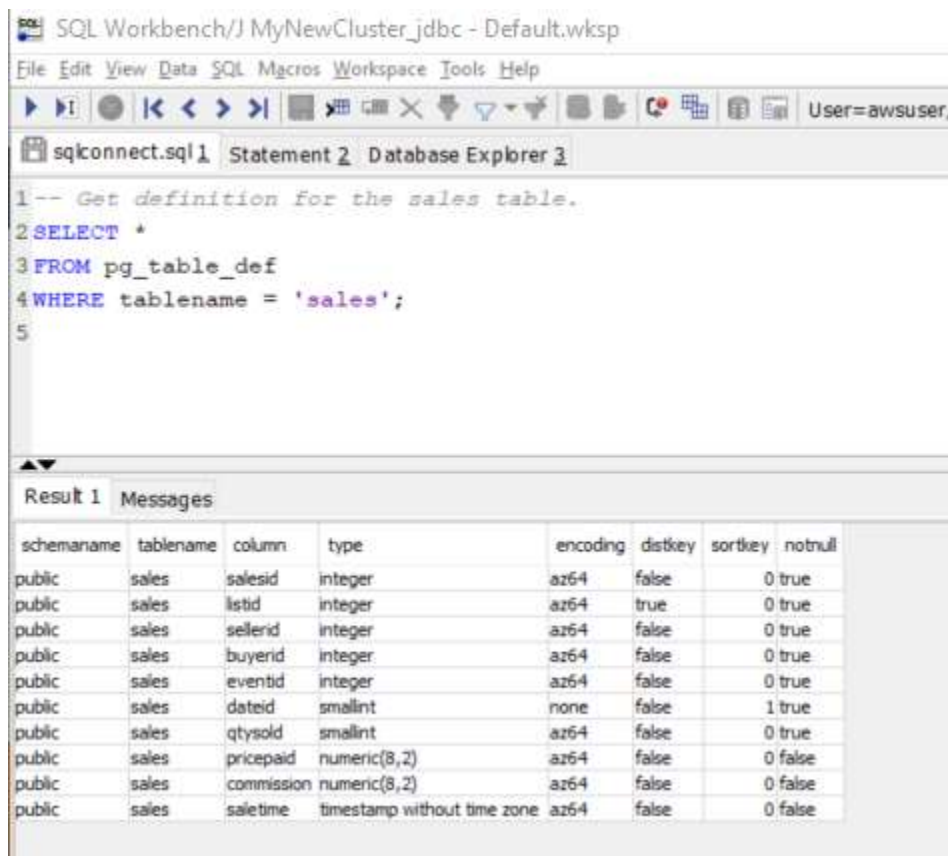


HOE 4 - Setup and Use SQL Workbench to Access Cluster remotely

In this HOE we will use the free SQL Workbench. Workbench/J is a free, DBMS-independent, cross-platform SQL query tool. It is written in Java and should run on any operating system that provides a Java Runtime Environment.

Its main focus is on running SQL scripts (either interactively or as a batch) and export/import features.



The screenshot shows the SQL Workbench/J interface. The title bar reads "SQL Workbench/J MyNewCluster_jdbc - Default.wksp". The menu bar includes File, Edit, View, Data, SQL, Macros, Workspace, Tools, and Help. The toolbar contains various icons for file operations, execution, and navigation. The main editor displays a SQL query in "Statement 2":

```
1 -- Get definition for the sales table.
2 SELECT *
3 FROM pg_table_def
4 WHERE tablename = 'sales';
5
```

Below the editor, the "Result 1" tab is active, showing a table with the following data:

| schemaname | tablename | column | type | encoding | distkey | sortkey | notnull |
|------------|-----------|------------|-----------------------------|----------|---------|---------|---------|
| public | sales | salesid | integer | az64 | false | 0 | true |
| public | sales | listid | integer | az64 | true | 0 | true |
| public | sales | sellerid | integer | az64 | false | 0 | true |
| public | sales | buyerid | integer | az64 | false | 0 | true |
| public | sales | eventid | integer | az64 | false | 0 | true |
| public | sales | dateid | smallint | none | false | 1 | true |
| public | sales | qtysold | smallint | az64 | false | 0 | true |
| public | sales | pricepaid | numeric(8,2) | az64 | false | 0 | false |
| public | sales | commission | numeric(8,2) | az64 | false | 0 | false |
| public | sales | saletime | timestamp without time zone | az64 | false | 0 | false |

