or more warning messages if tables contain data types that are not recognized by ORE. This is normal.

```
library(ORE)
ore.connect("moviedemo", "orcl", "localhost", "welcome1", all=TRUE)
```

```
> library(ORE)
Loading required package: OREbase

Attaching package: 'OREbase'

The following objects are masked from 'package:base':

  cbind, data.frame, eval, interaction, order, paste, pmax, pmin,
  rbind, table

Loading required package: OREembed
Loading required package: OREstats
Loading required package: MASS
Loading required package: OREgraphics
Loading required package: OREeda
Loading required package: OREmodels
Loading required package: OREdm
Loading required package: lattice
Loading required package: OREPredict
Loading required package: ORExml
> ore.connect("moviedemo","orcl","localhost","welcome1",all=TRUE)
Loading required package: ROracle
Loading required package: DBI
Warning messages:
1: table "MOVIEDEMO"."CUSTOMER" contains unsupported data types
2: table "MOVIEDEMO"."PROSPECTS" contains unsupported data types
3: table "MOVIEDEMO"."CUSTOMER_RFM" contains unsupported data types
R> ■
```

- In the following steps, you perform a variety of statistical analysis on the moviedemo data to better understand your customer profiles.

**Note:** Depending on your local environment, the R command prompt may display as “>” or “R>”. In either case, the command prompt is correct.

#### A. View the contents of the database schema:

```
ore.ls()
```

```
R> ore.ls()
[1] "ACTIVITY"                 "CAST"                      "CORPUS_TABLE"           "CREW"
[5] "CUSTOMER_SEGMENT"          "CUSTOMER_TARGET"          "CUSTOMER_V"             "CUST_R
ATING"
[9] "GENRE"                     "MOVIE"                    "MOVIE_ASSOC_RES"        "MOVIE_
CAST"
[13] "MOVIE_CREW"              "MOVIE_FACT"               "MOVIE_FACT_EXT_TAB_FILE" "MOVIE_
FACT_EXT_TAB_HIVE"
[17] "MOVIE_FACT_HDFS_EXT_TAB" "MOVIE_FACT_HDFS_V"        "MOVIE_FACT_LOCAL"       "MOVIE_
FACT_MW_HDFS_EXT_TAB"
[21] "MOVIE_FACT_V"            "MOVIE_GENRE"              "MOVIE_LTV"              "MOVIE_
SESSIONS_TAB"
[25] "MOVIE_SIMILARITY"        "MOVIE_SIMILARITY2"        "MOVIE_SIMILARITY_SAMP"   "NOSQL_
GENRE"
[29] "NOSQL_GENRE_MOVIE"       "NOSQL_MOVIE"              "ONTIME_S"               "SESSIO
NS"
[33] "SESSIONS_HDFS_EXT_TAB"   "TIMES"                   "TIMES_HOURS"
R> ■
```

## B. Determine if the CUSTOMER\_V table exists:

```
ore.exists("CUSTOMER_V")
```

```
R> ore.exists("CUSTOMER_V")
[1] TRUE
```

## C. Determine that table's dimensions and summary statistics. The dimensions and summary are computed in the database with only the results being retrieved:

```
dim(CUSTOMER_V)
names(CUSTOMER_V)
```

```
R> dim(CUSTOMER_V)
[1] 4848   39
```

```
R> names(CUSTOMER_V)
 [1] "CUST_ID"           "LAST_NAME"        "FIRST_NAME"
 "STREET_ADDRESS"      "POSTAL_CODE"       "STATE_PROVINCE_ID"
 [6] "CITY_ID"           "CITY"             "STATE_PROVINCE_ID"
 "STATE_PROVINCE"       "COUNTRY_ID"        "CONTINENT"
 [11] "COUNTRY"          "CONTINENT_ID"      "CONTINENT"
 "AGE"                 "COMMUTE_DISTANCE" "EMAIL"
 [16] "CREDIT_BALANCE"   "EDUCATION"        "EMAIL"
 "FULL_TIME"           "GENDER"           "INCOME_LEVEL"
 [21] "HOUSEHOLD_SIZE"   "INCOME"           "INCOME_LEVEL"
 "INSUFF_FUNDS INCIDENTS" "JOB_TYPE"        "MORTGAGE_AMT"
 [26] "LATE_MORT_RENT_PMTS" "MARITAL_STATUS"   "MORTGAGE_AMT"
 "NUM_CARS"            "NUM_MORTGAGES"    "RENT_OWN"
 [31] "PET"               "PROMOTION_RESPONSE" "RENT_OWN"
 "SEG"                 "WORK_EXPERIENCE"   "YRS_RESIDENCE"
 [36] "YRS_CURRENT_EMPLOYER" "YRS_CUSTOMER"    "YRS_RESIDENCE"
 "COUNTRY_CODE"
```

## D. To determine answers to the following questions about our customers:

- 1) Which gender (male or female) is better represented?
- 2) Is the customer base skewed toward young customers or old customers?
- 3) Are customers highly educated?

```
summary(CUSTOMER_V[, c("GENDER", "INCOME", "AGE", "EDUCATION")])
```

```
R> summary(CUSTOMER_V[, c("GENDER", "INCOME", "AGE", "EDUCATION")])
   GENDER      INCOME         AGE          EDUCATION
Male :2479  Min.   : 29  Min.   :16.00  LessThanHS :877
Female:2369  1st Qu.:22450  1st Qu.:32.00  High School:816
              Median :46612  Median :48.00   Associates :811
              Mean   :50056  Mean   :48.23   Masters    :792
              3rd Qu.:75123  3rd Qu.:65.00   Bachelors  :768
              Max.   :119987  Max.   :80.00   Doctorate  :749
                                      Unknown   : 35
```

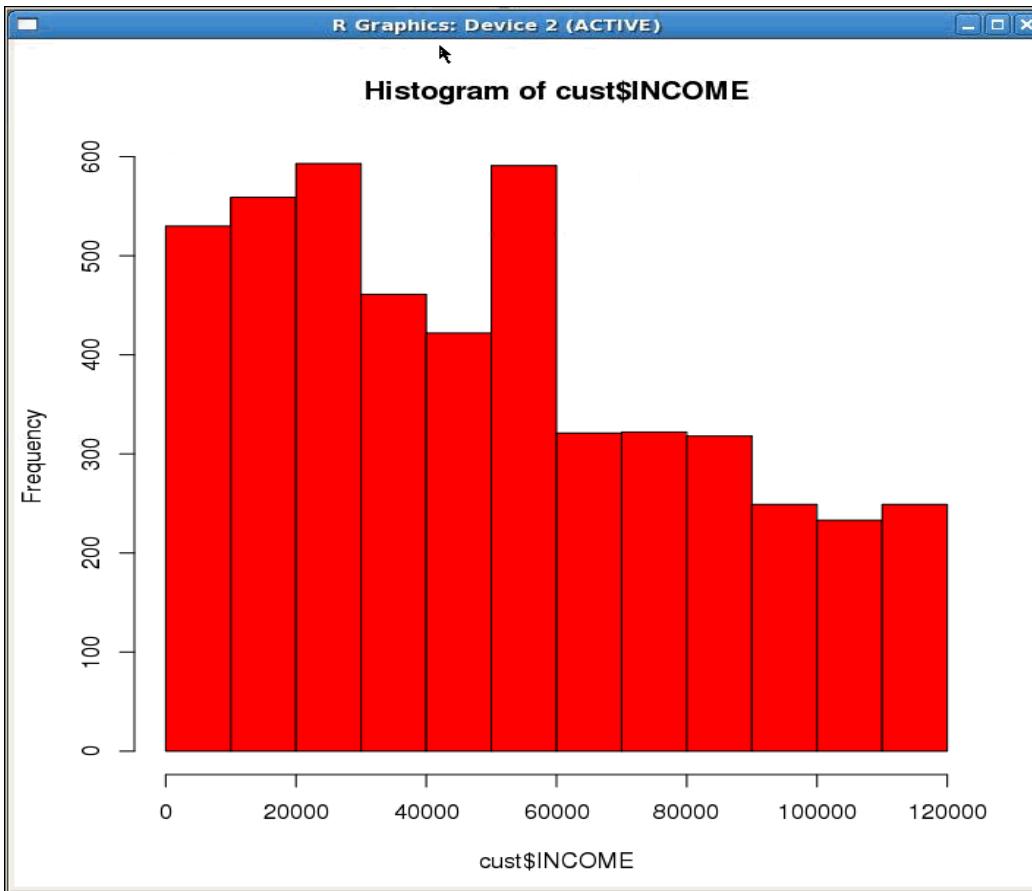
E. Execute the commands shown below to answer to the following questions:

- 1) Are customers generally upper-income or lower-income?
- 2) Are there any surprises in the distribution of customer incomes?
- 3) What is the income range of the middle 50% of customers?

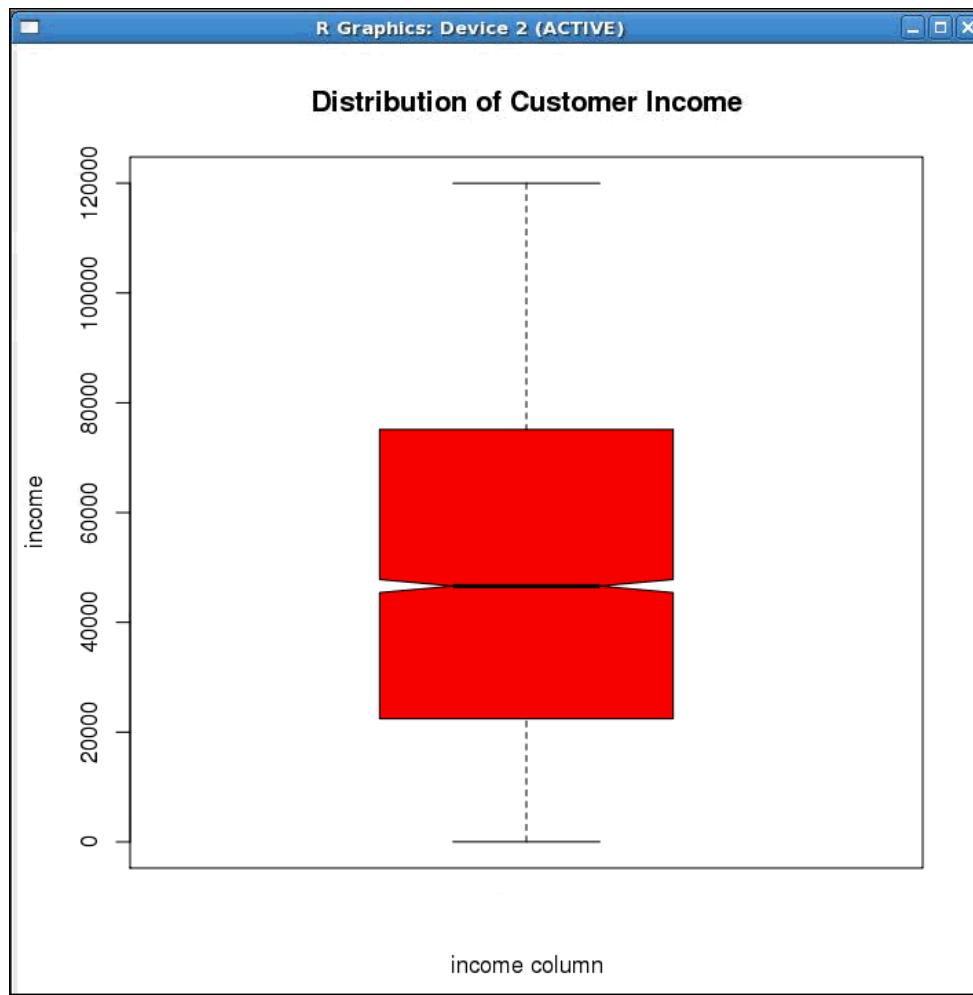
**Note:** The first two commands generate the histogram, and the third command generates a boxplot.

```
cust <- CUSTOMER_V  
hist(cust$INCOME,col="red")  
  
boxplot(cust$INCOME,xlab="income column",ylab="income",  
main="Distribution of Customer Income",  
col="red",notch=TRUE)
```

```
R> cust <- CUSTOMER_V  
R>  
R> hist(cust$INCOME,col="red")
```



```
R> boxplot(cust$INCOME,xlab="income column",ylab="income",
+           main="Distribution of Customer Income",
+           col="red",notch=TRUE)
```



F.

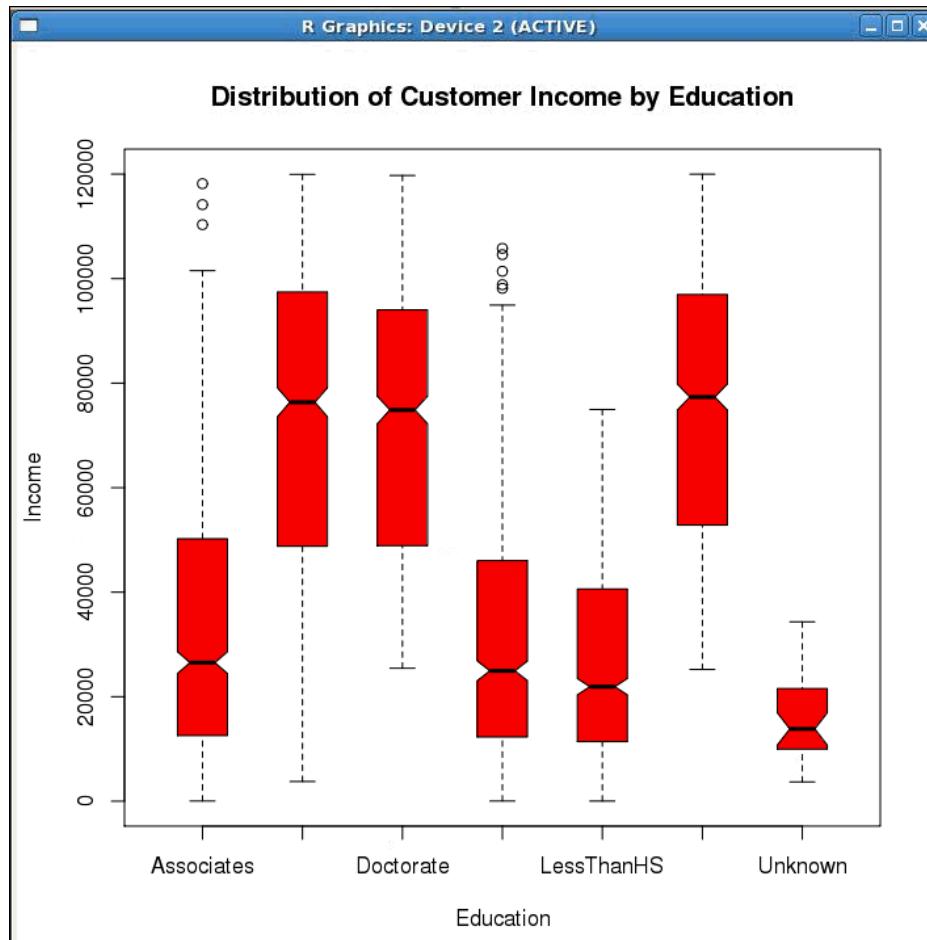
Answer the following question using the commands: Should we consider segmenting customers based on education and income?

