

Using RDF Knowledge Graphs in the Oracle Public Cloud

1 OVERVIEW

This document provides detailed steps to setup and use RDF Semantic Graph as part of an Oracle Database Cloud Service. The steps are based on Oracle Database version 18.0.0.0 and may vary if a different database version is used. In addition, these steps assume a Windows 10 desktop as the client computer.

This how-to covers the following tasks:

- Provisioning a Database Cloud Service (DBCS) instance
- Using Putty to login to the newly created DBCS instance
- Connecting SQL Developer to the DBCS instance
- Using the RDF Semantic Graph plugin for SQL Developer to setup and configure RDF Semantic Graph
- Using the RDF Semantic Graph plugin to bulk load and query a publicly-available RDF dataset
- Setting up a Fuseki SPARQL endpoint on the DBCS instance to provide a W3C-standard REST interface

2 BEFORE YOU GET STARTED

The following tasks should be completed before starting this how-to:

- Become familiar with DBCS, Oracle Spatial and Graph, and RDF Semantic Graph
 - DBCS Help Center: <https://docs.oracle.com/en/cloud/paas/database-dbaas-cloud/index.html>
 - Oracle Spatial and Graph:
<http://www.oracle.com/technetwork/database/options/spatialandgraph/overview/index.html>
 - RDF Semantic Graph: <https://docs.oracle.com/en/database/oracle/oracle-database/18/rdfrm/index.html>
- Obtain an Oracle Cloud subscription or trial account including DBCS and Java Cloud Service (JCS)
 - Available at: <http://cloud.oracle.com>
- Download and install Puttygen and Putty onto your client computer
 - Available at: <http://www.putty.org>
- Download and install Cygwin onto your client computer
 - Available at: <https://www.cygwin.com/>
- Download and install WinSCP onto your client computer
 - Available at: <https://winscp.net/eng/index.php>

- Download and install Oracle SQL Developer onto your client computer
 - Available at: <http://www.oracle.com/technetwork/developer-tools/sql-developer/overview/index.html>

3 PROVISIONING AN ORACLE CLOUD DATABASE SERVICE (DBCS) INSTANCE

The steps below illustrate how to provision an Oracle DBCS instance.

Log into your Oracle Cloud account at cloud.oracle.com. We will be using a Traditional Cloud Account for this exercise.

The screenshot shows the Oracle Cloud Sign In page in Mozilla Firefox. The URL in the address bar is https://cloud.oracle.com/en_US/sign-in. The page has a header with the ORACLE Cloud logo and navigation links for Contact, Chat, English, Estimate, Buy, and Try for Free. Below the header, there are dropdown menus for Applications, Platform, Infrastructure, and Resources, along with a search bar.

Cloud Account section:

- Traditional Cloud Account dropdown menu.
- US Commercial 2 (us2) dropdown menu.
- Administrators** list:
 - Perform administrative tasks for individual services
 - Monitor key metrics and notifications for services
 - Manage users and roles for cloud services
 - Access the console for services you administer
- Users** list:
 - Launchpad to all of your applications and services
 - Change your identity details and preferences

Trouble Signing In? (Orange callout box):

Need a list of all your Oracle Cloud accounts and services? We can help.

You'll receive an email listing the details of all your Oracle Cloud accounts and services that are associated with the specified email address.

[Account Details >](#)

Order Management section:

Complete orders and initiate cloud service provisioning

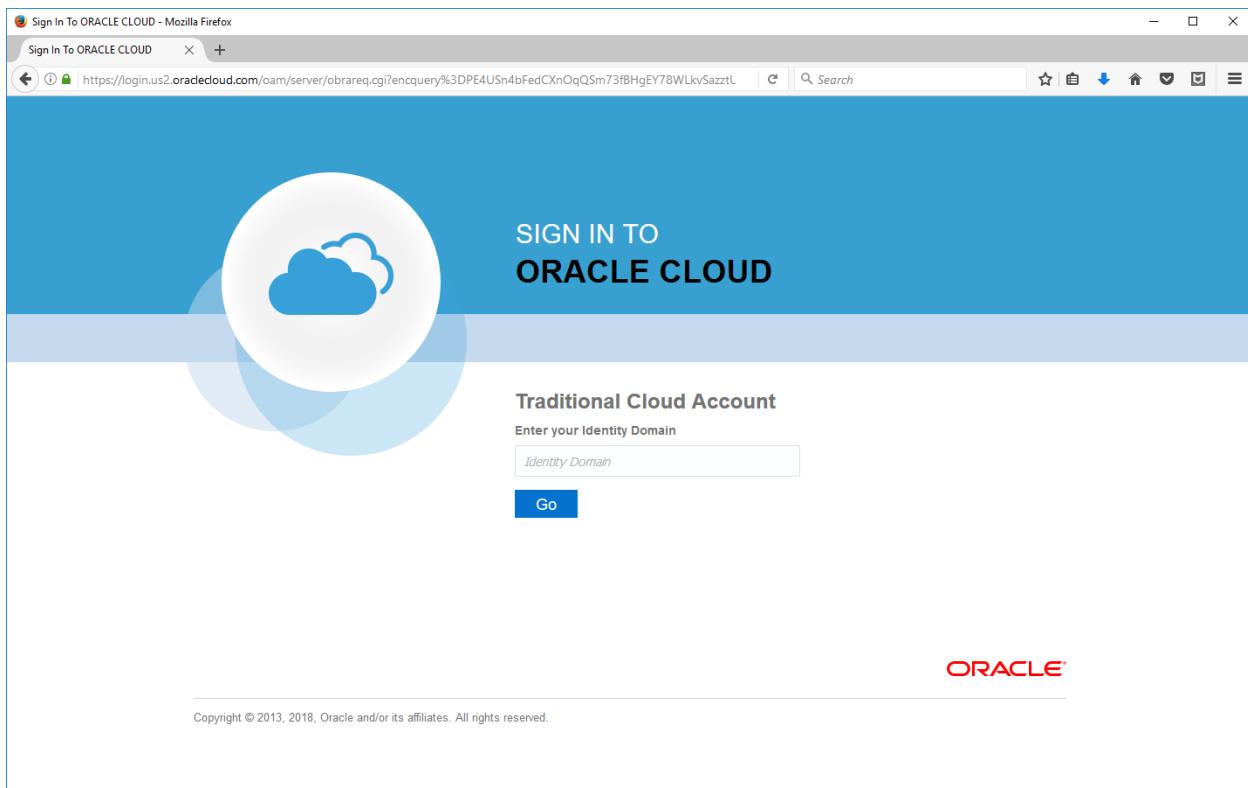
Activate paid services and trial requests

Monitor cloud services across data centers

[My Account >](#)

At the bottom of the page, there is a footer with links for Integrated Cloud Applications & Platform Services, About Oracle, Contact Us, Terms of Use and Privacy, and Cookie Preferences. There are also social media sharing icons for Facebook, LinkedIn, Twitter, and YouTube.

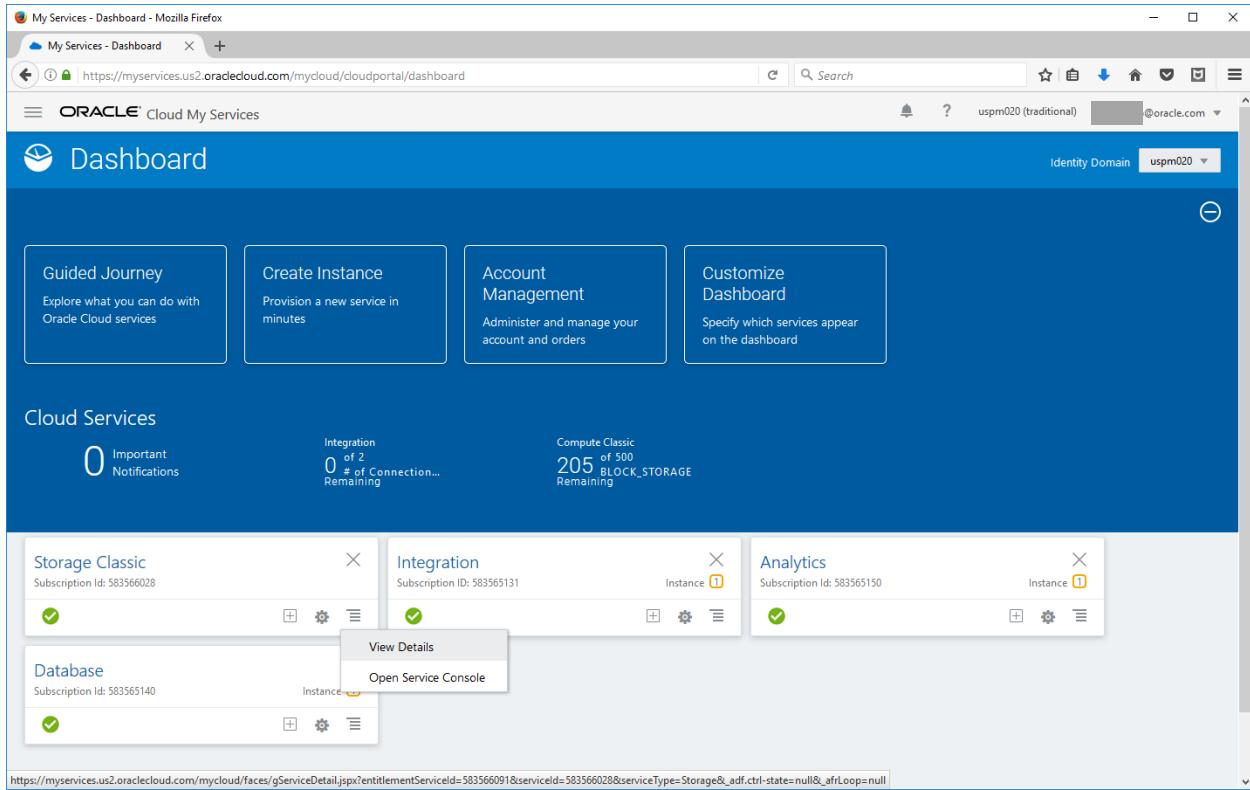
Enter your identity domain and then enter your login and password on the next screen.



After logging in, you will see your Dashboard for managing cloud services. You can click on Customize Dashboard to choose which services will appear.

A screenshot of a Mozilla Firefox browser window titled "My Services - Dashboard - Mozilla Firefox". The address bar shows the URL "https://myservices.us2.oraclecloud.com/mycloud/cloudportal/dashboard". The main content area is titled "ORACLE Cloud My Services" and "Dashboard". It features several cards: "Guided Journey" (Explore what you can do with Oracle Cloud services), "Create Instance" (Provision a new service in minutes), "Account Management" (Administer and manage your account and orders), and "Customize Dashboard" (Specify which services appear on the dashboard). Below this, there's a section for "Cloud Services" with "Important Notifications" (0), "Integration" (0 of 2 Remaining), and "Compute Classic" (205 of 500 Remaining). A grid of service cards includes "Storage Classic" (Subscription ID: 583566028), "Integration" (Subscription ID: 583565131, Instance 1), "Analytics" (Subscription ID: 583565150, Instance 0), "Database" (Subscription ID: 583565140, Instance 1), and others.

Before creating a DBCS instance, we will create a storage container for backups. Note that this step is optional, as there are several other backup mechanisms available for DBCS. Click the action button on the Storage Classic widget and select View Details.



The Details page for your Storage Classic services will appear. Note the REST Endpoint URL, which will be needed later to setup your DBCS backup.

Click Open Service Console to bring up the Storage Containers page.

Service Detail - Mozilla Firefox

Service Detail

https://myservices.us2.oraclecloud.com/mycloud/faces/gServiceDetail.jsp?entitlementServiceId=583566091&serviceId=58

Search

ORACLE Cloud My Services

Dashboard Users Notifications uspm020 (traditional) @oracle.com

Service: Oracle Cloud Infrastructure Object Storage Classic

Open Service Console

Overview Billing Metrics Resource Quotas Monitoring Metrics Business Metrics Documents

Overview Information

Category	Oracle IaaS Public Cloud Services
Identity Domain Name	uspm020
Identity Domain Id	uspm020
Subscription	Trial (Expires: 1-Aug-2018 5:51 PM EDT)

Additional Information

Plan	Oracle Cloud Infrastructure Object Storage Classic	Status	Active
Service Start Date	1-Aug-2017	Buyer	dbdevcloud_www@oracle.com
Service End Date	1-Aug-2018	Domain SFTP Host & Port	sftp.us2.cloud.oracle.com:22
Subscription ID	583566028	Domain SFTP User Name	[REDACTED]
Service Instance ID	583566091	REST Endpoint	https://uspm020.storage.oraclecloud.com/v1/Storage-uspm020
Customer Account	Oracle (US)	Auth V1 Endpoint	https://uspm020.storage.oraclecloud.com/auth/v1.0
CSI Number	Not available		

About Oracle Contact Us Terms of Use
Copyright © 2013, 2018, Oracle and/or its affiliates. All rights reserved.

RSS Twitter YouTube Facebook

Click Create Container to create a new storage container.

Oracle Public Cloud - Storage Console - Mozilla Firefox

Oracle Public Cloud - Stor... X +

https://storageconsole.us2.oraclecloud.com/instances/aHR0cHM6Ly91c3BtMDlwLnN0b3JhZ2Uub3JhY2xlY2xvdWQuY29tL...

Search

ORACLE Cloud Services @oracle.com

Storage Classic REST APIs

Containers Account

Container List

Summary 7 Containers 219 Objects 705 mb Storage Used

Create Container

Name	Objects	Size
_apaaS	150	333 KB
container1	14	17.9 KB
DBaaS	43	694 MB
DBCSS	1	1.31 KB
OAC1	11	11.1 MB
OPC_TEST_CONTAINER	0	0 Bytes
raster_image_test_ivan	0	0 Bytes

About Oracle | Contact Us | Legal Notices | Terms Of Use | Your Privacy Rights
Copyright © 2018 Oracle and/or its affiliates. All rights reserved

English ▾

Name	Objects	Size
_apaaS	150	333 KB
container1	14	17.9 KB
DBaaS	43	694 MB
DBCSS	1	1.31 KB
OAC1	11	11.1 MB
OPC_TEST_CONTAINER	0	0 Bytes
raster_image_test_ivan	0	0 Bytes

Enter a name for your storage container and click Create. We will name the container RDF_Container1 in this case. RDF_Container1 should be visible in your Container List afterwards.

Oracle Public Cloud - Storage Console - Mozilla Firefox

Oracle Public Cloud - Stor... X +

https://storageconsole.us2.oraclecloud.com/instances/aHR0dHM6Ly91c3BtMDlwLnN0b3JhZ2Uub3JhY2xlY2xvdWQuY29tL...

Search

ORACLE Cloud Services @oracle.com

Storage Classic REST APIs

Containers Account

Container List

Summary 7 Containers 219 705 mb

Create Storage Container

Name	Objects	Size
_apaaS	150	333 KB
container1	14	17.9 KB
DBaaS	43	694 MB
DBCSS	1	1.31 KB
OAC1	11	11.1 MB
OPC_TEST_CONTAINER	0	0 Bytes
raster_image_test_ivan	0	0 Bytes

Name: RDF_Container1

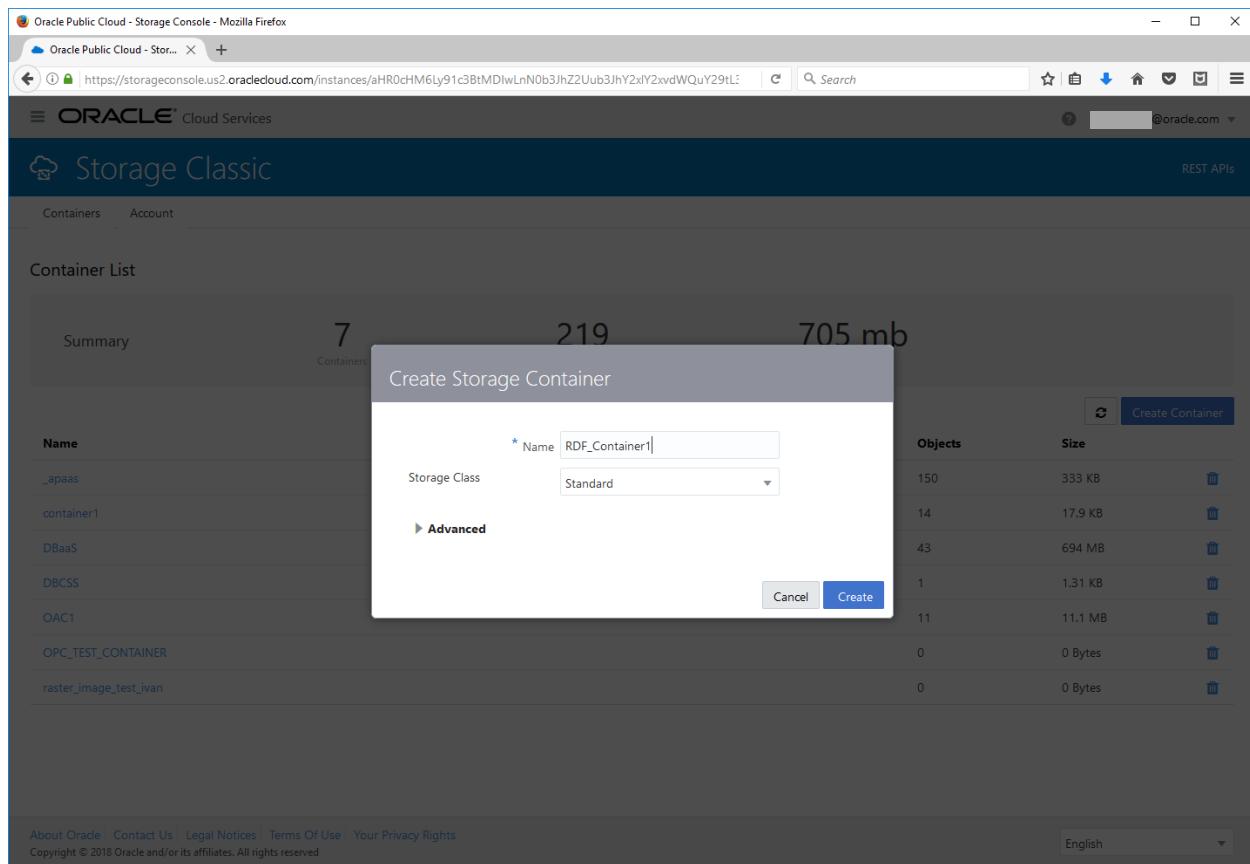
Storage Class: Standard

Advanced

Cancel Create

About Oracle | Contact Us | Legal Notices | Terms Of Use | Your Privacy Rights
Copyright © 2018 Oracle and/or its affiliates. All rights reserved.

English



Oracle Public Cloud - Storage Console - Mozilla Firefox

Oracle Public Cloud - Stor... X +

<https://storageconsole.us2.oraclecloud.com/instances/aHR0cHM6Ly91c3BtMDlwLnN0b3JhZ2Uub3JhY2xlY2xvdWQuY29tLz>

ORACLE® Cloud Services @oracle.com ▾

Storage Classic

REST APIs

Containers Account

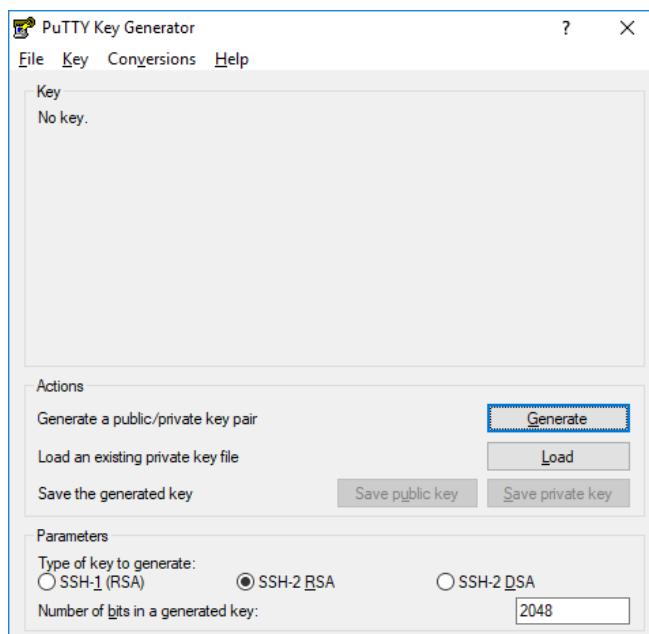
Container List

Name	Objects	Size
_apaaS	150	333 KB
container1	14	17.9 KB
DBaaS	43	694 MB
DBCSS	1	1.31 KB
OAC1	11	11.1 MB
OPC_TEST_CONTAINER	0	0 Bytes
raster_image_test_ivan	0	0 Bytes
RDF_Container1	0	0 Bytes

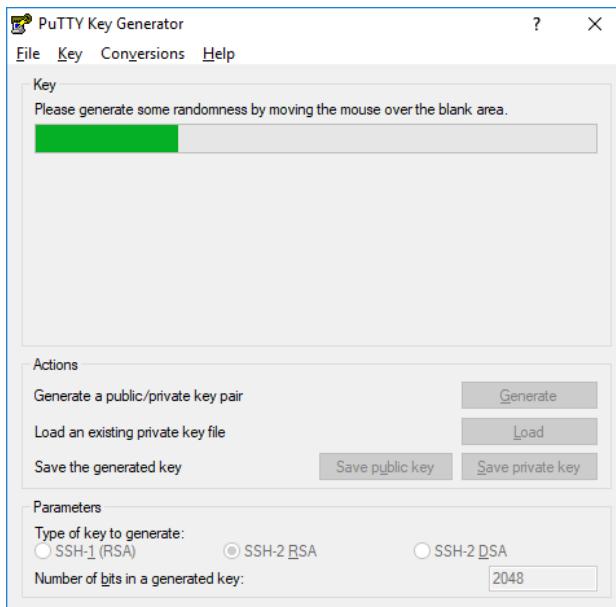
Create Container

About Oracle | Contact Us | Legal Notices | Terms Of Use | Your Privacy Rights
Copyright © 2018 Oracle and/or its affiliates. All rights reserved English ▾

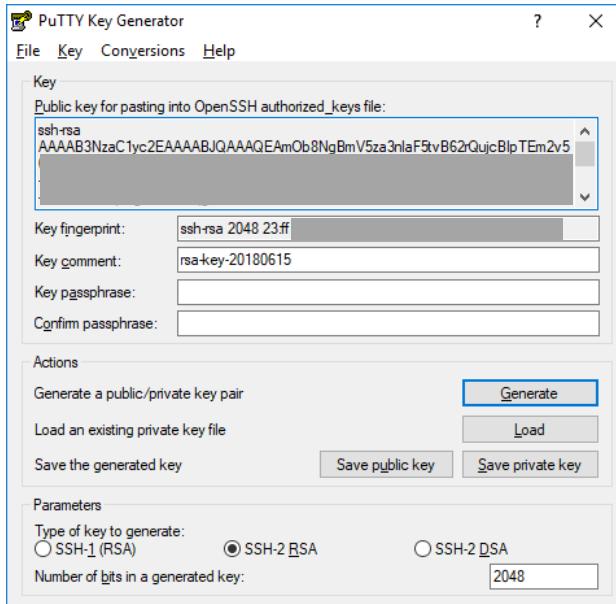
In the next step, we will create a DBCS instance. As a prerequisite, we will need to generate an SSH public/private key pair. You can use PuTTYgen for this on Windows. Open PuTTYgen and click Generate.



Follow the instructions to generate randomness.

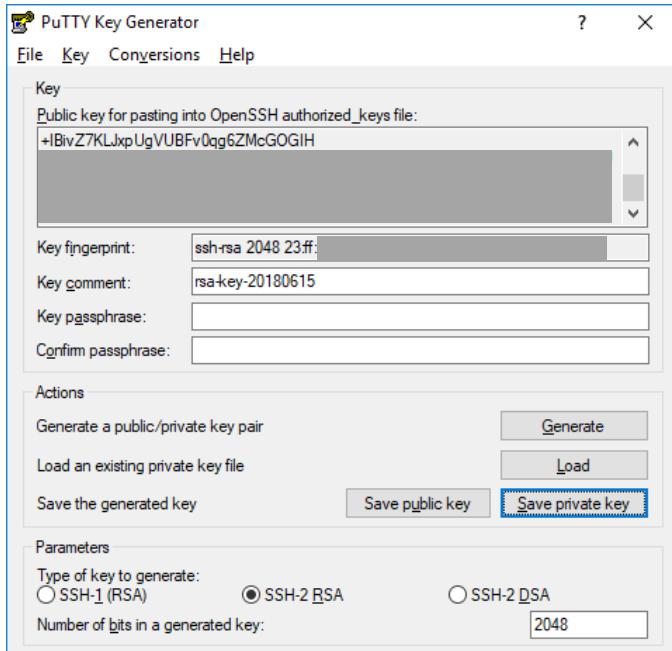


After generation, create a <key_name>.pub file and use Notepad or a similar text editor to copy and paste the entire contents of the Public key text into the myPubKey.pub file (be sure to scroll down to get everything). Save the <key_name>.pub file.