You can download the appropriate library version for your deployment here

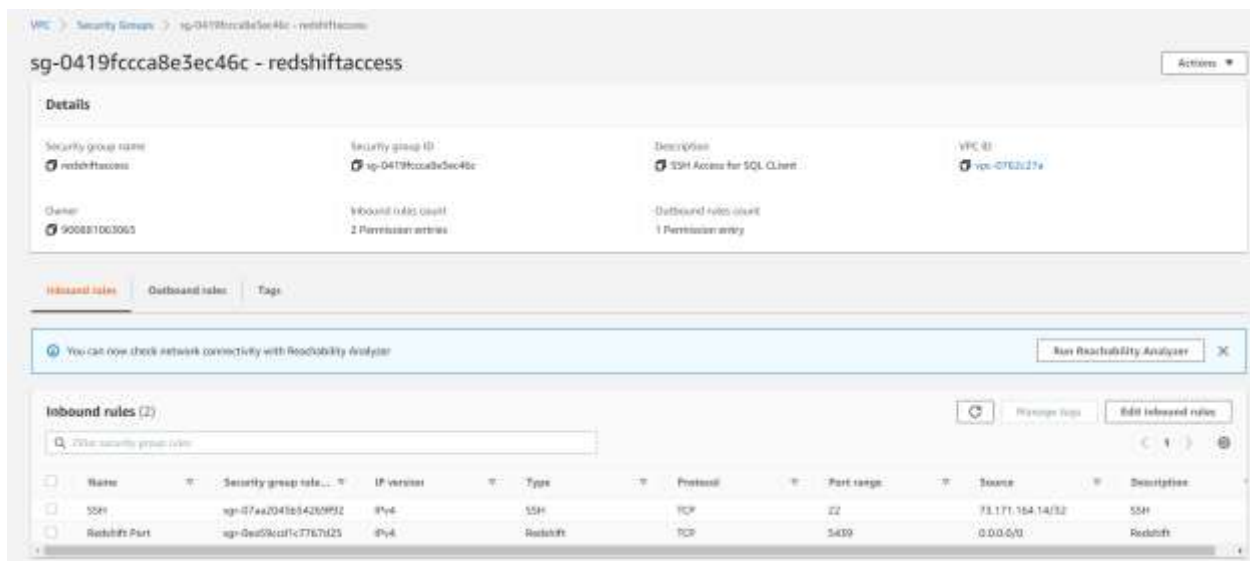
<https://www.sql-workbench.eu/>

There is also the requirement for drivers for Redshift.

<https://docs.aws.amazon.com/redshift/latest/mgmt/configure-jdbc-connection.html#download-jdbc-driver>

Note: Before you can access the cluster remotely(outside of AWS) make sure you have setup your security group in the VPC you're using to allow for Redshift port 5439

(Follow the Network and Security HOE)



The screenshot shows the AWS Management Console interface for a security group named 'sg-0419fcca8e3ec46c - redshiftaccess'. The console displays the following details:

- Security group name:** redshiftaccess
- Security group ID:** sg-0419fcca8e3ec46c
- Description:** SSH Access for SQL Client
- VPC ID:** vpc-c7f62c23a
- Owner:** 900881003063
- Inbound rules count:** 2 Permission entries
- Outbound rules count:** 1 Permission entry

The console also shows a notification: "You can now check network connectivity with Reachability Analyzer".

The **Inbound rules (2)** section displays the following rules:

| Name | Security group rule... | IP version | Type | Protocol | Port range | Source | Description |
|---------------|------------------------|------------|----------|----------|------------|------------------|-------------|
| SSH | sg-07aa2043e54269932 | IPv4 | SSH | TCP | 22 | 73.171.164.14/32 | SSH |
| Redshift Port | sg-0e059c0f1c7767025 | IPv4 | Redshift | TCP | 5439 | 0.0.0.0/0 | Redshift |