**Note:**

- First, use the overloaded function `split()` to partition the data in Oracle Database.
- Then, use the resulting list to illustrate the answer by producing a boxplot for income by education.

```
cust.split <- with(cust, split(INCOME,as.factor(EDUCATION)))
boxplot(cust.split, xlab="Education",ylab="Income",boxwex = 0.5,
main="Distribution of Customer Income by Education",
col="red",notch=TRUE)
```

```
R> cust.split <- with(cust, split(INCOME,as.factor(EDUCATION)))
R>
R> boxplot(cust.split, xlab="Education",ylab="Income",boxwex = 0.5,
+           main="Distribution of Customer Income by Education",
+           col="red",notch=TRUE)
```



G. Using the commands below, answer the following questions:

- 1) Are there more single or married customers in each education group?
- 2) Can you use the overloaded table function on the ore.frame object `cust` to build a contingency table of counts at each combination of factor levels to show the table numerically?

```
cust.tab <- with (cust, table(EDUCATION,MARITAL_STATUS))  
cust.tab
```

```
R> cust.tab <- with (cust, table(EDUCATION,MARITAL_STATUS))  
R>  
R> cust.tab  
      MARITAL_STATUS  
EDUCATION      M   S  
  Associates    389 422  
  Bachelors     364 404  
  Doctorate     365 384  
  High School   366 450  
  LessThanHS    376 501  
  Masters       369 423  
  Unknown        1   34
```

H. For customers, are there correlations between age, income, and the number of years since first becoming a customer?

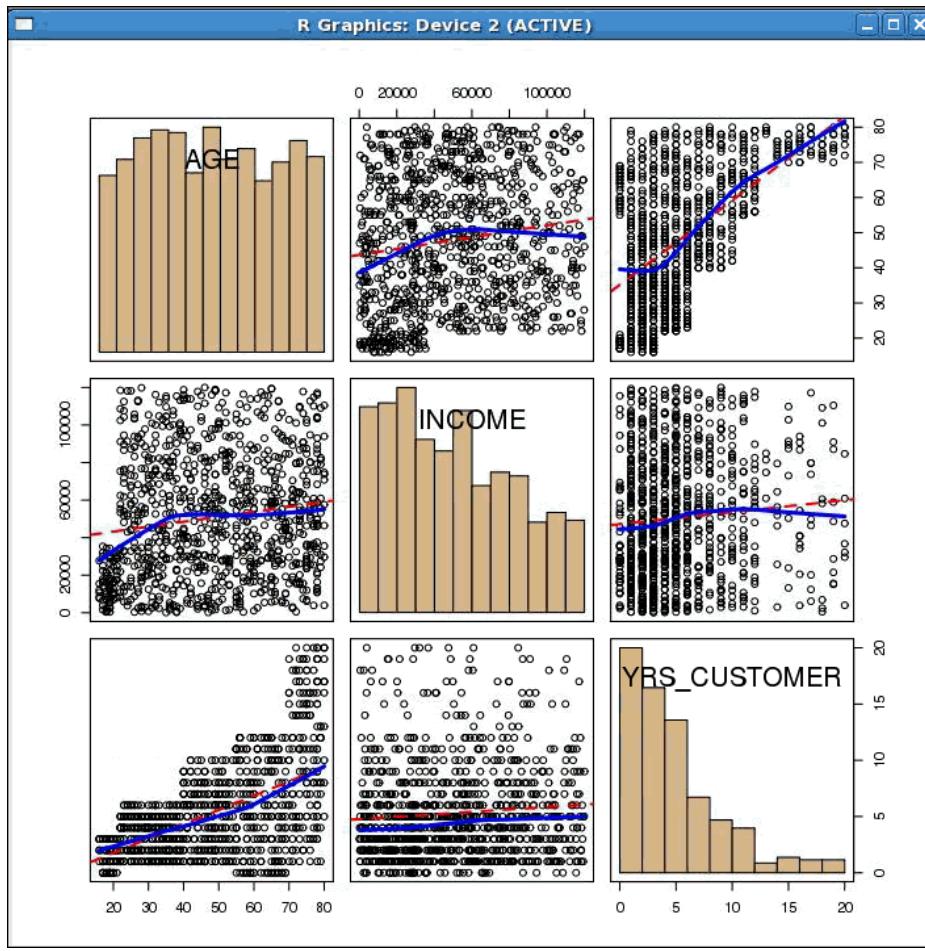
In this example:

- You produce a “pairs” plot that produces scatterplots of pairs of columns. From the `CUSTOMER_V` table, you sample 20% of the customers and select the columns `AGE`, `INCOME`, and `YRS_CUSTOMER`.
- Then, using the `pairs()` function, you not only produce a scatterplot but also draw a regression line (in red) and a lowess curve in blue. Along the diagonal, a histogram of each column’s data is plotted.

**Note:** You will notice that age correlates strongly with the number of years as a customer. Income and age show a mild correlation.

```
C1 <- CUSTOMER_V
row.names(C1) <- C1$CUST_ID
N <- nrow(C1)
s <- sample(1:N,N*0.2)
with(C1[s,],
pairs(cbind(AGE, INCOME, YRS_CUSTOMER),
panel=function(x,y) {
points(x,y)
abline(lm(y~x),lty="dashed",col="red",lwd=2)
lines(lowess(x,y),col="blue",lwd=3)
}),
diag.panel=function(x) {
par(new=TRUE)
hist(x,main="",axes=FALSE, col="tan")
}
))
```

```
R> C1 <- CUSTOMER_V
R>
R> row.names(C1) <- C1$CUST_ID
R>
R> N <- nrow(C1)
R>
R> s <- sample(1:N,N*0.2)
R>
R> with(C1[s,],
+
+     pairs(cbind(AGE, INCOME, YRS_CUSTOMER),
+
+             panel=function(x,y) {
+
+                 points(x,y)
+
+                 abline(lm(y~x),lty="dashed",col="red",lwd=2)
+
+                 lines(lowess(x,y),col="blue",lwd=3)
+
+             },
+
+             diag.panel=function(x){
+
+                 par(new=TRUE)
+
+                 hist(x,main="",axes=FALSE, col="tan")
+
+             }
+
+         ))
```



- I. Next, you answer the question “Which actor has the most movie titles to his or her credit?”
- You draw on three tables: MOVIE\_CAST, CAST, and MOVIE.
  - Then, you join these tables, aggregate based on the actor’s name, and selecting those with a count greater than 110.

```
MC <- MOVIE_CAST
C1 <- CAST
M1 <- MOVIE[,c("MOVIE_ID", "TITLE", "YEAR")]
m1 <- merge(MC,C1,by="CAST_ID") [,c("NAME", "MOVIE_ID", "CAST_ID")]
names(m1) <- c("ACTOR", "MOVIE_ID", "CAST_ID")
ACTORS <- merge(m1,M1,by="MOVIE_ID") [,c("CAST_ID", "ACTOR",
"MOVIE_ID", "TITLE", "YEAR")]
#row.names(ACTORS) <- c(ACTORS$MOVIE_ID,ACTORS$CAST_ID)
MOVIE_ACTORS <- ACTORS[,c("ACTOR", "TITLE", "YEAR")]
aggdata <- aggregate(MOVIE_ACTORS$ACTOR,
by = list(MOVIE_ACTORS$ACTOR),
FUN = length)
aggdata[aggdata$x > 50,]
```

```
R> MC <- MOVIE_CAST
R>
R> C1 <- CAST
R>
R>
R> M1 <- MOVIE[,c("MOVIE_ID", "TITLE", "YEAR")]
R>
R>
R>
R> m1 <- merge(MC,C1,by="CAST_ID") [,c("NAME", "MOVIE_ID", "CAST_ID")]
R>
R> names(m1) <- c("ACTOR", "MOVIE_ID", "CAST_ID")
R>
R>
R> ACTORS <- merge(m1,M1,by="MOVIE_ID") [,c("CAST_ID", "ACTOR",
+ "MOVIE_ID", "TITLE", "YEAR")]
R>
R> #row.names(ACTORS) <- c(ACTORS$MOVIE_ID,ACTORS$CAST_ID)
R>
R>
R>
R> MOVIE_ACTORS <- ACTORS[,c("ACTOR", "TITLE", "YEAR")]
R>
R> aggdata <- aggregate(MOVIE_ACTORS$ACTOR,
+ by = list(MOVIE_ACTORS$ACTOR),
+ FUN = length)
R>
R> aggdata[aggdata$x > 50,]
      Group.1   x
Robert De Niro Robert De Niro 54
```

J.

Now, you answer the question “Which are the most popular movie genres based on the number of movies produced in that genre?”

- Using the MOVIE\_GENRE and GENRE ore.frame objects, merge (join) the data so that you can use genre names instead of IDs.
- Use the overloaded function aggregate on the joined ore.frame to count the number of movies in each genre. The barplot window can be widened so that more labels can be shown.

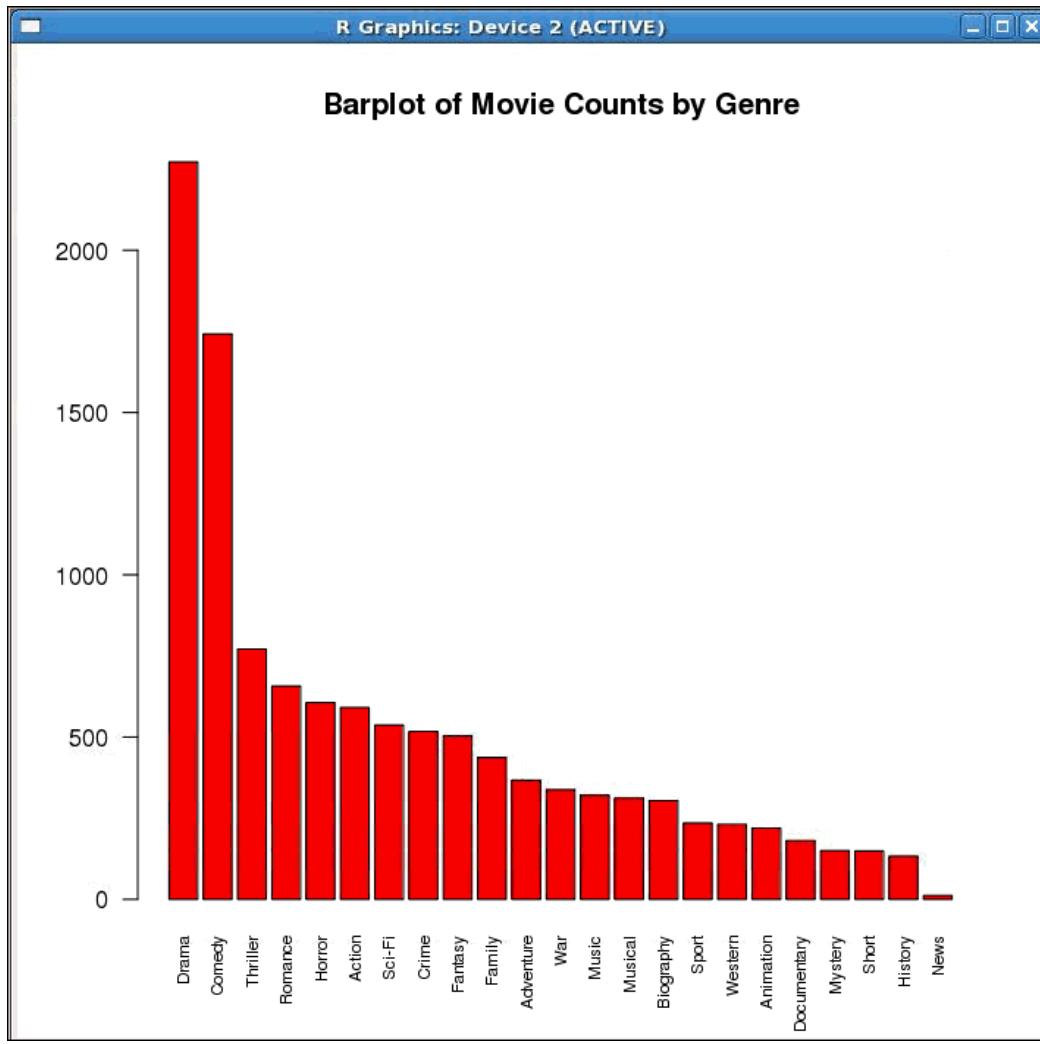
```
MG <- MOVIE_GENRE
G1 <- GENRE
m1 <- merge(MG,G1, by="GENRE_ID")
genre.cnts <- with (m1, aggregate(NAME,
by = list(NAME),
FUN = length))
class(genre.cnts)
names(genre.cnts) <- c("genre", "cnt")
genre.cnts
gcnts <- ore.pull(genre.cnts)
gcnts.sorted <- gcnts[order(gcmts$cnt,decreasing=TRUE),]
barplot(height=gcnts.sorted$cnt, names=gcnts.sorted$genre,
main="Barplot of Movie Counts by Genre",
col="red",cex.names=0.7,las=2)
```

```
R> MG <- MOVIE_GENRE
R>
R> G1 <- GENRE
R>
R> m1 <- merge(MG,G1, by="GENRE_ID")
R>
R> genre.cnts <- with (m1, aggregate(NAME,
+                               by = list(NAME),
+
+                               FUN = length))
R>
R> class(genre.cnts)
[1] "ore.frame"
attr(,"package")
[1] "OREbase"
R>
R> names(genre.cnts) <- c("genre", "cnt")
R>
```

```
R> genre.cnts
      genre   cnt
Action       Action  591
Adventure    Adventure 367
Animation    Animation 219
Biography    Biography 304
Comedy       Comedy 1742
Crime        Crime  517
Documentary Documentary 181
Drama        Drama 2272
Family       Family 437
Fantasy      Fantasy 504
History      History 133
Horror       Horror 606
Music        Music 321
Musical      Musical 312
Mystery      Mystery 150
News         News  12
Romance      Romance 657
Sci-Fi       Sci-Fi 537
Short        Short 149
Sport        Sport 235
Thriller     Thriller 771
War          War  338
Western      Western 231
R>
```

```
R> gcnts <- ore.pull(genre.cnts)
R>
R> gcnts.sorted <- gcnts[order(gcnts$cnt,decreasing=TRUE),]
R>
R> barplot(height=gcnts.sorted$cnt, names=gcnts.sorted$genre,
+           main="Barplot of Movie Counts by Genre",
+           col="red",cex.names=0.7,las=2)
```

K.



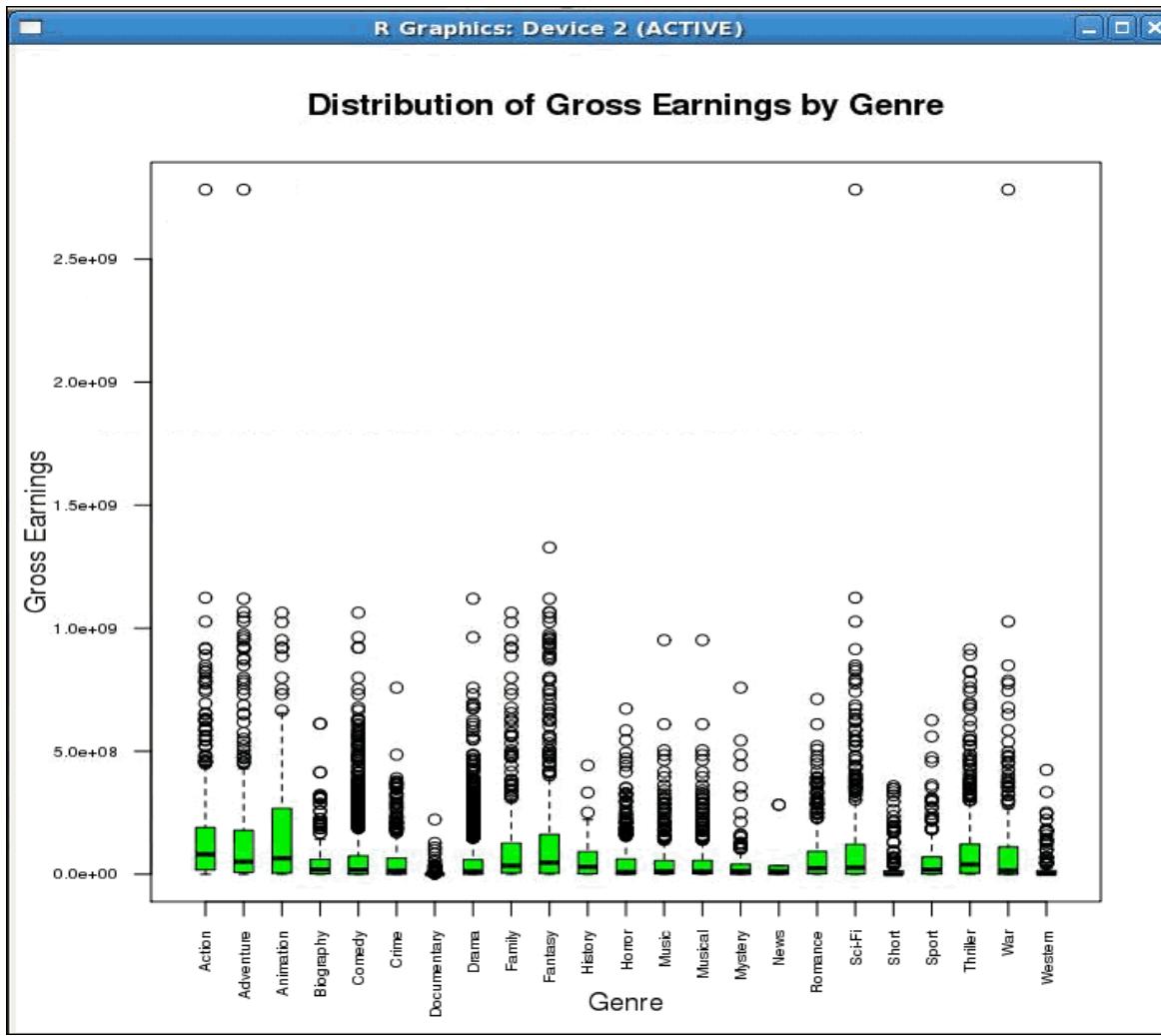
Finally, which movie genre generally has the lowest gross income? Use the:

- Overloaded `merge()` function on the `ore.frame` objects `MOVIE_GENRE`, `GENRE`, and `MOVIE` to get a sense of the distribution of gross earnings in a genre.
- `dim` function to see the size of the resulting data set.
- `split()` function and then graph the distribution of popularity using the `boxplot()` function.

**Note:** ORE enables all of the heavy computational work to be performed in the database. Only summary data is brought to the client to graph the statistics.

```
g <- merge(MOVIE_GENRE,GENRE,by="GENRE_ID")
m <- merge(MOVIE,g[,2:3],by="MOVIE_ID")
m$GENRE <- m$NAME
dim(m)
m.split <- split(m$GROSS,m$GENRE)
boxplot(m.split, ylab="Gross Earnings",xlab="Genre",col="green",
main="Distribution of Gross Earnings by Genre",
cex.axis=0.6, boxwex=.5, las=2)
```

```
R> g <- merge(MOVIE_GENRE,GENRE,by="GENRE_ID")
R>
R> m <- merge(MOVIE,g[,2:3],by="MOVIE_ID")
R>
R> m$GENRE <- m$NAME
R>
R> dim(m)
[1] 11586      8
R>
R> m.split <- split(m$GROSS, m$GENRE)
R>
R> boxplot(m.split, ylab="Gross Earnings",xlab="Genre",col="green",
+           main="Distribution of Gross Earnings by Genre",
+
+           cex.axis=0.6, boxwex=.5, las=2)
```



4. Quit your R session invoking the `q()` function. When prompted to save the workspace image, respond with `n` (no) and press the Enter key.
5. Then, close the terminal window with the `exit` command at the \$ prompt.



