Building with Amazon Redshift clusters

**SPL-TF-200-DBWARC-1 - Version 1.0.4**

The lab dataset is being provided to you by permission of IMDb and is subject to the terms of the AWS Digital Training Agreement (available at <https://aws.amazon.com/training/digital-training-agreement>). You are expressly prohibited from copying, modifying, selling, exporting, or using this dataset in any way other than for the purpose of completing this lab.

© 2024 Amazon Web Services, Inc. or its affiliates. All rights reserved. This work may not be reproduced or redistributed, in whole or in part, without prior written permission from Amazon Web Services, Inc. Commercial copying, lending, or selling is prohibited. All trademarks are the property of their owners.

Note: Do not include any personal, identifying, or confidential information into the lab environment. Information entered may be visible to others.

Corrections, feedback, or other questions? Contact us at [*AWS Training and Certification*](https://support.aws.amazon.com/#/contacts/aws-training).

**Overview**

Amazon Redshift is a fast, fully managed, petabyte-scale data warehouse service that makes it simple and cost-effective to efficiently analyze all your data using your existing business intelligence (BI) tools. It is optimized for datasets ranging from a few hundred gigabytes to a petabyte or more. One of the advancements that allows Amazon Redshift clusters to analyze so much data is Amazon Redshift Spectrum. This feature allows Amazon Redshift to analyze massive volumes of data stored in an Amazon Simple Storage Service (Amazon S3) data lake.

This lab uses the IMDb dataset. IMDb is the go-to platform for movie fans across the globe. It is the online database of information related to films, television programs, video games, and streaming content including cast, production crew, plot summaries, trivia, fan and critic reviews, and ratings.

**Objectives**

After completing this lab, you will be able to:

* Use the SQL Workbench for Amazon Redshift
* Understand the COPY command for loading data and working with compression
* Use a manifest file for data imports
* Archive data using the UNLOAD command
* Use ANALYZE and VACUUM operations
* Use the Amazon Redshift console to explore query statistics

**Pre-requisites**

This lab requires:

* Access to a notebook computer with Wi-Fi and Microsoft Windows, macOS X, or Linux (Ubuntu, SuSE, or Red Hat).
* **Note:** You can use an iPad or tablet device to access these directions in the lab console.
* An Internet browser such as Chrome, Firefox, or IE9+.
* **Note:** Previous versions of Internet Explorer are not supported.
* An SSH client such as PuTTY.

**Technical knowledge prerequisites**

To successfully complete this lab, you should be familiar with:

* Familiarity with SQL statements and basic operations
* Familiarity with the AWS Management Console
* A remote connection client (such as Remote Desktop Connection, which is included with most versions of Windows) to connect to your server

**Note:** If you are running macOS X, you can download the Microsoft Remote Desktop app from the App Store. See [*RDP Client for Mac*](https://apps.apple.com/us/app/microsoft-remote-desktop-10/id1295203466?mt=12).

In this lab, you will use the AWS Management Console and SQL Workbench to experiment with different table layouts and schema designs. You will use the COPY command for data load operations.

**Duration**

This lab requires **60 minutes** to complete.

**AWS services not used in this lab**

AWS services that are not used in this lab are disabled in the lab environment. In addition, the capabilities of the services used in this lab are limited to what the lab requires. Expect errors when accessing other services or performing actions beyond those provided in this lab guide.

**Start lab**

1. To launch the lab, at the top of the page, choose **Start lab**.

**Caution:** You must wait for the provisioned AWS services to be ready before you can continue.

1. To open the lab, choose **Open Console**.

You are automatically signed in to the AWS Management Console in a new web browser tab.

**WARNING:** **Do not change the Region unless instructed.**

COMMON SIGN-IN ERRORS

**Error: You must first sign out**



If you see the message, **You must first log out before logging into a different AWS account:**

* Choose the **click here** link.
* Close your **Amazon Web Services Sign In** web browser tab and return to your initial lab page.
* Choose **Open Console** again.

**Error: Choosing Start Lab has no effect**

In some cases, certain pop-up or script blocker web browser extensions might prevent the **Start Lab** button from working as intended. If you experience an issue starting the lab:

* Add the lab domain name to your pop-up or script blocker’s allow list or turn it off.
* Refresh the page and try again.

**Task 1: Connect to the cluster**

In this task, you will log in to the AWS Management Console, open the Amazon Redshift service, and explore the IMDb cluster. After that, you will use a remote connection client to access the Amazon Elastic Compute Cloud (Amazon EC2) Windows CommandHost instance to connect to the cluster to perform data activities.

REVIEW OF IMPORTANT AMAZON REDSHIFT CONCEPTS

An Amazon Redshift data warehouse is an enterprise-class relational database query and management system. Amazon Redshift supports client connections with many types of applications, including BI, reporting, data, and analytics tools. When you perform analytic queries, you are retrieving, comparing, and evaluating large amounts of data in multiple-stage operations to produce a final result. Amazon Redshift achieves efficient storage and optimum query performance through a combination of massively parallel processing, columnar data storage, and very efficient, targeted data compression encoding schemes.

High-performance Amazon Redshift clusters rely on proper compression and data distribution. Compression conserves storage space. A compression encoding specifies the type of compression that is applied to a column of data values as rows are added to a table. If no compression is specified in a CREATE TABLE or ALTER TABLE statement, Amazon Redshift automatically assigns compression encoding as follows:

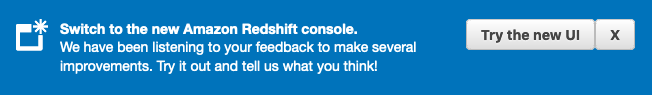
* Columns that are defined as sort keys are assigned RAW (no compression) compression.
* Columns that are defined as BOOLEAN, REAL, or DOUBLE PRECISION data types are assigned RAW compression.
* Columns that are defined as SMALLINT, INTEGER, BIGINT, DECIMAL, DATE, TIMESTAMP, or TIMESTAMPTZ data types are assigned AZ64 compression.
* Columns that are defined as CHAR or VARCHAR data types are assigned LZO compression.

When data is loaded into a table, Amazon Redshift distributes the rows of the table to each of the compute nodes according to the table’s distribution style. The goal in planning the data distribution is to select a table distribution style that minimizes the impact of redistribution by colocating data where it needs to be before the query is run.

CONNECT TO THE AMAZON REDSHIFT CLUSTER

1. In the AWS Management Console, click **Services** and select **Amazon Redshift**.

* **Note:** If you see a red “AccessDenied” banner, you can safely ignore this message and continue the lab.
* **Note:** You may see a banner about using the new Amazon Redshift console or returning to the original console. You may use either console to complete this lab. To use the new console, click **Try the new UI** and use the following instructions. To use the original console, click **Back to original console**, and then [click here](https://labs.skillbuilder.aws/sa/lab/arn%3Aaws%3Alearningcontent%3Aus-east-1%3A470679935125%3Ablueprintversion%2FSPL-TF-200-DBWARC-1%3A1.0.4-fac40e80/en-US#previousUI-1) for instructions using the original console. Note that the new console has icons in the left navigation pane while the original console does not.



* **Note:** If you use the new console, close the **Welcome** popup when the page loads.

1. In the left navigation pane, click **Clusters**.
2. Click the checkbox to the left of the cluster.
3. At the top of the screen, click **Actions** and select **Manage IAM roles**.
4. For **Available IAM roles**, select the **RedshiftAccessRole-xxxxx** role, and click **Associate IAM role**
5. Click **Save changes**

This is the AWS Identity and Access Management (IAM) role that authorizes Amazon Redshift to access the external data catalog and Amazon S3 on your behalf. You must associate this role with your Amazon Redshift cluster for data load and unload activities to work. For more information, see [*Create an IAM Role for Amazon Redshift*](https://docs.aws.amazon.com/redshift/latest/dg/c-getting-started-using-spectrum-create-role.html).