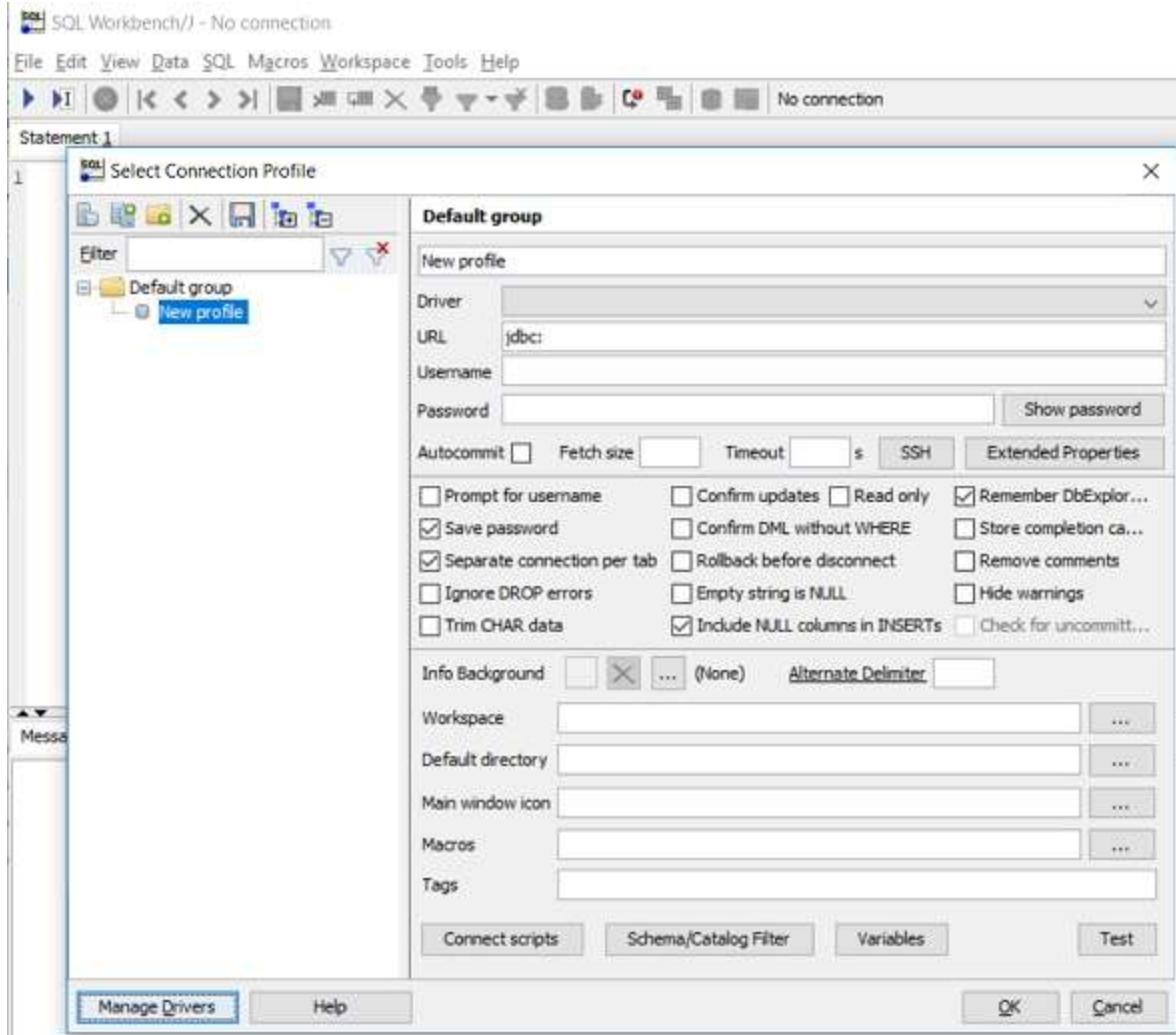
| This PC > Desktop > redshift-jdbc42-2.1.0.1 | | | | |
|---|---------------------------------------|--------------------|---------------------|----------|
| | Name | Date modified | Type | Size |
| | aws-java-sdk-core-1.12.23 | 10/12/2021 6:39 PM | Executable Jar File | 997 KB |
| | aws-java-sdk-redshift-1.12.23 | 10/12/2021 6:39 PM | Executable Jar File | 1,346 KB |
| | aws-java-sdk-redshift-internal-1.12.x | 10/12/2021 6:39 PM | Executable Jar File | 31 KB |
| | aws-java-sdk-sts-1.12.23 | 10/12/2021 6:39 PM | Executable Jar File | 147 KB |
| | commons-codec-1.15 | 10/12/2021 6:39 PM | Executable Jar File | 346 KB |
| | commons-logging-1.2 | 10/12/2021 6:39 PM | Executable Jar File | 61 KB |
| | httpclient-4.5.13 | 10/12/2021 6:39 PM | Executable Jar File | 763 KB |
| | httpcore-4.4.13 | 10/12/2021 6:39 PM | Executable Jar File | 321 KB |
| | jackson-annotations-2.12.3 | 10/12/2021 6:39 PM | Executable Jar File | 74 KB |
| | jackson-core-2.12.3 | 10/12/2021 6:39 PM | Executable Jar File | 357 KB |
| | jackson-databind-2.12.3 | 10/12/2021 6:39 PM | Executable Jar File | 1,481 KB |
| | jackson-dataformat-cbor-2.12.3 | 10/12/2021 6:39 PM | Executable Jar File | 61 KB |
| | joda-time-2.8.1 | 10/12/2021 6:39 PM | Executable Jar File | 608 KB |
| | redshift-jdbc42-2.1.0.1 | 10/12/2021 6:39 PM | Executable Jar File | 978 KB |