## **Practices for Lesson 24: Introducing Oracle Big Data Discovery**

**Chapter 24**

## Practices for Lesson 24

---

There are no practices for this lesson.

## **Practices for Lesson 25: Introduction to the Oracle Big Data Appliance (BDA)**

**Chapter 25**

## Practices for Lesson 25

---

### Practices Overview

In this practice, you use your web browser to access and review some of the useful Big Data Appliance resources such as the **Getting Real About Big Data: Build Versus Buy** white paper and the Oracle BDA 3\_D demonstration.

## Guided Practice 25-1: Introduction to Oracle BDA

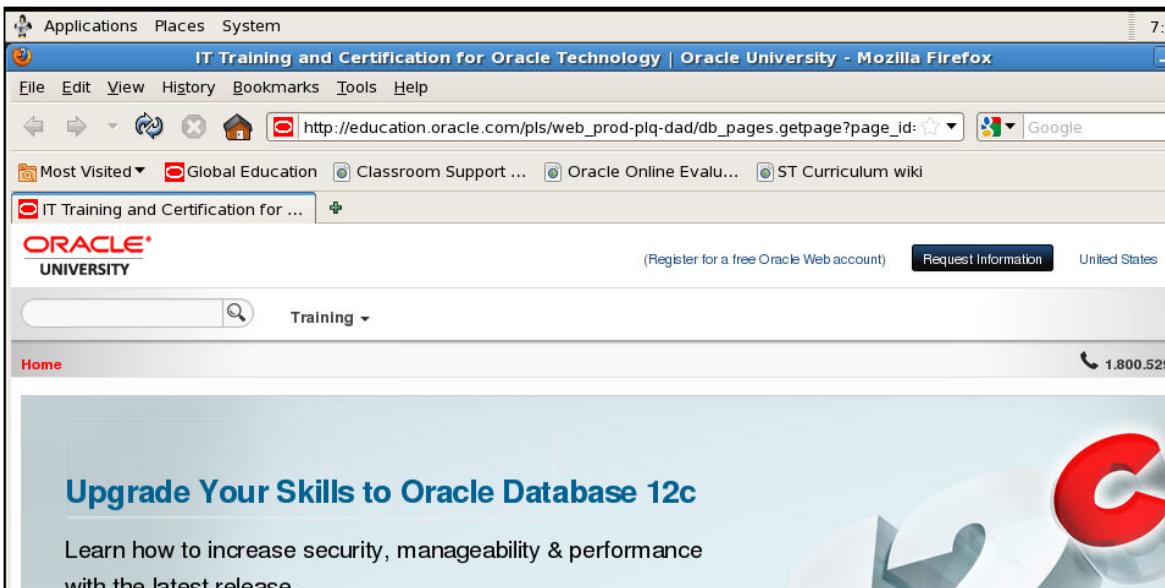
### Overview

In this practice, you use your web browser to access and review some of the useful Big Data Appliance resources such as the **Getting Real About Big Data: Build Versus Buy** white paper and the Oracle BDA 3D demonstration.

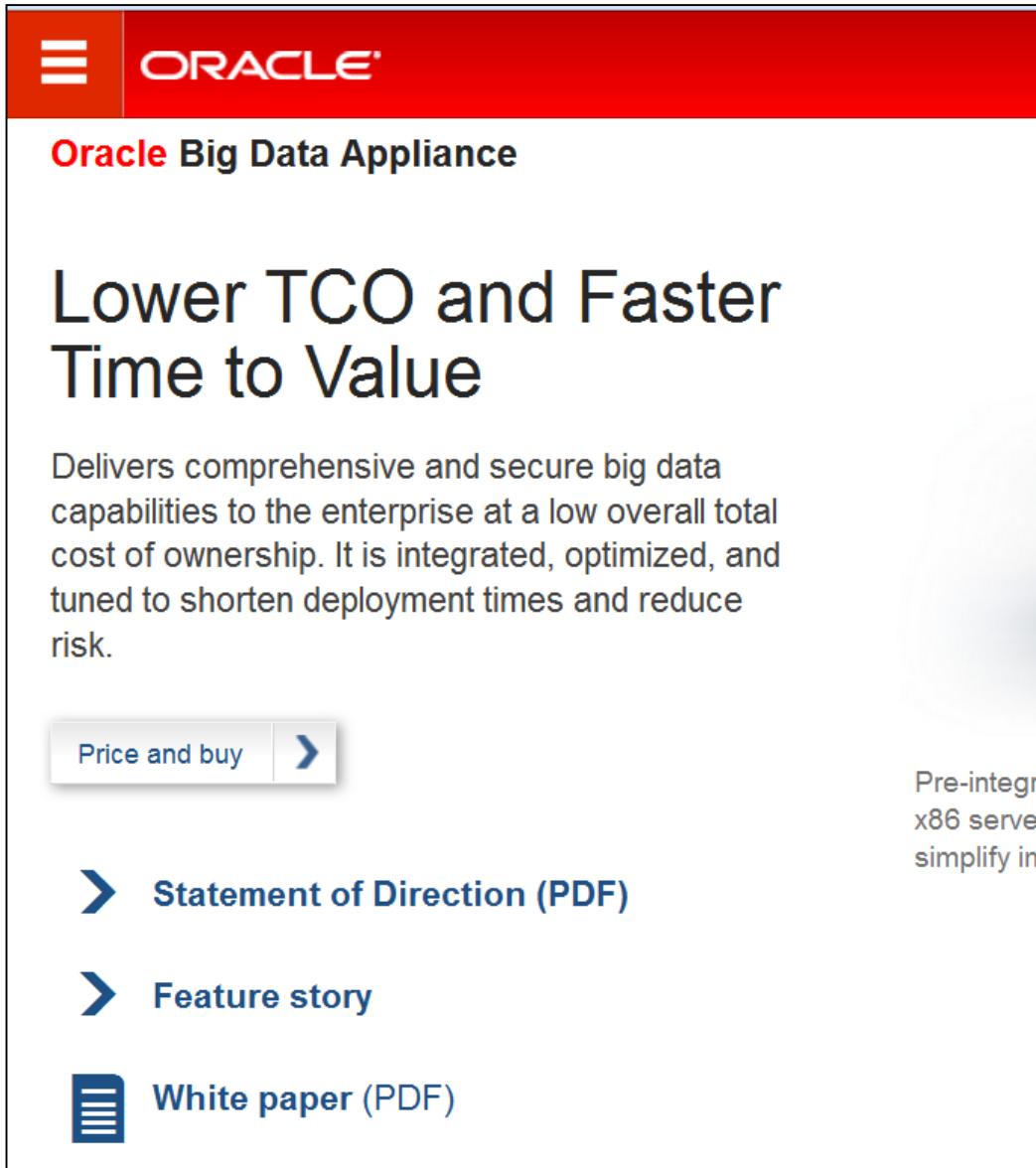
### Assumptions

### Tasks

1. Start your Mozilla Firefox web browser on your host machine desktop (not the one in your BDLite VM).



2. Access, review, and bookmark (on the Bookmarks toolbar) the following Oracle Big Data page at: <https://www.oracle.com/engineered-systems/big-data-appliance/index.html>



The image shows the Oracle Big Data Appliance landing page. At the top is the Oracle logo. Below it is the title "Oracle Big Data Appliance". The main headline reads "Lower TCO and Faster Time to Value". A descriptive text block follows, stating: "Delivers comprehensive and secure big data capabilities to the enterprise at a low overall total cost of ownership. It is integrated, optimized, and tuned to shorten deployment times and reduce risk." To the right of this text is a vertical sidebar with the heading "Pre-integrated x86 server simplifies implementation". Below the main headline are three call-to-action buttons: "Price and buy" (with a blue arrow icon), "Statement of Direction (PDF)" (with a blue arrow icon), "Feature story" (with a blue arrow icon), and "White paper (PDF)" (with a blue document icon). The "White paper (PDF)" button is highlighted with a red border.

**Oracle Big Data Appliance**

# Lower TCO and Faster Time to Value

Delivers comprehensive and secure big data capabilities to the enterprise at a low overall total cost of ownership. It is integrated, optimized, and tuned to shorten deployment times and reduce risk.

Pre-integrated x86 server simplifies implementation

[Price and buy](#) ➤

➤ [Statement of Direction \(PDF\)](#)

➤ [Feature story](#)

 [White paper \(PDF\)](#)

3. Click the White Paper (PDF) link to view the **Getting Real About Big Data: Build Versus Buy** document. Spend 10-15 minutes to review the content of the white paper. Make sure you read the “Do-it-yourself Hadoop Can be Difficult” and the “Real Costs and Benefits of Big Data Infrastructure” sections.



A callout box highlights the "White paper (PDF)" link from the previous slide. The box has a red border and contains the same three items: "Statement of Direction (PDF)", "Feature story", and "White paper (PDF)". The "White paper (PDF)" link is specifically highlighted with a red border around its icon and text.

➤ [Statement of Direction \(PDF\)](#)

➤ [Feature story](#)

 [White paper \(PDF\)](#)

The screenshot shows a web browser window with the following details:

- Page header: Page: 3 of 16
- Page title: Do-it-yourself Hadoop Can Be Difficult
- Text content:
  - Web 2.0 companies, such as Google and Yahoo, have successfully built big data infrastructures from scratch. Those same firms also have been primary participants in the birth and nurturing of Hadoop, the Apache open source project deservedly given credit for catalyzing the big data movement. Hadoop has matured over the past year, due in part to feature enhancements and in part to growing support from a widening range of both startups and more established IT vendors.
  - For many organizations, Hadoop-based solutions will represent the first new explicitly big data investment. However, successful Hadoop implementations put into full production using do-it-yourself infrastructure components may require more time and effort than initially expected. Hadoop lures many big data hopefuls due to its apparent low infrastructure cost and easy access; as an open source technology, anyone can download Hadoop for free, and can spin up a simple Hadoop infrastructure in the cloud or on-premises. Unfortunately, many organizations also quickly find that they lack the time, resources, and expertise to make do-it-yourself Hadoop infrastructure work for big data, with reported shortages of skilled staff in IT architecture and planning, business intelligence and analytics, and/or database administration at approximately 20% of companies.<sup>4</sup>
  - The still somewhat rare and expensive expertise comes in two forms: (1) The Hadoop engineer, who can architect an initial Hadoop infrastructure, feed applicable data in, help the data analyst squeeze useful analytics out from the data repository, and evolve and manage the whole infrastructure over time; and (2) The data scientist or analyst, who knows how to render the tools of statistics in the context of big data analytics, and also can lead the human and business process of discovery and collaboration in order to yield actionable results.
  - Thus, despite the hope and hype, Hadoop on a commodity stack of hardware does not always offer a lower cost ride to big data analytics, or at least not as quickly as hoped. ESG asserts that many organizations implementing Hadoop infrastructures based on human expertise plus a purchased commodity hardware and software infrastructure may experience unexpected costs, slower speed to market, and unplanned complexity throughout the lifecycle.
- Page footer: Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Page: 7 of 16 Automatic Zoom

## The Real Costs and Benefits of Big Data Infrastructure

### A Model for Hadoop Big Data Infrastructure Cost Analysis: Build Versus Buy

What would an enterprise experience in terms of costs if (a) it rolled its own Hadoop infrastructure versus (b) it used an appliance that was purpose-designed and integrated for enterprise-class big data? One of the possible myths about Hadoop has been that companies can save money by rolling out and managing their own Hadoop commodity infrastructure. Of course, big data using Hadoop isn't just about clusters, it is about infrastructure: The infrastructure includes, for example, systems management software, networking, and extra capacity for a variety of analytics processing purposes.

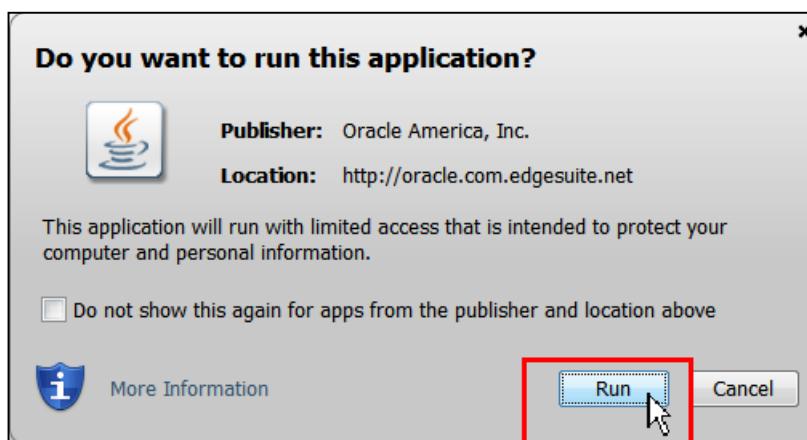
A very important note is that "soft costs" of labor time to evaluate, procure, test, deploy, and integrate a full stack of hardware and software is not examined here. These costs in time and money can be extreme, ranging to hundreds of hours or more, and this topic alone is one of the most compelling reasons to consider a purpose-built big data appliance. That said, many organizations look to their existing staff to manage this effort, and would heavily discount or exclude these human costs, real as they may be.

ESG conducted a review of the Oracle Big Data Appliance for a three-year total cost comparison with a closely matched commodity "build" infrastructure. While the exact pricing will of course vary by vendor and over time, this model uses the closest directly comparable equipment, including HP Proliant DL-series servers and Infiniband networking, and costs that are publically available at the time of writing. Evaluators should always use this kind of ROI model as a reference while conducting their own calculations based on their specific environmental requirements. This exercise is primarily a validation of the comparable costs and financial calculations provided by Oracle, and a similar comparison may be found at Oracle blog, [Price Comparison for Big Data Appliance and Hadoop](#).

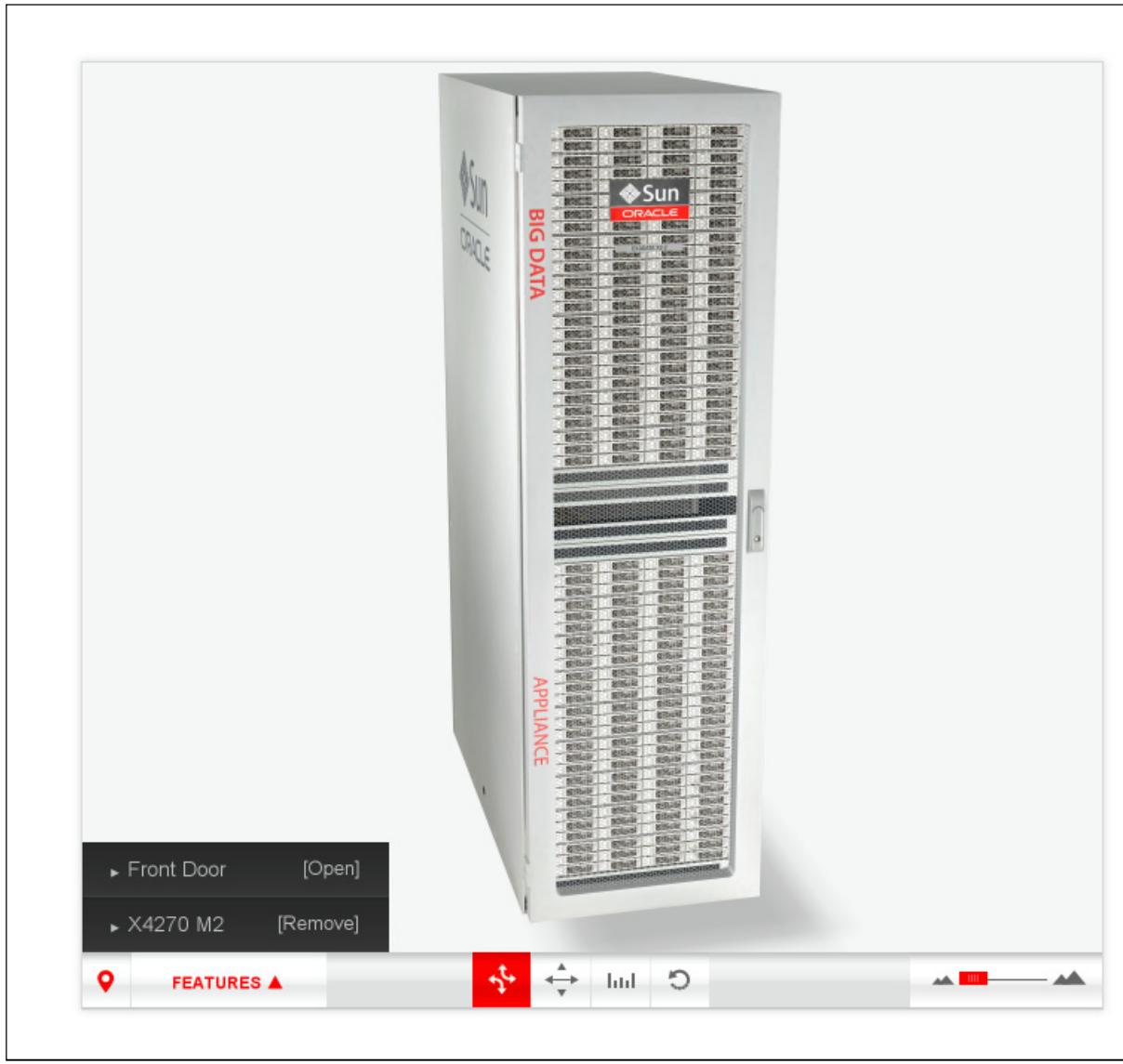
Though some readers will argue that lower cost servers and networking components are frequently chosen in real-world deployments, these options can be lacking in performance and reliability features and aren't as directly comparable to the equipment included in the Oracle offering. The reason that the Oracle Big Data Appliance itself works well for this comparison is (a) it would serve well as an infrastructure for a medium-sized big data project, as depicted in Table 1, and (b) the cost and infrastructural details of the "buy" option—Oracle Big Data Appliance in this case—are publicly disclosed.