Click Save private key. You can optionally enter a passphrase that will need to be entered whenever the private key is used. This will save your private key in .ppk format.

In addition, click Conversions and select Export OpenSSH Key to also save the key in OpenSSH format.



After this, you should have created three files: <key_name>.pub, <key_name>.ppk and <key_name>.openssh.

We will create the DBCS instance in the next step. First, open your cloud dashboard. In the Database widget, click Action and select Open Service Console.

The screenshot shows the Oracle Cloud My Services Dashboard. At the top, there are tabs for Guided Journey, Account Management, Customize Dashboard, and Create Instance. The main area displays several service cards:

- Storage Classic**: Subscription Id: 583566028
- Integration**: Subscription ID: 583565131
- Analytics**: Subscription Id: 583565150
- Database**: Subscription Id: 583565140

A context menu is open over the Database card, listing three options:

- View Details
- Open Service Console
- Maintenance and Service Requests

At the bottom of the page, there are links for About Oracle, Contact Us, Terms Of Use, and copyright information: Copyright © 2013, 2018, Oracle and/or its affiliates. All rights reserved. The URL https://dbaaS.oraclecloud.com/dbaaS/faces/dbRunner.jspx?_adf.ctrl-state=undefined&_afrLoop=undefined is also visible.

You will see any existing DBCS instances that you have. Click Create Instance to create a new DBCS instance.

Oracle Database Cloud Service Details - Mozilla Firefox

Oracle Database Cloud Ser... × +

https://dbaas.oraclecloud.com/dbaas/faces/dbRunner.jsp?_adf.ctrl-state=undefined

Search

ORACLE CLOUD My Services

Oracle Database Cloud Service

Welcome! REST APIs

Instances Activity SSH Access

As of Jun 12, 2018 5:08:33 PM UTC

Summary	1 Instances	0 OCpus	0 GB Memory	139 GB Storage	1 Public IPs
---------	-------------	---------	-------------	----------------	--------------

Instances

Search by instance name or tags

Create Instance

DB121	Status: Instance Stopped	Submitted On: Mar 4, 2018 4:22:32 AM UTC	OCpus: 1*
	Version: 12.1.0.2	Memory: 7.5 GB*	Edition: Enterprise Edition - High...
			Storage: 139 GB

* Configured values for CPU and Memory are indicated here. Since this service instance is stopped, these resources are currently not in use.

Service Create and Delete History

About Oracle | Contact Us | Legal Notices | Terms of Use | Your Privacy Rights
Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Oracle Database Cloud Service Details - Mozilla Firefox
Oracle Database Cloud Ser... + https://dbaas.oracledcloud.com/dbaas/faces/dbRunner.jsp?_adf.ctrl-state=undefined

ORACLE CLOUD My Services

Create Instance

Instance Details Confirm

Cancel Next >

Instance
Provide basic service instance information.

* Instance Name: RDFDBCS1	* Service Level: Oracle Database Cloud Service
Description: 18c RDF Database	Metering Frequency: Monthly
Notification Email: [REDACTED]@oracle.com	* Software Release: Oracle Database 18c
Region: No Preference	* Software Edition: Enterprise Edition - High Performance
Tags: [REDACTED]	* Database Type: Single Instance

Fill in the form with your desired values. We will be using Oracle Database version 18c. You will need either Enterprise Edition - High Performance or Enterprise Edition - Extreme Performance to get the Spatial and Graph option, which includes RDF Semantic Graph.

Oracle Database Cloud Service Details - Mozilla Firefox

Oracle Database Cloud Service...

https://dbaas.oraclecloud.com/dbaas/faces/dbRunner.jsp?_adf.ctrl-state=undefined

Create Instance

Instance Details

Instance Details Confirm

Instance Details

Provide details for this Oracle Database Cloud Service instance.

Database Configuration

- * DB Name: ORCL
- * PDB Name: PDB1
- * Administration Password:
- * Confirm Password:
- * Usable Database Storage (GB): 25
- Total Data File Storage (GB): 88.5
- * Compute Shape: OC4 - 2.0 OCPU, 15.0 GB RAM
- * SSH Public Key: 18_1_RDFDBCS1_public_key

Backup and Recovery Configuration

- * Backup Destination: Cloud Storage Only
- * Cloud Storage Container: ige-uspm020/RDF_Container1
- * Username: @oracle.com
- * Password:
- Create Cloud Storage Container
- Total Estimated Monthly Storage (GB): 140

Initialize Data From Backup

- * Create Instance from Existing Backup: No

Advanced Settings

- * Listener Port: 1521
- * Timezone: (UTC) Coordinated Universal Time
- * Character Set: AL32UTF8 - Unicode Universal
- * National Character Set: AL16UTF16 - Unicode UTF-16
- Enable Oracle GoldenGate
- Include "Demos" PDB

About Oracle | Contact Us | Legal Notices | Terms of Use | Your Privacy Rights
 https://dbaas.oraclecloud.com/dbaas/faces/dbRunner.jsp?_adf.ctrl-state=undefined#

Fill in the Instance Details. Choose the <key name>.pub file to enter the SSH public key you created earlier, and enter <REST endpoint URL>/<Container Name> for the Cloud Storage Container you created earlier. Click Next to bring up the confirmation page.

Oracle Database Cloud Service Details - Mozilla Firefox
Oracle Database Cloud Service... https://dbaas.oracledcloud.com/dbaas/faces/dbRunner.jsp?_adf.ctrl-state=undefined

ORACLE CLOUD My Services

Create Instance

Previous Cancel Create >

Confirmation
Confirm your responses and create service instance.

Instance

Instance Name:	RDFDBCS1
Description:	18c RDF Database
Bring Your Own License:	No
Service Level:	Oracle Database Cloud Service
Metering Frequency:	Monthly
Software Release:	Oracle Database 18c
Software Edition:	Enterprise Edition - High Performance
Compute Shape:	OC4 - 2.0 OCPU, 15.0 GB RAM
SSH Public Key:	18_1_RDFDBCS1_public_key.pub
Use High Performance Storage:	No
Assign Public IP:	Yes

Database Configuration

DB Name:	ORCL
PDB Name:	PDB1
Usable Database Storage (GB):	25
Total Data File Storage (GB):	88.5
Listener Port:	1521
Timezone:	(UTC) Coordinated Univers...
Character Set:	AL32UTF8 - Unicode Univer...
National Character Set:	AL16UTF16 - Unicode UTF-1...
Include "Demos" PDB:	No
Include GoldenGate:	No

Backup and Recovery Configuration

Backup Destination:	Cloud Storage Only
Cloud Storage Container:	https://uspm020.storage.o...
Username:	[REDACTED]@oracle.com

Notification

Notification Email: matthew.perry@oracle.com

www.oracle.com

The screenshot shows the 'Create Instance' wizard in the Oracle Database Cloud Service. The 'Details' step is selected. The 'Instance' section shows configuration like Instance Name (RDFDBCS1), Description (18c RDF Database), and Service Level (Oracle Database Cloud Service). The 'Database Configuration' section includes DB Name (ORCL), PDB Name (PDB1), and storage settings. The 'Backup and Recovery Configuration' section specifies Cloud Storage Only as the destination. The 'Notification' section lists a notification email address. A 'Create' button is visible at the top right.

Click Create to create the instance.

Oracle Database Cloud Service Details - Mozilla Firefox

Oracle Database Cloud Ser... + New Tab

https://dbaas.oraclecloud.com/dbaas/faces/dbRunner.jspx

Search

ORACLE CLOUD My Services

Oracle Database Cloud Service

Welcome! REST APIs

Instances Activity SSH Access

As of Jun 12, 2018 6:22:53 PM UTC

Summary	2 Instances	2 OCPUs	15 GB Memory	289 GB Storage	2 Public IPs
---------	-------------	---------	--------------	----------------	--------------

Instances

Search by instance name or tags

Create Instance

Instance Name	Version	Edition	Created On	OCPUs	Memory	Storage
RDFDBCS1	18.0.0.0	Enterprise Edition - High...	Jun 12, 2018 5:43:40 PM UTC	2	15 GB	150 GB
DB121	12.1.0.2	Enterprise Edition - High...	Mar 4, 2018 4:22:32 AM UTC	1*	7.5 GB*	139 GB

* Configured values for CPU and Memory are indicated here. Since this service instance is stopped, these resources are currently not in use.

Service Create and Delete History

About Oracle | Contact Us | Legal Notices | Terms of Use | Your Privacy Rights
Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

The screenshot shows the Oracle Database Cloud Service Instances page. At the top, there's a summary bar with counts for Instances (2), OCPUs (2), Memory (15 GB), Storage (289 GB), and Public IPs (2). Below this is a search bar and a 'Create Instance' button. The main area lists two instances: 'RDFDBCS1' and 'DB121'. Each instance has details like Version, Edition, Created On, and Resource Specs (OCPUs, Memory, Storage). A note at the bottom states that configured values for CPU and Memory are indicated here, but since the instance is stopped, these resources are currently not in use. At the bottom of the page, there's a footer with links to About Oracle, Contact Us, Legal Notices, Terms of Use, and Your Privacy Rights, along with a copyright notice for 2018.

After about 30 minutes, the DBCS instance will be ready to use. You can click the instance name (RDFDBCS1 in this case) to see details.

The screenshot shows the Oracle Database Cloud Service Details page in Mozilla Firefox. The URL is https://dbaas.oraclecloud.com/dbaas/faces/dbRunner.jspx?_adf.ctrl-state=undefined. The main content area is titled "Instance Overview" and displays the following information:

Nodes	OCPUs	Memory	Storage
1	2	15 GB	150 GB

Below this, there are several configuration details:

- Status: Ready
- Version: 18.0.0.0
- Connect String: RDFDBCS1:1521/PDB1.uspm02...
- Edition: Enterprise Edition - High Performance
- Backup Destination: Cloud Storage Only
- Cloud Storage Container: https://uspm020.storage.o...
- PDB Name: PDB1
- Container Name: ORCL
- Character Set: AL32UTF8 - Unicode Univers...
- National Character Set: AL16UTF16 - Unicode UTF-1...
- SQL *Net Port: 1521
- Timezone: Coordinated Universal Time
- Show less...

On the left sidebar, under "Administration", it shows 1 Patches available and 0 Snapshots available. Under "Resources", it shows the Host Name: RDFDBCS1, Public IP: 129. [REDACTED], SID: ORCL, OCPUs: 2, Memory: 15 GB, and Storage: 150 GB.

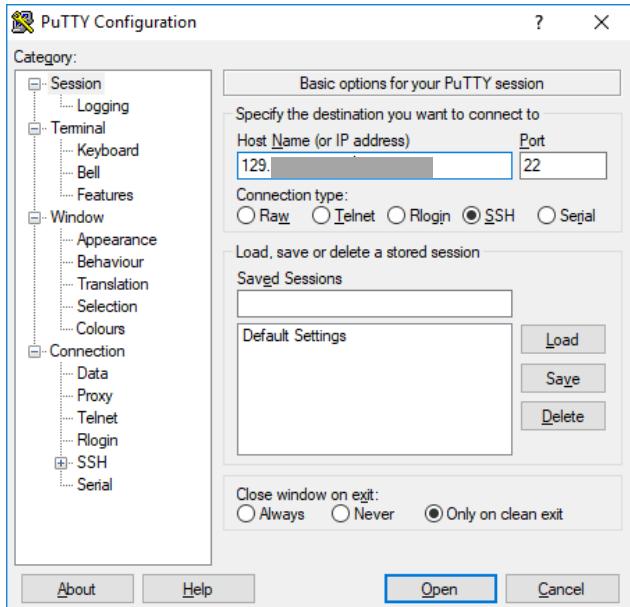
At the bottom of the page, there are links to About Oracle, Contact Us, Legal Notices, Terms of Use, and Your Privacy Rights, along with a copyright notice: Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Note the Public IP, Connect String and SQL *Net port. You will need this information to make a connection to the DBCS instance.

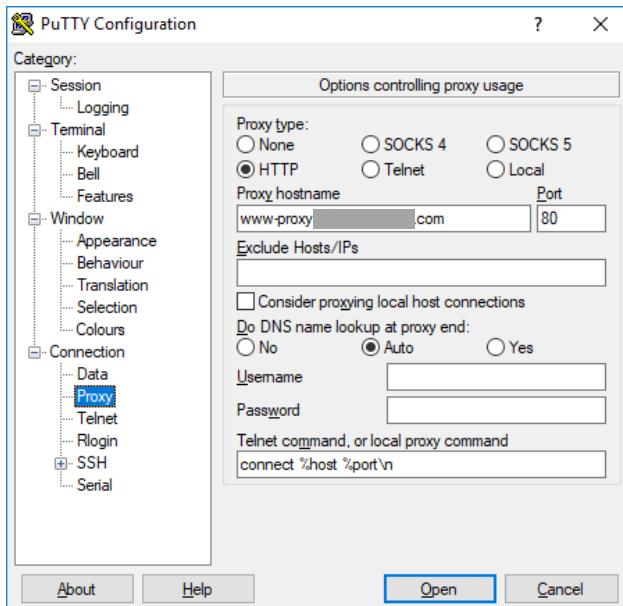
4 CONNECTING TO THE DBCS INSTANCE WITH PUTTY

Next, we will use PuTTY to make an SSH connection to our DBCS instance, start SQL *Plus, and check the RDF Semantic Graph installation in our database.

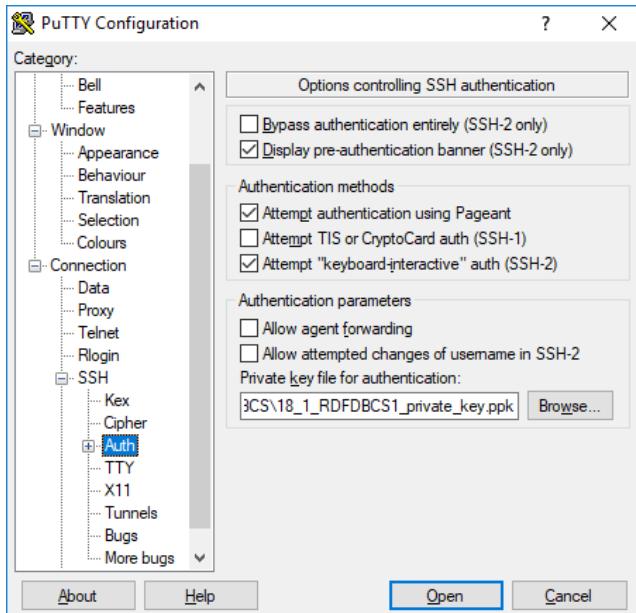
Open PuTTY and click Session in the tree on the left-hand side. Enter the public IP for your DBCS instance for Host Name and enter 22 for Port.



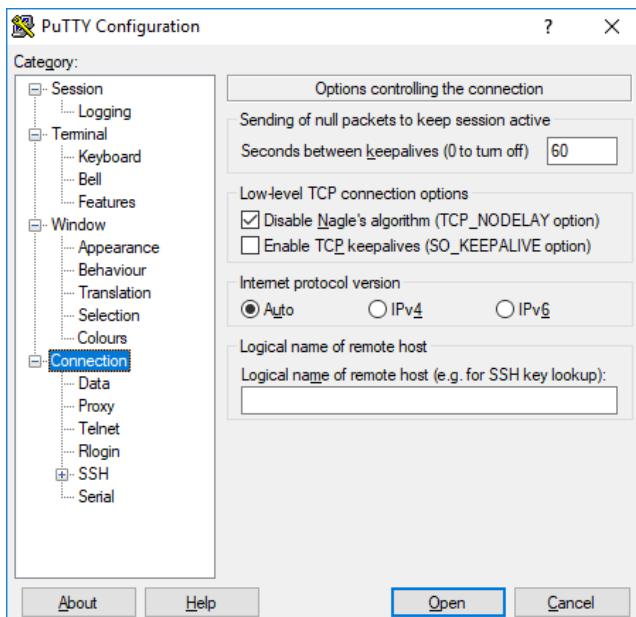
If necessary, expand Proxy and configure any proxy settings for your network.



Expand SSH and click Auth. Click Browse and select your .ppk private key file for Private key file for authentication.

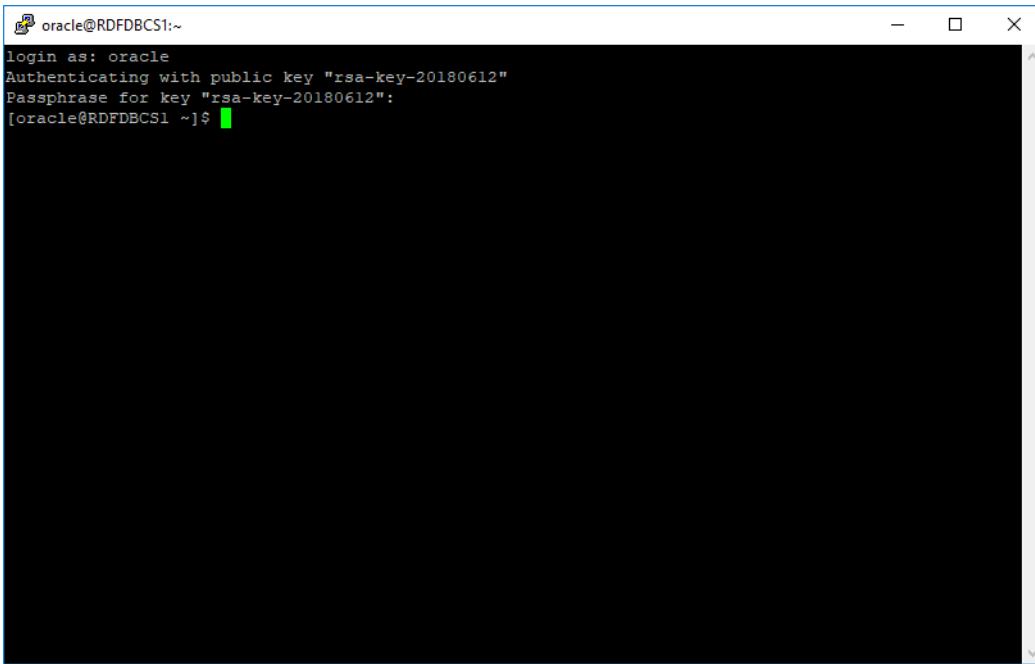


You can also click Connection and specify a non-zero keepalive value to prevent the SSH session from ending due to inactivity.



Tip: it is a good idea to click Session and save this PuTTY configuration for future use.

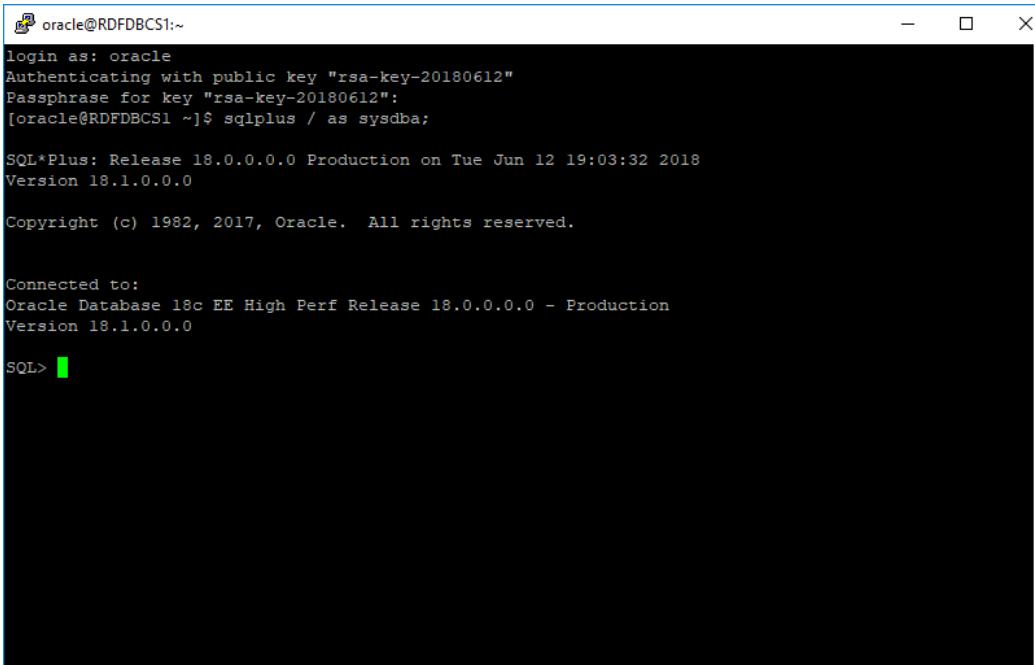
Now, click Open to start the SSH session. Enter oracle for the login name. Then enter your passphrase if you used one when creating your public/private key pair.



```
oracle@RDFDBCS1:~  
login as: oracle  
Authenticating with public key "rsa-key-20180612"  
Passphrase for key "rsa-key-20180612":  
[oracle@RDFDBCS1 ~]$
```

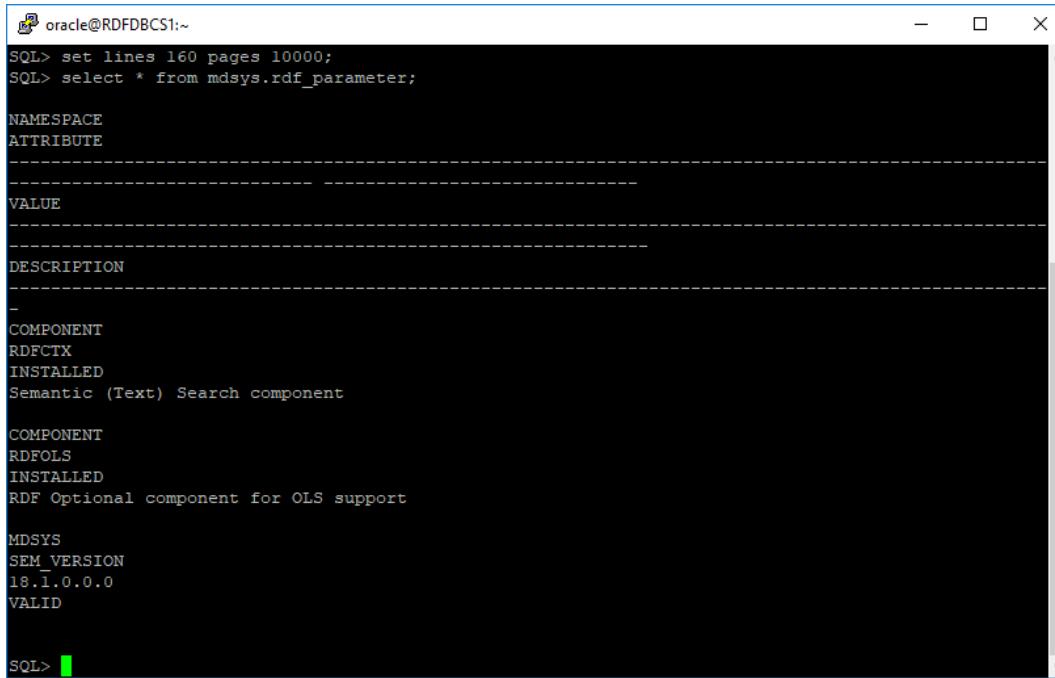
Start SQL*Plus with the following command:

```
sqlplus / as sysdba
```



```
oracle@RDFDBCS1:~  
login as: oracle  
Authenticating with public key "rsa-key-20180612"  
Passphrase for key "rsa-key-20180612":  
[oracle@RDFDBCS1 ~]$ sqlplus / as sysdba;  
  
SQL*Plus: Release 18.0.0.0.0 Production on Tue Jun 12 19:03:32 2018  
Version 18.1.0.0.0  
  
Copyright (c) 1982, 2017, Oracle. All rights reserved.  
  
Connected to:  
Oracle Database 18c EE High Perf Release 18.0.0.0.0 - Production  
Version 18.1.0.0.0  
  
SQL>
```

Execute the query `SELECT * FROM MDSYS.RDF_PARAMETER` to verify the RDF Semantic Graph installation. You should see a valid 18.1.0.0.0 installation.