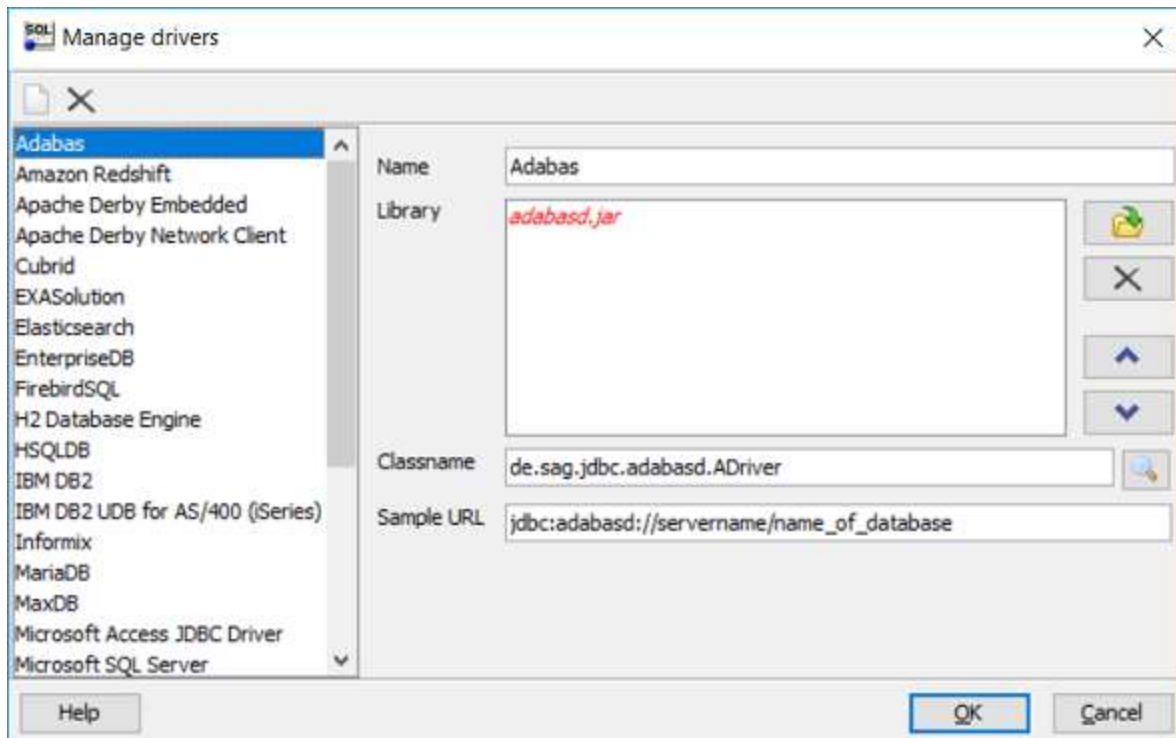
When your ready launch SQL Workbench using **SQLWorkbench64.exe**

“Select Connection Profile” dialog screen can ask you to create the profile for database connection that you want to work with.