4. Access the Oracle BDA 3-D demo by using the following URL:  
<http://oracle.com.edgesuite.net/producttours/3d/big-data-appliance/index.html>

**Note:** If you are prompted with the following message asking you if you want to run this application, click **Run**.



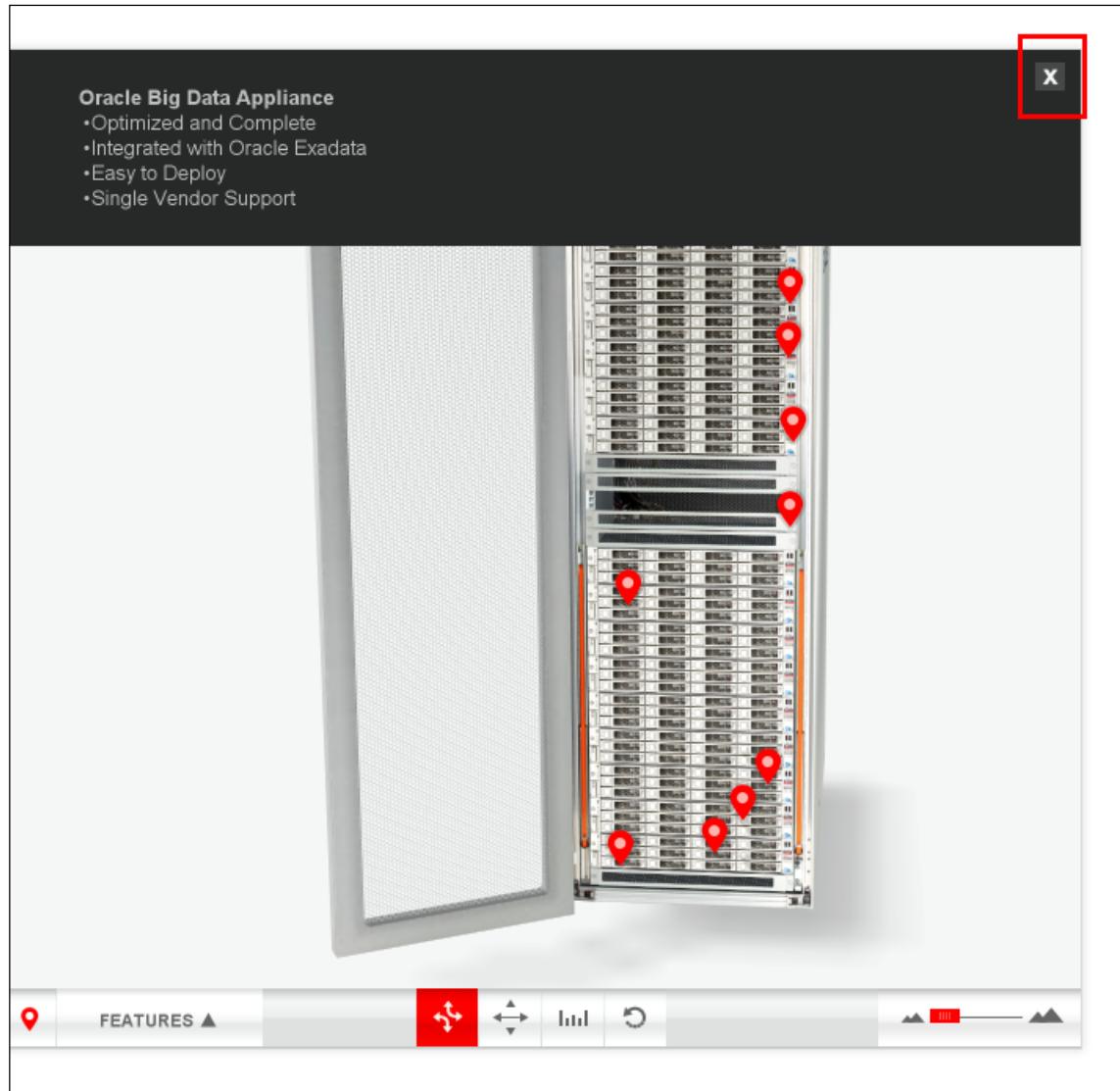
This demo shows the older BDA X4270 M2 and its components and is only meant to get you familiar with the BDA look and feel. A newer 3D demo using the latest BDA X5-2 will be available soon but was not yet available at the time this practice was written. As soon as the latest demo is available, the new link will be provided to the instructor who can then pass that information to you.



5. Explore the interactive demo as follows:

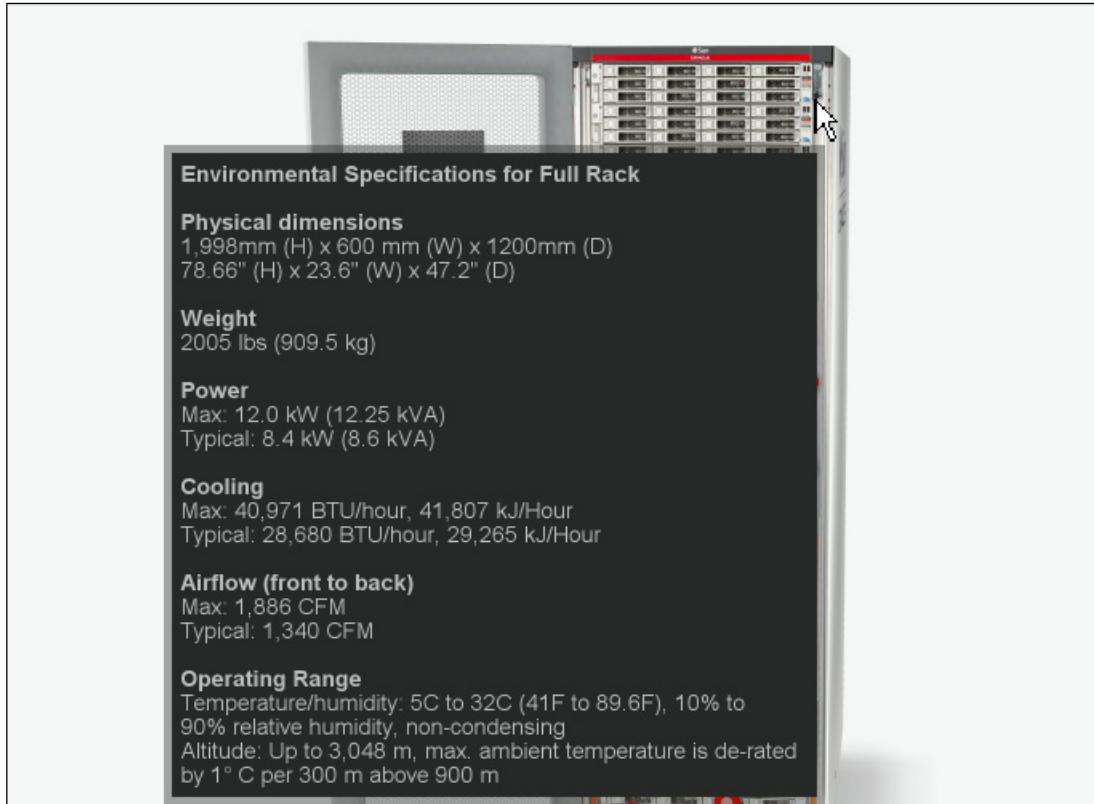
- Click the **Front Door [Open]** link, and then click the x control to close the black window.



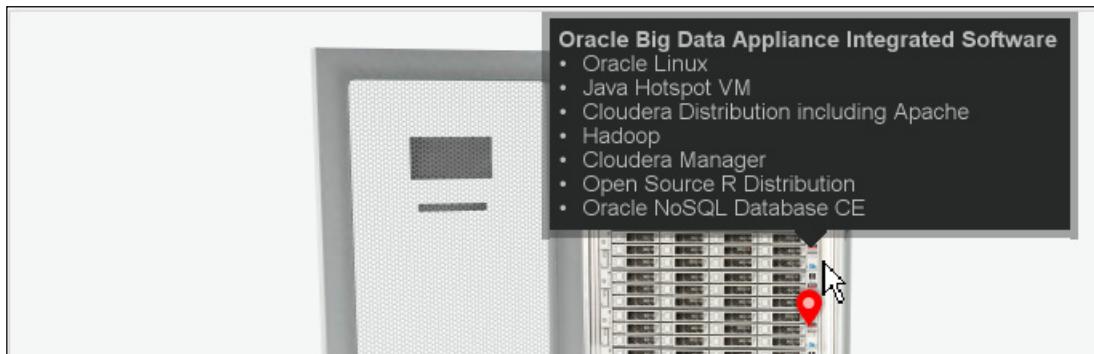


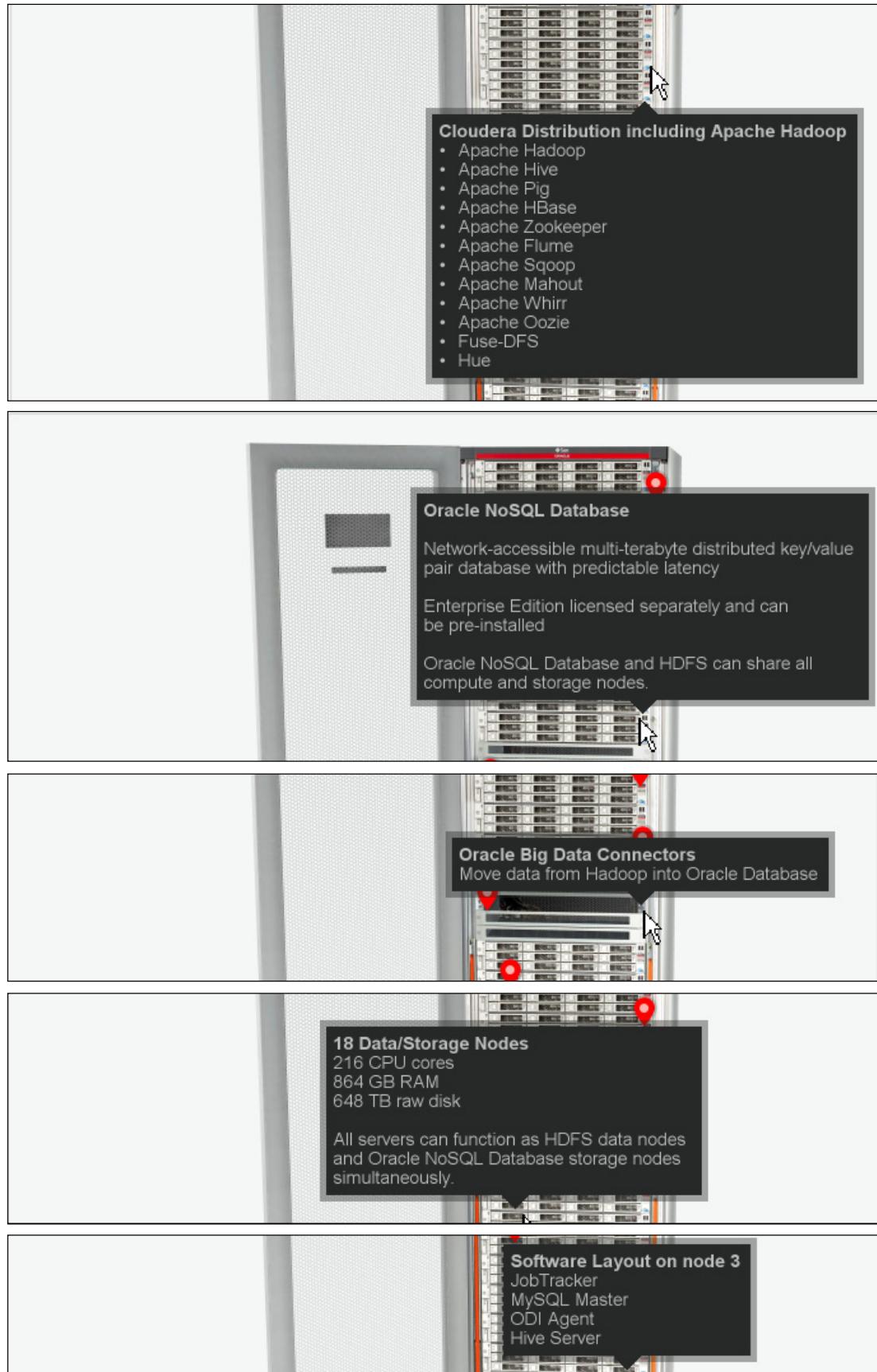
- b. Move your mouse cursor over any of the red icons (Find Notes) on the Big Data Appliance picture to display information about that component. For example, click the top red icon to display information about the **Environmental Specifications for Full Rack**.





- c. Explore the remaining red icons, especially the bottom half of the picture, which displays information about the number of data nodes and the software installed on the critical nodes in the BDA.