 Congratulations! You have successfully connected to a Redshift cluster.

**Task 2: Connect to the Windows instance**

In this task, you will use a remote connection client (such as Remote Desktop Connection, which is included with most versions of Windows) to connect to a Windows EC2 instance. This instance has an application installed that you will use to query the Amazon Redshift cluster.

If you are running macOS X, you can download the Microsoft Remote Desktop app from the App Store. See [*RDP Client for Mac*](https://apps.apple.com/us/app/microsoft-remote-desktop-10/id1295203466?mt=12).

**Note:** If you are unable to use a Remote Desktop client or have outbound port 3389 open from your network, you can [connect to your windows instance using Apache Guacamole](https://labs.skillbuilder.aws/sa/lab/arn%3Aaws%3Alearningcontent%3Aus-east-1%3A470679935125%3Ablueprintversion%2FSPL-TF-200-DBWARC-1%3A1.0.4-fac40e80/en-US#connect_guacamole).

CONNECT TO THE REMOTE SYSTEM

1. In the **AWS Management Console**, click **Services** and select **EC2**.
2. In the left navigation pane, click **Instances**.

The **WindowsInstance** row should be selected. To ensure that the instance is fully operational, verify that the **Status Checks** column shows *2/2 checks passed*. If it doesn’t, wait a minute or so and click the refresh  icon to update the status.

1. Click **Connect**
2. Select the **RDP client** tab.
3. Click **Download Remote Desktop File**
4. Save the file to your computer, and then open it.
5. When prompted that the publisher can’t be identified, click **Connect**
6. When prompted for credentials, enter the following:

* **Username:** Keep the default username

Administrator

* **Password:** Copy and paste the **AdministratorPassword** value from the left side of the lab page

1. Click **OK**
2. When prompted to connect despite certificate errors, click **Yes**

**Note:** The window may be black when you first open it. Be patient, as it may take a minute or two to load the desktop.

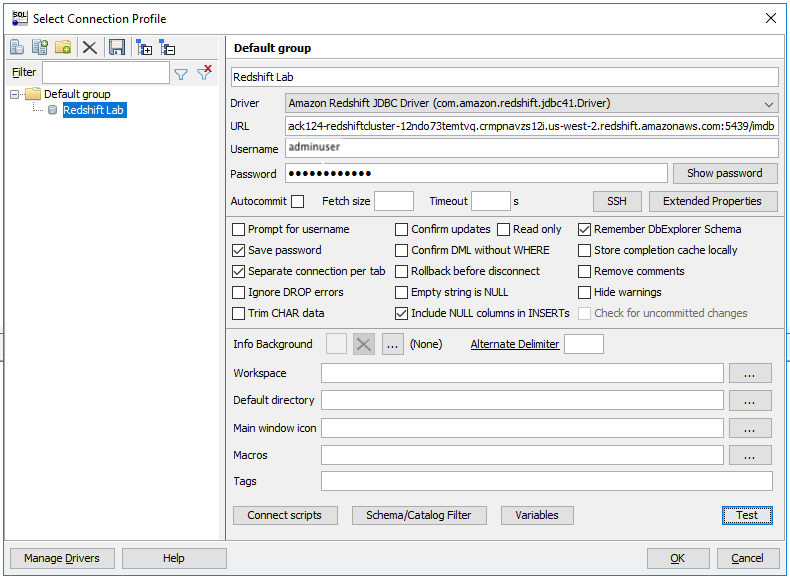
OPEN AND CONFIGURE SQL WORKBENCH

**Note:** When working in the remote connection client, the right-click menu may not work. To copy and paste, use hot keys. For Windows, use CTRL+C to copy and CTRL+V to paste. For macOS, use Command+C to copy and Command+V to paste.

1. On the desktop of your remote connection session, double-click the **SQLWorkbench** shortcut.

This opens the SQL Workbench application. You will be asked to create a new connection profile because this is the first time you have used the application.

The window should look like the following:



1. At the bottom of the **Select Connection Profile** window, click **Manage Drivers**
2. Configure the following:

* **Name:** Enter

Amazon Redshift JDBC Driver

* To the right of the **Library** box, click the folder  icon
* Navigate to your desktop and select the *RedshiftJDBC41-1.1.10.1010* file
* Click **Open**
* Click **OK**

1. In the **Select Connection Profile** window, configure the following:

* Delete **New profile** from the top field and enter

Amazon Redshift profile

* **Driver:** Select **Amazon Redshift JDBC Driver** (You must select the one you added, with “JDBC” in the title. Do NOT accept the default “Amazon Redshift” driver.)
* **URL:** Delete the text in this field, then copy and paste the **JDBCClusterEndpoint** value from the left side of the lab page
* **Username:** Enter

adminuser

* **Password:** Enter

Pa33w0rd!

* The JDBC URL looks similar to: jdbc:redshift://redshiftdemo.clsqyy8ilhmx.us-west-2.redshift.amazonaws.com:5439/redshiftdemo\_

1. Click **OK**

 Congratulations! You have established a connection to the Amazon Redshift cluster.

**Troubleshooting** If you have trouble, please verify that you:

* Entered the password correctly. Note that the password is case sensitive.
* Included the *jdbc:* piece at the beginning of the JDBC URL.
* Cleared the URL field before pasting in your cluster’s URL.

 Congratulations! You have successfully connected to the Windows instance.

**Task 3: Import data**

In this task, you will use multiple files for data import activities. The COPY command loads the data in parallel from multiple files, dividing the workload among the nodes in your cluster. When you load all the data from a single large file, Amazon Redshift is forced to perform a serialized load, which is much slower. Splitting a large data file into smaller, equally sized files, between 1 MB and 1 GB after compression, allows Amazon Redshift to parallelize the load and reduce the overall operation time. For optimum parallelism, the ideal size is between 1 and 125 MB after compression.

This approach has been applied to the IMDb data that is stored in an Amazon S3 bucket. The following is an example of the file structure in the Amazon S3 bucket for **title\_genres**. The data in each file is in tab separated value (TSV) format. Notice the data is split into three files: aa, ab, and ac. This is mirrored in the data that is compressed with GZIP: aa.gz, ab.gz, and ac.gz.

title-genres

├── split

│   ├── tg\_aa

│   ├── tg\_ab

│   └── tg\_ac

└── split-with-gzip

├── tg\_aa.gz

├── tg\_ab.gz

└── tg\_ac.gz

**Learn more:** For more information about loading data from Amazon S3, see <https://docs.aws.amazon.com/redshift/latest/dg/t_Loading-data-from-S3.html>.

UNDERSTAND THE DATA

The following image displays the schema for the IMDb data you will be working with throughout this lab.