For this simple configuration, we will first define the Amazon Redshift JDBC driver to the SQL Workbench tool.

Click Manage Drivers button at the bottom of the screen.





Press “Create new entry” icon at the top of the screen of Manage Drivers

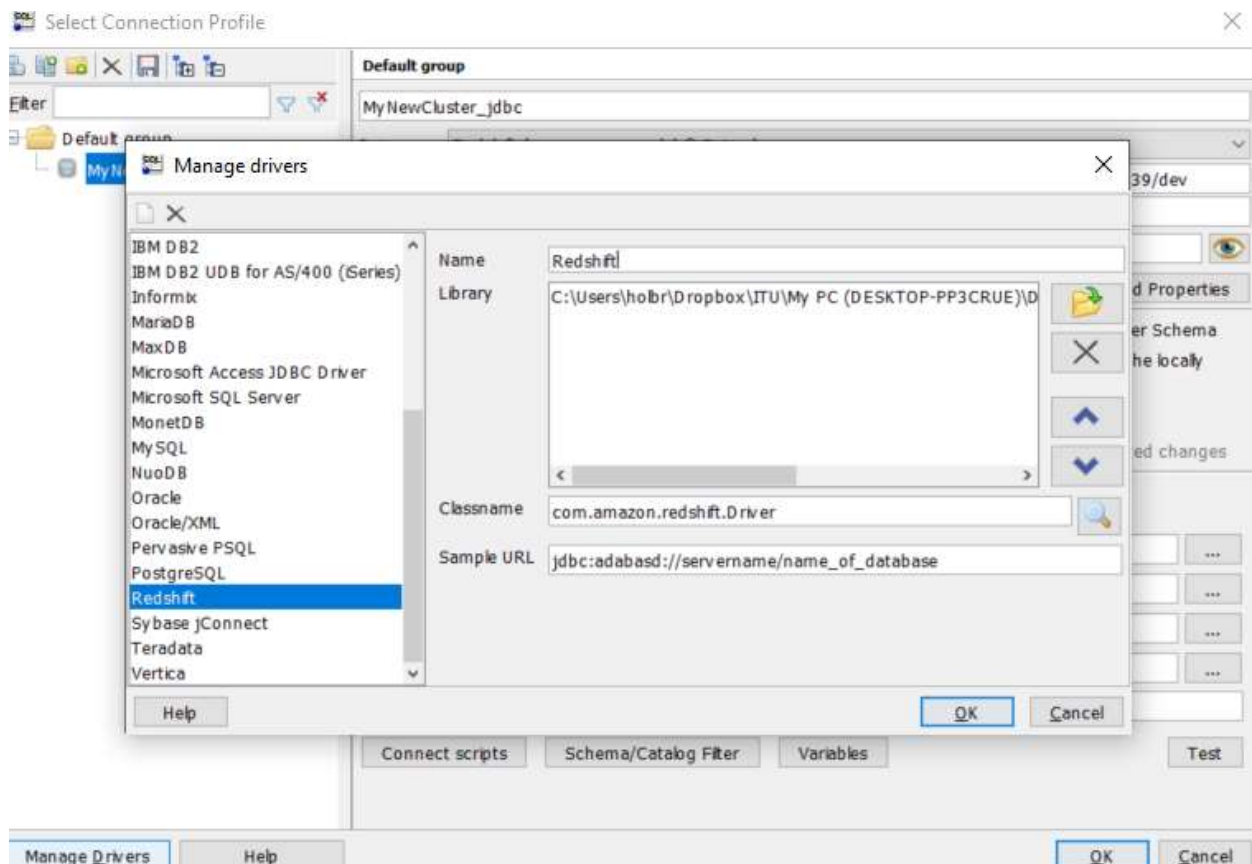
Fill the input text areas as seen below.

Using file selection icon, point to the .jar file you have download as JDBC Driver for Amazon Redshift in previous steps.

The class name field will be populated as **com.amazon.redshift.jdbc**.

A driver which is the default class name. You can also accept this class name for JDBC driver.

Then press the OK button.



Create a Profile to Connect Amazon Redshift Database

After the JDBC specified is defined, we can create a new profile to connect to a specific Amazon Redshift database.