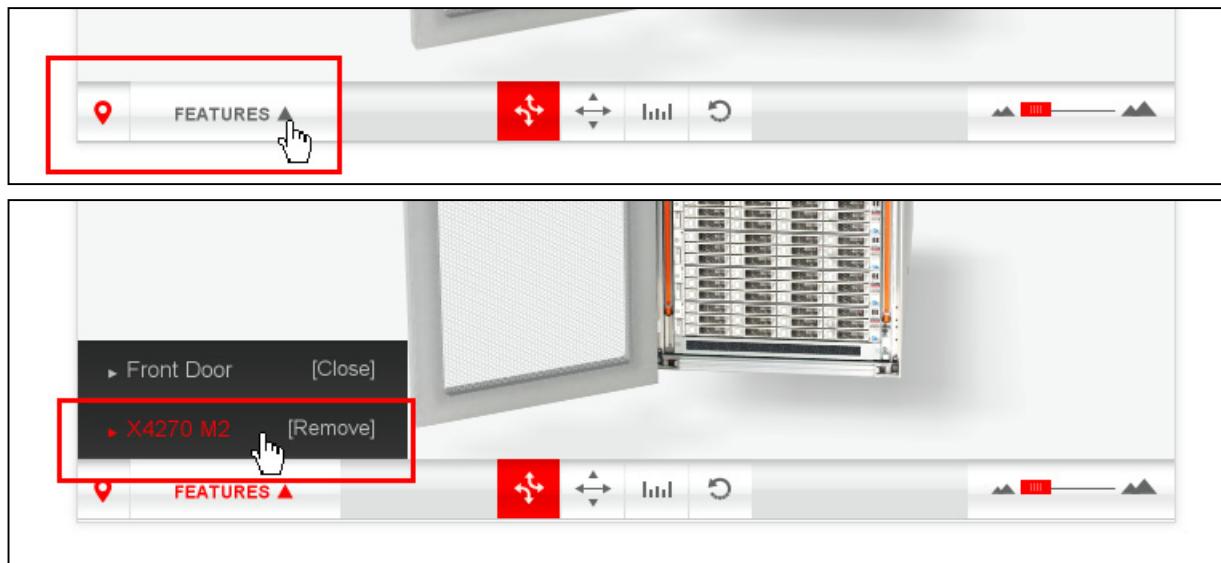


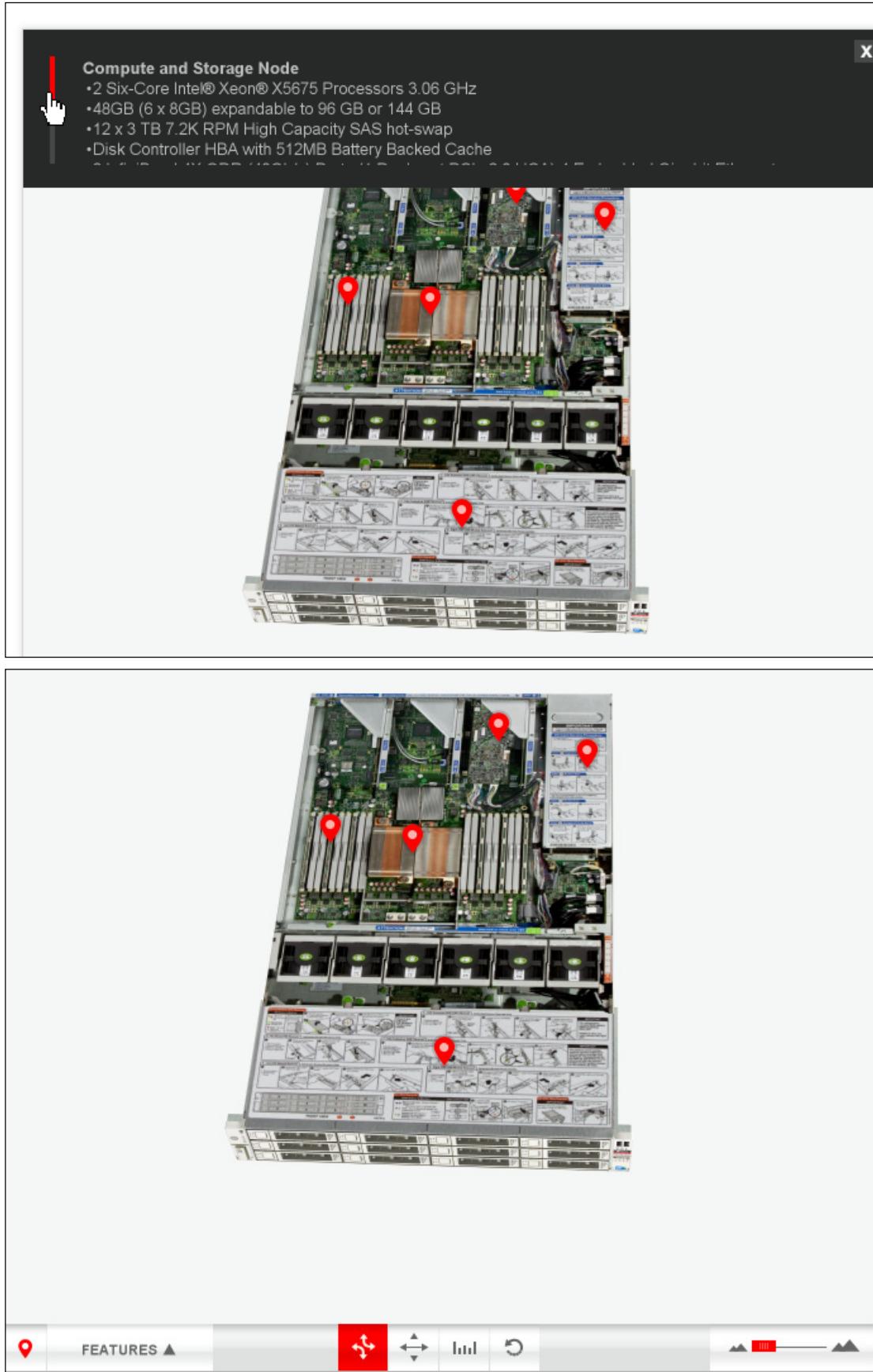


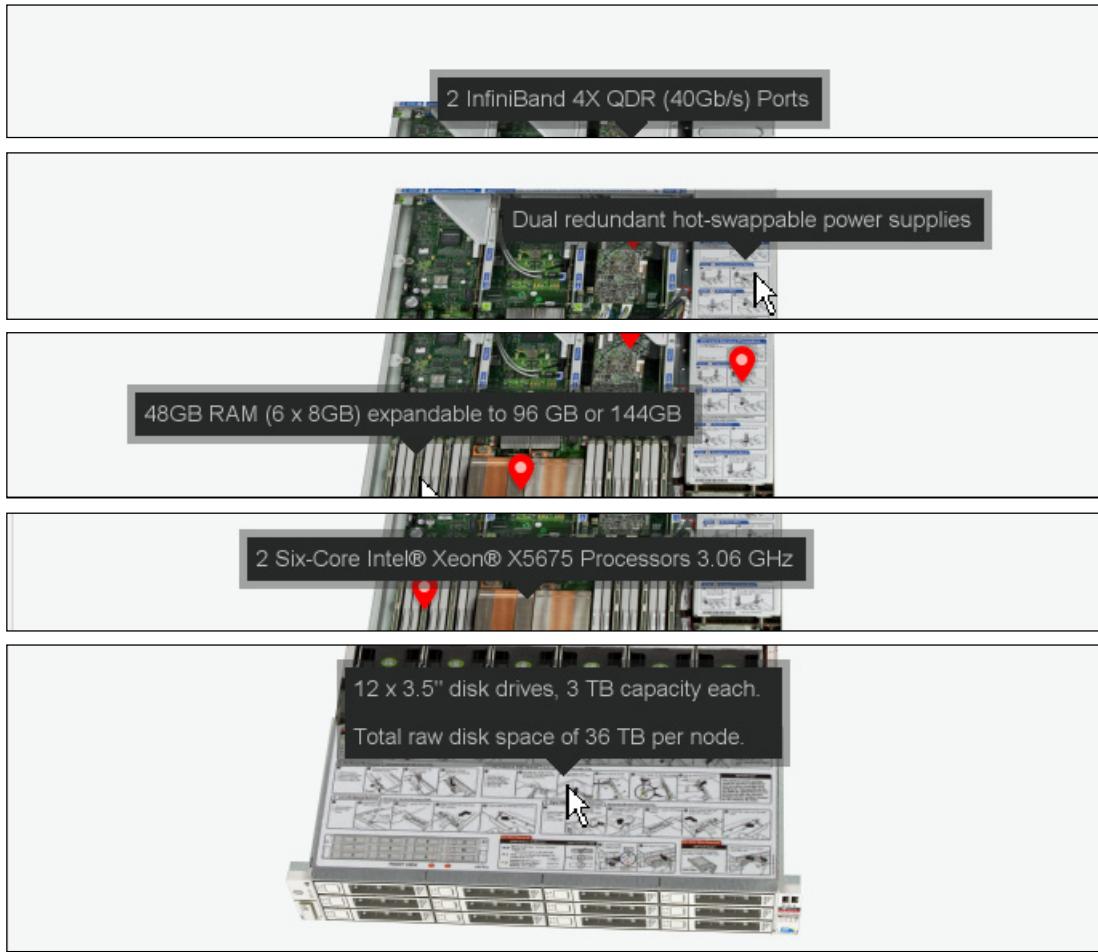
- d. Explore the tools at the bottom of the page.



6. Click the black triangle next to **FEATURES** and then click the **X4270M2** link. Explore the components as in the previous steps. Note the red scroll bar to the right of the screen. Click the **x** control to exit the **Compute and Storage Node** window.









## **Practices for Lesson 26: Managing Oracle BDA**

**Chapter 26**

## Practices for Lesson 26

---

### Practices Overview

In these practices, you will examine the ways to:

- Monitor MapReduce jobs
- Monitor the health of HDFS
- Use the Hive Query Editor (HUE)

## Guided Practice 26-1: Monitoring MapReduce Jobs

### Overview

In this practice, you will monitor the MapReduce jobs that are executed by CDH.

### Assumptions

HDFS service is started and running.

### Tasks

- To monitor MapReduce jobs, open your web browser and enter the URL as shown below:

<http://localhost:8088>

**Note:** In the above URL:

- localhost is the machine where the YARN resource manager runs. Instead of localhost, you can also provide the machine name.
- 8088 is the default port number for the user interface

The screenshot shows the 'All Applications' page of the Cloudera Manager interface. The left sidebar has a 'Cluster Metrics' section with tabs for 'About', 'Nodes', and 'Applications'. Under 'Applications', there are sub-tabs for 'Scheduler' and 'Tools'. The main content area displays a table of completed MapReduce jobs. The columns include: ID, User, Name, Application Type, Queue, StartTime, FinishTime, State, FinalStatus, Progress, and a 'History' link. The table lists six completed jobs, each with a unique ID, user (oracle), job name, application type (MAPREDUCE), queue (root.oracle), start and finish times (e.g., Wed, 08 Apr 2015 11:14:05 GMT), state (FINISHED), final status (SUCCEEDED), progress (100%), and a history link.

ID	User	Name	Application Type	Queue	StartTime	FinishTime	State	FinalStatus	Progress	History
application_1427904466940_0045	oracle	select * from movieapp.log_json where ra...4(Stage-1)	MAPREDUCE	root.oracle	Wed, 08 Apr 2015 11:14:05 GMT	Wed, 08 Apr 2015 11:14:16 GMT	FINISHED	SUCCEEDED	100%	<a href="#">History</a>
application_1427904466940_0044	oracle	insert overwrite table default.movie...MOVIE_1(Stage-1)	MAPREDUCE	root.oracle	Wed, 08 Apr 2015 10:41:56 GMT	Wed, 08 Apr 2015 10:42:05 GMT	FINISHED	SUCCEEDED	100%	<a href="#">History</a>
application_1427904466940_0043	oracle	QueryResult.jar	MAPREDUCE	root.oracle	Wed, 08 Apr 2015 10:41:44 GMT	Wed, 08 Apr 2015 10:41:53 GMT	FINISHED	SUCCEEDED	100%	<a href="#">History</a>
application_1427904466940_0042	oracle	SELECT count(*) FROM movie_fact(Stage-1)	MAPREDUCE	root.oracle	Wed, 08 Apr 2015 08:47:21 GMT	Wed, 08 Apr 2015 08:47:39 GMT	FINISHED	SUCCEEDED	100%	<a href="#">History</a>
application_1427904466940_0041	oracle	OraLoader	MAPREDUCE	root.oracle	Wed, 08 Apr 2015 06:41:36 GMT	Wed, 08 Apr 2015 06:42:02 GMT	FINISHED	SUCCEEDED	100%	<a href="#">History</a>
application_1427904466940_0040	oracle	select count(*) from moviedemo.movieapp...2(Stage-1)	MAPREDUCE	root.oracle	Wed, 08 Apr 2015 06:28:08 GMT	Wed, 08 Apr 2015 06:28:21 GMT	FINISHED	SUCCEEDED	100%	<a href="#">History</a>

**Note:** The number of rows in the application might vary.

- Click the About link in the left pane under the Cluster drop-down list. This link has information about the cluster.

**About the Cluster**

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	Active Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes
45	0	0	45	0	0 B	4 GB	0 B	1	0	0	0	0

User Metrics for dr.who

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Containers Pending	Containers Reserved	Memory Used	Memory Pending	Memory Reserved
0	0	0	45	0	0	0	0 B	0 B	0 B

Cluster overview

Cluster ID: 1427904466940  
**ResourceManager state:** STARTED  
**ResourceManager HA state:** active  
**ResourceManager started on:** 1-Apr-2015 12:07:46  
**ResourceManager version:** 2.3.0-cdh5.1.2 from 8e266e052e423af592871e2dfe09d54c03f6a0e8 by Jenkins source checksum 49178e87abfc66a38a85aa4bb2dd21b on 2014-08-26T01:44Z  
**Hadoop version:** 2.3.0-cdh5.1.2 from 8e266e052e423af592871e2dfe09d54c03f6a0e8 by Jenkins source checksum ec11b8ec19ca2bf3e7cb1be4ee182 on 2014-08-26T01:36Z

- Click the Nodes link in the left pane under the Cluster drop-down list. This link has information about the various nodes of the cluster.

**Nodes of the cluster**

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	Active Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes
45	0	0	45	0	0 B	4 GB	0 B	1	0	0	0	0

User Metrics for dr.who

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Containers Pending	Containers Reserved	Memory Used	Memory Pending	Memory Reserved
0	0	0	45	0	0	0	0 B	0 B	0 B

Show 20 - entries

Rack	Node State	Node Address	Node HTTP Address	Last health-update	Health-report	Containers	Mem Used	Mem Avail	Version
/default-rack	RUNNING	bigdatalite.localdomain:36705	bigdatalite.localdomain:8042	9-Apr-2015 05:33:40		0	0 B	4 GB	2.3.0-cdh5.1.2

Showing 1 to 1 of 1 entries

**Note:** BigDataLite is usually deployed on a single cluster, so this page will show details of a single cluster. An actual BDA is usually deployed on a multiple cluster.

- Under the Tools drop-down on the left pane, click Local Logs. This has the link to all the logs. You can also click any of the links in the logs directory to view the corresponding logs.

**Directory: /logs/**

container/_	4096 bytes Aug 12, 2014 2:50:04 PM
containers/_	4096 bytes Apr 8, 2015 7:14:24 AM
userlogs/_	4096 bytes Aug 14, 2014 9:28:09 AM
yarn-yarn-nodemanager-bigdatalite.localdomain.log	2198974 bytes Apr 8, 2015 12:07:47 PM
yarn-yarn-nodemanager-bigdatalite.localdomain.out	703 bytes Apr 1, 2015 12:07:40 PM
yarn-yarn-nodemanager-bigdatalite.localdomain.out.1	703 bytes Apr 1, 2015 11:53:26 AM
yarn-yarn-nodemanager-bigdatalite.localdomain.out.2	703 bytes Mar 24, 2015 12:48:17 PM
yarn-yarn-nodemanager-bigdatalite.localdomain.out.3	703 bytes Mar 24, 2015 12:33:54 PM
yarn-yarn-nodemanager-bigdatalite.localdomain.out.4	703 bytes Feb 23, 2015 2:38:40 PM
yarn-yarn-nodemanager-bigdatalite.localdomain.out.5	703 bytes Feb 23, 2015 1:14:19 PM
yarn-yarn-resourcemanager-bigdatalite.localdomain.log	1343301 bytes Apr 9, 2015 5:36:11 AM
yarn-yarn-resourcemanager-bigdatalite.localdomain.out	2078 bytes Apr 1, 2015 3:08:57 PM
yarn-yarn-resourcemanager-bigdatalite.localdomain.out.1	703 bytes Apr 1, 2015 11:53:32 AM
yarn-yarn-resourcemanager-bigdatalite.localdomain.out.2	703 bytes Mar 24, 2015 12:48:23 PM
yarn-yarn-resourcemanager-bigdatalite.localdomain.out.3	703 bytes Mar 24, 2015 12:34:00 PM
yarn-yarn-resourcemanager-bigdatalite.localdomain.out.4	2078 bytes Feb 24, 2015 6:02:09 AM
yarn-yarn-resourcemanager-bigdatalite.localdomain.out.5	703 bytes Feb 23, 2015 1:14:25 PM

## Guided Practice 26-2: Monitoring the Health of HDFS

### Overview

In this practice, you will monitor the health of the HDFS.

### Assumptions

HDFS service is started and running.

### Tasks

1. You can monitor the health of the Hadoop file system by using the DFS health utility.
2. To monitor the health of the HDFS, open your web browser and enter the URL as shown below:

<http://localhost:50070>

**Note:** In the above URL:

- localhost is the machine where the YARN resource manager runs. Instead of localhost, you can also provide the machine name.
- 50070 is the default port number for the user interface.

Started: Wed Apr 01 12:07:15 EDT 2015  
Version: 2.3.0-cdh5.1.2, r8e266e052e423a5f92871e2dfe09d54c03f6a0e8  
Compiled: 2014-08-26T01:36Z by Jenkins from (no branch)  
Cluster ID: CID-7c49dbeb-8c26-4a7d-b4c2-72b4a05268b9  
Block Pool ID: BP-703742109-127.0.0.1-1398459391664

3. Scroll down the Overview page to view summary and details about the name node.

## Summary

Security is off.  
Safemode is off.  
1992 files and directories, 1141 blocks = 3133 total filesystem object(s).  
Heap Memory used 80.64 MB of 151 MB Heap Memory. Max Heap Memory is 889 MB.  
Non Heap Memory used 32.57 MB of 32.69 MB Committed Non Heap Memory. Max Non Heap Memory is 130 MB.

Configured Capacity:	98.43 GB
DFS Used:	951.69 MB
Non DFS Used:	5.96 GB
DFS Remaining:	91.54 GB
DFS Used%:	0.94%
DFS Remaining%:	93%
Block Pool Used:	951.69 MB
Block Pool Used%:	0.94%
DataNodes usages% (Min/Median/Max/stdDev):	0.94% / 0.94% / 0.94% / 0.00%
Live Nodes	1 (Decommissioned: 0)
Dead Nodes	0 (Decommissioned: 0)
Decommissioning Nodes	0
Number of Under-Replicated Blocks	124
Number of Blocks Pending Deletion	0

## NameNode Journal Status

Current transaction ID: 100143

Journal Manager	State
FileJournalManager(root=/u02/dfs/nn)	EditLogFileOutputStream(/u02/dfs/nn/current/edits_inprogress_000000000000100143)

## NameNode Storage

Storage Directory	Type	State
/u02/dfs/nn	IMAGE_AND_EDITS	Active

Hadoop, 2014. Legacy UI

4. Click the DataNodes tab. This tab has information about the data node.

Hadoop Overview Datanodes Snapshot Startup Progress Utilities -

## Datanode Information

In operation

Node	Last contact	Admin State	Capacity	Used	Non DFS Used	Remaining	Blocks	Block pool used	Failed Volumes	Version
bigdatalite.localdomain (127.0.0.1:50010)	1	In Service	98.43 GB	951.69 MB	5.96 GB	91.54 GB	1139	951.69 MB (0.94%)	0	2.3.0-cdh5.1.2

Decommissioning

Node	Last contact	Under replicated blocks	Blocks with no live replicas	Under Replicated Blocks In files under construction
------	--------------	-------------------------	------------------------------	--------------------------------------------------------

Hadoop, 2014. Legacy UI

5. Click the Snapshot tab. This tab has information about the snapshots, if any.

**Snapshot Summary**

Snapshottable directories: 0

Path	Snapshot Number	Snapshot Quota	Modification Time	Permission	Owner	Group
------	-----------------	----------------	-------------------	------------	-------	-------

Snapshotted directories: 0

Snapshot ID	Snapshot Directory	Modification Time
-------------	--------------------	-------------------

Hadoop, 2014. [Legacy UI](#)

- Click the Startup Progress tab. This tab has information about the startup of the machine.