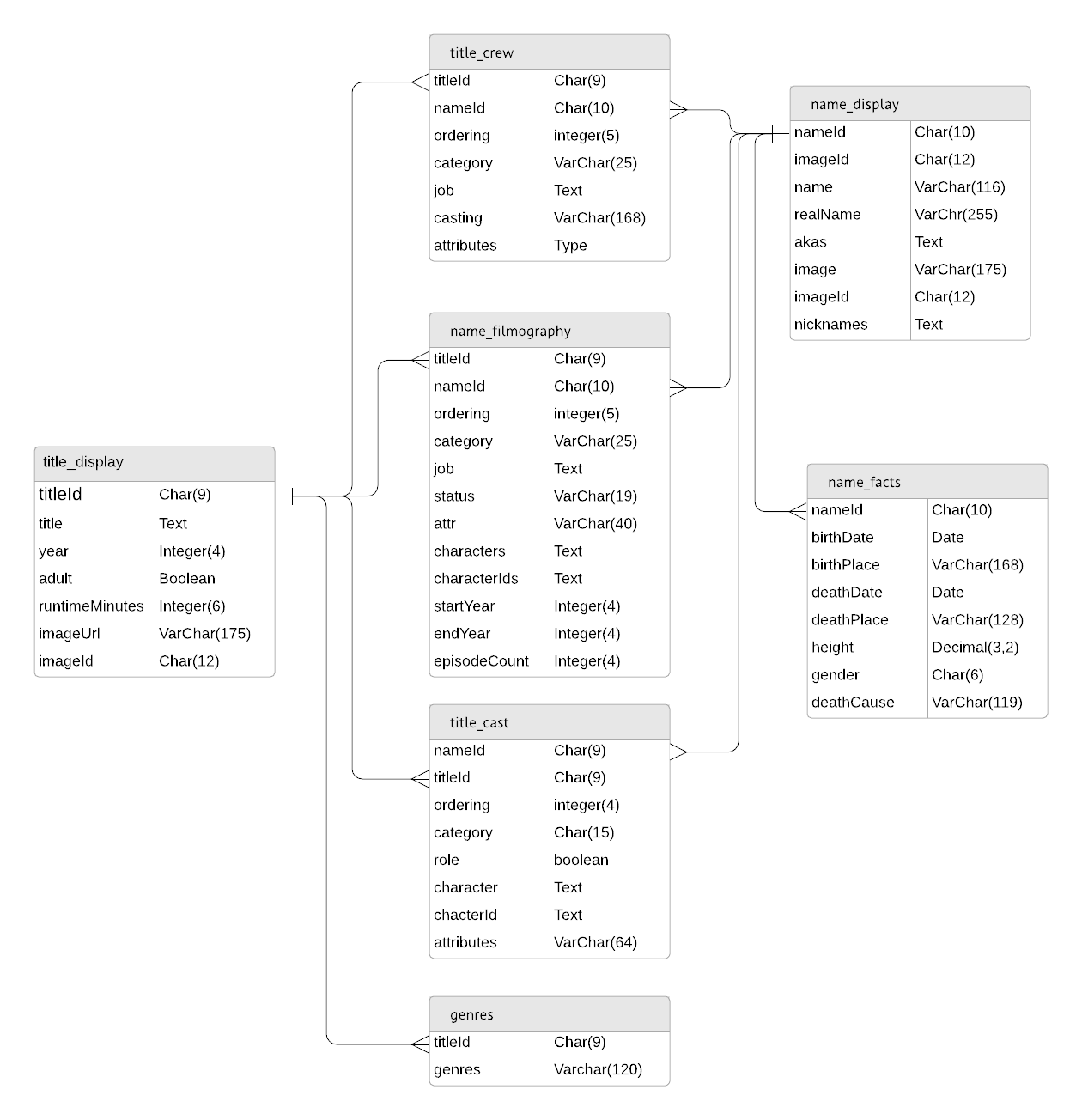
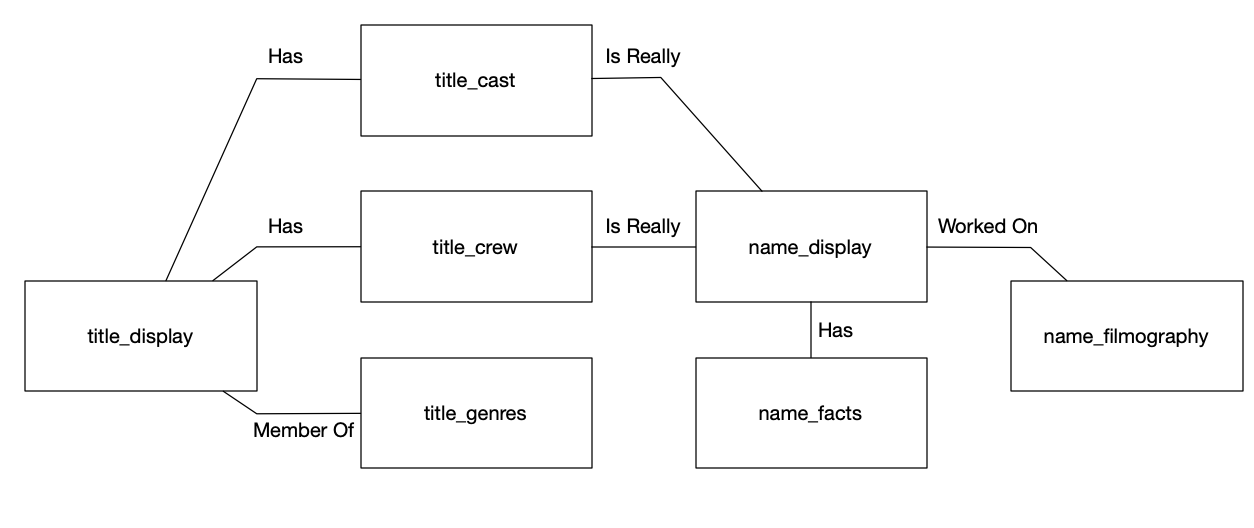


Table explanation:

* **title\_display:** Records data for each title, assigning each a titleId.
* **name\_facts:** Records facts around a person. Assigns a unique nameId to each person.
* **name\_display:** Records how a person’s information is displayed. Shares the nameId with name\_facts.
* **name\_filmography:** Relates a person (nameId) to a title (titleId).
* **title\_genres:** Relates a title (titleId) to genres.
* **title\_crew:** Relates a title (titleId) to a person (nameId) and the job held on the crew.
* **title\_cast:** Relates the title (titleId) to a person (nameId) and the casting they held.

UNDERSTAND THE RELATIONSHIPS

The following diagram can help you understand how the relationships are used in the existing database:



CREATE THE SCHEMA

Now you will create a schema for the data and determine the schema path.

1. **Copy/Paste:** In the **Statement 1** tab of the **SQL Workbench** window, copy and paste the following SQL statements:

BEGIN;

create schema if not exists imdb authorization adminuser;

set search\_path = imdb;

COMMIT;

1. To run the statements, click the play  button just above the **Statement 1** tab.

IMPORT THE UNCOMPRESSED DATA

The data stored in the Amazon S3 bucket must be loaded into the Amazon Redshift cluster. The data was exported in two forms: one without any compression added and the second with GZIP compression. The next few steps take you through loading both datasets.

1. **Copy/Paste:** Delete everything in the **Statement 1** tab. Then, copy and paste in the following SQL statements:

BEGIN;

set search\_path = imdb;

CREATE TABLE imdb.name\_display (

"nameId" char(10) NOT NULL,

"name" varchar(128) NOT NULL,

"realName" varchar(255) DEFAULT NULL,

"akas" varchar(2000),

"image" varchar(192) DEFAULT NULL,

"imageId" char(12) DEFAULT NULL,

"nicknames" varchar(2000)

);

CREATE TABLE imdb.name\_display\_gzip (

"nameId" char(10) NOT NULL,

"name" varchar(128) NOT NULL,

"realName" varchar(255) DEFAULT NULL,

"akas" varchar(2000),

"image" varchar(192) DEFAULT NULL,

"imageId" char(12) DEFAULT NULL,

"nicknames" varchar(2000)

);

CREATE TABLE imdb.name\_facts (

"nameId" char(10) NOT NULL,

"birthDate" date DEFAULT NULL,

"birthPlace" varchar(192) DEFAULT NULL,

"deathDate" date DEFAULT NULL,

"deathPlace" varchar(128) DEFAULT NULL,

"height" float DEFAULT NULL,

"gender" char(6) DEFAULT NULL,

"deathCause" varchar(128) DEFAULT NULL

);

CREATE TABLE imdb.name\_facts\_gzip (

"nameId" char(10) NOT NULL,

"birthDate" date DEFAULT NULL,

"birthPlace" varchar(192) DEFAULT NULL,

"deathDate" date DEFAULT NULL,

"deathPlace" varchar(128) DEFAULT NULL,

"height" float DEFAULT NULL,

"gender" char(6) DEFAULT NULL,

"deathCause" varchar(128) DEFAULT NULL

);

CREATE TABLE imdb.name\_filmography (

"nameId" char(10) NOT NULL,

"ordering" integer NOT NULL,

"titleId" char(9) NOT NULL,

"category" varchar(32) NOT NULL,

"job" varchar(4000), -- You can use 'max' for the columns the values are big.