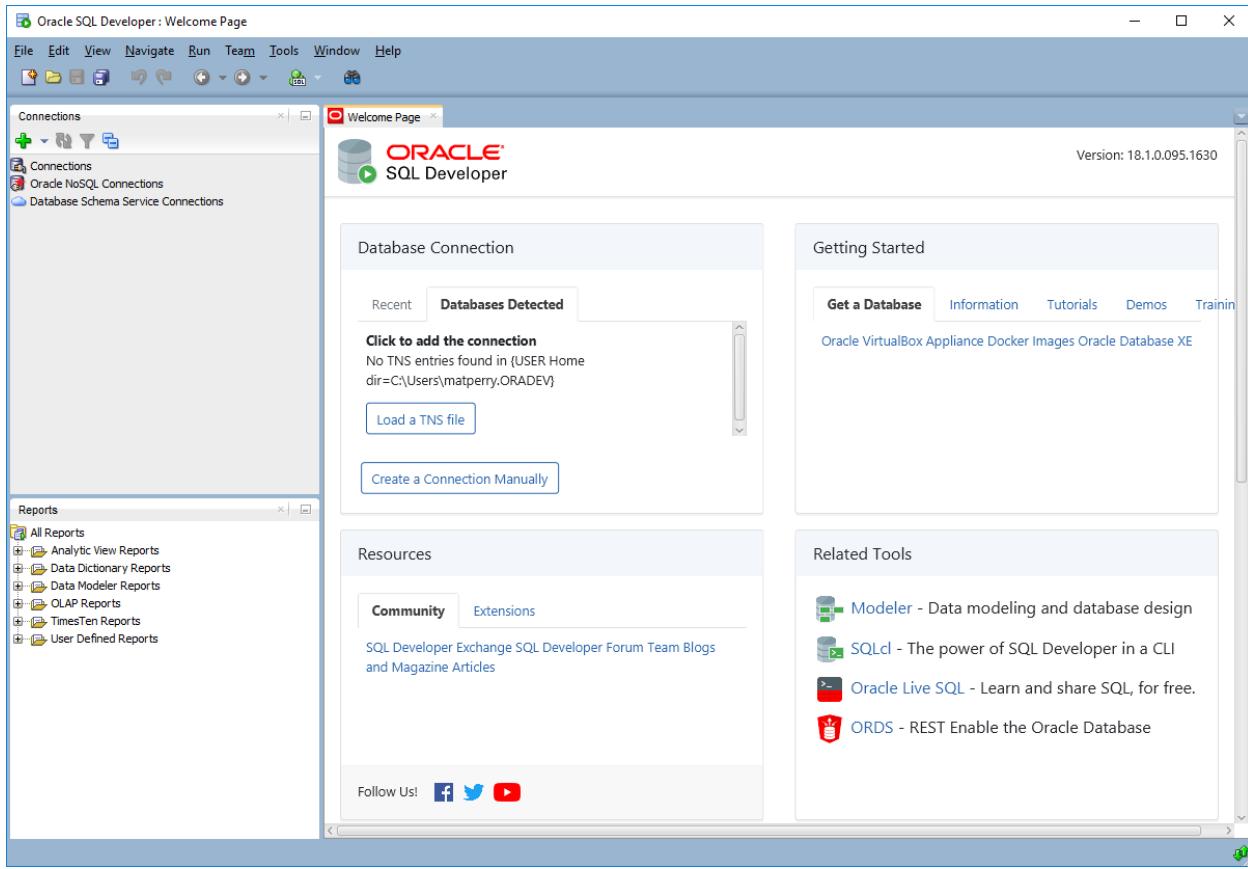
```
oracle@RDFDBCS1:~  
SQL> set lines 160 pages 10000;  
SQL> select * from mdsys.rdf_parameter;  
  
NAMESPACE  
ATTRIBUTE  
-----  
  
VALUE  
-----  
  
DESCRIPTION  
-----  
-  
COMPONENT  
RDFCTX  
INSTALLED  
Semantic (Text) Search component  
  
COMPONENT  
RDFOLS  
INSTALLED  
RDF Optional component for OLS support  
  
MDSYS  
SEM_VERSION  
18.1.0.0.0  
VALID  
  
SQL>
```

We have now verified that RDF Semantic Graph is properly installed on our DBCS instance.

5 USING SQL DEVELOPER FOR RDF SEMANTIC GRAPH ON THE DBCS INSTANCE

We will now use SQL Developer to setup and configure our RDF Semantic Graph installation.

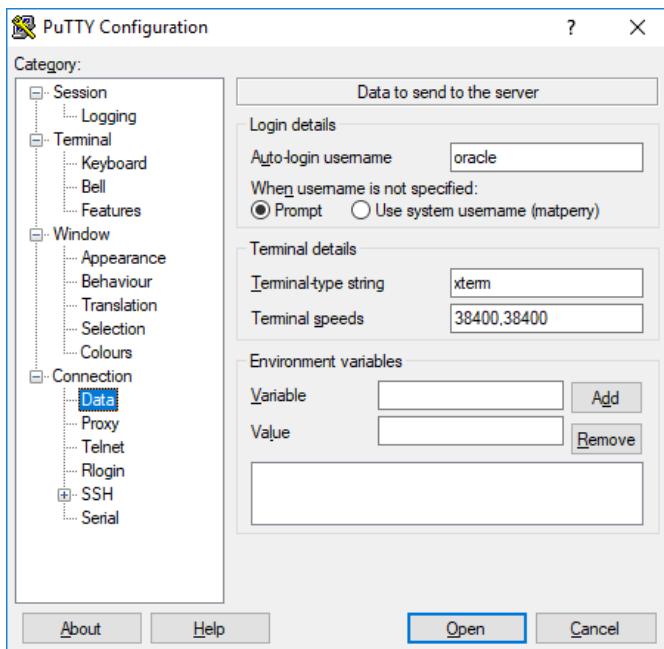
Open SQL Developer. We are using version 18.1.0.095 in this example.



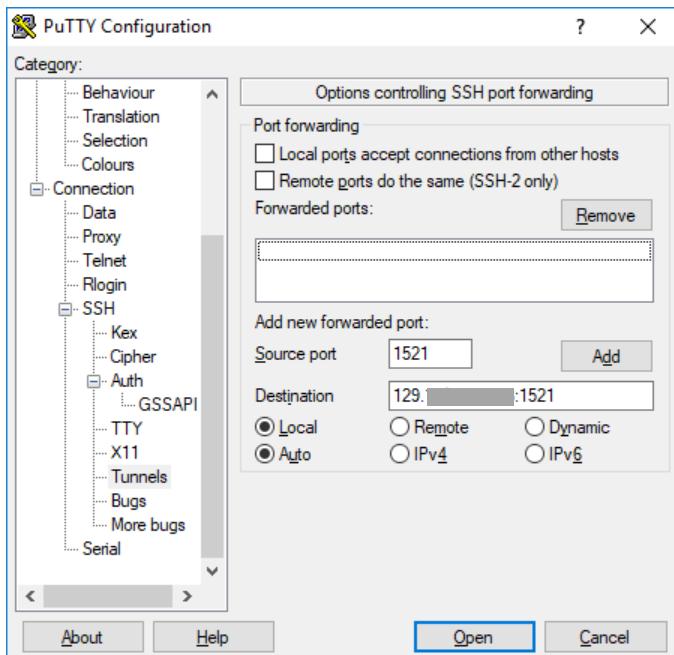
5.1 CONNECTING TO THE DBCS INSTANCE

First, we need to create a connection to our DBCS instance. There are a few ways to create a connection. In this tutorial, we will create an SSH tunnel and configure port forwarding for our SQL*Net port. See the following Oracle Help Center [article](#) for more details on other ways to connect SQL Developer to a DBCS instance.

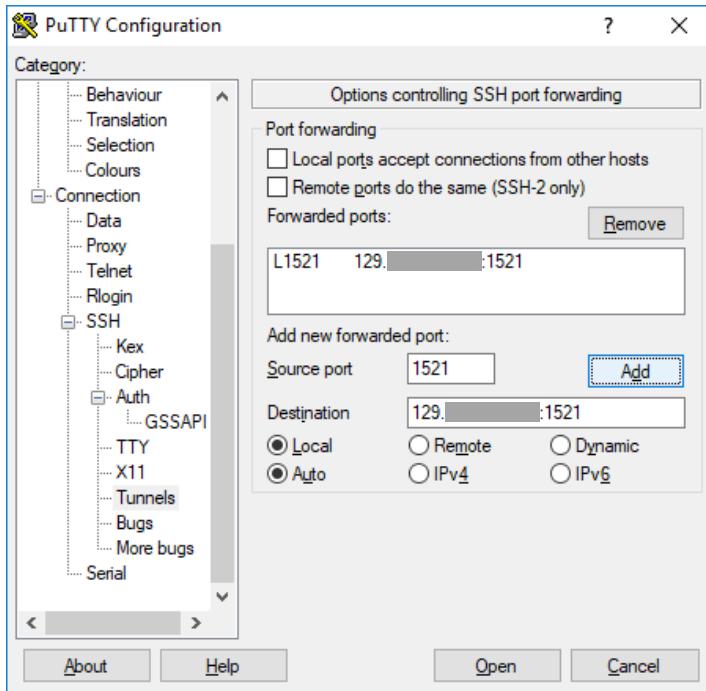
We will use PuTTY to create an SSH tunnel for the SQL*Net port (1521 in this example). Open PuTTY and repeat all the steps from before to configure an SSH connection to your DBCS instance. Next, click Data under Connection and enter oracle for Auto-login username.



Click Tunnels and enter a Source port to tunnel and use the public IP and SQL*Net port for your DBCS instance for Destination. Click Add.



After the forwarded port has been added, click Open.



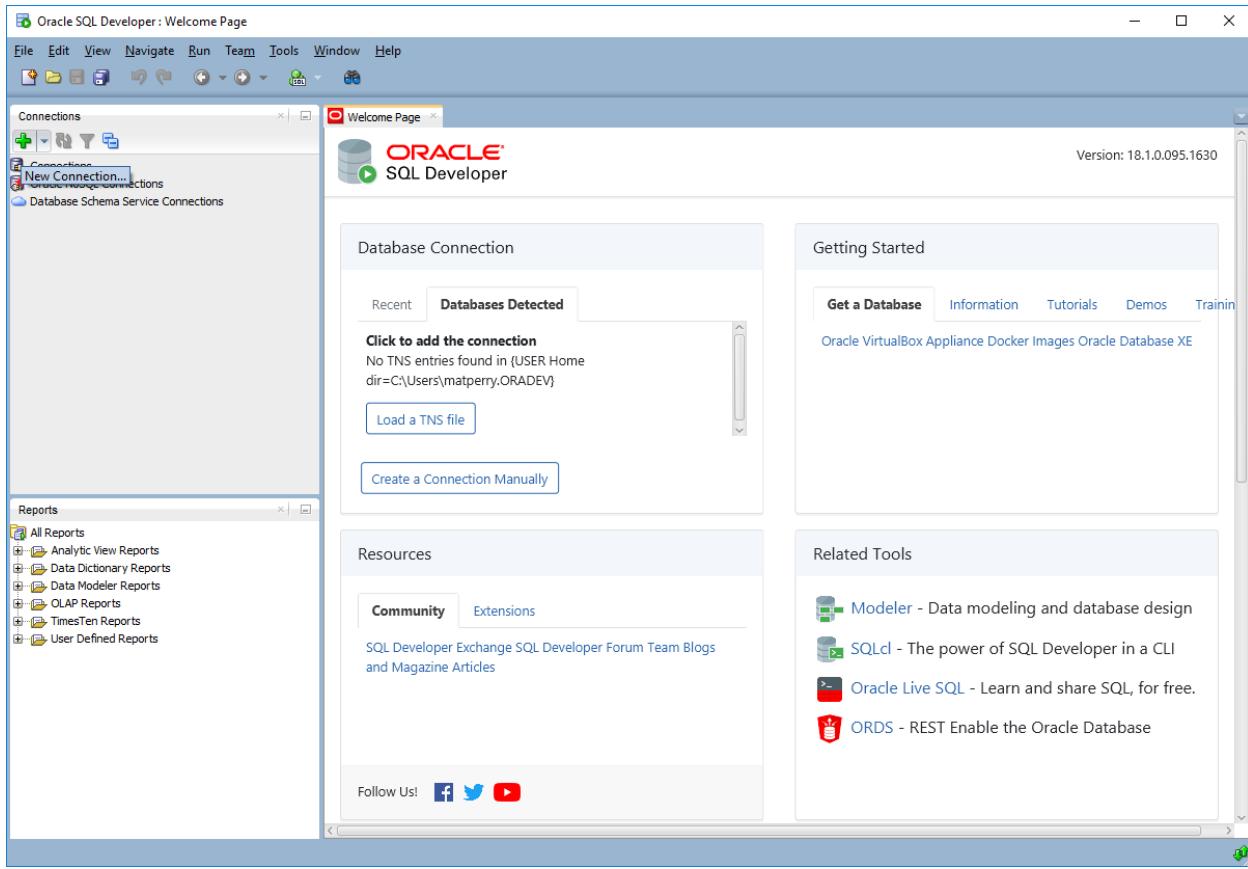
If applicable, enter the passphrase for your private key. Leave this window open.

A terminal window titled 'oracle@RDFDBCS1:~' is displayed. It shows the following text:

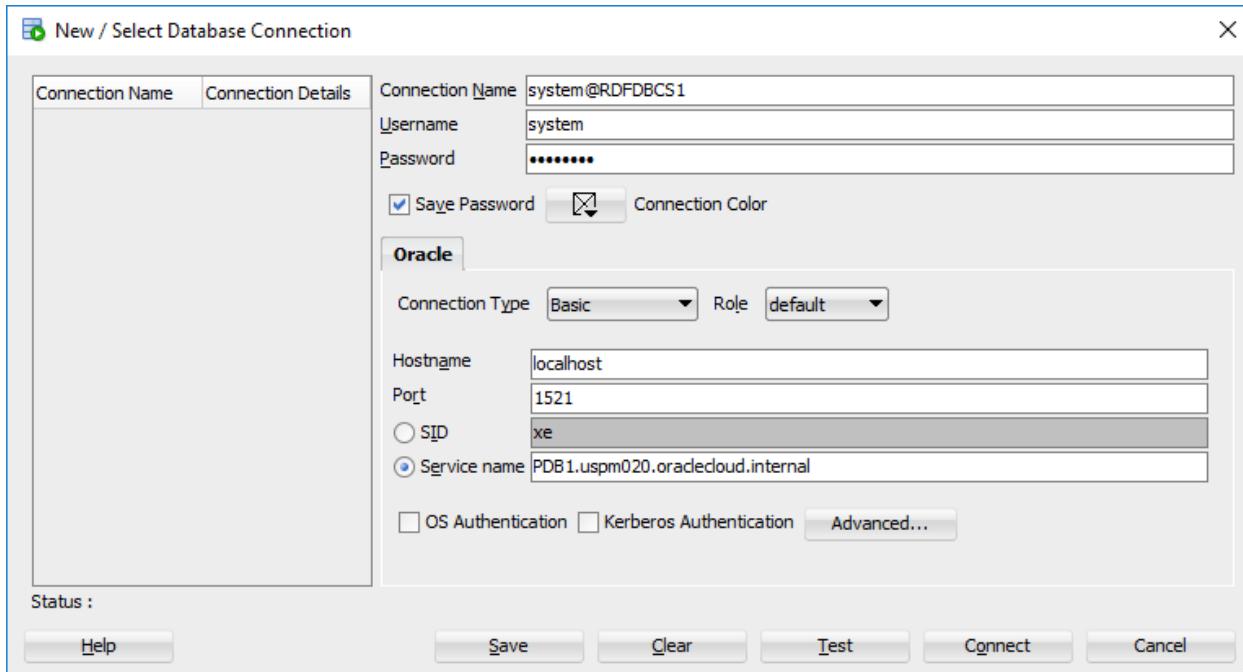
```
Using username "oracle".
Authenticating with public key "rsa-key-20180612"
Passphrase for key "rsa-key-20180612":
```

The last line ends with a green cursor bar, indicating where the user should type their passphrase.

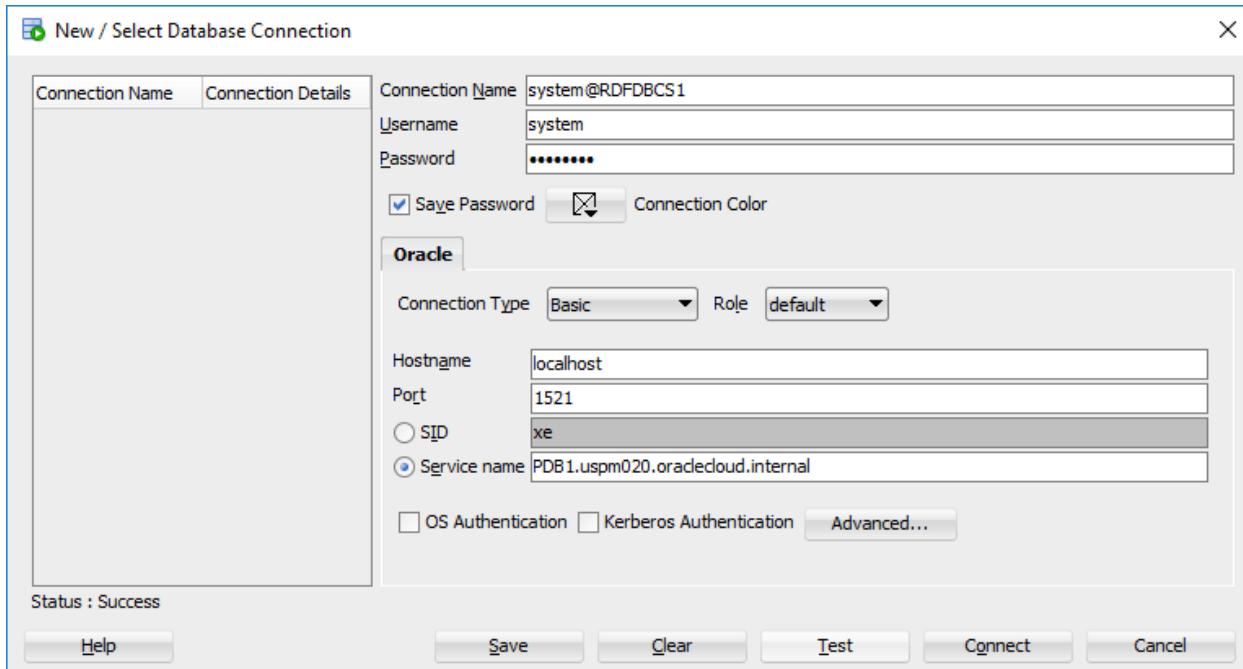
Now we will create a connection in SQL Developer that uses the SSH tunnel we created for port 1521. Click the green plus under Connections to create a new connection.



Choose a name for the connection. Use system as the Username and use the admin password for your DBCS instance. Use localhost for Hostname, the SQL*Net port from your DBCS instance for Port (1521 in this example), and the service name from your DBCS connection string (the portion of the connect string after <port>/).



Click Test to test the connection.



Save the connection if the test was successful, and click Connect to open the connection. This will open up a SQL Worksheet for the system connection.

Now we will verify the RDF Semantic Graph installation. Type the following query into your SQL Worksheet and click the green triangle to run the query.

```
SELECT * FROM MDSYS.RDF_PARAMETER;
```

The query result should show a valid 18.1.0.0.0 installation of RDF Semantic Graph.

The screenshot shows the Oracle SQL Developer interface. The 'Worksheet' tab is active, displaying the following SQL query:

```
SELECT * FROM MDSYS.RDF_PARAMETER;
```

The 'Query Result' tab shows the output of the query:

NAMESPACE	ATTRIBUTE	VALUE	DESCRIPTION
1	COMPONENT	RDFCTX	INSTALLED Semantic (Text) Search component
2	COMPONENT	RDFOLSL	INSTALLED RDF Optional component for OLS support
3	MDSYS	SEM_VERSION	18.1.0.0.0 VALID

Next, we will create the Semantic Network to prepare the database for storing RDF data. As a prerequisite, we need to create a tablespace for the Semantic Network. We can do this by running the following SQL statement as system:

```
create bigfile tablespace rdftbs
datafile '?/dbs/rdftbs.dat' size 512M reuse autoextend on next 512M maxsize
10G
extent management local
segment space management auto;
```

The screenshot shows the Oracle SQL Developer interface. The top menu bar includes File, Edit, View, Navigate, Run, Source, Team, Tools, Window, and Help. The Connections sidebar lists 'system@RDFDBCS1', 'Oracle NoSQL Connections', and 'Database Schema Service Connections'. The Reports sidebar lists 'All Reports' with categories like Analytic View Reports, Data Dictionary Reports, Data Modeler Reports, OLAP Reports, TimesTen Reports, and User Defined Reports. The central Worksheet pane contains a query script:

```
SELECT * FROM MDSYS.RDF_PARAMETER;

create bigfile tablespace rdftbs
datafile '?/dbs/rdftbs.dat' size 512M reuse autoextend on next 512M maxsize 10G
extent management local
segment space management auto;
```

The Script Output pane below shows the result of the execution:

```
Tablespace RDFTBS created.
```

At the bottom right of the interface, status indicators show 'Line 6 Column 33', 'Insert', 'Modified', and 'Windows: C:\...'.

5.2 USING THE RDF SEMANTIC GRAPH PLUGIN

Now we can use the RDF Semantic Graph component of SQL Developer to create the Semantic Network. Expand the system connection by clicking the plus sign next to the connection name, and then scroll down to the RDF Semantic Graph component.

The screenshot shows the Oracle SQL Developer interface. The 'Connections' sidebar on the left lists various database objects like Synonyms, Public Synonyms, Database Links, etc. The 'Reports' sidebar also lists several report types. The main workspace contains a 'Worksheet' tab with a 'Query Builder' interface. A SQL script is being run:

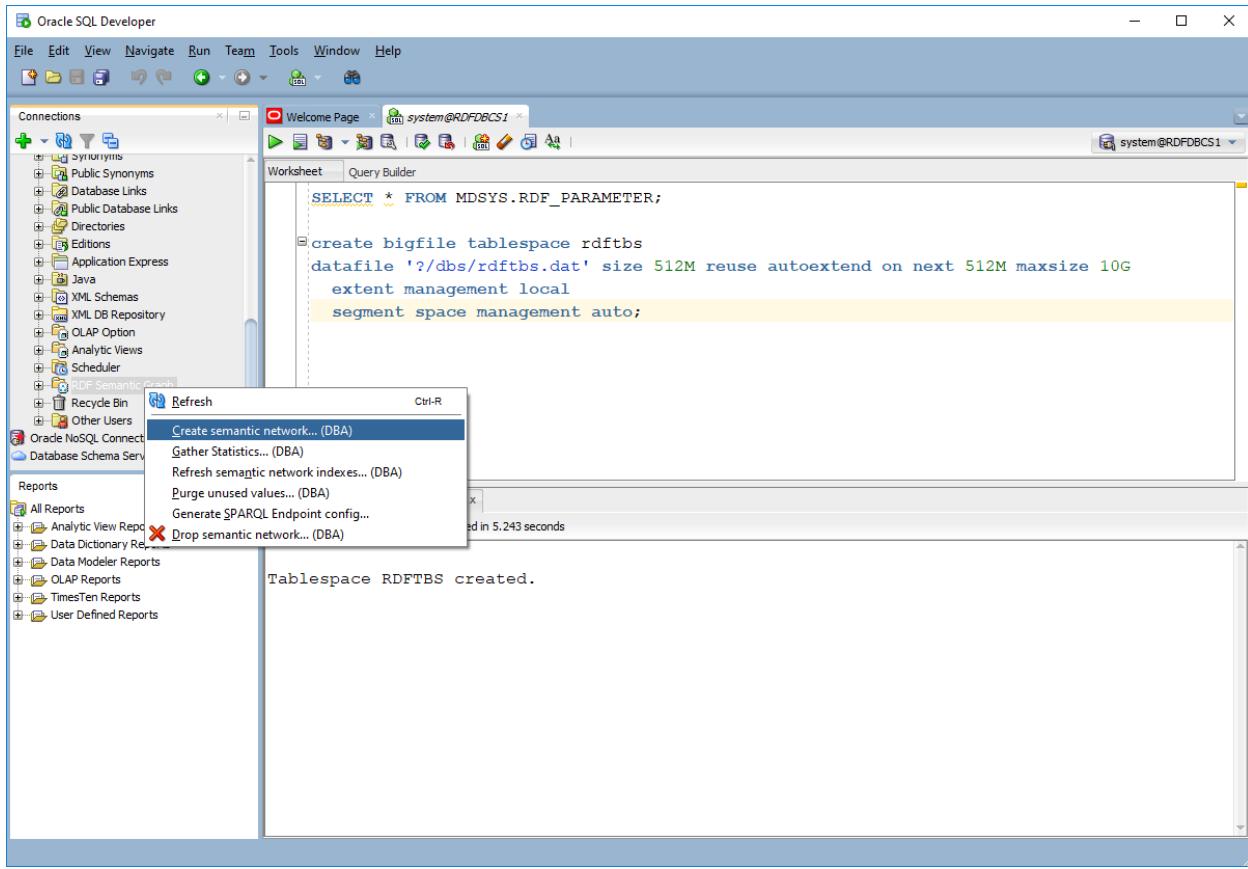
```
SELECT * FROM MDSYS.RDF_PARAMETER;

create bigfile tablespace rdftbs
datafile '?/dbs/rdftbs.dat' size 512M reuse autoextend on next 512M maxsize 10G
extent management local
segment space management auto;
```

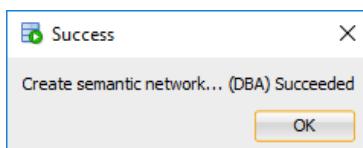
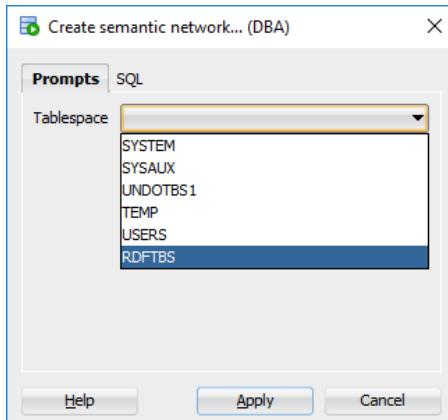
The 'Script Output' tab at the bottom shows the result of the command:

```
Tablespace RDFTBS created.
```

Right-click on RDF Semantic Graph and select Create Semantic Network.