**Startup Progress**

Elapsed Time: 0 sec, Percent Complete: 100%

Phase	Completion	Elapsed Time
Loading fsimage /u02/dfs/nn/current/fsimage_000000000000088172 129.39 KB	100%	0 sec
inodes (0/0)	100%	
delegation tokens (0/0)	100%	
cache pools (0/0)	100%	
Loading edits	100%	0 sec
/u02/dfs/nn/current/editds_0000000000000088173-0000000000000088194 1 MB (22/22)	100%	
Saving checkpoint	100%	0 sec
Safe mode	100%	0 sec
awaiting reported blocks (0/917)	100%	

Hadoop, 2014. [Legacy UI](#)

- On the Utilities tab, click the “Browse the file system” link. Using this webpage, you can view and browse the file system. Enter / in the input box and click Go! The root directory of the file systems will be displayed. You can click the various Name links to get details about the corresponding directories and files under the link.

**Browse Directory**

/

Permission	Owner	Group	Size	Replication	Block Size	Name
drwxr-xr-x	hbase	hadoop	0 B	0	0 B	hbase
drwxrwxrwx	hdfs	hadoop	0 B	0	0 B	share
drwxr-xr-x	solr	solr	0 B	0	0 B	solr
drwxrwxrwt	hdfs	hadoop	0 B	0	0 B	tmp
drwxr-xr-x	hdfs	hadoop	0 B	0	0 B	user
drwxr-xr-x	hdfs	hadoop	0 B	0	0 B	var

Hadoop, 2014.

## Guided Practice 26-3: Using HUE

### Overview

In this practice, you will use the Hive Query Editor (HUE).

### Assumptions

Hive and HUE services are started and running.

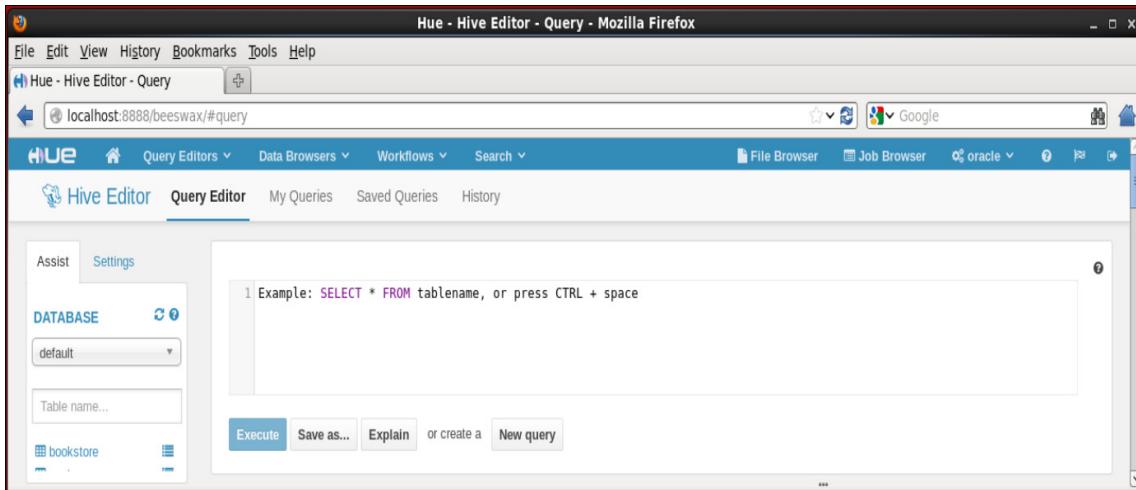
### Tasks

1. HUE enables developers to type and execute HiveQL queries using the query editor.
2. To use HUE, open your web browser and enter the URL as shown below:

<http://localhost:8888/beeswax/#query>

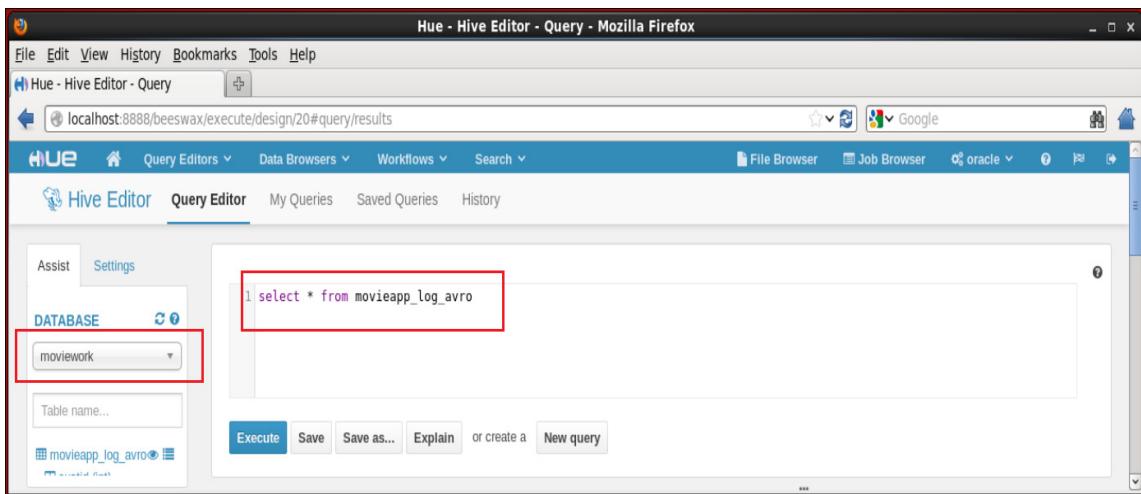
**Note:** In the above URL:

- localhost is the machine where the YARN resource manager runs. Instead of localhost, you can also provide the machine name.
- 8888 is the default port number for the user interface.

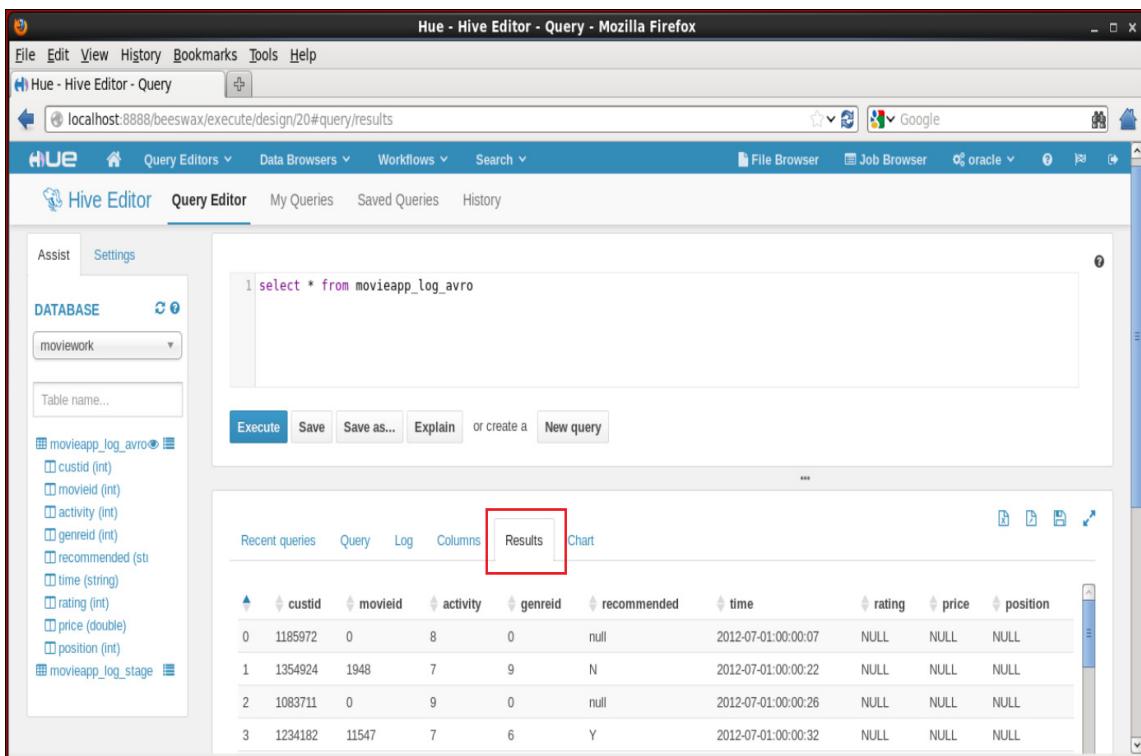


**Note:** Enter the credentials and sign in to HUE, if not already signed in. For credentials refer to the <file:///home/oracle/GettingStarted/StartHere.html> link.

3. Select the appropriate database from the drop-down list on the left menu and type the query in the text area.



- Click the Execute button. The results will be shown on the Results tab as shown below.





## **Practices for Lesson 27: Balancing MapReduce Jobs**

**Chapter 27**

## Practices for Lesson 27

---

### Practices Overview

In this practice, you will use the Oracle BDA Perfect Balance feature and generate and view some reports.

## Practice 27-1: Balancing MapReduce Jobs

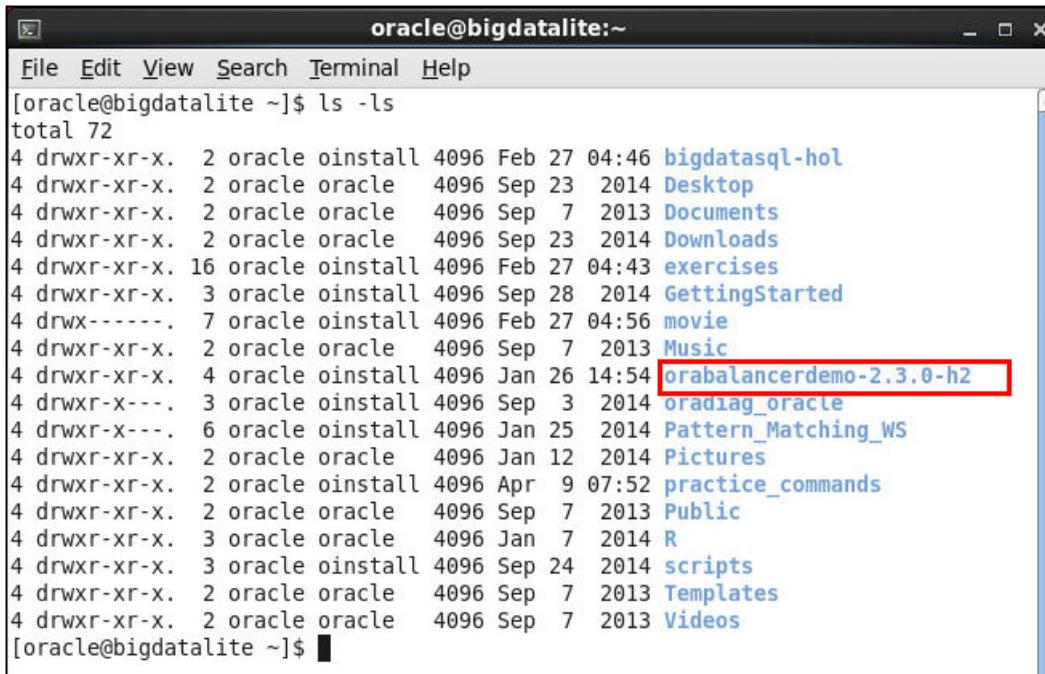
### Overview

In this practice, you use the Oracle BDA Perfect Balance feature. This practice uses some of the MoviePlex application data.

### Assumptions

#### Tasks

1. The `/oracle/home/orabalancerdemo-2.3.0-h2` directory contains the files that you will use in this practice.



```
oracle@bigdatalite:~$ ls -ls
total 72
4 drwxr-xr-x. 2 oracle oinstall 4096 Feb 27 04:46 bigdatasql-hol
4 drwxr-xr-x. 2 oracle oracle 4096 Sep 23 2014 Desktop
4 drwxr-xr-x. 2 oracle oracle 4096 Sep 7 2013 Documents
4 drwxr-xr-x. 2 oracle oracle 4096 Sep 23 2014 Downloads
4 drwxr-xr-x. 16 oracle oinstall 4096 Feb 27 04:43 exercises
4 drwxr-xr-x. 3 oracle oinstall 4096 Sep 28 2014 GettingStarted
4 drwxr----- 7 oracle oinstall 4096 Feb 27 04:56 movie
4 drwxr-xr-x. 2 oracle oracle 4096 Sep 7 2013 Music
4 drwxr-xr-x. 4 oracle oinstall 4096 Jan 26 14:54 orabalancerdemo-2.3.0-h2
4 drwxr-xr-x. 3 oracle oinstall 4096 Sep 3 2014 oradiag_oracle
4 drwxr-xr-x. 6 oracle oinstall 4096 Jan 25 2014 Pattern_Matching_WS
4 drwxr-xr-x. 2 oracle oracle 4096 Jan 12 2014 Pictures
4 drwxr-xr-x. 2 oracle oinstall 4096 Apr 9 07:52 practice_commands
4 drwxr-xr-x. 2 oracle oracle 4096 Sep 7 2013 Public
4 drwxr-xr-x. 3 oracle oracle 4096 Jan 7 2014 R
4 drwxr-xr-x. 3 oracle oinstall 4096 Sep 24 2014 scripts
4 drwxr-xr-x. 2 oracle oracle 4096 Sep 7 2013 Templates
4 drwxr-xr-x. 2 oracle oracle 4096 Sep 7 2013 Videos
[oracle@bigdatalite ~]$
```

2. Set `BALANCERDEMO_HOME` to make it easier to use the resources in the newly created `orabalancerdemo-2.3.0-h2` directory.

**Note:** For all the steps to work successfully in this practice, make sure you issue all the commands in this practice in the same terminal window.

```
BALANCERDEMO_HOME=/home/oracle/orabalancerdemo-2.3.0-h2
```

```
INITIATING: orabalancerdemo-2.3.0-h2/tld/orabalancerdemo-2.3.0.tar
[oracle@bigdatalite ~]$ BALANCERDEMO_HOME=/home/oracle/orabalancerdemo-2.3.0-h2
[oracle@bigdatalite ~]$
```

3. Copy the practice data to HDFS as follows:

```
cd $BALANCERDEMO_HOME/examples/movie
hadoop fs -put movieData
```

```
oracle@bigdatalite:~/orabalancerdemo-2.3.0-h2/examples/movie
File Edit View Search Terminal Help
[oracle@bigdatalite movie]$ ls -ls
total 3468
 84 -r--r--r--. 1 oracle oinstall    85338 Jan 26 14:54 HandsOnLab.pdf
  4 drwxr-xr-x. 2 oracle oinstall    4096 Mar 27 03:03 movieData
3360 -rw-r--r--. 1 oracle oinstall 3439263 Jan 26 14:54 movieData.zip
  8 -rwxr-xr-x. 1 oracle oinstall    6799 Jan 26 14:54 moviedemo
 12 -rw-r--r--. 1 oracle oinstall   9658 Jan 26 14:54 README.txt
[oracle@bigdatalite movie]$
```

```
[oracle@bigdatalite movie]$ hadoop fs -put movieData
[oracle@bigdatalite movie]$
```

4. Verify that the movieData data file is copied successfully to HDFS.

```
hadoop fs -ls
hadoop fs -ls /user/oracle/movieData
```

**Note:** Your output might look different than what is shown in the following partial screen capture.

```
[oracle@bigdatalite movie]$ hadoop fs -ls
Found 15 items
drwxr-xr-x  - oracle oracle      0 2015-04-14 05:05 .
drwx----- - oracle oracle      0 2014-08-25 05:55 .Trash
drwx----- - oracle oracle      0 2015-04-15 02:39 .staging
drwxr-xr-x  - oracle oracle      0 2015-04-14 05:12 insur_cust_ltv_sample
drwxr-xr-x  - oracle oracle      0 2015-04-15 08:10 movieData
drwxr-xr-x  - oracle oracle      0 2015-04-14 06:49 movie_Tact
drwxr-xr-x  - oracle oracle      0 2015-04-14 06:50 movie_fact_query
drwxr-xr-x  - oracle oracle      0 2014-01-12 18:15 moviedemo
drwxr-xr-x  - oracle oracle      0 2014-09-24 09:38 moviework
drwxr-xr-x  - oracle oracle      0 2014-09-08 15:50 oggdemo
drwxr-xr-x  - oracle oracle      0 2015-04-14 06:12 olbcache
```

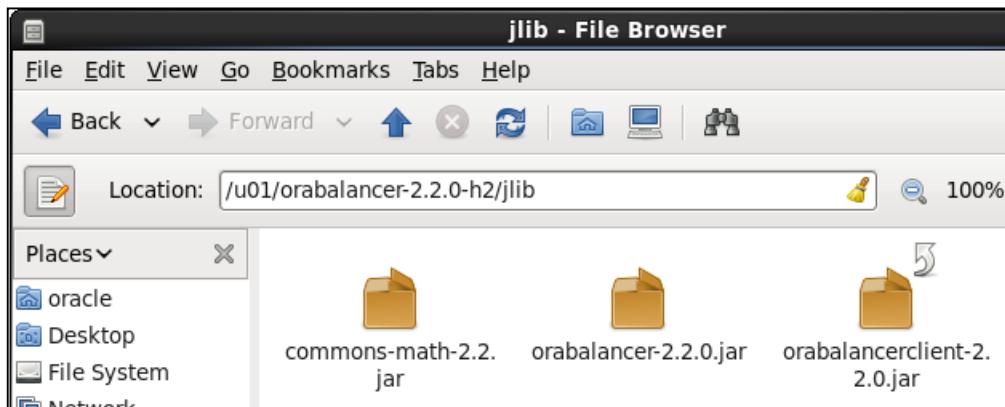
```
[oracle@bigdatalite movie]$ hadoop fs -ls /user/oracle/movieData
Found 5 items
-rw-r--r--  1 oracle oracle  12737318 2015-02-20 06:51 /user/oracle/movieData/
part-00001
-rw-r--r--  1 oracle oracle     438 2015-02-20 06:51 /user/oracle/movieData/
part-00002
-rw-r--r--  1 oracle oracle     432 2015-02-20 06:51 /user/oracle/movieData/
part-00003
-rw-r--r--  1 oracle oracle     202 2015-02-20 06:51 /user/oracle/movieData/
part-00004
-rw-r--r--  1 oracle oracle  2522962 2015-02-20 06:51 /user/oracle/movieData/
part-00005
[oracle@bigdatalite movie]$
```

5. Set the environment variables in order to run Perfect Balance and Job Analyzer.
  - a. Set `BALANCER_HOME` to make it easier for you to use the resources that are available in the `orabalancer`.