"status" varchar(19) DEFAULT NULL,

"attr" varchar(40) DEFAULT NULL,

"characters" varchar(2000),

"characterIds" varchar(2000),

"startYear" integer DEFAULT NULL,

"endYear" integer DEFAULT NULL,

"episodeCount" integer DEFAULT NULL

);

CREATE TABLE imdb.name\_filmography\_gzip (

"nameId" char(10) NOT NULL,

"ordering" integer NOT NULL,

"titleId" char(9) NOT NULL,

"category" varchar(32) NOT NULL,

"job" varchar(4000), -- You can use 'max' for the columns the values are big.

"status" varchar(19) DEFAULT NULL,

"attr" varchar(40) DEFAULT NULL,

"characters" varchar(2000),

"characterIds" varchar(2000),

"startYear" integer DEFAULT NULL,

"endYear" integer DEFAULT NULL,

"episodeCount" integer DEFAULT NULL

);

CREATE TABLE imdb.title\_cast (

"titleId" char(9) NOT NULL,

"ordering" int NOT NULL,

"nameId" char(10) NOT NULL,

"category" char(15) NOT NULL,

"role" int DEFAULT NULL,

"characters" varchar(2500),

"characterIds" varchar(250),

"attributes" varchar(64) DEFAULT NULL

);

CREATE TABLE imdb.title\_cast\_gzip (

"titleId" char(9) NOT NULL,

"ordering" int NOT NULL,

"nameId" char(10) NOT NULL,

"category" char(15) NOT NULL,

"role" int DEFAULT NULL,

"characters" varchar(2500),

"characterIds" varchar(250),

"attributes" varchar(64) DEFAULT NULL

);

CREATE TABLE imdb.title\_crew (

"titleId" char(9) NOT NULL,

"ordering" int NOT NULL,

"nameId" char(10) NOT NULL,

"category" varchar(28) DEFAULT NULL,

"job" varchar(4000),

"casting" varchar(192) DEFAULT NULL,

"attributes" varchar(32) DEFAULT NULL

);

CREATE TABLE imdb.title\_crew\_gzip (

"titleId" char(9) NOT NULL,

"ordering" int NOT NULL,

"nameId" char(10) NOT NULL,

"category" varchar(28) DEFAULT NULL,

"job" varchar(4000),

"casting" varchar(192) DEFAULT NULL,

"attributes" varchar(32) DEFAULT NULL

);

CREATE TABLE imdb.title\_display (

"titleId" char(9) NOT NULL,

"title" text NOT NULL,

"year" int DEFAULT NULL,

"adult" int DEFAULT NULL,

"runtimeMinutes" int DEFAULT NULL,

"imageUrl" varchar(175) DEFAULT NULL,

"imageId" char(12) DEFAULT NULL,

"type" char(12) DEFAULT NULL,

"originalTitle" text

);

CREATE TABLE imdb.title\_display\_gzip (

"titleId" char(9) NOT NULL,

"title" text NOT NULL,

"year" int DEFAULT NULL,

"adult" int DEFAULT NULL,

"runtimeMinutes" int DEFAULT NULL,

"imageUrl" varchar(175) DEFAULT NULL,

"imageId" char(12) DEFAULT NULL,

"type" char(12) DEFAULT NULL,

"originalTitle" text

);

CREATE TABLE imdb.title\_genres (

"titleId" char(10) NOT NULL,

"genres" varchar(250) NOT NULL

);

CREATE TABLE imdb.title\_genres\_gzip (

"titleId" char(10) NOT NULL,

"genres" varchar(250) NOT NULL

);

COMMIT;

1. To run the statements, click the play  button just above the **Statement 1** tab.

**Note:** If you receive an “Invalid operation” message for no obvious reason, remove the code by selecting all of the code in the tab and pressing CTRL+X. Type

COMMIT

 and click the play button. Then, press CTRL+V to paste the contents back into the tab, and click the play button to run the code again.

Next, you will load the data for each individual table into the new **imdb.title\_genres** table, which was created by the last script you ran.

1. Delete everything in the **Statement 1** tab.

**Note:** You need to replace sections in the code so that it is specific to your environment. A helper webpage has been created to make this easier for you.

1. Copy the **ReplaceWebsite** value from the left side of the lab page and paste it into a new browser tab.
2. Fill in the text boxes for **LabDataBucket**, **RedshiftAccessRoleArn**, and **Region** with the corresponding values from the left side of the lab page.
3. **Copy/Paste:** Copy and paste the following code block into the large text box:

BEGIN;

COPY imdb.name\_display FROM 's3://(LabDataBucket)/name-display/split/'

iam\_role '(RedshiftAccessRoleArn)'

delimiter '\t' timeformat 'YYYY-MM-DD HH:MI:SS' NULL AS 'NULL' region '(Region)';

COPY imdb.name\_facts FROM 's3://(LabDataBucket)/name-facts/split/'

iam\_role '(RedshiftAccessRoleArn)'

delimiter '\t' timeformat 'YYYY-MM-DD HH:MI:SS' NULL AS 'NULL' region '(Region)';

COPY imdb.name\_filmography FROM 's3://(LabDataBucket)/name-filmography/split/'

iam\_role '(RedshiftAccessRoleArn)'

delimiter '\t' timeformat 'YYYY-MM-DD HH:MI:SS' NULL AS 'NULL' region '(Region)';

COPY imdb.title\_cast FROM 's3://(LabDataBucket)/title-cast/split/'

iam\_role '(RedshiftAccessRoleArn)'

delimiter '\t' timeformat 'YYYY-MM-DD HH:MI:SS' NULL AS 'NULL' region '(Region)';

COPY imdb.title\_crew FROM 's3://(LabDataBucket)/title-crew/split/'

iam\_role '(RedshiftAccessRoleArn)'

delimiter '\t' timeformat 'YYYY-MM-DD HH:MI:SS' NULL AS 'NULL' region '(Region)';

COPY imdb.title\_display FROM 's3://(LabDataBucket)/title-display/split/'

iam\_role '(RedshiftAccessRoleArn)'

delimiter '\t' timeformat 'YYYY-MM-DD HH:MI:SS' NULL AS 'NULL' region '(Region)';

COPY imdb.title\_genres FROM 's3://(LabDataBucket)/title-genres/split/'

iam\_role '(RedshiftAccessRoleArn)'

delimiter '\t' timeformat 'YYYY-MM-DD HH:MI:SS' NULL AS 'NULL' region '(Region)';

COMMIT;

1. Below the large text box, click **Replace and Copy**.

This updates the code to use your unique variables and copies the result to your clipboard.

1. Return to the **remote connection session**.
2. Delete everything in the **Statement 1** tab. Then, paste in the updated code block.