On the main menu, follow menu options **“File > Connect window”**

On the **“New Profile”** screen we will define a new configuration to connect to your target Amazon Redshift cluster and database.

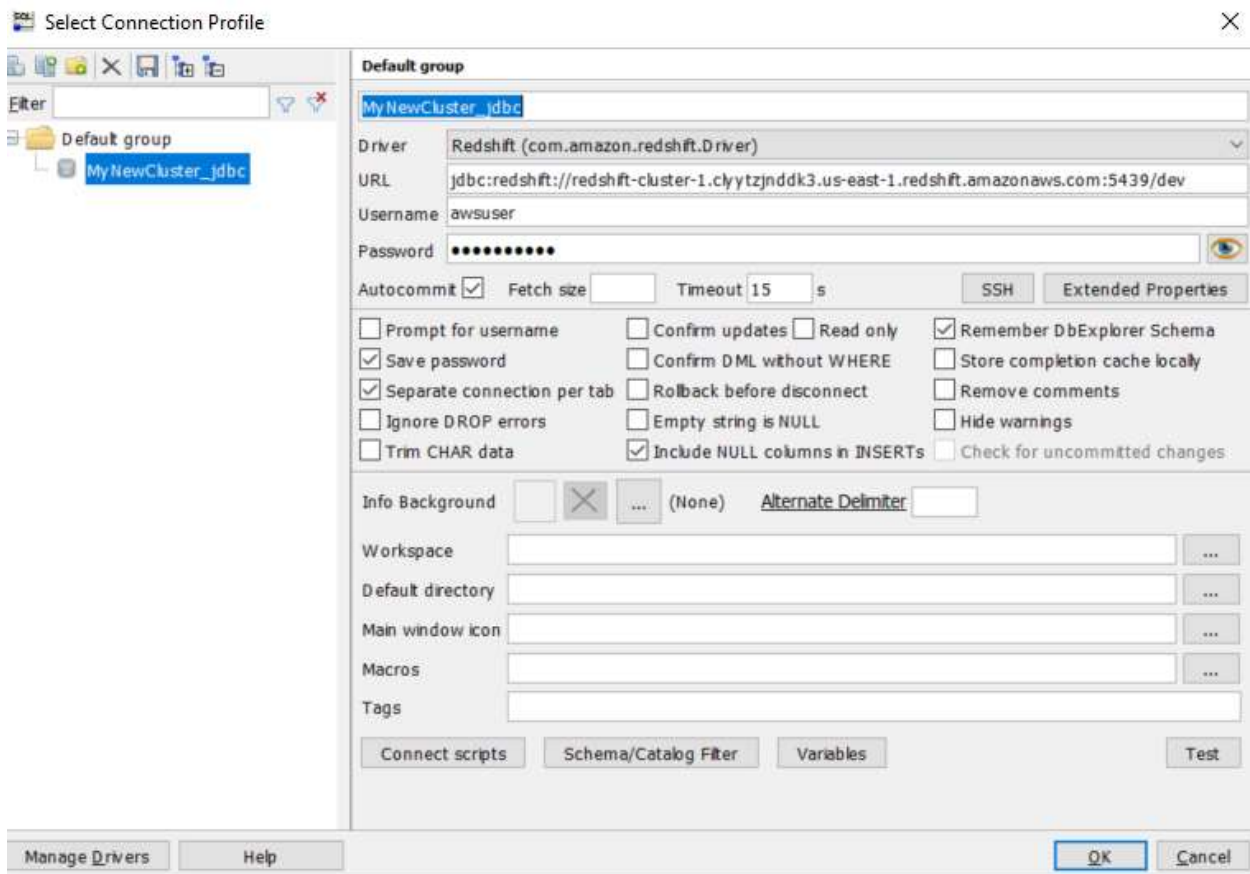
Give a descriptive name to your database connection profile.

In the Driver combo box, choose the driver entry you have created in the previous step (Amazon Redshift).