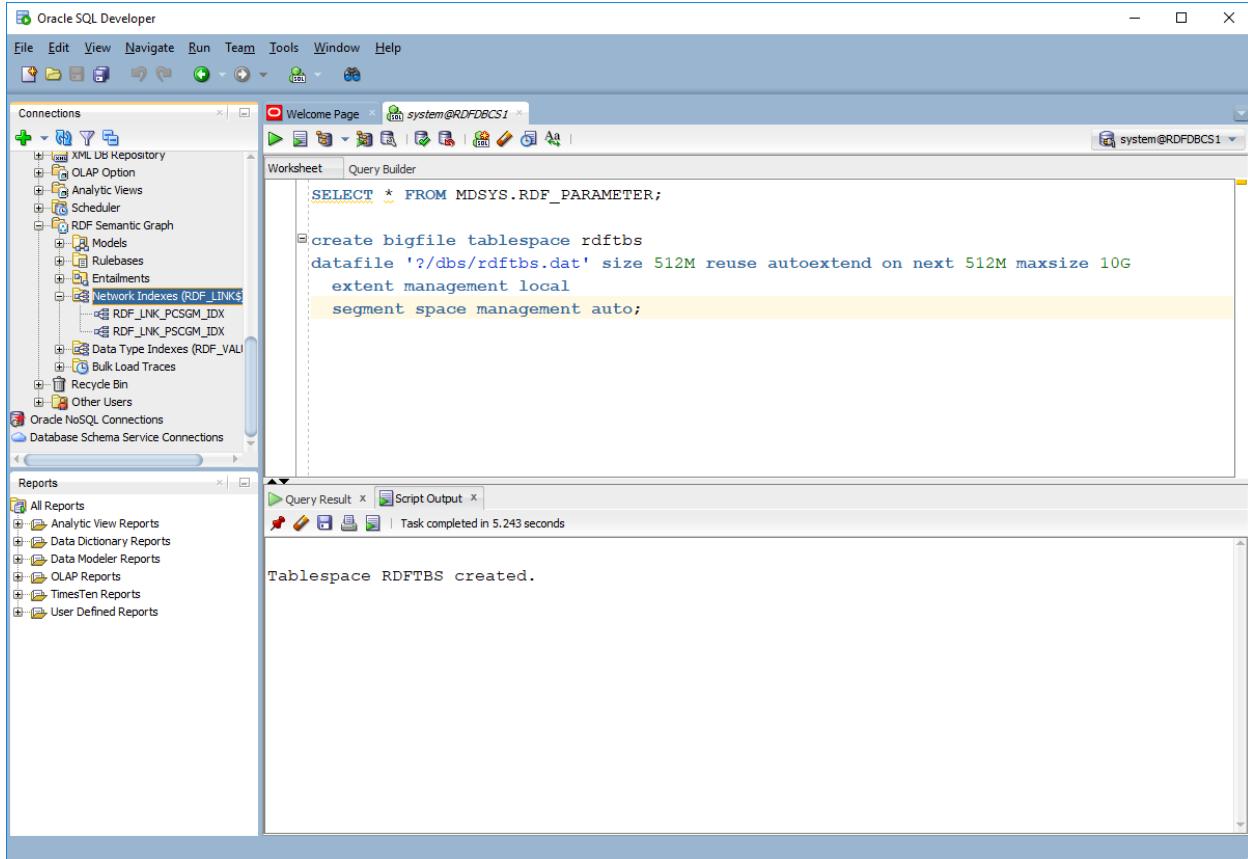


Use the drop down menu to select the tablespace that we created earlier and click Apply.



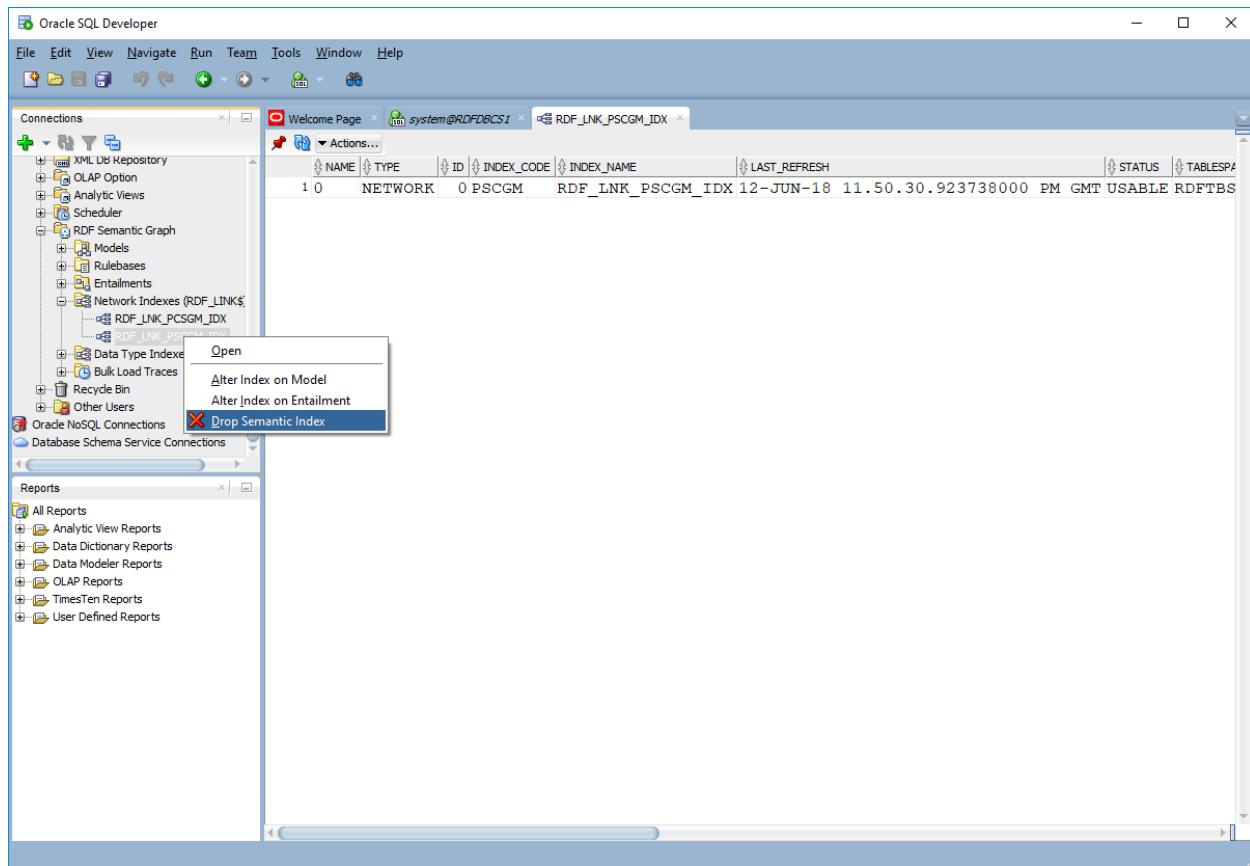
Next, we will make some changes to the default index configuration for better general-purpose query performance. Expand Network Indexes under RDF Semantic Graph to see index codes for the current network indexes. Each letter in an index code corresponds to a component of an RDF quad: S – subject, P – predicate, C – canonical object, G – graph, M – model. By default, two indexes are created: a

mandatory PCSGM unique index and a PSCGM index. This indexing scheme works very well when SPARQL triple patterns have constants in the predicate position, but this scheme may encounter performance problems if variables appear in the predicate position.

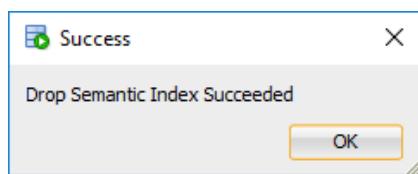
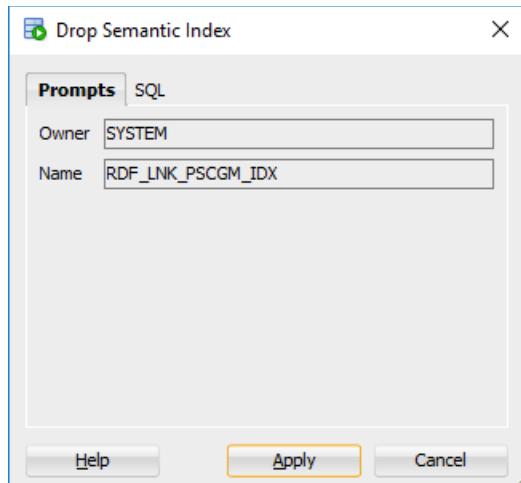


For a more general scheme, a three-index combination of PCSGM, SPCGM and CPSGM works well, so we will drop the PSCGM index and add SPCGM and CPSGM indexes.

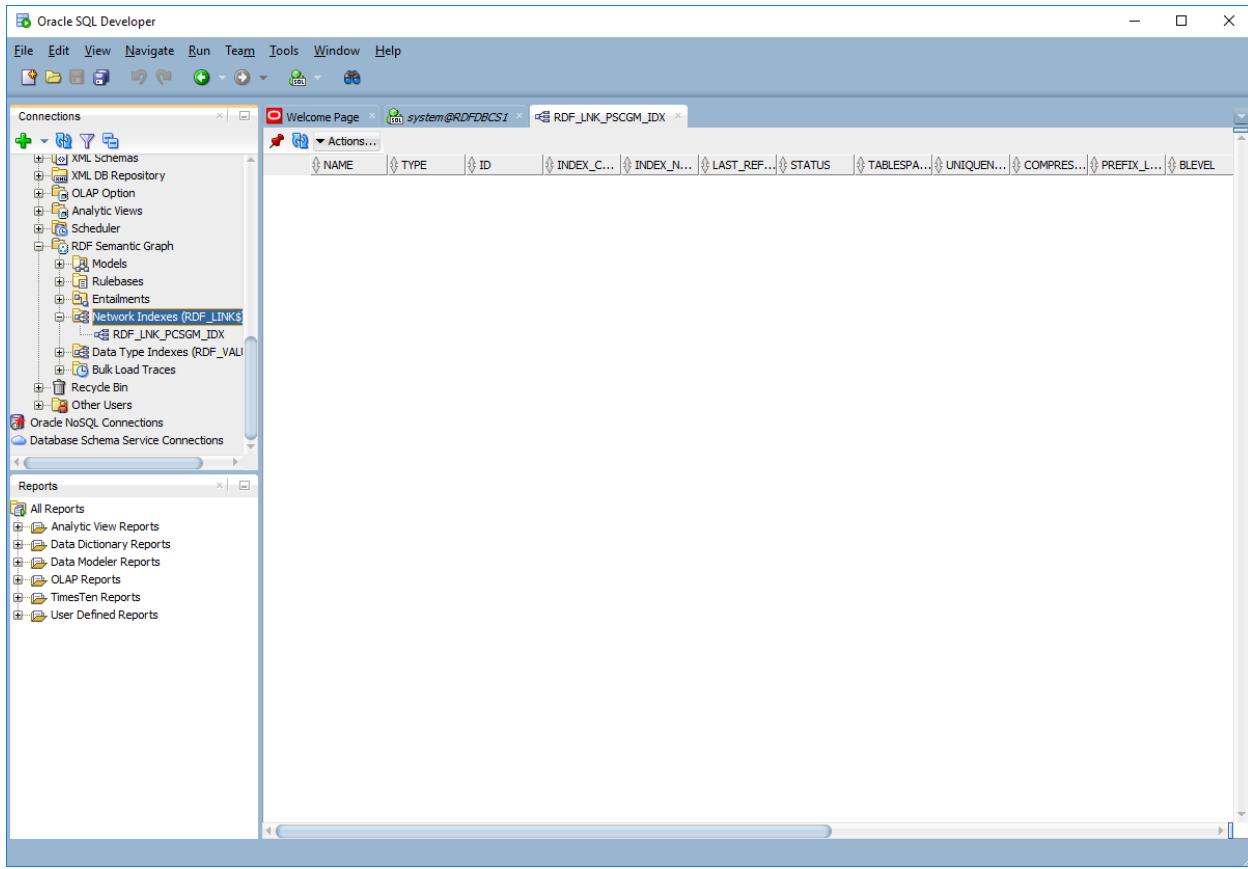
Right click on RDF_LNK_PSCGM_IDX and select Drop Semantic Index.



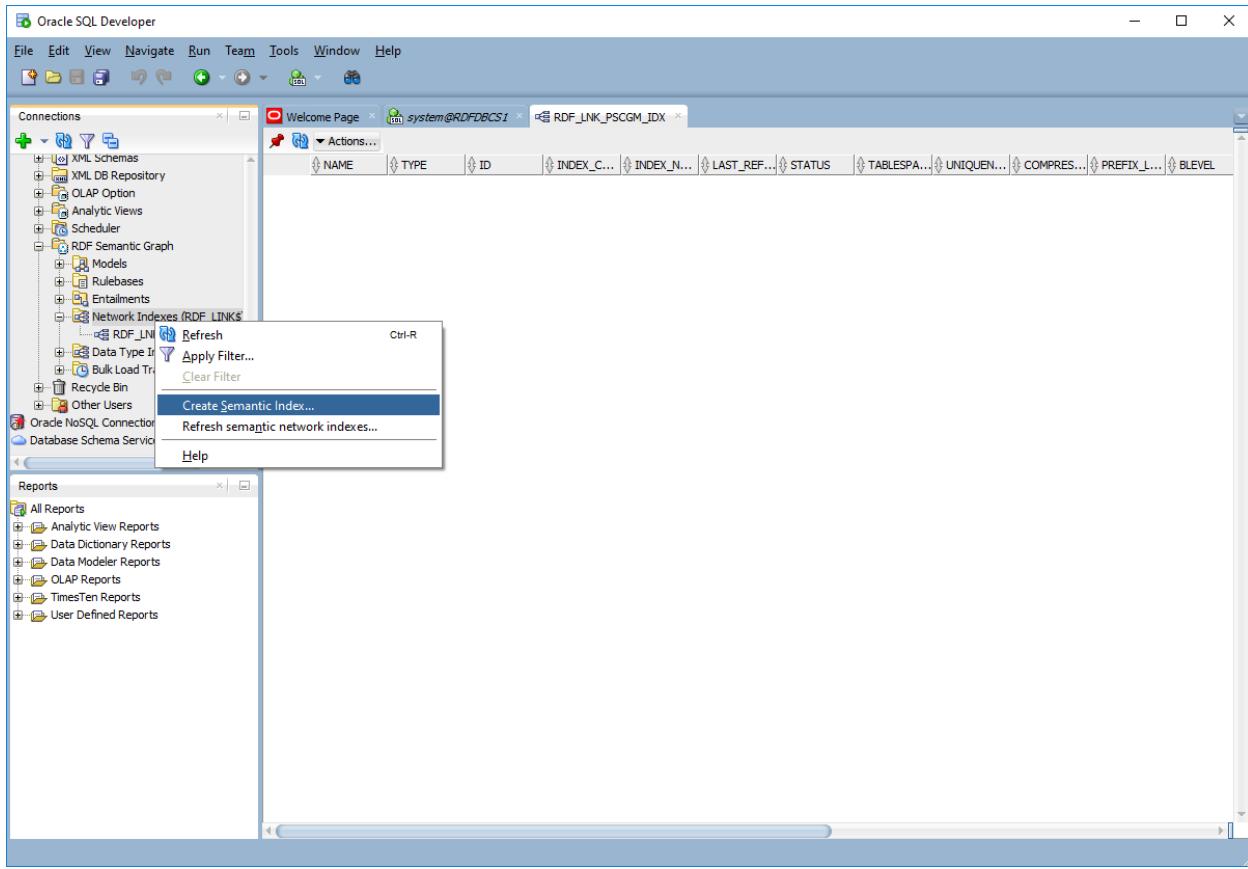
Click Apply.



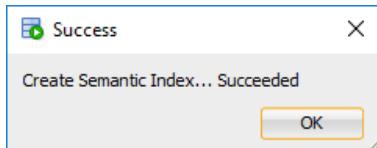
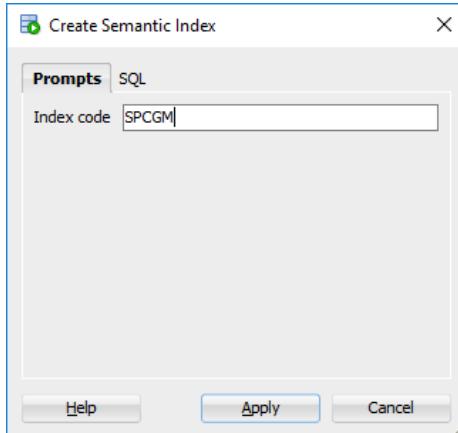
Only the PCSGM index should appear under Network Indexes now.



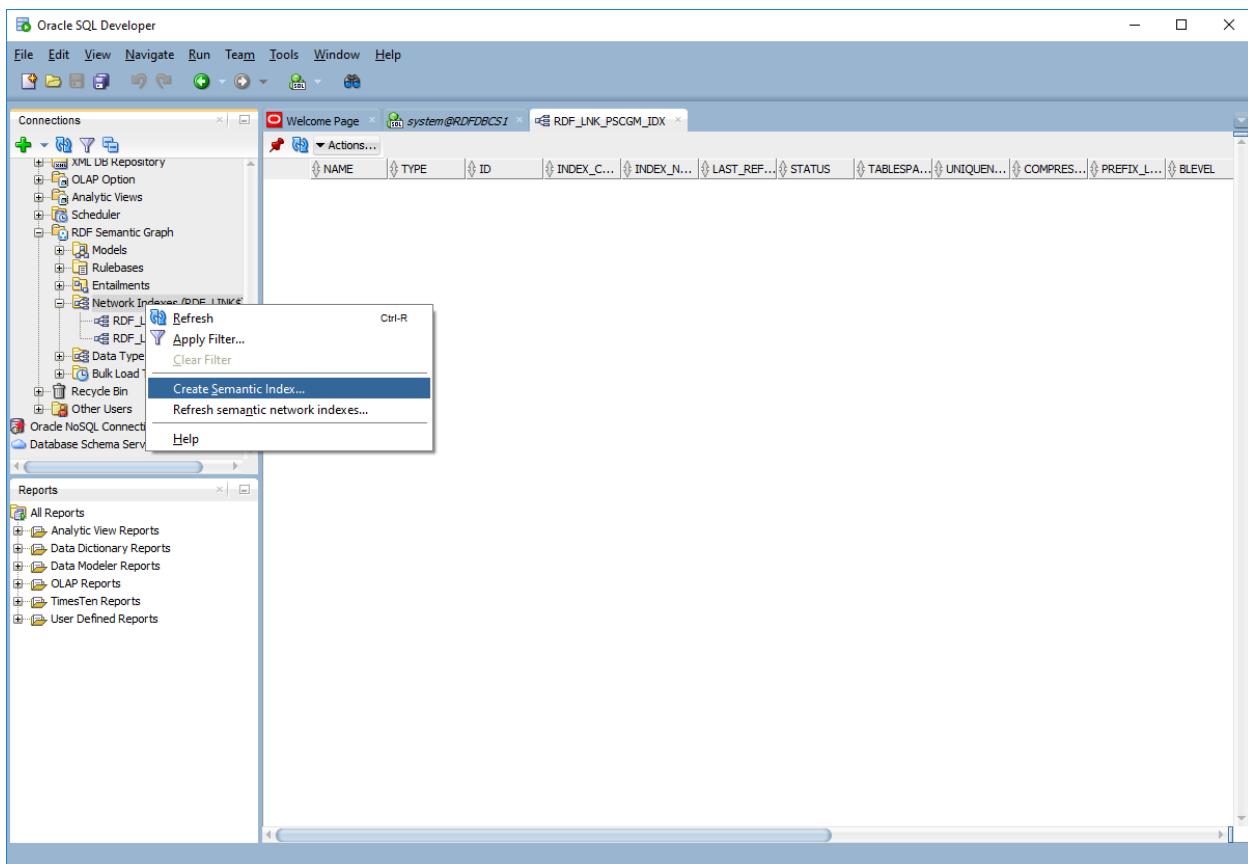
Right click on Network Indexes and select Create Semantic Index.



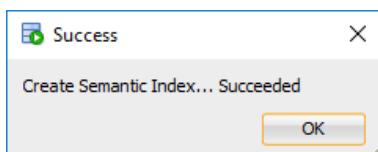
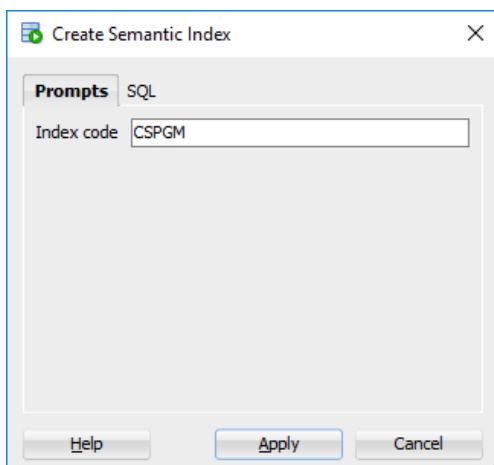
Enter SPCGM as the Index code and click Apply.