**Note:** Instead of using the helper webpage, you could replace the code sections within SQL Workbench. To do so:

* Copy the previous code block and paste it into the **Statement 1** tab. Click **Edit** > **Replace** and uncheck **Regular Expression** in the **Replace** window.
* Replace the following placeholders with the corresponding values from the left side of the lab page:
* (LabDataBucket)
* (RedshiftAccessRoleArn)
* (Region)

1. To run the statements, click the play  button just above the **Statement 1** tab.

**Note:** If you get an error that a statement did not run correctly, double-check your code. Once you have corrected the problem, try again.

**Expected Sample output:**

BEGIN executed successfully

(many statements here)

Script execution finished

Total script execution time: 1m 42s

1. Note the **Total script execution time** for data import for all the tables. This is the last line in the **Message** output at the bottom of the screen. You will compare this to the time it takes to import the compressed data.

IMPORT THE COMPRESSED DATA

1. **Copy/Paste:** Return to the helper webpage. Delete everything in the large text box, and then copy and paste the following SQL statements into it:

BEGIN;

COPY imdb.name\_display\_gzip FROM 's3://(LabDataBucket)/name-display/split-with-gzip/'

iam\_role '(RedshiftAccessRoleArn)'

delimiter '\t' timeformat 'YYYY-MM-DD HH:MI:SS' NULL AS 'NULL' region '(Region)' gzip;

COPY imdb.name\_facts\_gzip FROM 's3://(LabDataBucket)/name-facts/split-with-gzip/'

iam\_role '(RedshiftAccessRoleArn)'

delimiter '\t' timeformat 'YYYY-MM-DD HH:MI:SS' NULL AS 'NULL' region '(Region)' gzip;

COPY imdb.name\_filmography\_gzip FROM 's3://(LabDataBucket)/name-filmography/split-with-gzip/'

iam\_role '(RedshiftAccessRoleArn)'

delimiter '\t' timeformat 'YYYY-MM-DD HH:MI:SS' NULL AS 'NULL' region '(Region)' gzip;

COPY imdb.title\_cast\_gzip FROM 's3://(LabDataBucket)/title-cast/split-with-gzip/'

iam\_role '(RedshiftAccessRoleArn)'

delimiter '\t' timeformat 'YYYY-MM-DD HH:MI:SS' NULL AS 'NULL' region '(Region)' gzip;

COPY imdb.title\_crew\_gzip FROM 's3://(LabDataBucket)/title-crew/split-with-gzip/'

iam\_role '(RedshiftAccessRoleArn)'

delimiter '\t' timeformat 'YYYY-MM-DD HH:MI:SS' NULL AS 'NULL' region '(Region)' gzip;

COPY imdb.title\_display\_gzip FROM 's3://(LabDataBucket)/title-display/split-with-gzip/'

iam\_role '(RedshiftAccessRoleArn)'

delimiter '\t' timeformat 'YYYY-MM-DD HH:MI:SS' NULL AS 'NULL' region '(Region)' gzip;

COPY imdb.title\_genres\_gzip FROM 's3://(LabDataBucket)/title-genres/split-with-gzip/'

iam\_role '(RedshiftAccessRoleArn)'

delimiter '\t' timeformat 'YYYY-MM-DD HH:MI:SS' NULL AS 'NULL' region '(Region)' gzip;

COMMIT;

1. Click **Replace and Copy**.
2. Return to the **remote connection session**.
3. Delete everything in the **Statement 1** tab. Then, paste in the updated code block.
4. To run the statement, click the play  button just above the **Statement 1** tab.

**Expected Sample output:**

BEGIN executed successfully

(many statements here)

Script execution finished

Total script execution time: 35.76s

Compare the total script execution time with how long it took to import the uncompressed data. Loading compressed files is often much faster than loading uncompressed files. Additionally, storing compressed files consumes less space in Amazon S3 and will cost you less in the long run.

TEST THE DATA LOAD

The following query returns information on the movies that actor Tom Hanks has been in. This query joins three tables: **name\_display\_gzip**, **title\_cast\_gzip**, and **title\_display\_gzip**.

1. **Copy/Paste:** Delete everything in the **Statement 1** tab. Then, copy and paste the following SQL statement:

select a.name, a.nameId, a.realName, c.title, c.year, c.runtimeMinutes, c.type, c.originalTitle, d.status

from imdb.name\_display\_gzip a, imdb.title\_cast\_gzip b, imdb.title\_display\_gzip c, imdb.name\_filmography\_gzip d

where a.nameId = b.nameId

and b.titleId = c.titleId

and c.titleId = d.titleId

and a.nameId = d.nameId

and a.name = 'Tom Hanks';

1. To run the statement, click the play  button just above the **Statement 1** tab.

**Challenge** Test the query by running it against the uncompressed tables. Did you notice any difference in the query response time?

 Congratulations! You have successfully imported the data.

**Task 4: Use the manifest file to import data**

In this task, you will learn how a manifest file is used to import data. You can use a manifest file to ensure that the COPY command loads all of the required files, and only the required files, for a data load. You can use a manifest to load files from different buckets or files that do not share the same prefix. Instead of supplying an object path for the COPY command, you supply the name of a JSON-formatted text file that explicitly lists the files to be loaded. The URL in the manifest must specify the bucket name and full object path for the file, not just a prefix.

In the next few steps, you will load the **title\_genres** table data using a manifest file. You will create a new table and load the data.

CREATE A NEW TABLE FOR THE DATA

1. **Copy/Paste:** Delete everything in the **Statement 1** tab. Then, copy and paste in the following SQL statements:

BEGIN;

CREATE TABLE imdb.title\_genres\_mnf (

"titleId" char(10) NOT NULL,

"genres" varchar(250) NOT NULL

);

COMMIT;

1. To run the statement, click the play  button just above the Statement 1 tab.

PREPARE THE MANIFEST FILE

1. **Copy/Paste:** Copy and paste the following script into a **text editor** on your local machine, not in the remote connection session.

{

"entries": [

{"url":"s3://(LabDataBucket)/title-genres/split-with-gzip/tg\_aa.gz", "mandatory":true},

{"url":"s3://(LabDataBucket)/title-genres/split-with-gzip/tg\_ab.gz", "mandatory":true},

{"url":"s3://(LabDataBucket)/title-genres/split-with-gzip/tg\_ac.gz", "mandatory":true}

]

}

1. Replace the

(LabDataBucket)

 placeholders with the corresponding value from the left side of the lab page.

1. Save the file to your local machine with the title

title\_genres.manifest

**Note:** Be sure that no other file extensions are added to the filename.

UPLOAD THE MANIFEST FILE

1. Return to the tab with the **AWS Management Console**, click **Services** and click **S3**.
2. Click the name of the bucket that matches the value of **LabDataBucket** from the left side of the lab page.
3. Upload the

title\_genres.manifest

 file into this bucket:

* Click **Upload**
* Click **Add files**
* Select the **title\_genres.manifest** file, and click **Open**
* Click **Upload**

1. **Copy/Paste:** Return to the helper webpage. Delete everything in the large text box, and then copy and paste the following SQL statement into it:

BEGIN;

COPY imdb.title\_genres\_mnf FROM 's3://(LabDataBucket)/title\_genres.manifest'

iam\_role '(RedshiftAccessRoleArn)'

delimiter '\t' timeformat 'YYYY-MM-DD HH:MI:SS' NULL AS 'NULL' region '(Region)' gzip manifest;

COMMIT;

1. Click **Replace and Copy**.
2. Return to the **remote connection session**.
3. In the **Statement 1** tab, delete everything currently in the text box, and paste the updated code.
4. To run the statement, click the play  button just above the Statement 1 tab.