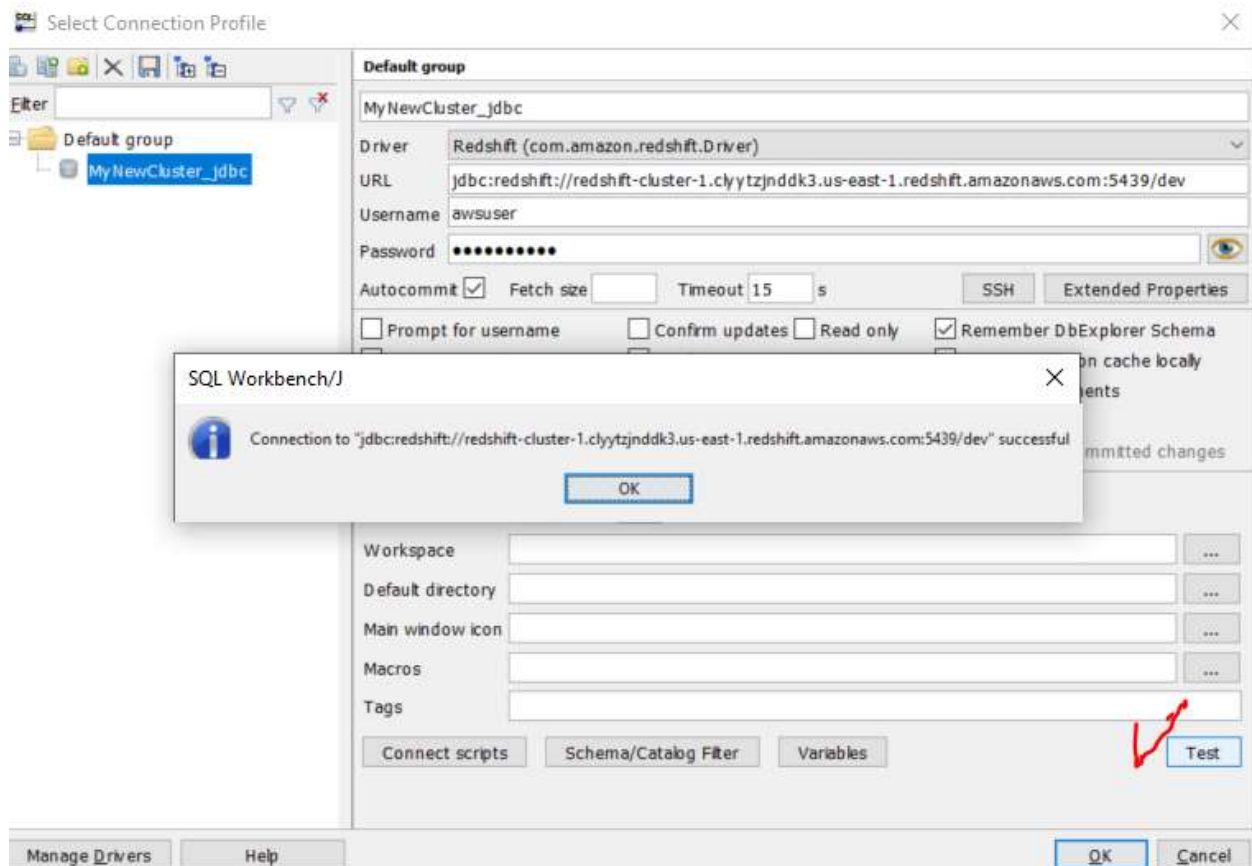
In URL entry, type the connection URL which is in a similar format as

jdbc:redshift://mydemocluster.clyytzjnddk3.us-east-1.redshift.amazonaws.com:5439/dev