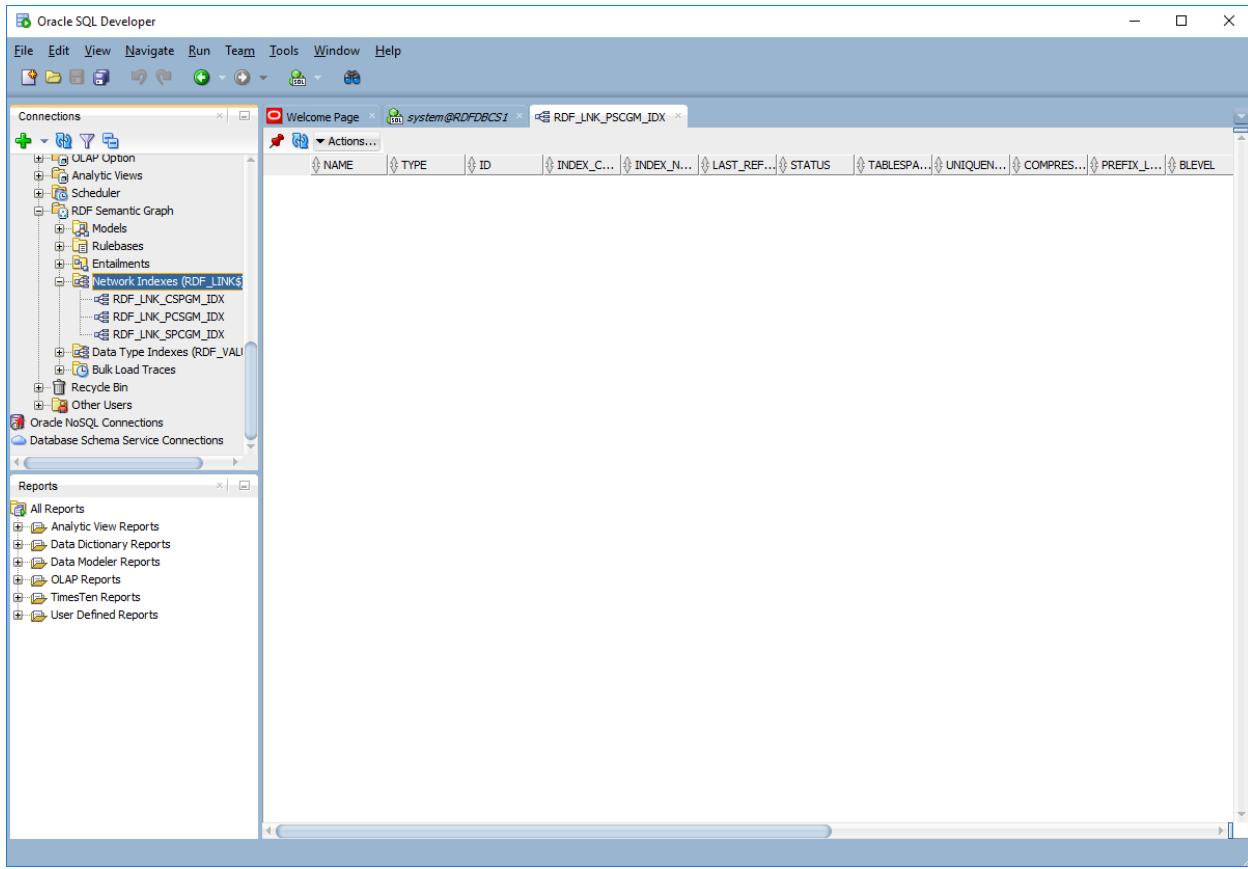
Right click on Network Indexes and select Create Semantic Index.



Enter CSPGM as the index code and click Apply.



You should now see CSPGM, PCSGM and SPCGM indexes under Network Indexes.



The next step is to create a regular non-administrator user to perform operations with the Semantic Network. We can create user RDFUSER by clicking on the SQL Worksheet tab for SYSTEM and running the following SQL statement as SYSTEM:

```
grant connect, resource, unlimited tablespace to rdfuser identified by  
rdfuser;
```

The screenshot shows the Oracle SQL Developer interface. The 'Worksheet' tab is active, displaying the following SQL script:

```

SELECT * FROM MDSYS.RDF_PARAMETER;

create bigfile tablespace rdf�bs
datafile '?/dbs/rdf�bs.dat' size 512M reuse autoextend on next 512M maxsize 10G
extent management local
segment space management auto;

grant connect, resource, unlimited tablespace to rdfuser identified by rdfuser;

```

The 'Script Output' tab below shows the results of the execution:

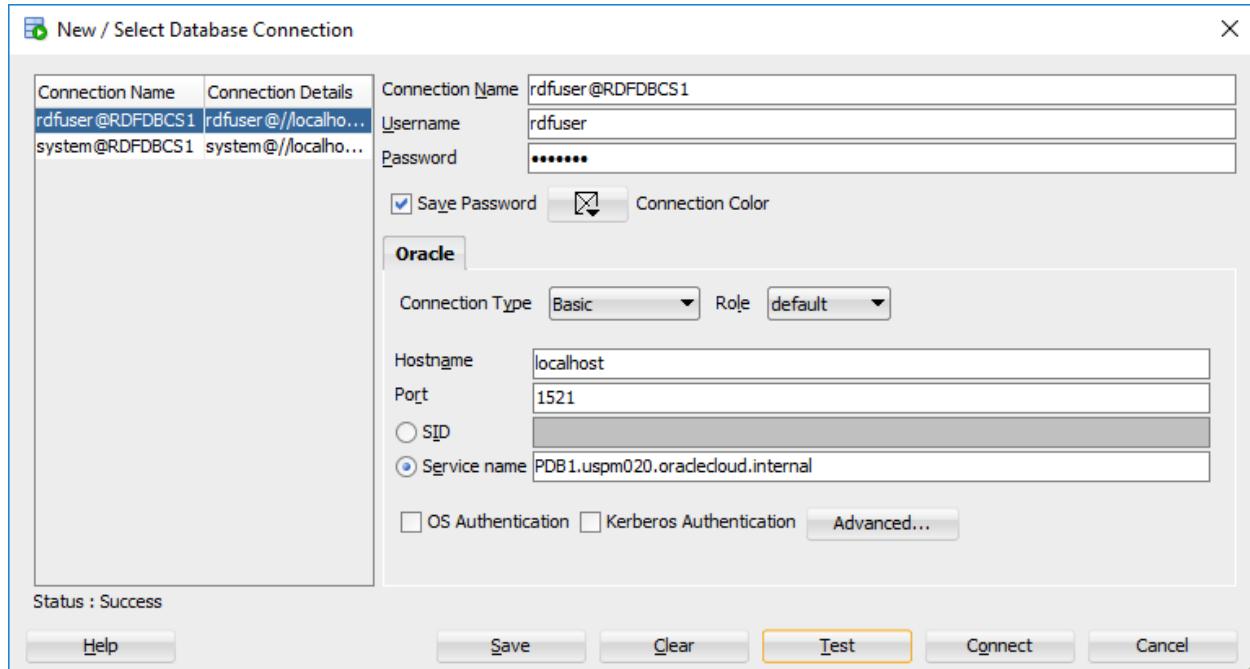
```

Tablespace RDFTBS created.

Grant succeeded.

```

Now create a new connection in SQL Developer for RDFUSER. Use rdfuser for the Username and Password, and use the same Hostname, Port and Service name as we did before for the SYSTEM connection.



Now we will use the RDFUSER account to perform a few operations. We will download a publicly available RDF dataset and transfer it to our DBCS instance. Then we will use SQL Developer to bulk load this RDF dataset and execute some SPARQL queries against it.

The data we will use comes from the Linked Geo Data project: <http://linkedgeodata.org/About>

The screenshot shows the LinkedGeoData.org website. On the left, there's a sidebar with links like About / News, Downloads, Online Access, RDF Mapping, Use Cases, LGD Browser, Publications, Community, Blog, and Contact / Imprint. The main content area has a yellow banner at the top with the date "2018 May 7: Linked Data interface operation back to normal" and "Quick Links: Downloads – SPARQL – Virtual-SPARQL by Sparqlify – HTML interface – Example Queries". Below this, there's a section about LinkedGeoData, news items (including one about new RDF versions of OpenStreetMap datasets), and a map interface where a specific location is selected. A sidebar on the right lists "Instances" related to the selected location, such as Lukaskirche, amity, place_of_worship, religion, and denomination. At the bottom of the page, there's a footer note: "This browser was developed by AKSW".

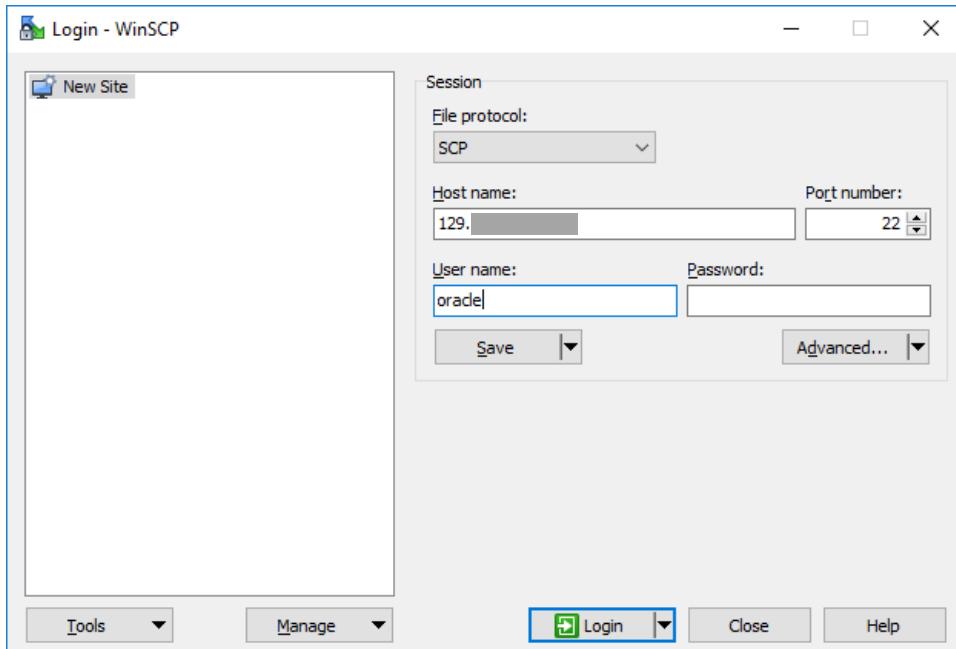
The project provides several downloadable datasets, which are indexed here:

<https://hobbitdata.informatik.uni-leipzig.de/LinkedGeoData/downloads.linkedgeodata.org/releases/>

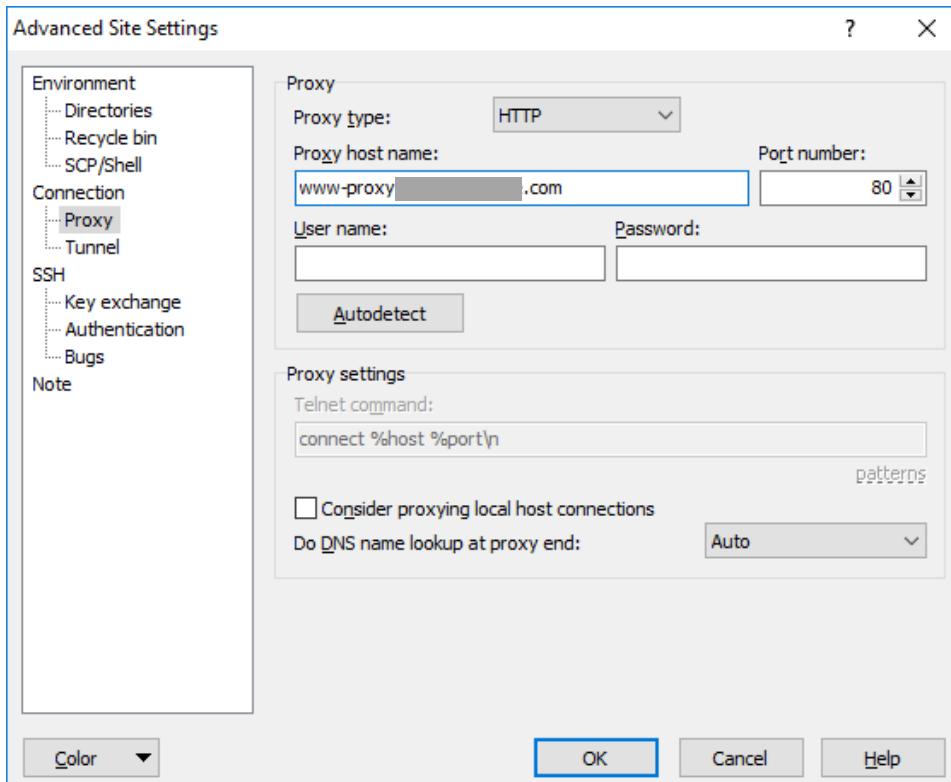
We will be using a 1.2 million triple dataset of sports facilities: <https://hobbitdata.informatik.uni-leipzig.de/LinkedGeoData/downloads.linkedgeodata.org/releases/2015-11-02/2015-11-02-SportThing.node.sorted.nt.bz2>

After downloading this file, you can use WinSCP to copy it to the DBCS instance.

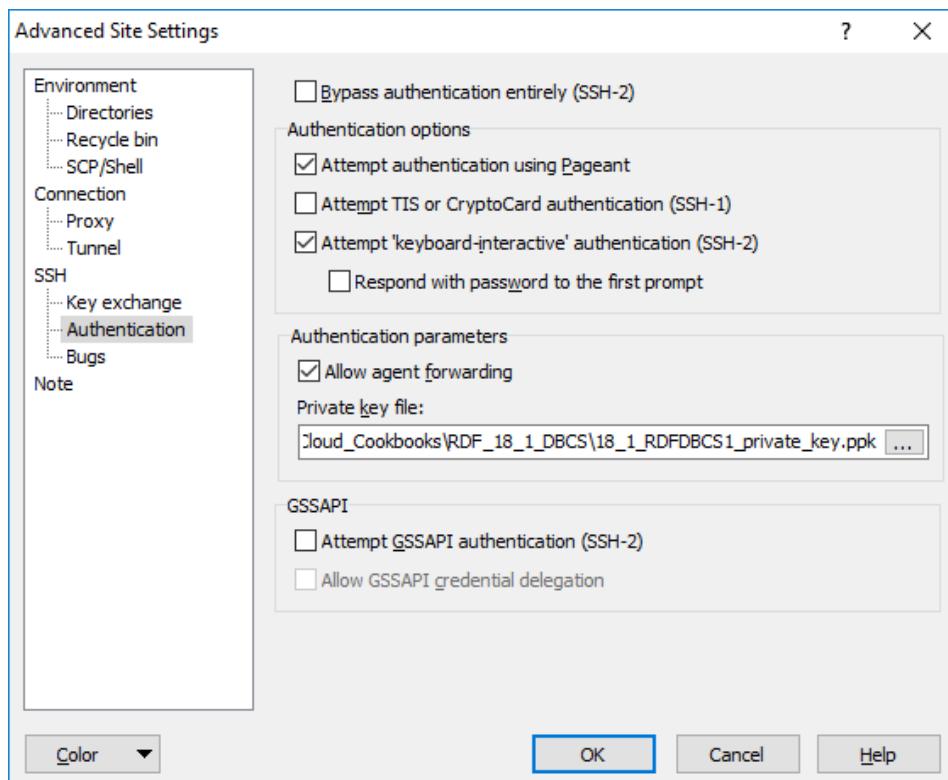
Open WinSCP. Choose SCP File protocol and enter your DBCS public IP for Host name. Enter oracle for User name and leave Password blank.



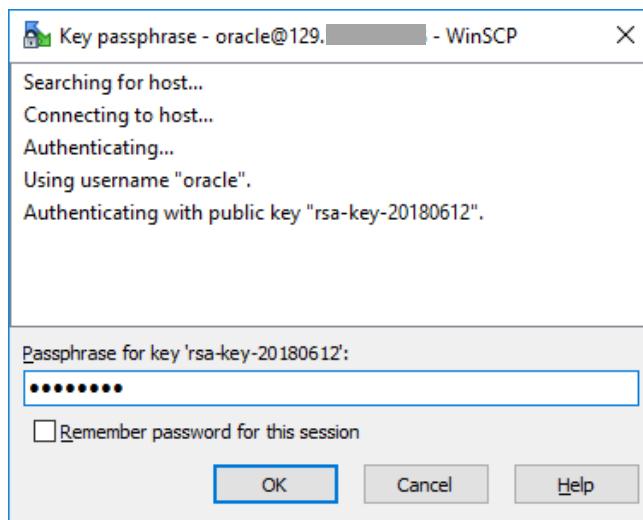
Click Advanced to configure SSH authentication and any proxy settings. Click Proxy and enter any proxy information for your network.



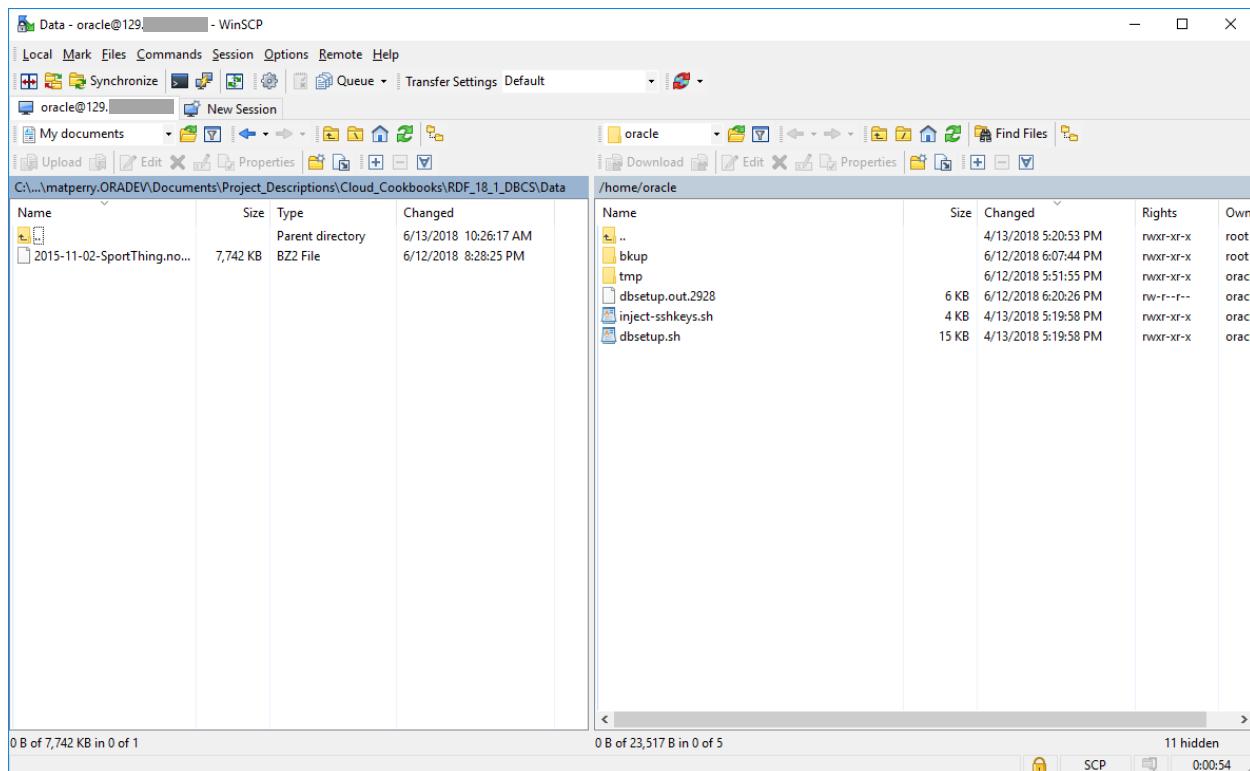
Click Authentication and add your .ppk private key file. Also, uncheck "Respond with password to the first prompt".



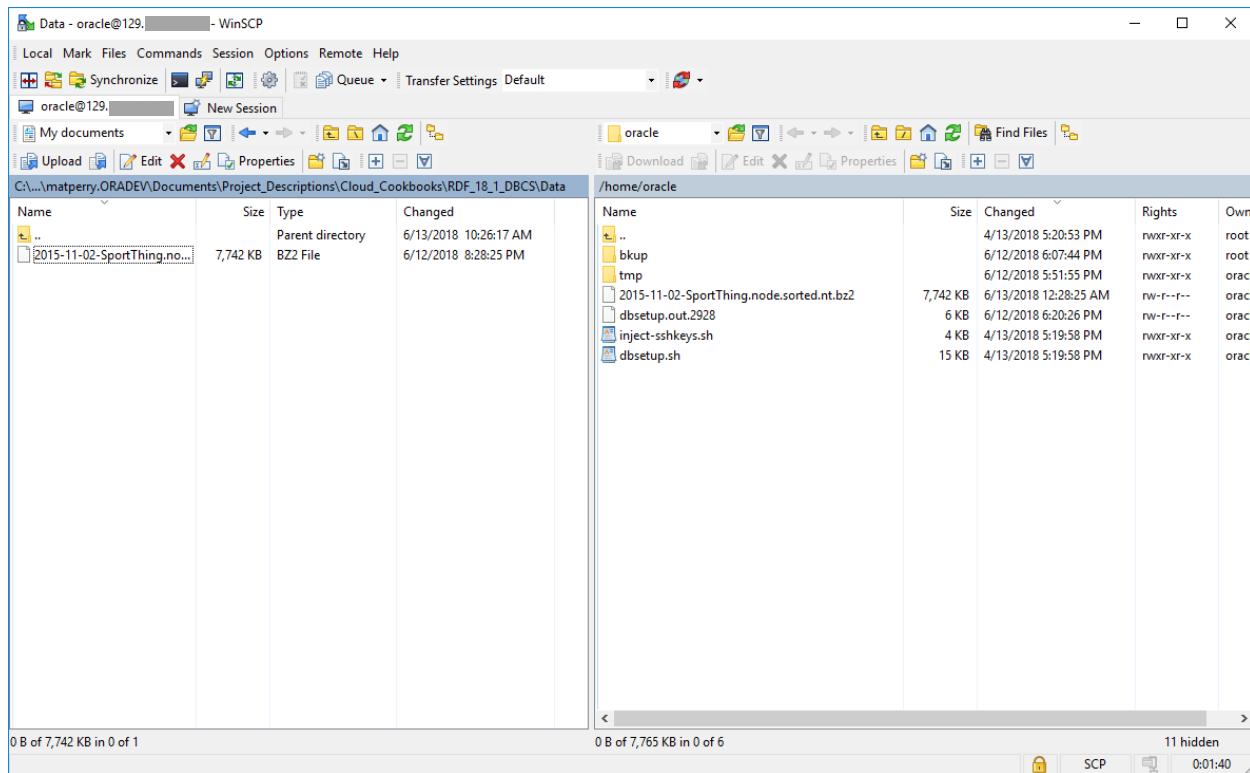
Click OK and then Login. Enter passphrase if necessary.



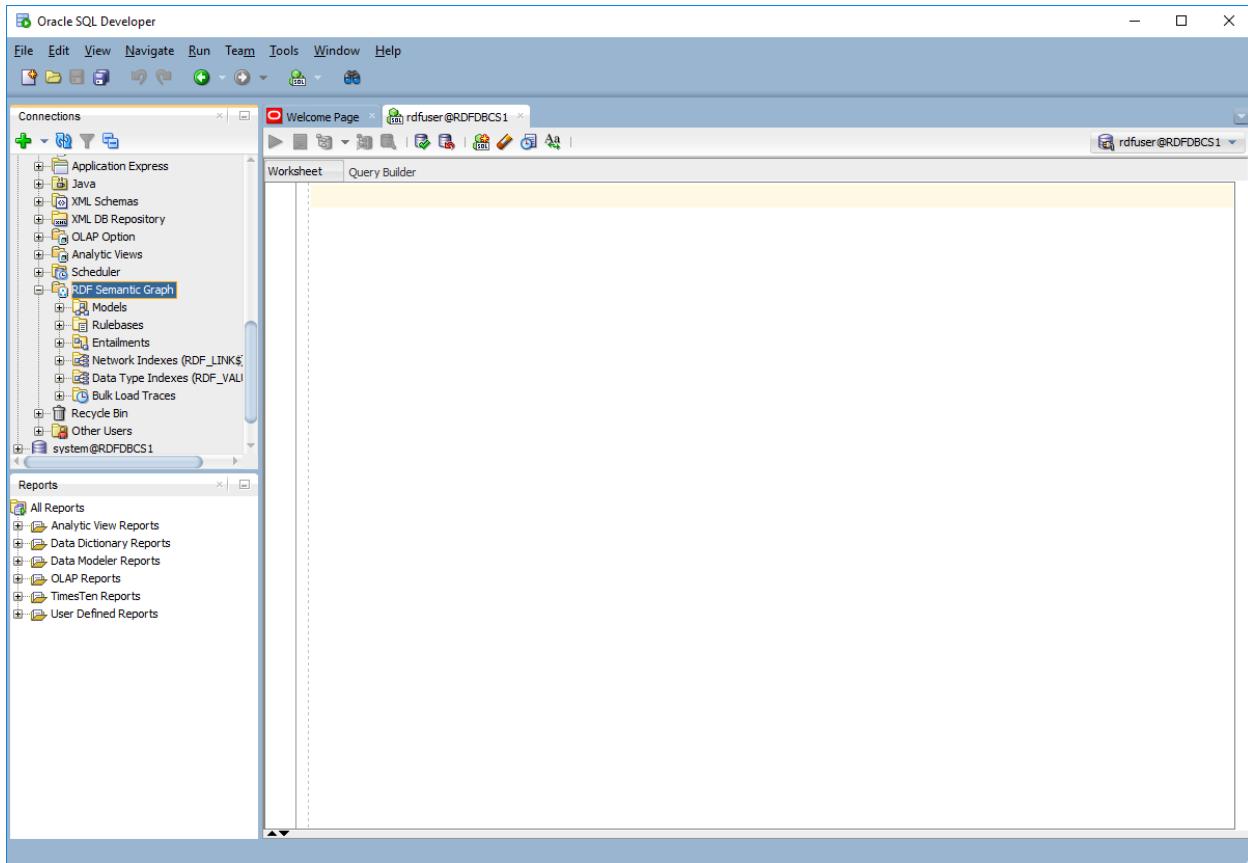
Browse to appropriate directories on the local and remote machines.