Next, you will verify the row count in the **title\_genres** table matches the row count from the **title\_genres\_mnf** manifest file.

1. **Copy/Paste:** Delete everything in the **Statement 1** tab. Then, copy and paste in the following SQL statements:

select count(\*) as counts1

from imdb.title\_genres\_mnf;

select count(\*) as counts2

from imdb.title\_genres;

1. To run the statement, click the play  button just above the Statement 1 tab.

When the statement finishes running, check the table counts on the **Result 1** and **Result 2** tabs. The counts should be the same.

 Congratulations! You have successfully manifested the file to import the data.

**Task 5: Unload data to Amazon S3**

To archive the data into Amazon S3, you will use the UNLOAD command. By default, UNLOAD writes data in parallel to multiple files, according to the number of slices in the cluster. To write data to a single file, specify PARALLEL OFF. UNLOAD writes the data serially, sorted absolutely according to the ORDER BY clause, if one is used. The maximum size for a data file is 6.2 GB. If the data size is greater than the maximum, UNLOAD creates additional files, up to 6.2 GB each. However, the UNLOAD command is designed to use parallel processing. AWS recommends leaving PARALLEL enabled for most cases, especially if the files will be used to load tables using a COPY command.

**Learn more:** For more information about UNLOAD, see <https://docs.aws.amazon.com/redshift/latest/dg/r_UNLOAD.html>.

In this task, you will free up some space in the Amazon Redshift cluster by archiving a portion of data to an Amazon S3 bucket. Some of the older data is no longer being used frequently; therefore, keeping that data in the cluster may not be cost-effective. Instead of keeping the data in the Amazon Redshift cluster, move it to Amazon S3. Once the data is archived, you can reclaim that space for new data operations.

ARCHIVE DATA TO AMAZON S3

This query extracts rows from the **name\_filmography** table and uploads them to an Amazon S3 bucket. The following query identifies all movies from 1990 to 1996 and unloads them to an Amazon S3 bucket.

1. **Copy/Paste:** Return to the helper webpage. Delete everything in the large text box, and then copy and paste the following SQL statement into it:

UNLOAD ('select \* from imdb.name\_filmography where titleid in (select titleid from imdb.title\_display where year between 1990 and 1996)')

to 's3://(LabDataBucket)/unload/name-filmography/tf\_'

iam\_role '(RedshiftAccessRoleArn)' MAXFILESIZE 5MB;

1. Click **Replace and Copy**.
2. Return to the **remote connection session**.
3. In the **Statement 1** tab, delete everything currently in the text box, and paste the updated code.
4. To run the statement, click the play  button just above the Statement 1 tab.

VERIFY THE DATA WAS SUCCESSFULLY UNLOADED

1. Return to the tab with the **AWS Management Console**, click **Services** and click **S3**.
2. Click the name of the bucket that matches the value of **LabDataBucket** from the left side of the lab page. Look for the **unload** folder.

**Note:** If you do not see the folder, refresh the page.

1. Open the **unload** folder by clicking its name.

Notice that it contains the **name-filmography** folder, which contains the unloaded files from the cluster.

 Congratulations! You have successfully unloaded the data to S3.

**Task 6: Use VACUUM and ANALYZE commands**

Whenever you add, delete, or modify a significant number of rows in an Amazon Redshift cluster, data becomes fragmented. Best practice is to run a VACUUM command and then an ANALYZE command to return the cluster to an organized state. A VACUUM operation recovers the space from deleted rows and restores the sort order. The ANALYZE command updates the statistics metadata, which the query optimizer uses to generate query execution plans.

In this task, you will delete the rows that were unloaded to the Amazon S3 bucket. This gives you the opportunity to reclaim the space for future operations.

1. Return to the **remote connection session**.
2. **Copy/Paste:** Delete everything in the **Statement 1** tab. Then, copy and paste in the following SQL statements:

BEGIN;

DELETE FROM imdb.name\_filmography where titleid in (select titleid from imdb.title\_display where year between 1990 and 1996);

COMMIT;

1. To run the statement, click the play  button just above the Statement 1 tab.

RUN A VACUUM OPERATION

The following query looks at the table stats before running the VACUUM operation.

1. **Copy/Paste:** Delete everything in the **Statement 1** tab. Then, copy and paste in the following SQL statement:

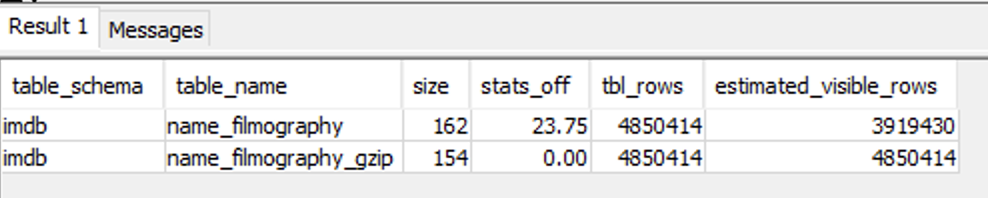
SELECT schema as table\_schema, "table" as table\_name, size, stats\_off, tbl\_rows, estimated\_visible\_rows

FROM svv\_table\_info d

WHERE "table" like 'name\_filmography%';

1. To run the statement, click the play  button just above the Statement 1 tab.

Note the **size** value for the **name\_filmography** table before performing the VACUUM operation.



1. **Copy/Paste:** Delete everything in the **Statement 1** tab. Then, copy and paste in the following SQL statements:

set autocommit on;

VACUUM FULL imdb.name\_filmography;

set autocommit off;

1. To run the statement, click the play  button just above the Statement 1 tab.

By default, VACUUM skips the sort phase for any table where more than 95 percent of the table’s rows are already sorted. Skipping the sort phase can significantly improve VACUUM performance. To change the default sort or delete threshold for a single table, include the table name and the TO threshold PERCENT parameter when you run VACUUM; for example, **VACUUM imdb.name\_filmography to 100 percent;**.

VERIFY THE VACUUM OPERATION

1. **Copy/Paste:** Delete everything in the **Statement 1** tab. Then, copy and paste in the following SQL statement. This will return the table statistics for the **name\_filmography** table.

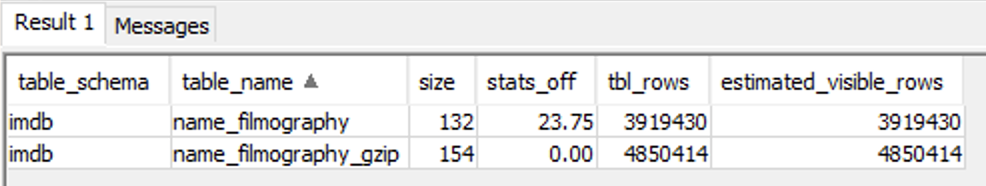
SELECT schema as table\_schema, "table" as table\_name, size, stats\_off, tbl\_rows, estimated\_visible\_rows

FROM svv\_table\_info d

WHERE "table" like 'name\_filmography%';

1. To run the statement, click the play  button just above the Statement 1 tab.

Note the new size of the **name\_filmography** table after the VACUUM operation.



Inspect the **size**, **stats\_off**, **tbl\_rows**, and **estimated\_visible\_rows** column values. These values provide the following information:

* **size:** Size of the table, in 1-MB data blocks.
* **stats\_off:** Number that indicates how stale the table’s statistics are; 0 is current, 100 is out of date.
* **tbl\_rows:** Total number of rows in the table. This value includes rows marked for deletion but not yet vacuumed.
* **estimated\_visible\_rows:** The estimated rows in the table. This value does not include rows marked for deletion.

You can use this general query to see details for all the VACUUM operations in your schema. This is a useful query for your production workloads.