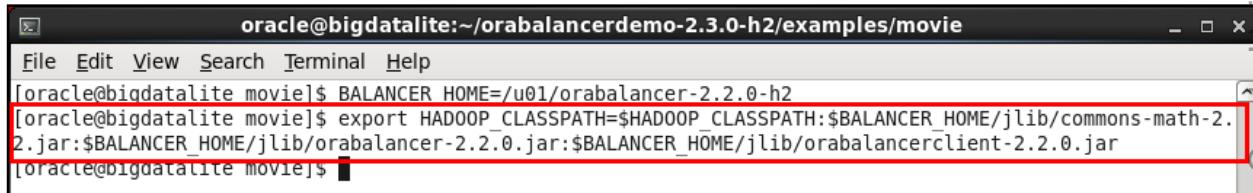
```
BALANCER_HOME=/u01/orabalancer-2.2.0-h2
```



- b. Change `HADOOP_CLASSPATH` so that it includes the JAR file required by Perfect Balance.



```
export  
HADOOP_CLASSPATH=$HADOOP_CLASSPATH:$BALANCER_HOME/jlib/commons-  
math-2.2.jar:$BALANCER_HOME/jlib/orabalancer-  
2.2.0.jar:$BALANCER_HOME/jlib/orabalancerclient-2.2.0.jar
```



- c. The JAR files that are required by Perfect Balance should be given precedence over the Hadoop JAR files. To enable this, set `HADOOP_USER_CLASSPATH_FIRST` to true.

```
export HADOOP_USER_CLASSPATH_FIRST=true
```

```
oracle@bigdatalite:~/orabalancerdemo-2.3.0-h2/examples/movie
File Edit View Search Terminal Help
[oracle@bigdatalite movie]$ BALANCER_HOME=/u01/orabalancer-2.2.0-h2
[oracle@bigdatalite movie]$ export HADOOP_CLASSPATH=$HADOOP_CLASSPATH:$BALANCER_HOME/jlib/commons-math-2.2.jar:$BALANCER_HOME/jlib/orabalancer-2.2.0.jar:$BALANCER_HOME/jlib/orabalancerclient-2.2.0.jar
[oracle@bigdatalite movie]$ export HADOOP_USER_CLASSPATH_FIRST=true
[oracle@bigdatalite movie]$
```

6. Run the MovieDemo MapReduce job, which clusters movie-ids by genre\_id. Applications can use the results of this MapReduce job to quickly access movie-ids for a particular genre\_id. After you run the job, note the job ID and the start of the map and reduce tasks.

```
hadoop jar $BALANCERDEMO_HOME/jlib/orabalancerdemo-2.3.0.jar
oracle.hadoop.balancer.examples.movie.MovieGenre -D
mapred.reduce.tasks=5 -D mapred.input.dir=movieData -D
mapred.output.dir=moviegenre_output
```

```
15/01/28 11:40:35 INFO client.RMProxy: Connecting to ResourceManager at localhost/127.0.0.1:8032
15/01/28 11:40:37 INFO input.FileInputFormat: Total input paths to process : 5
15/01/28 11:40:37 INFO mapreduce.JobSubmitter: number of splits:5
15/01/28 11:40:37 INFO Configuration.deprecation: mapred.reduce.tasks is deprecated. Instead, use mapreduce.job.reduces
15/01/28 11:40:37 INFO Configuration.deprecation: mapred.input.dir is deprecated. Instead, use mapreduce.input.fileinputformat.dir
15/01/28 11:40:37 INFO Configuration.deprecation: mapred.output.dir is deprecated. Instead, use mapreduce.output.fileoutputformat.dir
15/01/28 11:40:37 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1420735348738_0016
15/01/28 11:40:37 INFO impl.YarnClientImpl: Submitted application application_1420735348738_0016
15/01/28 11:40:37 INFO mapreduce.Job: The url to track the job: http://bigdatalite.localdomain:8088/proxy/application_1420735348738_0016/
15/01/28 11:40:37 INFO mapreduce.Job: Running job: job_1420735348738_0016
15/01/28 11:40:44 INFO mapreduce.Job: Job job_1420735348738_0016 running in uber mode : false
15/01/28 11:40:44 INFO mapreduce.Job: map 0% reduce 0%
15/01/28 11:40:56 INFO mapreduce.Job: map 60% reduce 0%
15/01/28 11:41:03 INFO mapreduce.Job: map 100% reduce 0%
15/01/28 11:41:06 INFO mapreduce.Job: map 100% reduce 20%
15/01/28 11:41:12 INFO mapreduce.Job: map 100% reduce 40%
15/01/28 11:41:14 INFO mapreduce.Job: map 100% reduce 60%
15/01/28 11:41:16 INFO mapreduce.Job: map 100% reduce 80%
15/01/28 11:41:19 INFO mapreduce.Job: map 100% reduce 100%
15/01/28 11:41:19 INFO mapreduce.Job: Job job_1420735348738_0016 completed successfully
15/01/28 11:41:19 INFO mapreduce.Job: Counters: 49
    File System Counters
        FILE: Number of bytes read=3590500
        FILE: Number of bytes written=8102395
        FILE: Number of read operations=0
        FILE: Number of large read operations=0
        FILE: Number of write operations=0
        HDFS: Number of bytes read=15262017
        HDFS: Number of bytes written=1860178
        HDFS: Number of read operations=30
        HDFS: Number of large read operations=0
        HDFS: Number of write operations=10
```

**Note:** Your results will show a different job ID and different application ID than the preceding screen captures.

```

Job Counters
Launched map tasks=5
Launched reduce tasks=5
Data-local map tasks=5
Total time spent by all maps in occupied slots (ms)=39560
Total time spent by all reduces in occupied slots (ms)=39406
Total time spent by all map tasks (ms)=39560
Total time spent by all reduce tasks (ms)=39406
Total vcore-seconds taken by all map tasks=39560
Total vcore-seconds taken by all reduce tasks=39406
Total megabyte-seconds taken by all map tasks=10127360
Total megabyte-seconds taken by all reduce tasks=10087936

Map-Reduce Framework
Map input records=359047
Map output records=359047
Map output bytes=2872376
Map output materialized bytes=3590620
Input split bytes=665
Combine input records=0
Combine output records=0
Reduce input groups=28
Reduce shuffle bytes=3590620
Reduce input records=359047
Reduce output records=3608
Spilled Records=718094
Shuffled Maps =25
Failed Shuffles=0
Merged Map outputs=25
GC time elapsed (ms)=773
CPU time spent (ms)=13390
Physical memory (bytes) snapshot=2168471552
Virtual memory (bytes) snapshot=8844177408
Total committed heap usage (bytes)=1891631104

```

```

Shuffle Errors
BAD_ID=0
CONNECTION=0
IO_ERROR=0
WRONG_LENGTH=0
WRONG_MAP=0
WRONG_REDUCE=0
File Input Format Counters
Bytes Read=15261352
File Output Format Counters
Bytes Written=1860178
[oracle@bigdatalite movie]$ ^C
[oracle@bigdatalite movie]$ █

```

## Running a MapReduce Job with Perfect Balance

- In this section, you will first run Job Analyzer on the log files of a completed MapReduce job. This helps you identify whether or not there is skew in the data. You will run Job Analyzer on the job that was completed in Step 6. On a Yarn cluster, you need the job ID for the Job Analyzer to identify the job logs. You can find the job ID in YARN or in Cloudera Manager. Both list the jobs you ran previously. In this practice, you just ran this job in step 6; so you can scroll up the terminal window screen and copy the job ID from the terminal output. You should see a line that looks similar to: "mapreduce.Job : Running job: JOB\_ID". For example, if the job ID is job\_1420735348738\_0016, the command to get a Job Analyzer report is the following:

**Note:** Your job ID # for the MapReduce job that you just ran in step 6 will be different than what is shown here. Job IDs are unique. If you re-run the same command

in practice 7, you will get a new job ID number. Replace the job\_1420735348738\_0016 shown in the following command to match your own job ID number.

```
hadoop jar $BALANCER_HOME/jlib/orabalancer-2.2.0.jar
oracle.hadoop.balancer.tools.JobAnalyzer -D
oracle.hadoop.balancer.application_id=job_1420735348738_0016
```

The preceding command runs the Job Analyzer on the logs generated by the MapReduce job. It creates a \_balancer directory in the job output directory.

```
hadoop fs -ls moviegenre_output
hadoop fs -ls moviegenre_output/_balancer
```

**Note:** Your output might look different than what is shown in the following partial screen capture.

```
[oracle@bigdatalite movie]$ hadoop fs -ls moviegenre_output
Found 7 items
-rw-r--r-- 1 oracle oracle          0 2015-02-20 07:00 moviegenre_output/ SUCCESS
drwxr-xr-x - oracle oracle          0 2015-02-20 07:07 moviegenre_output/_balancer
-rw-r--r-- 1 oracle oracle 392653 2015-02-20 06:59 moviegenre_output/part-r-00000
-rw-r--r-- 1 oracle oracle 399611 2015-02-20 06:59 moviegenre_output/part-r-00001
-rw-r--r-- 1 oracle oracle 276328 2015-02-20 07:00 moviegenre_output/part-r-00002
-rw-r--r-- 1 oracle oracle 423957 2015-02-20 07:00 moviegenre_output/part-r-00003
-rw-r--r-- 1 oracle oracle 367629 2015-02-20 07:00 moviegenre_output/part-r-00004
[oracle@bigdatalite movie]$ hadoop fs -ls moviegenre_output/_balancer
Found 3 items
-rw-r--r-- 1 oracle oracle          23 2015-02-20 07:00 moviegenre_output/_balancer/application_id
-rw-r--r-- 1 oracle oracle         7146 2015-02-20 07:07 moviegenre_output/_balancer/jobanalyzer-report.html
-rw-r--r-- 1 oracle oracle        12227 2015-02-20 07:07 moviegenre_output/_balancer/jobanalyzer-report.xml
```

**Note:** Your job IDs will be different than what is shown in this practice and the generated reports.

8. Copy and view the generated Job Analyzer report, step8-jobanalyzer-report.html, from HDFS to your local file system as follows:

```
hadoop fs -get moviegenre_output/_balancer/jobanalyzer-
report.html step8-jobanalyzer-report.html
```

```
$ hadoop fs -get moviegenre_output/_balancer/jobanalyzer-report.html step8-jobanalyzer-report.html
$
```

```
[oracle@bigdatalite movie]$ ls -l
total 3476
-r--r--r--. 1 oracle oinstall 85338 Jan 26 14:54 HandsOnLab.pdf
drwxr-xr-x. 2 oracle oinstall 4096 Feb 20 06:50 movieData
-rw-r--r--. 1 oracle oinstall 3439263 Jan 26 14:54 movieData.zip
-rw xr-x. 1 oracle oinstall 6799 Jan 26 14:54 moviedemo
-rw-r--r--. 1 oracle oinstall 9658 Jan 26 14:54 README.txt
-rw-r--r--. 1 oracle oinstall 7146 Feb 20 07:13 step8-jobanalyzer-report.html
[oracle@bigdatalite movie]$ pwd
/home/oracle/orabalancerdemo-2.3.0-h2/examples/movie
```

```
firefox step8-jobanalyzer-report.html
```

```
[oracle@bigdatalite movie]$ hadoop fs -get moviegenre output/ bala
[oracle@bigdatalite movie] $ firefox step8-jobanalyzer-report.html
```

The report is displayed as shown in the following partial screen capture.

**Note:** Your report information might look different.

**Job Information**

Job Name	orabalancerdemo-2.3.0.jar
Job Id	job_1420735348738_0016
Start Time	2015-01-28 11:40:42
Finish Time	2015-01-28 11:41:18

**Time Information**

Map Phase	00:00:17
Reduce Phase	00:00:21
Shuffle	00:00:20
Merge	00:00:14
Reduce	00:00:14
Job	00:00:35

**Reduce Tasks Metrics Summary**

Task ID	Time		
	Start	Finish	Elapsed
0	11:40:56	11:41:04	00:00:08
1	11:41:02	11:41:11	00:00:09
2	11:41:03	11:41:13	00:00:09
3	11:41:05	11:41:15	00:00:09
4	11:41:12	11:41:18	00:00:05

**Reduce Tasks Metrics**

Task ID	Elapsed Time			Input				Output			
	Shuffle	Merge	Reduce	Shuffle Bytes	Keys	Records	Records	Bytes			
	count	%	count	%	count	%	count	%			
0	753,360	21	6	21	75,333	21	757	21	392,653	21	
1	765,260	21	6	21	76,523	21	769	21	399,611	21	
2	522,620	15	5	18	52,259	15	526	15	276,328	15	
3	903,540	25	5	18	90,351	25	906	25	423,957	23	
4	645,840	18	6	21	64,581	18	650	18	367,629	20	
Total	00:00:20	00:00:14	00:00:14	3,590,620	-	28	-	359,047	-	1,860,178	-
Average	00:00:06	00:00:00	00:00:01	718,124	-	6	-	71,809	-	372,036	-

The Job Analyzer gathers information about the job and publishes this report. The Reduce Tasks Metrics Summary table includes elapsed times of each Reduce task. This is a **BASIC\_REPORT**.

**Note:** Close your web browser when you are done reviewing the report.

- In this step, you will enable the Job Analyzer during the MapReduce job run to get a **REDUCER\_REPORT**. A **REDUCER\_REPORT** includes information about the number of rows processed by each Reduce task. When you enable the Job Analyzer during the job run, more information is available to the Job Analyzer. This is reflected in the **REDUCER\_REPORT**.