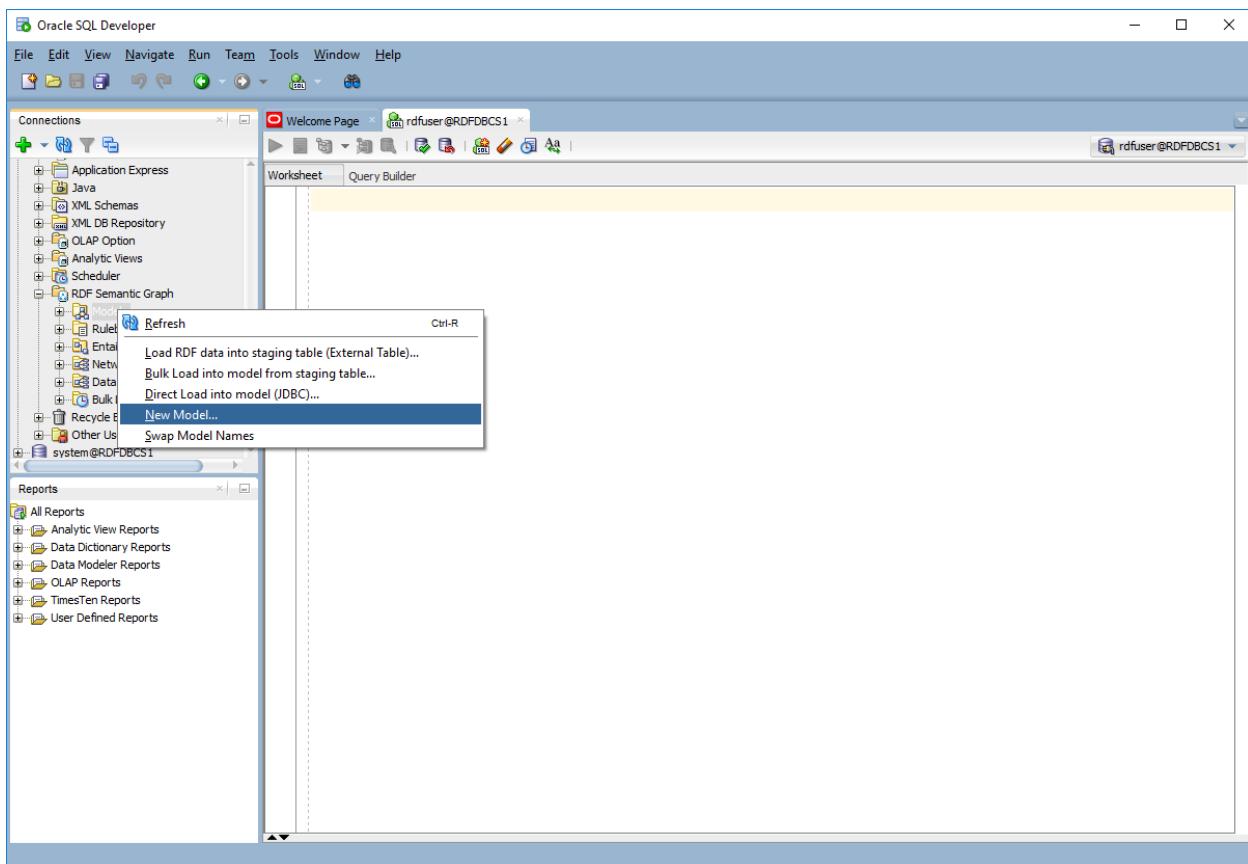
Drag the downloaded file from Linked Geo Data from left to right to do the transfer. In this case, we are storing the data in /home/oracle on the remote DBCS machine.



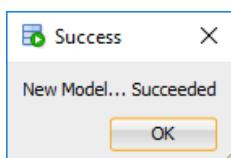
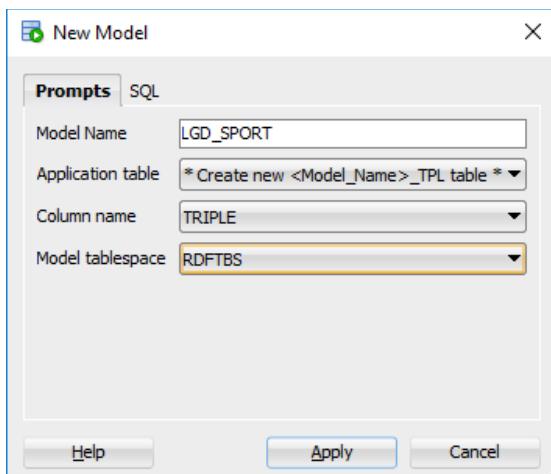
Next, we will create a semantic model to hold our RDF dataset. Open the RDFUSER connection in SQL Developer and expand the RDF Semantic Graph component.



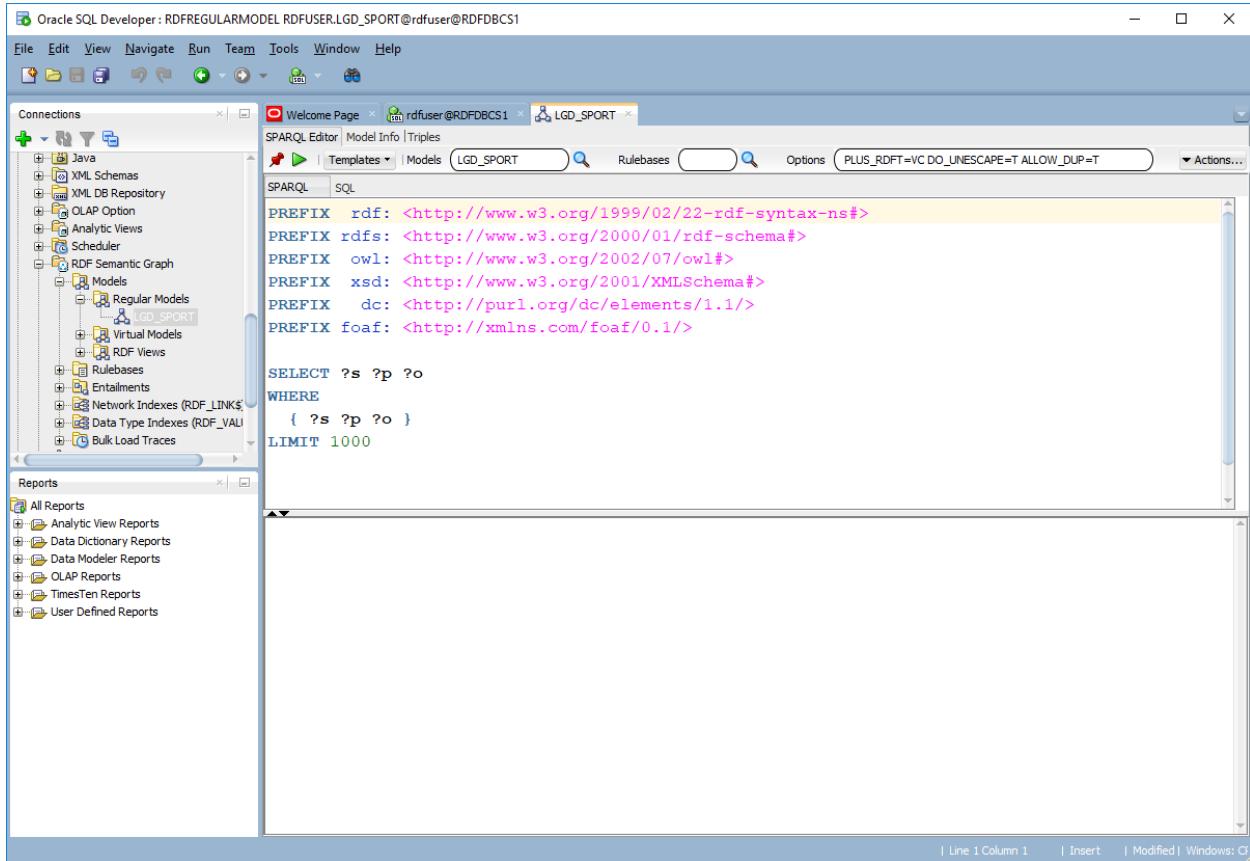
Right click on Models and choose New Model.



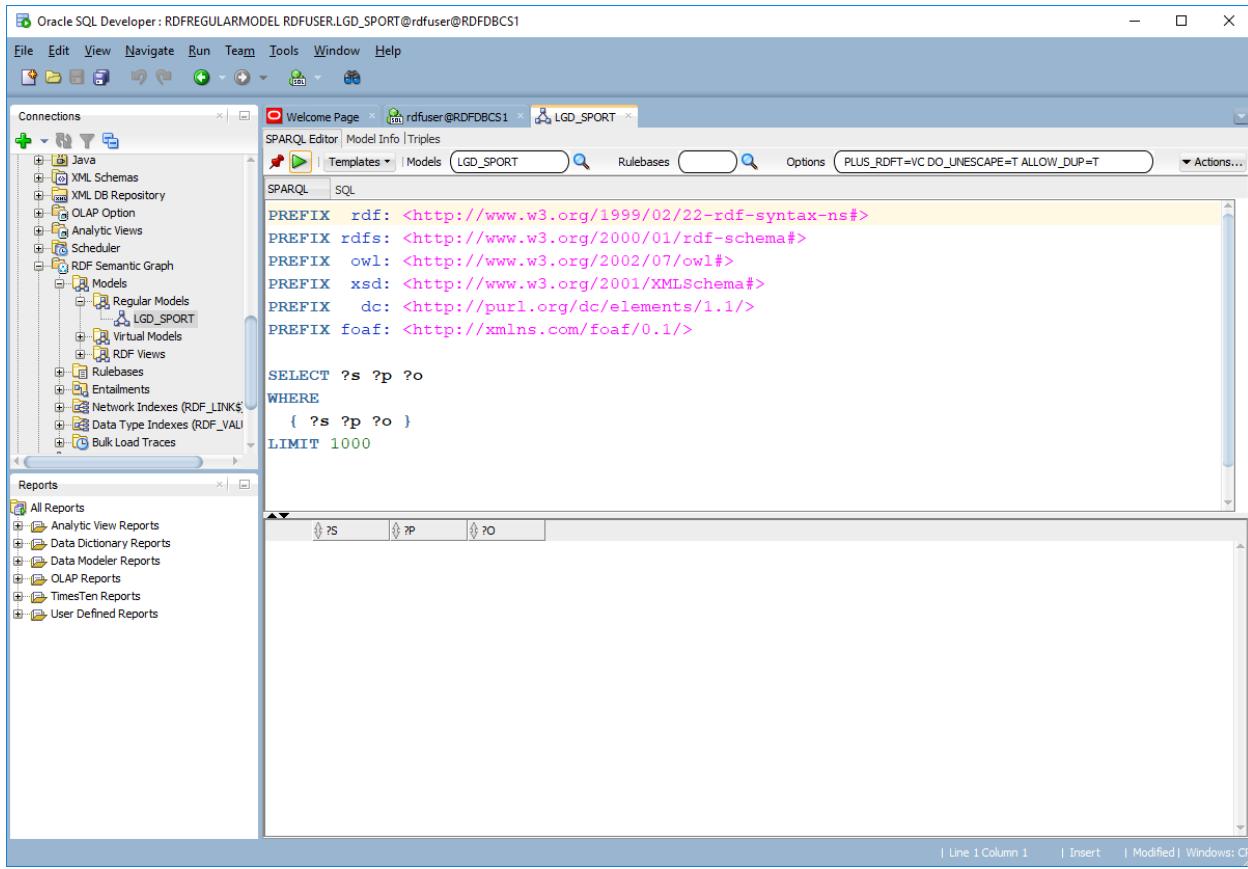
Enter a Model Name and choose to create a new Application table with TRIPLE column. Choose the tablespace you used when creating the semantic network for Model tablespace. Click Apply.



Expand Models and Regular Models and click the LGD_SPORT model we created. This will bring up the SPARQL query editor for LGD_SPORT.



We can click the green triangle to run the default query and see that the model is empty.

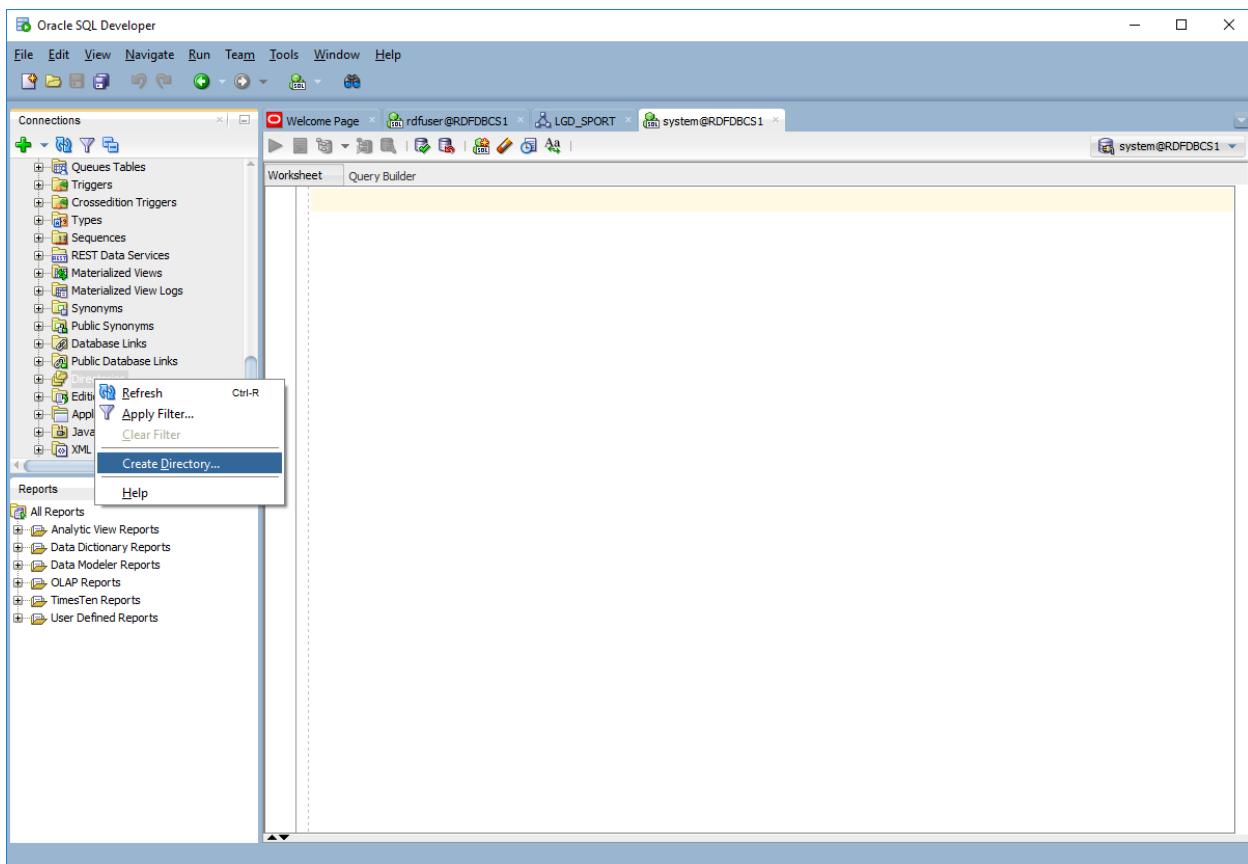


Now, we will bulk load the downloaded RDF file. The bulk load process consists of two major steps:

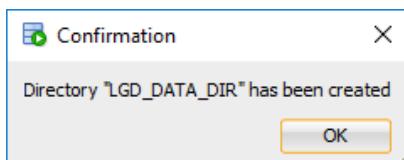
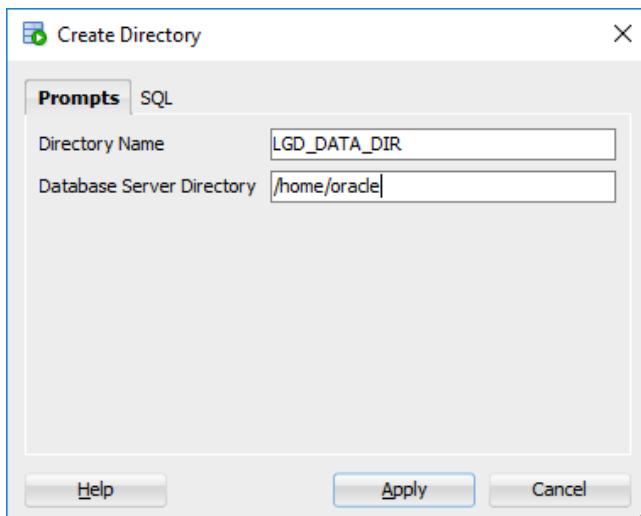
1. Loading the file from the file system into a simple staging table in the database.
2. Loading data from the staging table into our semantic model.

The first step involves loading from an external table, so we need to use the SYSTEM connection to create a DIRECTORY in the database and grant privileges on it to RDFUSER.

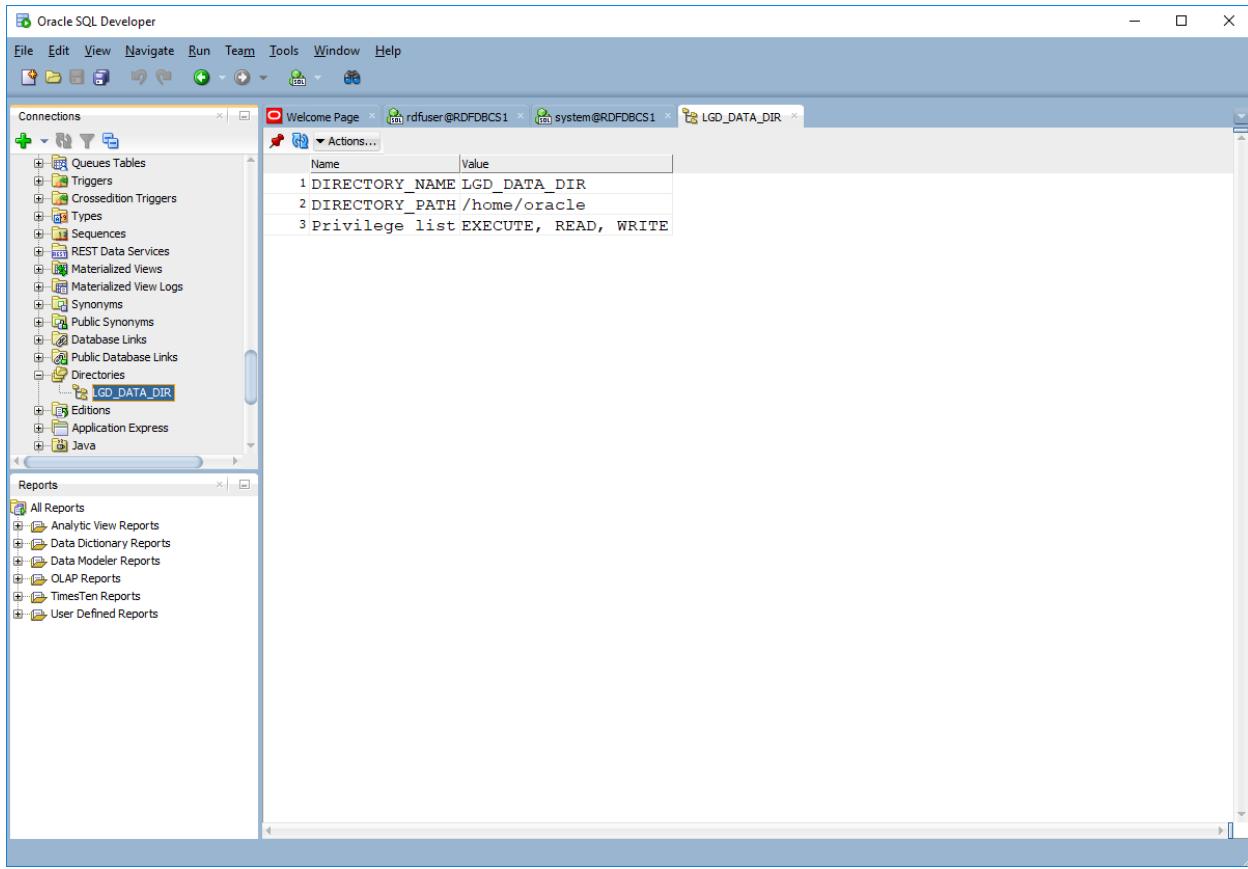
Expand the SYSTEM connection and right-click on Directories. Then choose Create Directory.



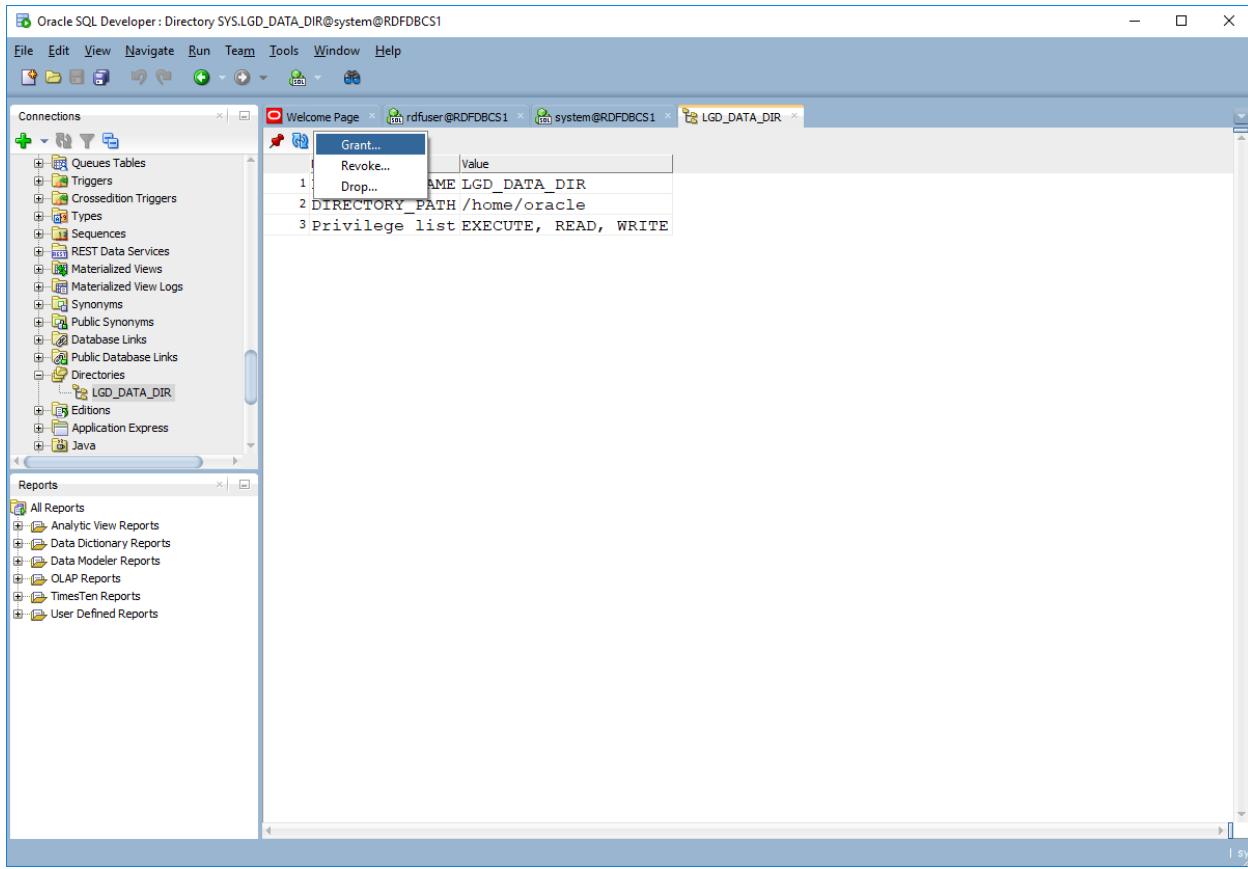
Enter a Directory Name and the full path for the directory on the database server. Click Apply.