**Consider Sample Code:**

SELECT vac\_start.userid

, vac\_start.xid

, vac\_start.table\_id

, tab.schema\_name AS schema\_name

, tab.table\_name AS table\_name

, TRIM(vac\_start.status) AS vac\_type

, TRIM(vac\_end.status) AS result

, NVL(vac\_end."rows",vac\_start."rows") AS end\_rows

, (end\_rows - vac\_start."rows") AS rows\_delta

, NVL(vac\_end."blocks",vac\_start."blocks") AS end\_blocks

, (end\_blocks - vac\_start."blocks") AS blocks\_delta

, vac\_start."eventtime" AS start\_time

, DATEDIFF(seconds,vac\_start."eventtime",NVL(vac\_start."eventtime",vac\_end."eventtime")) AS duration

FROM stl\_vacuum vac\_start

LEFT JOIN stl\_vacuum vac\_end

ON vac\_start.userid = vac\_end.userid

AND vac\_start.table\_id = vac\_end.table\_id

AND vac\_start.xid = vac\_end.xid

AND (vac\_end.status NOT LIKE 'Start%'

AND vac\_end.status NOT LIKE 'Skip%')

LEFT JOIN (SELECT TRIM(pgn.nspname) AS schema\_name

, TRIM(name) AS table\_name

, tbl.id AS table\_id

FROM stv\_tbl\_perm tbl

JOIN pg\_class pgc ON pgc.oid = tbl.id

JOIN pg\_namespace pgn ON pgn.oid = pgc.relnamespace

GROUP BY 1,2,3

) tab

ON tab.table\_id = vac\_start.table\_id

WHERE vac\_start.status LIKE 'Start%'

OR vac\_start.status LIKE 'Skip%'

ORDER BY start\_time DESC

RUN AN ANALYZE OPERATION

The ANALYZE operation updates the statistical metadata that the query optimizer uses to choose execution plans. In most cases, you don’t need to explicitly run the ANALYZE command. Amazon Redshift monitors changes to your workload and automatically updates statistics in the background. In addition, the COPY command performs an analysis automatically when it loads data into an empty table.

To explicitly analyze a table or the entire database, run the ANALYZE command.

1. **Copy/Paste:** Delete everything in the **Statement 1** tab. Then, copy and paste in the following SQL statement:

ANALYZE verbose imdb.name\_filmography;

1. To run the statement, click the play  button just above the Statement 1 tab.

 ANALYZE best practices:

* Run the ANALYZE command before running queries.
* Run the ANALYZE command on the database routinely at the end of every regular load or update cycle.
* Run the ANALYZE command on any new tables that you create and any existing tables or columns that undergo significant change.
* Consider running ANALYZE operations on different schedules for different types of tables and columns, depending on their use in queries and their propensity to change.

 Congratulations! You have successfully used the VACUUM and ANALYZE commands.

**Task 7: Optimize with SORTKEY and DISTKEY**

Amazon Redshift stores your data on disk in sorted order according to the sort key. The Amazon Redshift query optimizer uses sort order when it determines query execution plans. Queries performed against sorted data are more efficient because they can skip entire blocks of data that fall outside of the requested range.

The column designated with DISTKEY is the column that is used to distribute values on each node. Rows with the same value in this column are guaranteed to be on the same node. The column designated with SORTKEY is the column that is used to sort duplicate values in the DISTKEY column.

CREATE TABLES WITH A SORTKEY AND DISTKEY

The following three tables contain the same data as in previous tasks. The difference is that a DISTKEY and SORTKEY have been assigned.

1. **Copy/Paste:** Delete everything in the **Statement 1** tab. Then, copy and paste in the following SQL statements:

BEGIN;

CREATE TABLE imdb.name\_display\_dskey (

"nameId" char(10) NOT NULL DISTKEY,

"name" varchar(128) NOT NULL,

"realName" varchar(255) DEFAULT NULL,

"akas" varchar(2000),

"image" varchar(192) DEFAULT NULL,

"imageId" char(12) DEFAULT NULL,

"nicknames" varchar(2000)

);

CREATE TABLE imdb.title\_display\_dskey(

"titleId" char(9) NOT NULL DISTKEY,

"title" text NOT NULL,

"year" int DEFAULT NULL,

"adult" int DEFAULT NULL,

"runtimeMinutes" int DEFAULT NULL,

"imageUrl" varchar(175) DEFAULT NULL,

"imageId" char(12) DEFAULT NULL,

"type" char(12) DEFAULT NULL,

"originalTitle" text

);

CREATE TABLE imdb.title\_cast\_dskey (

"titleId" char(9) NOT NULL DISTKEY,

"ordering" int NOT NULL,

"nameId" char(10) NOT NULL,

"category" char(15) NOT NULL,

"role" int DEFAULT NULL,

"characters" varchar(2500),

"characterIds" varchar(250),

"attributes" varchar(64) DEFAULT NULL

)

SORTKEY (titleid, ordering);

COMMIT;

**Note:** Notice lines 4, 13, and 24. This is where the DISTKEY is assigned in each table. Line 33 assigns the SORTKEY to the **titleid** and **ordering** columns in the **title\_cast\_dskey** table.

1. To run the statement, click the play  button just above the Statement 1 tab.

Next, you will import the data into these new tables.

1. **Copy/Paste:** Return to the helper webpage. Delete everything in the large text box, and then copy and paste the following SQL statements into it:

BEGIN;

COPY imdb.title\_cast\_dskey FROM 's3://(LabDataBucket)/title-cast/split/'

iam\_role '(RedshiftAccessRoleArn)'

delimiter '\t' timeformat 'YYYY-MM-DD HH:MI:SS' NULL AS 'NULL' region '(Region)';

COPY imdb.name\_display\_dskey FROM 's3://(LabDataBucket)/name-display/split/'

iam\_role '(RedshiftAccessRoleArn)'

delimiter '\t' timeformat 'YYYY-MM-DD HH:MI:SS' NULL AS 'NULL' region '(Region)';

COPY imdb.title\_display\_dskey FROM 's3://(LabDataBucket)/title-display/split/'

iam\_role '(RedshiftAccessRoleArn)'

delimiter '\t' timeformat 'YYYY-MM-DD HH:MI:SS' NULL AS 'NULL' region '(Region)';

COMMIT;

1. Click **Replace and Copy**.
2. Return to the **remote connection session**.
3. In the **Statement 1** tab, delete everything currently in the text box, and paste the updated code.
4. To run the statement, click the play  button just above the Statement 1 tab.

TEST THE DISTKEY AND SORTKEY

You will begin by running a baseline query against the original tables where no SORTKEY or DISTKEY is added. This query is to list all the actors in each title in the order they appear in the movie.

1. **Copy/Paste:** Delete everything in the **Statement 1** tab. Then, copy and paste in the following SQL statement:

select b.title, a.name, c.ordering

from imdb.name\_display a, imdb.title\_display b,imdb.title\_cast c

where a.nameid = c.nameid

and b.titleid = c.titleid

order by c.titleid, c.ordering

1. To run the statement, click the play  button just above the Statement 1 tab.

Note the query response time, which is located in the bottom-right corner of the window.

Now you will run the same query against the new tables, which are using SORTKEY and DISTKEY.

1. **Copy/Paste:** Delete everything in the **Statement 1** tab. Then, copy and paste in the following SQL statement:

select b.title, a.name, c.ordering

from imdb.name\_display\_dskey a, imdb.title\_display\_dskey b,imdb.title\_cast\_dskey c

where a.nameid = c.nameid

and b.titleid = c.titleid

order by c.titleid, c.ordering

1. To run the statement, click the play  button just above the Statement 1 tab.

Note the query response time for this query and compare it with the benchmark query response time. The improvement you see is because of the new DISTKEY and SORTKEY.