**Note:** The following code is one long command. To ensure that there are no extra spaces in the command, copy the code from the `lab_27_01.txt` file in the `practice_commands` folder under the `/home/oracle` home directory, and then paste it into the command line in your terminal window, and then press the Enter key.

```
hadoop jar $BALANCERDEMO_HOME/jlib/orabalancerdemo-2.3.0.jar
oracle.hadoop.balancer.examples.movie.MovieGenre -D
mapred.reduce.tasks=5 -D mapred.input.dir=movieData -D
```

```
mapred.output.dir=moviegenre_report -D  
oracle.hadoop.balancer.autoAnalyze=REDUCER_REPORT -libjars  
$BALANCER_HOME/jlib/commons-math-  
2.2.jar,$BALANCER_HOME/jlib/orabalancer-2.2.0.jar
```

The preceding command creates a \_balancer directory in the job output directory while running the job.

```
15/01/28 12:15:47 INFO Configuration.deprecation: mapred.reduce.tasks is deprecated. Instead, use mapreduce.job.  
15/01/28 12:15:47 INFO Configuration.deprecation: mapred.input.dir is deprecated. Instead, use mapreduce.input.  
dir  
15/01/28 12:15:47 INFO Configuration.deprecation: mapred.output.dir is deprecated. Instead, use mapreduce.output.  
tputdir  
15/01/28 12:15:47 INFO client.RMProxy: Connecting to ResourceManager at localhost/127.0.0.1:8032  
15/01/28 12:15:48 INFO input.FileInputFormat: Total input paths to process : 5  
15/01/28 12:15:48 INFO mapreduce.JobSubmitter: number of splits:5  
15/01/28 12:15:48 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1420735348738_0017  
15/01/28 12:15:49 INFO impl.YarnClientImpl: Submitted application application_1420735348738_0017  
15/01/28 12:15:49 INFO mapreduce.Job: The url to track the job: http://bigdatalite.localdomain:8088/proxy/applic  
0017/  
15/01/28 12:15:49 INFO mapreduce.Job: Running job: job_1420735348738_0017  
15/01/28 12:15:56 INFO mapreduce.Job: Job job_1420735348738_0017 running in uber mode : false  
15/01/28 12:15:56 INFO mapreduce.Job: map 0% reduce 0%  
15/01/28 12:16:07 INFO mapreduce.Job: map 20% reduce 0%  
15/01/28 12:16:08 INFO mapreduce.Job: map 60% reduce 0%  
15/01/28 12:16:15 INFO mapreduce.Job: map 80% reduce 0%  
15/01/28 12:16:16 INFO mapreduce.Job: map 100% reduce 0%  
15/01/28 12:16:20 INFO mapreduce.Job: map 100% reduce 20%  
15/01/28 12:16:27 INFO mapreduce.Job: map 100% reduce 40%  
15/01/28 12:16:29 INFO mapreduce.Job: map 100% reduce 60%  
15/01/28 12:16:33 INFO mapreduce.Job: map 100% reduce 80%  
15/01/28 12:16:36 INFO mapreduce.Job: map 100% reduce 100%  
15/01/28 12:16:38 INFO mapreduce.Job: Job job_1420735348738_0017 completed successfully  
15/01/28 12:16:39 INFO mapreduce.Job: Counters: 49  
File System Counters  
FILE: Number of bytes read=3590500  
FILE: Number of bytes written=8124755  
FILE: Number of read operations=0  
FILE: Number of large read operations=0  
FILE: Number of write operations=0  
HDFS: Number of bytes read=15262017  
HDFS: Number of bytes written=1869765  
HDFS: Number of read operations=30  
HDFS: Number of large read operations=0  
HDFS: Number of write operations=15
```

```

Job Counters
    Launched map tasks=5
    Launched reduce tasks=5
    Data-local map tasks=5
    Total time spent by all maps in occupied slots (ms)=40866
    Total time spent by all reduces in occupied slots (ms)=49633
    Total time spent by all map tasks (ms)=40866
    Total time spent by all reduce tasks (ms)=49633
    Total vcore-seconds taken by all map tasks=40866
    Total vcore-seconds taken by all reduce tasks=49633
    Total megabyte-seconds taken by all map tasks=10461696
    Total megabyte-seconds taken by all reduce tasks=12706048

Map-Reduce Framework
    Map input records=359047
    Map output records=359047
    Map output bytes=2872376
    Map output materialized bytes=3590620
    Input split bytes=665
    Combine input records=0
    Combine output records=0
    Reduce input groups=28
    Reduce shuffle bytes=3590620
    Reduce input records=359047
    Reduce output records=3608
    Spilled Records=718094
    Shuffled Maps =25
    Failed Shuffles=0
    Merged Map outputs=25
    GC time elapsed (ms)=1013
    CPU time spent (ms)=20340
    Physical memory (bytes) snapshot=2231046144
    Virtual memory (bytes) snapshot=8928280576
    Total committed heap usage (bytes)=1880621056

Shuffle Errors
    BAD_ID=0
    CONNECTION=0
    IO_ERROR=0
    WRONG_LENGTH=0
    WRONG_MAP=0
    WRONG_REDUCE=0

File Input Format Counters
    Bytes Read=15261352
File Output Format Counters
    Bytes Written=1860178
[oracle@bigdatalite movie]$ █

```

#### 10. Copy and view the Job Analyzer report.

```

hadoop fs -get moviegenre_report/_balancer/jobanalyzer- \
report.html step10-jobanalyzer-report.html

```

```

[oracle@bigdatalite movie]$ hadoop fs -get moviegenre_report/_balancer/jobanalyzer-report.html step10-jobanalyzer-report.html
[oracle@bigdatalite movie]$ ls -ls
total 3496
  84 -r--r--r--. 1 oracle oinstall   85338 Jan 26 14:54 HandsOnLab.pdf
  4 drwxr-xr-x. 2 oracle oinstall   4096 Feb 20 06:50 movieData
3360 -rw-r--r--. 1 oracle oinstall 3439263 Jan 26 14:54 movieData.zip
   8 -rwxr-xr-x. 1 oracle oinstall   6799 Jan 26 14:54 movieDemo
  12 -rw-r--r--. 1 oracle oinstall   9658 Jan 26 14:54 README.txt
  20 -rw-r--r--. 1 oracle oinstall 19613 Feb 20 07:29 step10-jobanalyzer-report.html
   8 -rw-r--r--. 1 oracle oinstall   7146 Feb 20 07:13 step8-jobanalyzer-report.html
[oracle@bigdatalite movie]$ █

```

```

firefox step10-jobanalyzer-report.html

```

The partial report is as follows:

**Job Information**

Job Name	orabalancerdemo-2.3.0.jar
Job Id	job_1420735348738_0017
Start Time	2015-01-28 12:15:54
Finish Time	2015-01-28 12:16:34

**Time Information**

Map Phase	00:00:18
Reduce Phase	00:00:25
Shuffle	00:00:23
Merge	00:00:17
Reduce	00:00:18
Job	00:00:40

**Reduce Tasks Metrics Summary**

Task ID	Time			%Load	
	Start	Finish	Elapsed	Observed	
0	12:16:09	12:16:19	00:00:09	21	
1	12:16:15	12:16:25	00:00:10	21	
2	12:16:16	12:16:28	00:00:11	15	
3	12:16:20	12:16:31	00:00:10	25	
4	12:16:27	12:16:34	00:00:06	18	

**Reduce Tasks Metrics**

Task ID	Elapsed Time			Input							
	Shuffle	Merge	Reduce	Shuffle Bytes		Keys		Records		ValueBytes	
				count	%	count	%	count	%	count	%
0	00:00:06	00:00:00	00:00:02	753,360	21	6	21	75,333	21	301,332	21
1	00:00:06	00:00:00	00:00:03	765,260	21	6	21	76,523	21	306,092	21
2	00:00:08	00:00:00	00:00:03	522,620	15	5	18	52,259	15	209,036	15
3	00:00:08	00:00:00	00:00:02	903,540	25	5	18	90,351	25	361,404	25
4	00:00:05	00:00:00	00:00:01	645,840	18	6	21	64,581	18	258,324	18
Total	00:00:23	00:00:17	00:00:18	3,590,620	-	28	-	359,047	-	1,436,188	-
Average	00:00:06	00:00:00	00:00:02	718,124	-	6	-	71,809	-	287,238	-

Note the additional information in the report. The Reduce Tasks Metrics Summary table includes a %Load Observed column, which is the number of rows processed by each Reduce task. You can clearly see that there is some skew—Reduce task #2 processes only 15 rows, while Reduce task #3 processes 25 rows (because the VM runs only on one node, the elapsed times do not reflect the skew. In a cluster, the elapsed time is proportional to the %Load Observed). The report also includes detailed metrics on each Reduce task in the second half of the report.

**Note:** Close your web browser when you are done reviewing the report.

- Re-run the MapReduce job with Perfect Balance enabled. This automatically sets the `oracle.hadoop.balancer.autoAnalyze=BASIC_REPORT` property, which enables the Job Analyzer. You can change `BASIC_REPORT` to `REDUCER_REPORT` to get a more detailed report.

```

hadoop jar $BALANCERDEMO_HOME/jlib/orabalancerdemo-2.3.0.jar
oracle.hadoop.balancer.examples.movie.MovieGenre -D
mapred.reduce.tasks=5 -D mapred.input.dir=movieData -D
mapred.output.dir=moviegenre_autobalance -D
oracle.hadoop.balancer.autoBalance=TRUE -D
oracle.hadoop.balancer.autoAnalyze=REDUCER_REPORT -libjars
$BALANCER_HOME/jlib/commons-math-
2.2.jar,$BALANCER_HOME/jlib/orabalancer-2.2.0.jar

```

```
[oracle@bigdatalite movie]$ hadoop jar $BALANCERDEMO_HOME/jlib/orabalancerdemo-2.3.0.jar oracle.hadoop.balancer.example -D mapred.reduce.tasks=5 -D mapred.input.dir=movieData -D mapred.output.dir=moviegenre_autobalance -D oracle.utoBalance=TRUE -D oracle.hadoop.balancer.autoAnalyze=REDUCER_REPORT -libjars $BALANCER_HOME/jlib/commons-math-2.2E/jlib/orabalancer-2.2.0.jar
15/01/28 13:06:08 INFO Configuration.deprecation: mapred.reduce.tasks is deprecated. Instead, use mapreduce.job.reduces
15/01/28 13:06:08 INFO Configuration.deprecation: mapred.input.dir is deprecated. Instead, use mapreduce.input.fileinputdir
15/01/28 13:06:08 INFO Configuration.deprecation: mapred.output.dir is deprecated. Instead, use mapreduce.output.fileoutputdir
15/01/28 13:06:09 INFO balancer.Balancer: Creating balancer
15/01/28 13:06:09 INFO balancer.Balancer: Starting Balancer
15/01/28 13:06:09 INFO Configuration.deprecation: mapred.map.tasks is deprecated. Instead, use mapreduce.job.maps
15/01/28 13:06:09 INFO Configuration.deprecation: mapred.max.split.size is deprecated. Instead, use mapreduce.input.split.maxsize
15/01/28 13:06:09 INFO input.FileInputFormat: Total input paths to process : 5
15/01/28 13:06:11 INFO balancer.Balancer: Balancer completed
15/01/28 13:06:12 INFO client.RMProxy: Connecting to ResourceManager at localhost/127.0.0.1:8032
15/01/28 13:06:12 INFO input.FileInputFormat: Total input paths to process : 5
15/01/28 13:06:12 INFO mapreduce.JobSubmitter: number of splits:5
15/01/28 13:06:12 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1420735348738_0019
15/01/28 13:06:13 INFO impl.YarnClientImpl: Submitted application application_1420735348738_0019
15/01/28 13:06:13 INFO mapreduce.Job: The url to track the job: http://bigdatalite.localdomain:8088/proxy/application_0019/
15/01/28 13:06:13 INFO mapreduce.Job: Running job: job_1420735348738_0019
15/01/28 13:06:20 INFO mapreduce.Job: Job job_1420735348738_0019 running in uber mode : false
15/01/28 13:06:20 INFO mapreduce.Job: map 0% reduce 0%
15/01/28 13:06:34 INFO mapreduce.Job: map 60% reduce 0%
15/01/28 13:06:43 INFO mapreduce.Job: map 80% reduce 0%
15/01/28 13:06:44 INFO mapreduce.Job: map 100% reduce 0%
15/01/28 13:06:47 INFO mapreduce.Job: map 100% reduce 20%
15/01/28 13:06:56 INFO mapreduce.Job: map 100% reduce 40%
15/01/28 13:06:57 INFO mapreduce.Job: map 100% reduce 60%
15/01/28 13:06:59 INFO mapreduce.Job: map 100% reduce 80%
15/01/28 13:07:03 INFO mapreduce.Job: map 100% reduce 100%
15/01/28 13:07:05 INFO mapreduce.Job: Job job_1420735348738_0019 completed successfully
15/01/28 13:07:05 INFO mapreduce.Job: Counters: 49
```

#### File System Counters

FILE: Number of bytes read=3590500  
 FILE: Number of bytes written=8139825  
 FILE: Number of read operations=0  
 FILE: Number of large read operations=0  
 FILE: Number of write operations=0  
 HDFS: Number of bytes read=15262017  
 HDFS: Number of bytes written=1871037  
 HDFS: Number of read operations=30  
 HDFS: Number of large read operations=0  
 HDFS: Number of write operations=15

#### Job Counters

Launched map tasks=5  
 Launched reduce tasks=5  
 Data-local map tasks=5  
 Total time spent by all maps in occupied slots (ms)=50881  
 Total time spent by all reduces in occupied slots (ms)=50264  
 Total time spent by all map tasks (ms)=50881  
 Total time spent by all reduce tasks (ms)=50264  
 Total vcore-seconds taken by all map tasks=50881  
 Total vcore-seconds taken by all reduce tasks=50264  
 Total megabyte-seconds taken by all map tasks=13025536  
 Total megabyte-seconds taken by all reduce tasks=12867584

```

Map-Reduce Framework
  Map input records=359047
  Map output records=359047
  Map output bytes=2872376
  Map output materialized bytes=3590620
  Input split bytes=665
  Combine input records=0
  Combine output records=0
  Reduce input groups=32
  Reduce shuffle bytes=3590620
  Reduce input records=359047
  Reduce output records=3611
  Spilled Records=718094
  Shuffled Maps =25
  Failed Shuffles=0
  Merged Map outputs=25
  GC time elapsed (ms)=1285
  CPU time spent (ms)=26760
  Physical memory (bytes) snapshot=2255773696
  Virtual memory (bytes) snapshot=8914972672
  Total committed heap usage (bytes)=1909981184
Shuffle Errors
  BAD_ID=0
  CONNECTION=0
  IO_ERROR=0
  WRONG_LENGTH=0
  WRONG_MAP=0
  WRONG_REDUCE=0
File Input Format Counters
  Bytes Read=15261352
File Output Format Counters
  Bytes Written=1860184
[oracle@bigdatalite movie]$ █

```

The preceding command creates a `_balancer` directory in the job output directory while running the job.

## 12. Copy and view the Job Analyzer report.

```

hadoop fs -get moviegenre_autobalance/_balancer/jobanalyzer-
report.html step12-jobanalyzer-report.html

```

```

[oracle@bigdatalite movie]$ hadoop fs -get moviegenre_autobalance/_balancer/jobanalyzer-report.html step12-jobanalyzer-report.html
[oracle@bigdatalite movie]$ ls -ls
total 3520
  84 -r--r--r--. 1 oracle oinstall    85338 Jan 26 14:54 HandsOnLab.pdf
  4 drwxr-xr-x. 2 oracle oinstall    4096 Feb 20 06:50 movieData
3360 -rw-r--r--. 1 oracle oinstall 3439263 Jan 26 14:54 movieData.zip
   8 -rwxr-xr-x. 1 oracle oinstall     6799 Jan 26 14:54 moviedemo
  12 -rw-r--r--. 1 oracle oinstall     9658 Jan 26 14:54 README.txt
  20 -rw-r--r--. 1 oracle oinstall 19613 Feb 20 07:29 step10-jobanalyzer-report.html
  24 -rw-r--r--. 1 oracle oinstall  23252 Feb 20 07:36 step12-jobanalyzer-report.html
   8 -rw-r--r--. 1 oracle oinstall    7146 Feb 20 07:13 step8-jobanalyzer-report.html
[oracle@bigdatalite movie]$ █

```

```

firefox step12-jobanalyzer-report.html

```

Screenshot of a web browser showing a MapReduce job analysis report. The report includes tables for Job Information, Time Information, Reduce Tasks Metrics Summary, and Reduce Tasks Metrics.

Job Information		Time Information	
Job Name	orabalancerdemo-2.3.0.jar	Map Phase	00:00:22
Job Id	job_1420735348738_0019	Reduce Phase	00:00:26
Start Time	2015-01-28 13:06:18	Shuffle	00:00:24
Finish Time	2015-01-28 13:07:02	Merge	00:00:17
		Reduce	00:00:18
		MapReduce	00:00:43
		Sampling	00:00:02
		Job	00:00:45

Reduce Tasks Metrics Summary						
Task ID	Time			%Load		
	Start	Finish	Elapsed	Predicted	±CI	Observed
0	13:06:35	13:06:45	00:00:09	20	3.9	20
1	13:06:43	13:06:54	00:00:11	20	3.5	20
2	13:06:44	13:06:56	00:00:11	20	3.6	20
3	13:06:46	13:06:57	00:00:11	20	3.7	21
4	13:06:55	13:07:02	00:00:06	20	3.9	19

Task ID	Elapsed Time			Input						
	Shuffle	Merge	Reduce	Shuffle Bytes		Keys		Records		ValueBytes
	count	%	count	%	count	%	count	%	count	%
0	731,860	20	5	16	73,183	20	292,732	20	7	7
1	701,240	20	7	22	70,121	20	280,484	20	7	7
2	723,100	20	7	22	72,307	20	289,228	20	7	7
3	745,280	21	6	19	74,525	21	298,100	21	7	7

The report was generated when Perfect Balance was enabled to balance the MapReduce job. The **%Load Observed** column shows a more evenly balanced distribution of rows.

- When you are done with this practice, delete all the generated files as follows, running one command at a time to ensure that there are no typing errors and that each command runs successfully.

```

hadoop fs -rm -r moviegenre_output
hadoop fs -rm -r moviegenre_report
hadoop fs -rm -r moviegenre_autobalance
rm -r $BALANCERDEMO_HOME/examples/movie/movieData
rm $BALANCERDEMO_HOME/examples/movie/step8-jobanalyzer-report.html
rm $BALANCERDEMO_HOME/examples/movie/step10-jobanalyzer-report.html
rm $BALANCERDEMO_HOME/examples/movie/step12-jobanalyzer-report.html

```



## **Practices for Lesson 28: Securing Your Data**

**Chapter 28**

## Practices for Lesson 28

---

There are no practices for this lesson.