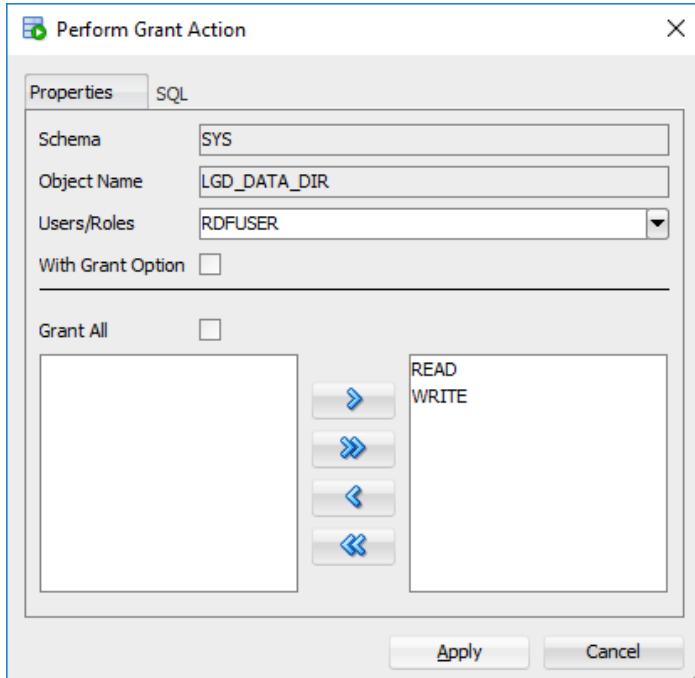
Expand Directories and click the directory name to see details.

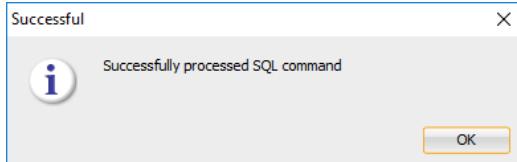


Now we need to grant privileges on this directory to RDFUSER. Click actions and select Grant.



Grant READ and WRITE privileges to RDFUSER. Click Apply.

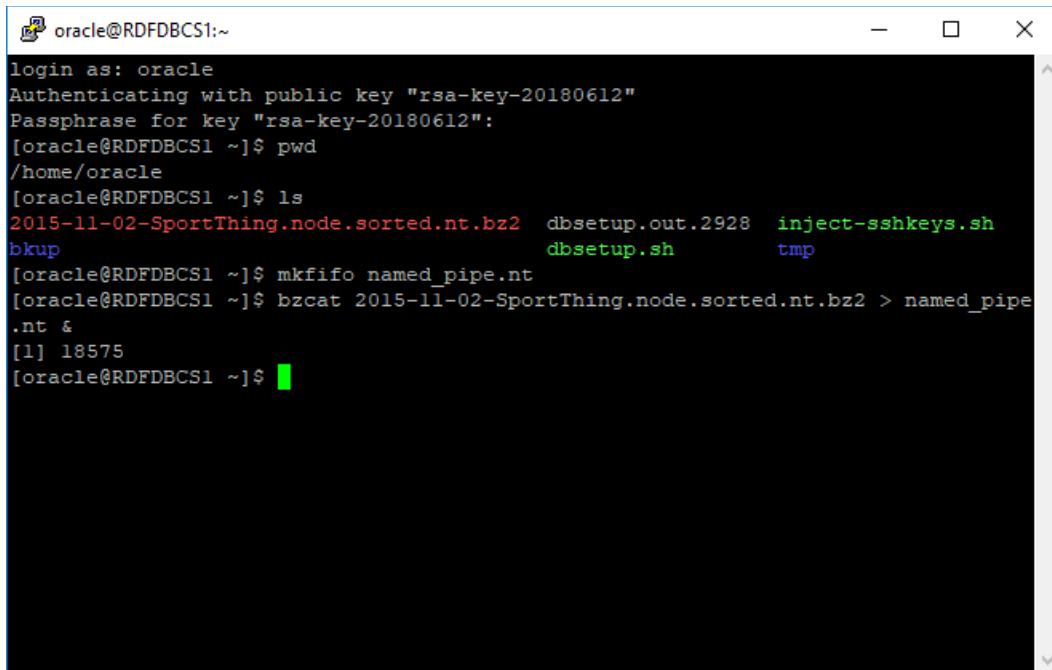




The downloaded RDF file is compressed, so we will use a Unix named pipe to stream the uncompressed data rather than storing an uncompressed version of the file.

Use PuTTY to make an SSH connection to the remote DBCS instance as the oracle user. Then execute the following commands to create a named pipe to stream the uncompressed data.

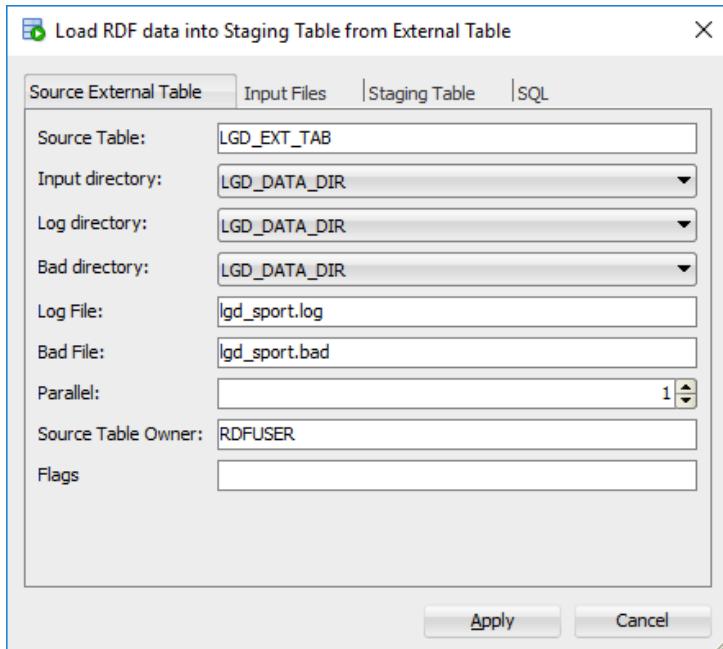
```
mkfifo named_pipe.nt  
bzcat 2015-11-02-SportThing.node.sorted.nt.bz2 > named_pipe.nt &
```

A screenshot of a terminal window titled "oracle@RDFDBCS1:~". The session shows the user logging in as oracle, navigating to their home directory, listing files, and then executing the commands to create a named pipe and stream data from a bz2 file into it.

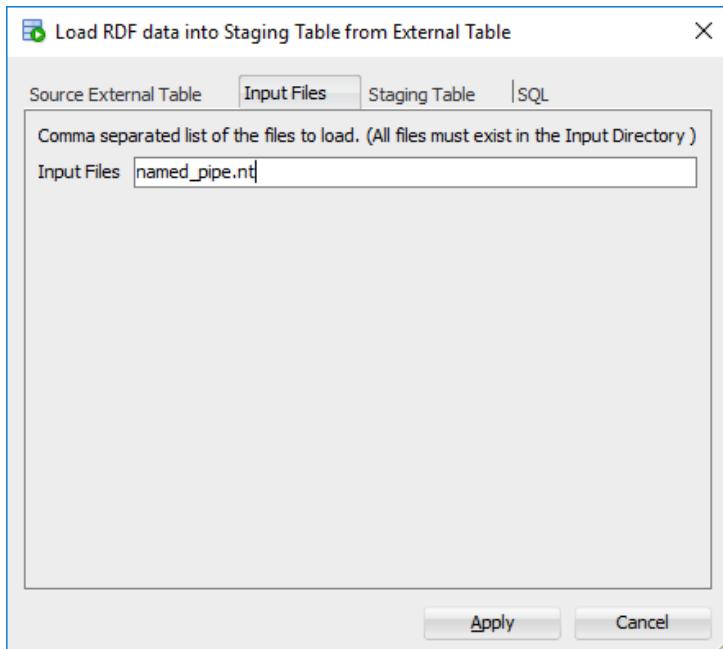
```
login as: oracle  
Authenticating with public key "rsa-key-20180612"  
Passphrase for key "rsa-key-20180612":  
[oracle@RDFDBCS1 ~]$ pwd  
/home/oracle  
[oracle@RDFDBCS1 ~]$ ls  
2015-11-02-SportThing.node.sorted.nt.bz2  dbsetup.out.2928  inject-sshkeys.sh  
bkup                                dbsetup.sh        tmp  
[oracle@RDFDBCS1 ~]$ mkfifo named_pipe.nt  
[oracle@RDFDBCS1 ~]$ bzcat 2015-11-02-SportThing.node.sorted.nt.bz2 > named_pipe  
.nt &  
[1] 18575  
[oracle@RDFDBCS1 ~]$
```

Now, expand the RDFUSER connection in SQL Developer and expand the RDF Semantic Graph component. Then right click on Models and select “Load RDF data into staging table (External Table).”

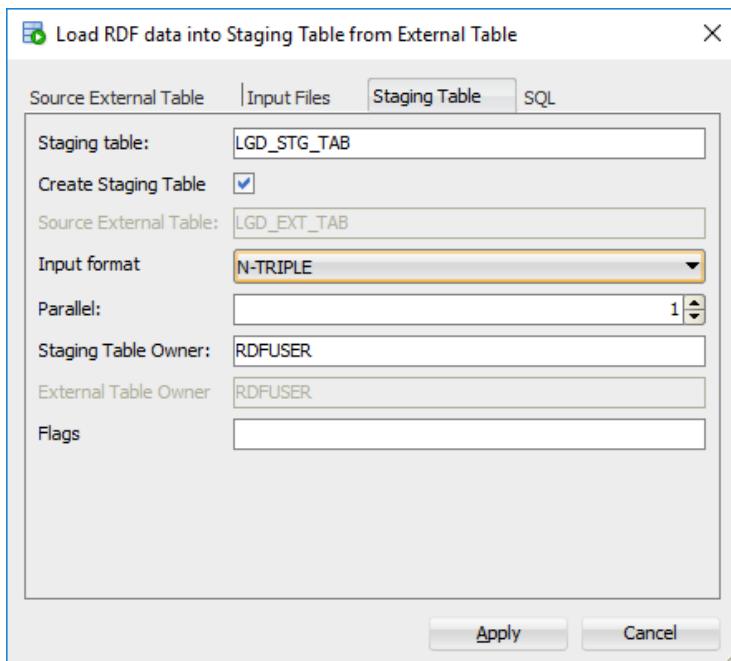
Choose a name for the external table to create (we are using LGD_EXT_TAB) and fill in the other fields on the Source External Table tab.



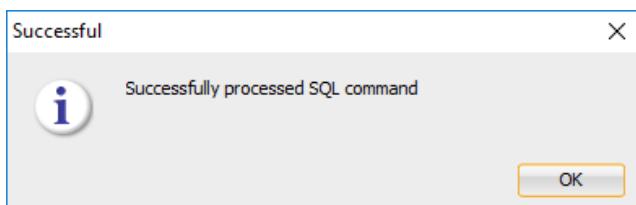
Enter the names of the files to load (we are using named_pipe.nt) on the Input Files tab.



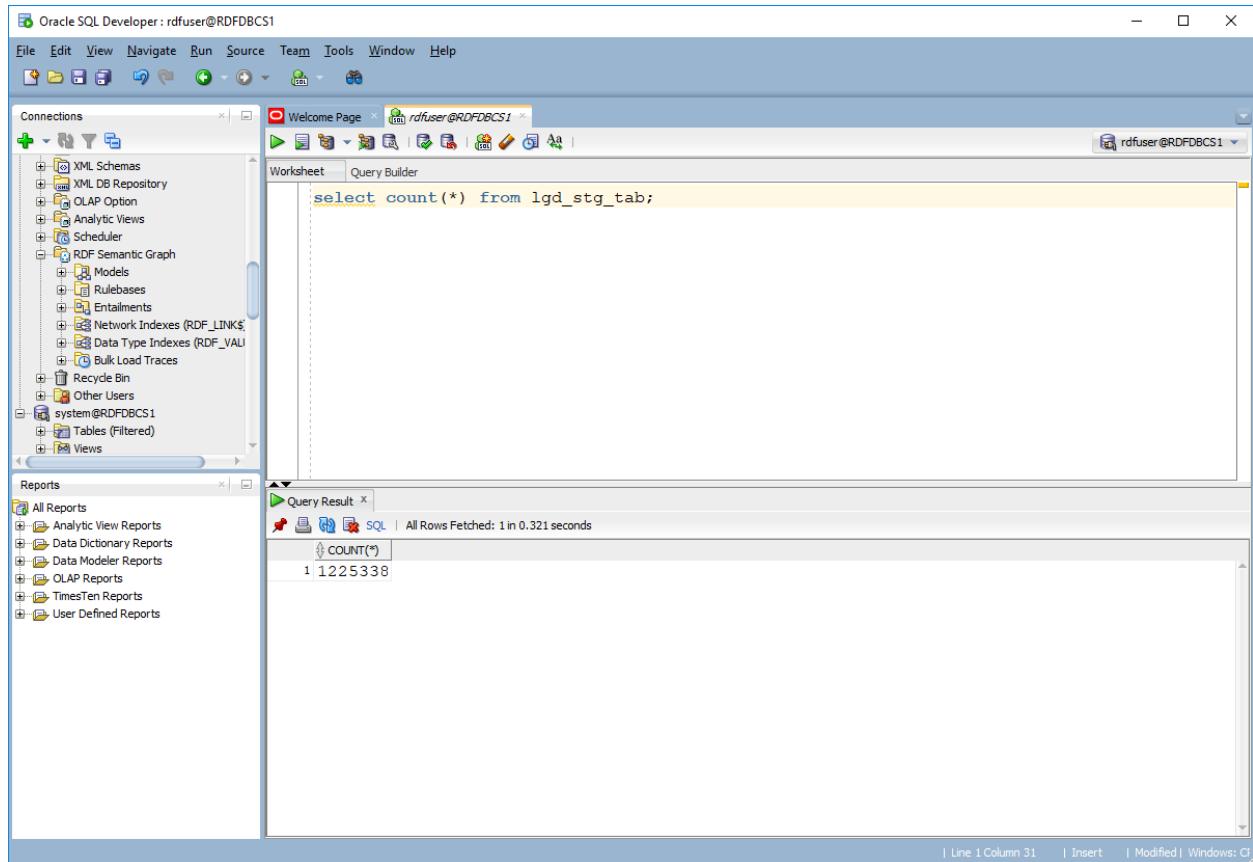
Finally, use the Staging Table tab to enter a name for the staging table that will be created (we are using LGD_STG_TAB) and choose the appropriate format.



Now click Apply to load the data into LGD_STG_TAB.



Check the contents of LGD_STG_TAB.



The screenshot shows the Oracle SQL Developer interface. The top menu bar includes File, Edit, View, Navigate, Run, Source, Team, Tools, Window, and Help. The Connections sidebar shows a connection named 'rdfuser@RDFDBCS1' under the 'RDF Semantic Graph' section. The Worksheet tab contains the following SQL code:

```

select count(*) from lgd_stg_tab;

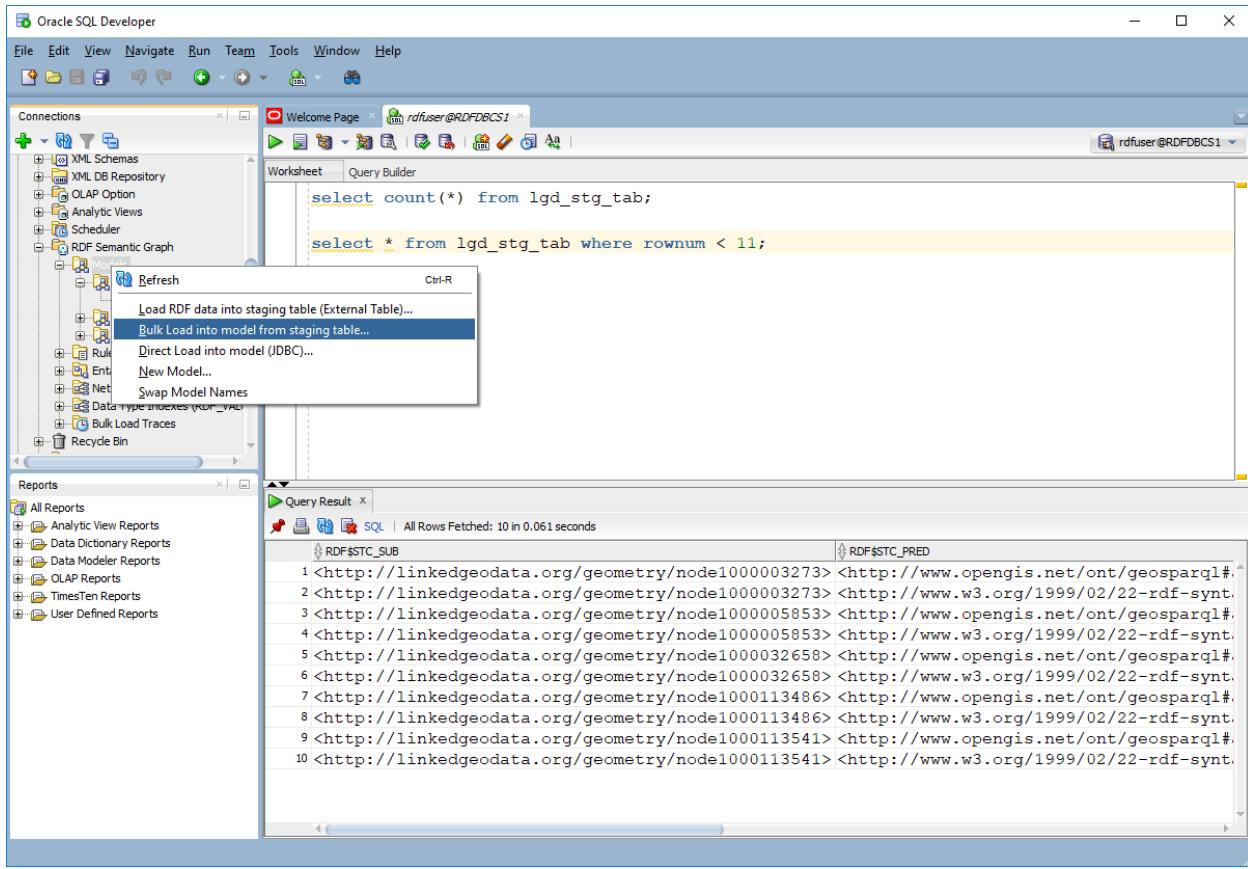
select * from lgd_stg_tab where rownum < 11;

```

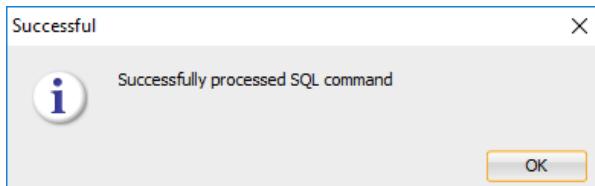
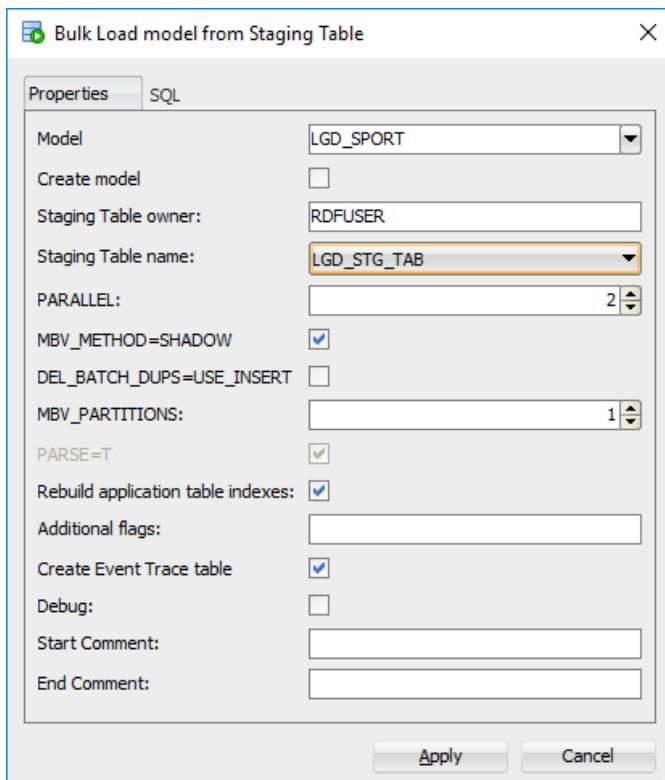
The Query Result tab displays the output of the second query, showing 10 rows of RDF triples. The columns are labeled 'RDF\$STC_SUB' and 'RDF\$STC_PRED'. The data consists of 10 lines of triples, each starting with a subject URI followed by a predicate URI.

RDF\$STC_SUB	RDF\$STC_PRED
1 <http://linkedgeodata.org/geometry/node1000003273>	<http://www.opengis.net/ont/geosparql#
2 <http://linkedgeodata.org/geometry/node1000003273>	<http://www.w3.org/1999/02/22-rdf-synt
3 <http://linkedgeodata.org/geometry/node1000005853>	<http://www.opengis.net/ont/geosparql#
4 <http://linkedgeodata.org/geometry/node1000005853>	<http://www.w3.org/1999/02/22-rdf-synt
5 <http://linkedgeodata.org/geometry/node1000032658>	<http://www.opengis.net/ont/geosparql#
6 <http://linkedgeodata.org/geometry/node1000032658>	<http://www.w3.org/1999/02/22-rdf-synt
7 <http://linkedgeodata.org/geometry/node1000113486>	<http://www.opengis.net/ont/geosparql#
8 <http://linkedgeodata.org/geometry/node1000113486>	<http://www.w3.org/1999/02/22-rdf-synt
9 <http://linkedgeodata.org/geometry/node1000113541>	<http://www.opengis.net/ont/geosparql#
10 <http://linkedgeodata.org/geometry/node1000113541>	<http://www.w3.org/1999/02/22-rdf-synt

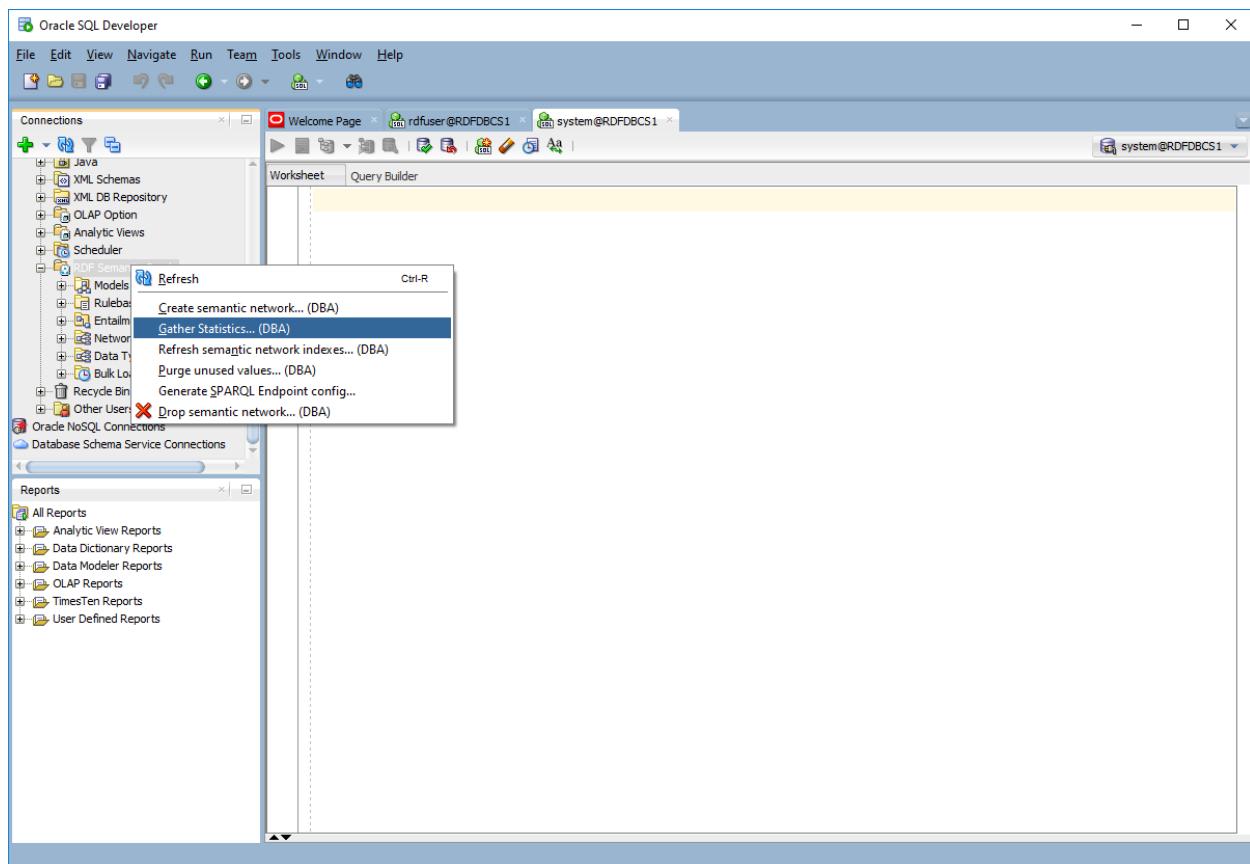
Next, we will load the data from LGD_STG_TAB into the LGD_SPORT semantic model. To bring up the bulk load interface, expand RDF Semantic Graph under the RDFUSER connection. Then, right-click on Models and select “Bulk Load into model from staging table”.