 Congratulations! You have successfully optimized with SORTKEY and DISTKEY.

**Task 8: View query statistics with the console**

In this task, you will open Amazon Redshift console to review the queries that you ran. The **Queries** tab shows a list of queries that you ran over a time period you specify. By default, the console displays queries that have run in the last 24 hours, including currently running queries.

1. Return to the tab with the **AWS Management Console**, click **Services** and click **Amazon Redshift**.
2. In the left navigation pane, click **Clusters**.
3. Click the cluster name.
4. Click on **Query monitoring** tab.
5. Scroll down to the **Queries and loads** section.

**Note:** If you are using the original console, at the top of the page, click the **Queries** tab.

1. In the dropdown box, select **All queries**.
2. Open any query by clicking the number ID in the **Query** column.

Explore the details and stats for the query.

 Congratulations! You have successfully viewed query statistics with the console.

**Task 9: Run specific scenario queries**

In this task, you will write queries based on a scenario. The idea here is to challenge yourself and write the query before looking at the answer.

1. Return to the **remote connection session**.
2. Write and perform queries to match the following requirements:

* List all movies for a specific genre.
  + [Click here](https://labs.skillbuilder.aws/sa/lab/arn%3Aaws%3Alearningcontent%3Aus-east-1%3A470679935125%3Ablueprintversion%2FSPL-TF-200-DBWARC-1%3A1.0.4-fac40e80/en-US#solution-1) to review the solution.
* List all the actors in movies that were made between 1990 and 1994 and are 2 hours in length or longer.
  + [Click here](https://labs.skillbuilder.aws/sa/lab/arn%3Aaws%3Alearningcontent%3Aus-east-1%3A470679935125%3Ablueprintversion%2FSPL-TF-200-DBWARC-1%3A1.0.4-fac40e80/en-US#solution-2) to review the solution.
* Find all the persons who were part of the movie “The Firm”.
  + [Click here](https://labs.skillbuilder.aws/sa/lab/arn%3Aaws%3Alearningcontent%3Aus-east-1%3A470679935125%3Ablueprintversion%2FSPL-TF-200-DBWARC-1%3A1.0.4-fac40e80/en-US#solution-3) to review the solution.

1. Once you have completed these challenges, close the remote connection session.

 Congratulations! You have successfully run specific scenario queries.

**Conclusion**

 Congratulations! You have successfully

* Used the SQL Workbench for Amazon Redshift
* Understood the COPY command for loading data and working with compression
* Used a manifest file for data imports
* Archived data using the UNLOAD command
* Used ANALYZE and VACUUM operations
* Used the Amazon Redshift console to explore query statistics

**End lab**

Follow these steps to close the console and end your lab.

1. Return to the **AWS Management Console**.
2. At the upper-right corner of the page, choose **AWSLabsUser**, and then choose **Sign out**.
3. Choose **End lab** and then confirm that you want to end your lab.

**Additional resources**

For more information about AWS Training and Certification, see [*https://aws.amazon.com/training/*](https://aws.amazon.com/training/).

*Your feedback is welcome and appreciated.*  
If you would like to share any feedback, suggestions, or corrections, please provide the details in our [*AWS Training and Certification Contact Form*](https://support.aws.amazon.com/#/contacts/aws-training).

**Appendix**

SOLUTION 1

select a.titleid, a.genres, b.title, b.year, b.runtimeminutes, b.type, b.originaltitle

from imdb.title\_genres\_gzip a, imdb.title\_display\_gzip b

where a.titleid = b.titleid

and a.genres = 'Drama';

SOLUTION 2

select c.year, a.name, c.title, c.runtimeMinutes

from imdb.name\_display\_gzip a, imdb.title\_cast\_gzip b, imdb.title\_display\_gzip c

where a.nameId = b.nameId

and b.titleId = c.titleId

and c.runtimeminutes >= 120

and c.year between 1990 and 1994

order by year, title;

SOLUTION 3

select a.name, a.nameId, a.realName, a.image, c.title, c.year, c.runtimeMinutes, c.type, c.originalTitle, d.status

from imdb.name\_display\_gzip a, imdb.title\_cast\_gzip b, imdb.title\_display\_gzip c, imdb.name\_filmography\_gzip d

where a.nameId = b.nameId

and b.titleId = c.titleId

and c.titleId = d.titleId

and a.nameId = d.nameId

and c.title = 'The Firm';

INSTRUCTIONS FOR USING THE ORIGINAL AMAZON REDSHIFT CONSOLE

1. In the left navigation pane, click **Clusters**.
2. To display the details for the cluster, click the checkbox to the left of the cluster name.

**Note:** If no clusters are listed, make sure that the AWS Region in the console matches the **Region** value from the left side of the lab page.

1. Just above the cluster information, click **Manage IAM roles**
2. For **Available roles**, select the **RedshiftAccessRole-xxxxx** role, and click **Apply changes**

**Note:** The full role name does not display until you select it.

This is the AWS Identity and Access Management (IAM) role that authorizes Amazon Redshift to access the external data catalog and Amazon S3 on your behalf. You must associate this role with your Amazon Redshift cluster for data load and unload activities to work.

[Click here](https://labs.skillbuilder.aws/sa/lab/arn%3Aaws%3Alearningcontent%3Aus-east-1%3A470679935125%3Ablueprintversion%2FSPL-TF-200-DBWARC-1%3A1.0.4-fac40e80/en-US#continueInstructions) to continue to the next task.

**Connect to the Windows Instance using Apache Guacamole**

* In the **Connection Details** section in the lab console, copy the **GuacamoleLink** and paste it into a browser.
* Go back to the lab console and copy the **GuacamolePassword** to the clipboard.
* Go to the Apache Guacamole sign in in the browser. Sign in by using the following steps:
* For **Username**, enter:

student

* For **Password**, paste the **GuacamolePassword** from the clipboard.
* Select **Log In**.

Your connection to your remote instance should start momentarily. Once you open a connection, you will see an image of the instance desktop. You can interact with this image just as you would your normal desktop, or any remote desktop client.

You are now connected to your Windows instance in the browser via Guacamole.

**Hint:** Web browsers don’t provide access to clipboard data, which means synchronization between your local clipboard and the remote clipboard is impossible. To copy and paste when using Guacamole, you must use the Clipboard editor. To open the Clipboard editor, press **Command -> Control -> Shift**.

Copy your text and paste it to the Clipboard editor. This will set the clipboard of your instance to what you just pasted. You can also edit the text that you place in the Clipboard editor before pasting into your remote desktop. To close the Clipboard editor, press **Command -> Control -> Shift**.

ICON KEY

Various icons are used throughout this lab to call attention to different types of instructions and notes. While not all of the icons will be used, the following list explains the purpose for each icon:

* **Command:** A command that you must run.
* **Expected output:** A sample output that you can use to verify the output of a command or edited file.
* **Note:** A note, tip, or important guidance.
* **Copy/Paste:** A code block that displays the contents of a script or file you need to copy and paste that has been pre-created for you. When you need to copy only a certain part of a code block, there will be numbered **TODO** comments in the code.
* **WARNING:** An action that is irreversible and could potentially impact the failure of a command or process (including warnings about configurations that cannot be changed after they are made).
* **Additional information:** Where to find more information.

[Return to the instructions](https://labs.skillbuilder.aws/sa/lab/arn%3Aaws%3Alearningcontent%3Aus-east-1%3A470679935125%3Ablueprintversion%2FSPL-TF-200-DBWARC-1%3A1.0.4-fac40e80/en-US#prerequisites)