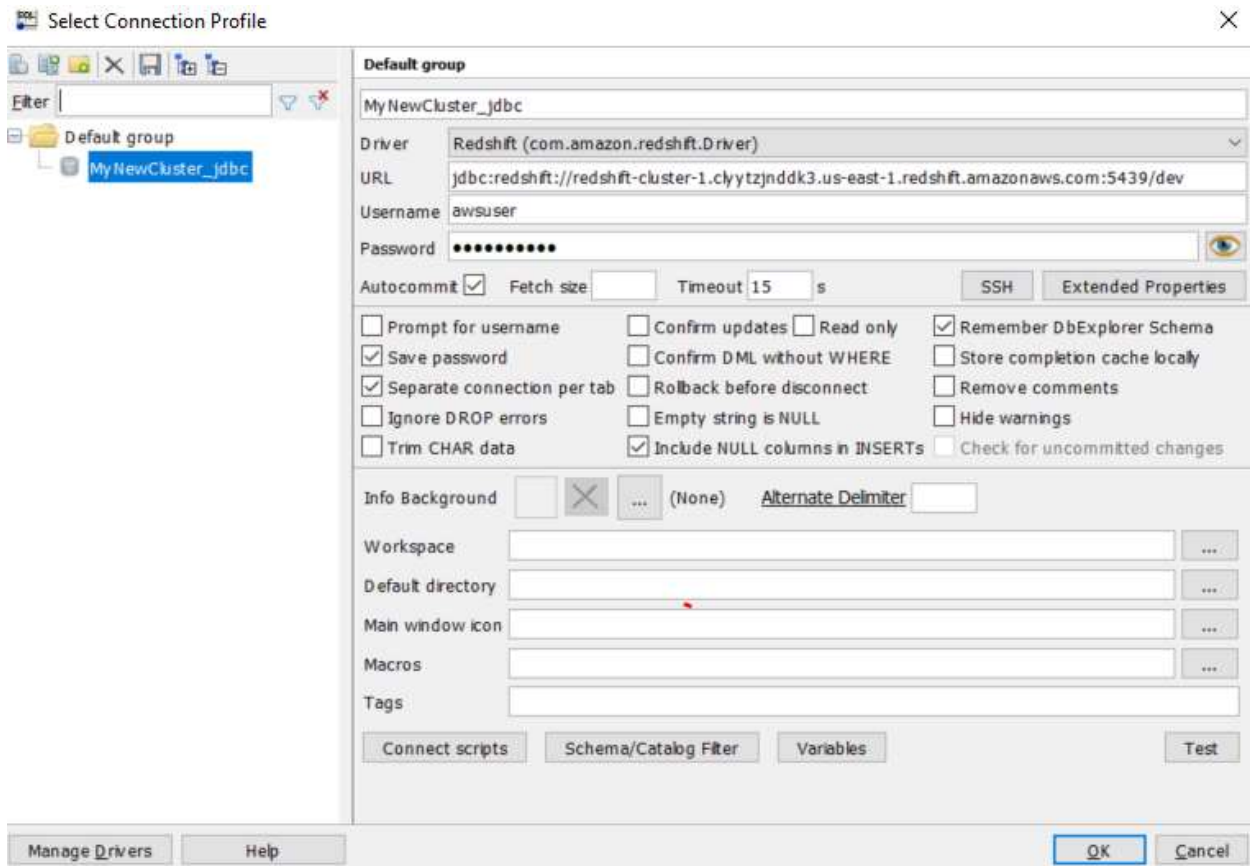
In username and password, textboxes enter the credentials of the database user you have to connect to Amazon Redshift cluster database.



Select Test button on-screen to see if your database connection works successfully.



Press OK to save your Amazon Redshift connection profile.



You can execute following SQL SELECT command too to see if the database connection is successfully working on SQL Workbench

-- Get definition for the sales table.

SELECT *

FROM pg_table_def

WHERE tablename = 'sales';

SQL Workbench/J MyNewCluster_jdbc - Default.wksp

File Edit View Data SQL Macros Workspace Tools Help

sqlconnect.sql 1 Statement 2 Database Explorer 3

```

1 -- Get definition for the sales table.
2 SELECT *
3 FROM pg_table_def
4 WHERE tablename = 'sales';
5

```

Result 1 Messages

| schemaname | tablename | column | type | encoding | distkey | sortkey | notnull |
|------------|-----------|------------|-----------------------------|----------|---------|---------|---------|
| public | sales | salesid | integer | az64 | false | 0 | true |
| public | sales | listid | integer | az64 | true | 0 | true |
| public | sales | sellerid | integer | az64 | false | 0 | true |
| public | sales | buyerid | integer | az64 | false | 0 | true |
| public | sales | eventid | integer | az64 | false | 0 | true |
| public | sales | dateid | smallint | none | false | 1 | true |
| public | sales | qtysold | smallint | az64 | false | 0 | true |
| public | sales | pricepaid | numeric(8,2) | az64 | false | 0 | false |
| public | sales | commission | numeric(8,2) | az64 | false | 0 | false |
| public | sales | saletime | timestamp without time zone | az64 | false | 0 | false |

End of Hands on Exercise