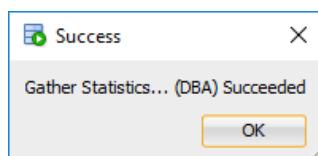
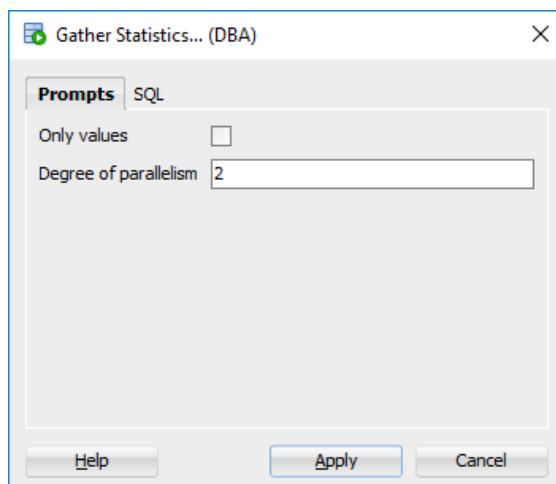
Enter LGD_SPORT for Model and unselect the Create model option since we have already created this semantic model. Also, choose the LGD_STG_TAB for Staging table name. Be careful not to select the external table (LGD_EXT_TAB), as it will also be listed. Consult the [user guide](#) for more information on the other options for bulk load. Click Apply, and the load will finish in a minute or so.



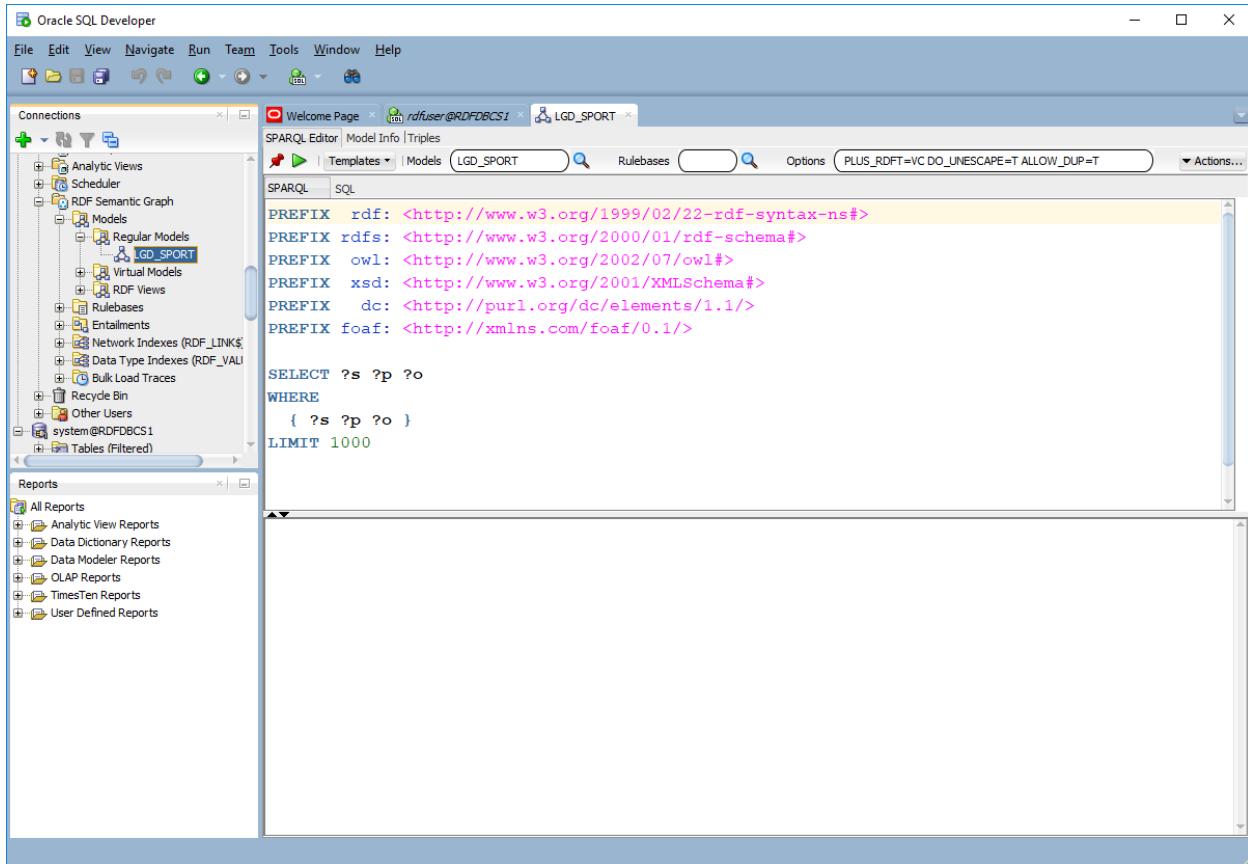
Now that we have completed the bulk load, it is a good idea to gather statistics on the whole RDF network. Only a privileged user can gather statistics on the whole RDF network. Expand the RDF Semantic Graph component under the SYSTEM connection, right click on RDF Semantic Graph and select Gather Statistics.



Enter the desired Degree of parallelism and click Apply.

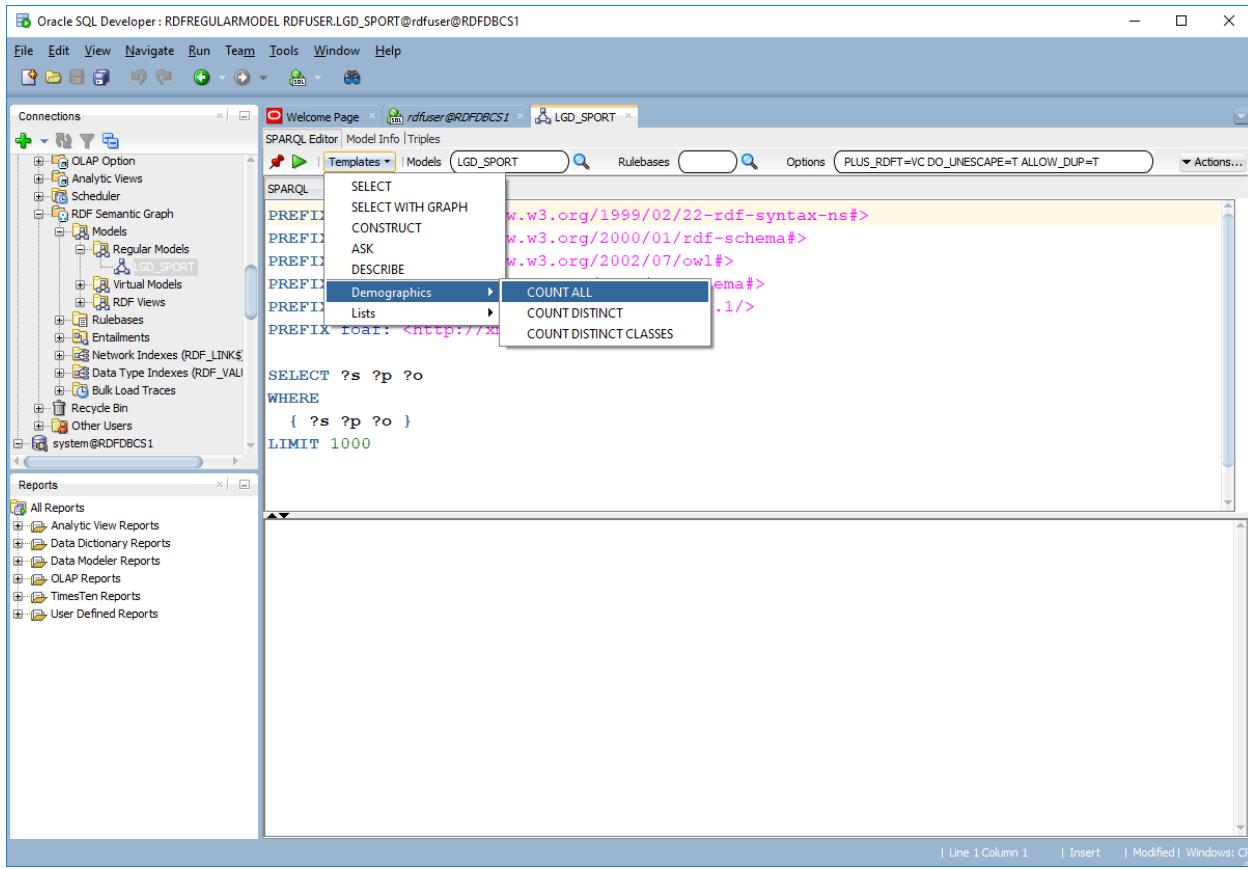


After gathering statistics, the data is ready to query. Go back to the RDFUSER connection and expand Models and Regular Models under RDF Semantic Graph. Clicking LGD_SPORT will open the SPARQL query editor for this semantic model.

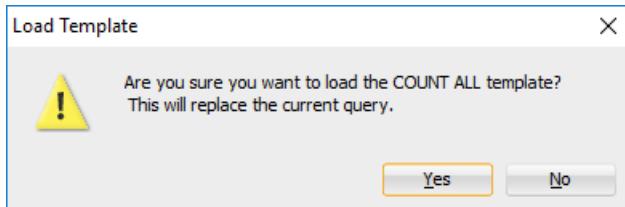


You can edit and execute SPARQL queries here. In addition, several pre-created templates are available.

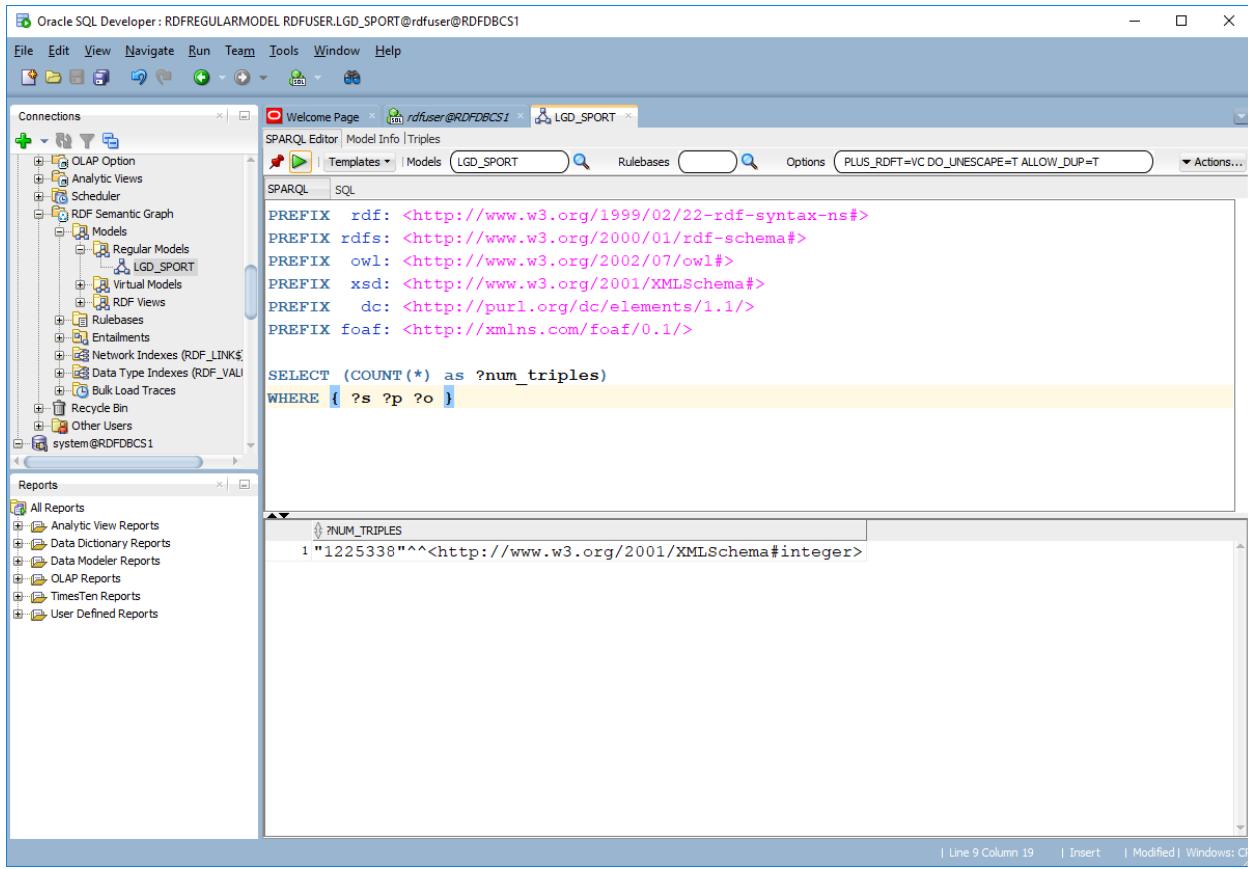
Count All triples.



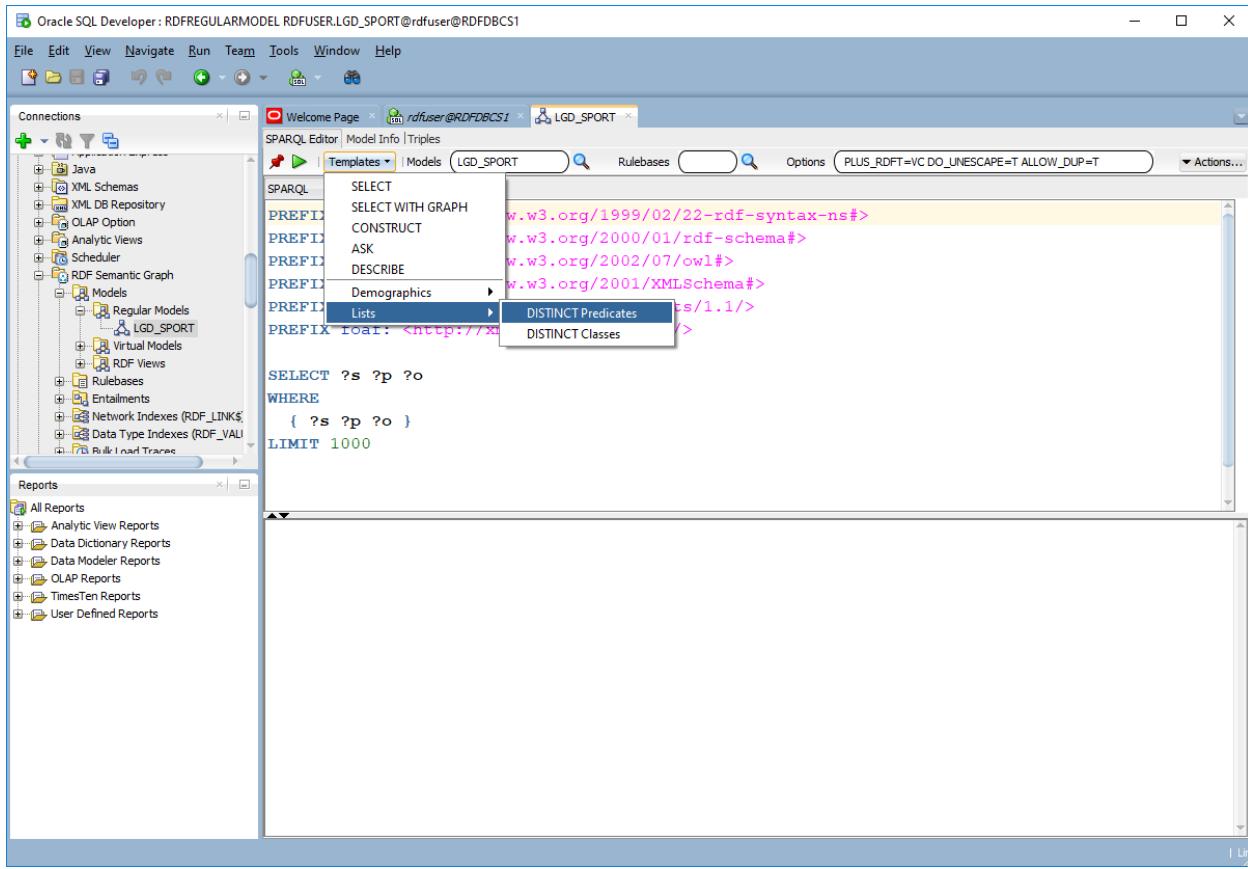
Click Yes when the warning comes up.



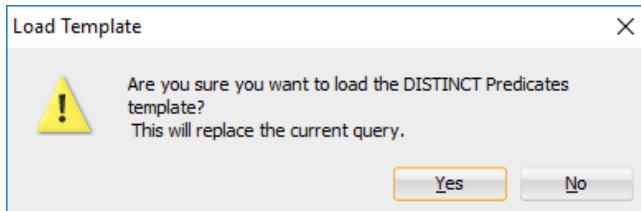
Click the green triangle to run the query.



List of distinct predicates.



Click Yes.



Click the green triangle to run the query.

The screenshot shows the Oracle SQL Developer interface with the following details:

- Title Bar:** Oracle SQL Developer : RDFREGULARMODEL RDFUSER.LGD_SPORT@rdfuser@RDFDBCS1
- Menu Bar:** File Edit View Navigate Run Team Tools Window Help
- Toolbar:** Includes icons for New, Open, Save, Print, and various developer tools.
- Connections Tab:** Shows a list of connections including Java, XML Schemas, XML DB Repository, OLAP Option, Analytic Views, Scheduler, and RDF Semantic Graph. Under RDF Semantic Graph, there is a 'Models' section with 'Regular Models' containing 'LGD_SPORT'.
- SPARQL Editor Tab:** The active tab. It has tabs for Templates, Models, LGD_SPORT (selected), Rulebases, and Options (with setting PLUS_RDFT=VC DO_UNESCAPE=T ALLOW_DUP=T). The SPARQL tab contains the following query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>

SELECT DISTINCT ?p
WHERE { ?s ?p ?o }

```

- Results Area:** Displays the results of the query as a list of triples. The first 14 results are shown:

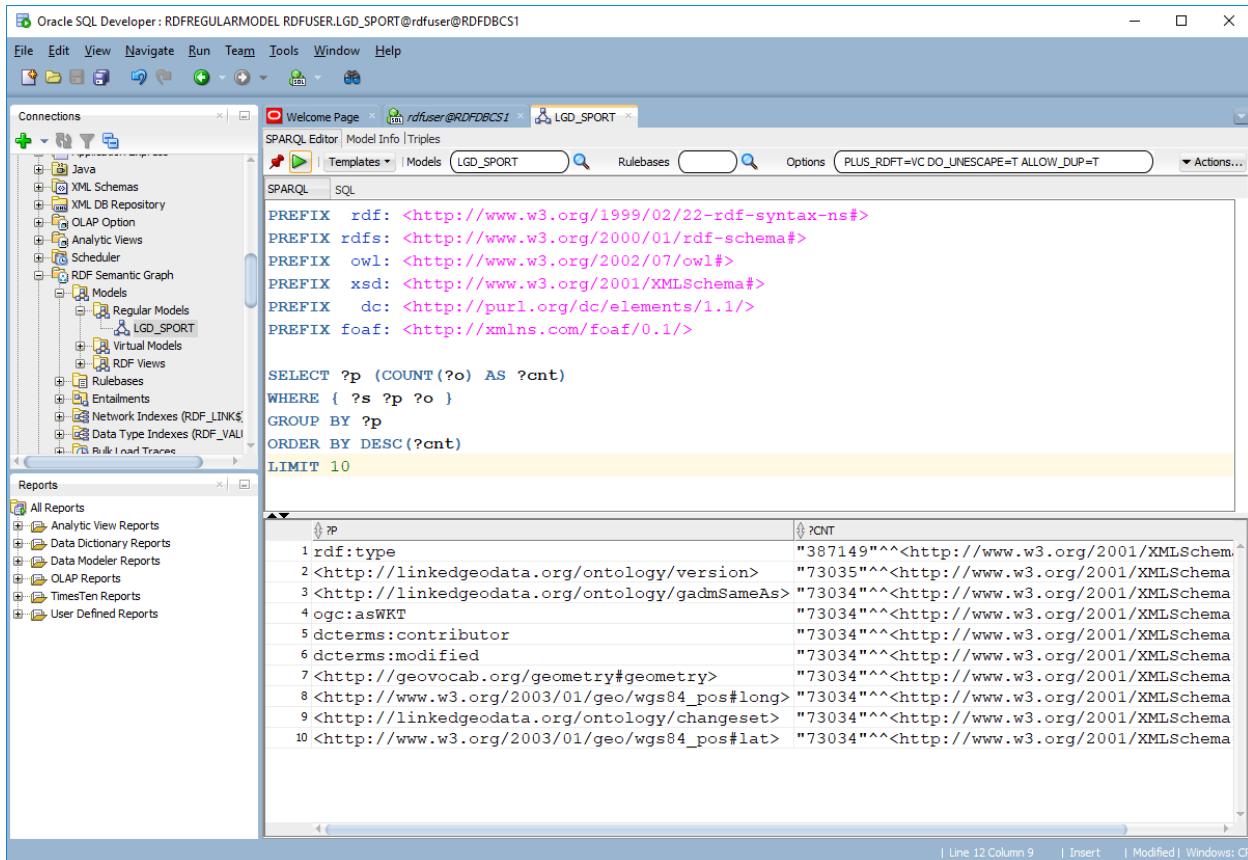
```

1 <http://linkedgeodata.org/ontology/gadmSameAs>
2 <http://linkedgeodata.org/ontology/type>
3 <http://www.w3.org/2003/01/geo/wgs84_pos#lat>
4 <http://linkedgeodata.org/ontology/surface>
5 <http://linkedgeodata.org/ontology/ref>
6 <http://linkedgeodata.org/ontology/addr%3AhouseNumber>
7 <http://linkedgeodata.org/ontology/boules>
8 <http://linkedgeodata.org/ontology/designation>
9 <http://linkedgeodata.org/ontology/scuba_diving%3Aentry%3Ashore>
10 <http://linkedgeodata.org/ontology/climbing%3Agrade%3AUAAA%3Amax>
11 <http://linkedgeodata.org/ontology/aircraft>
12 <http://linkedgeodata.org/ontology/table_tennis%3Atabletop>
13 <http://linkedgeodata.org/ontology/sport_3>
14 <http://linkedgeodata.org/ontology/scuba_diving%3Adifficulty>

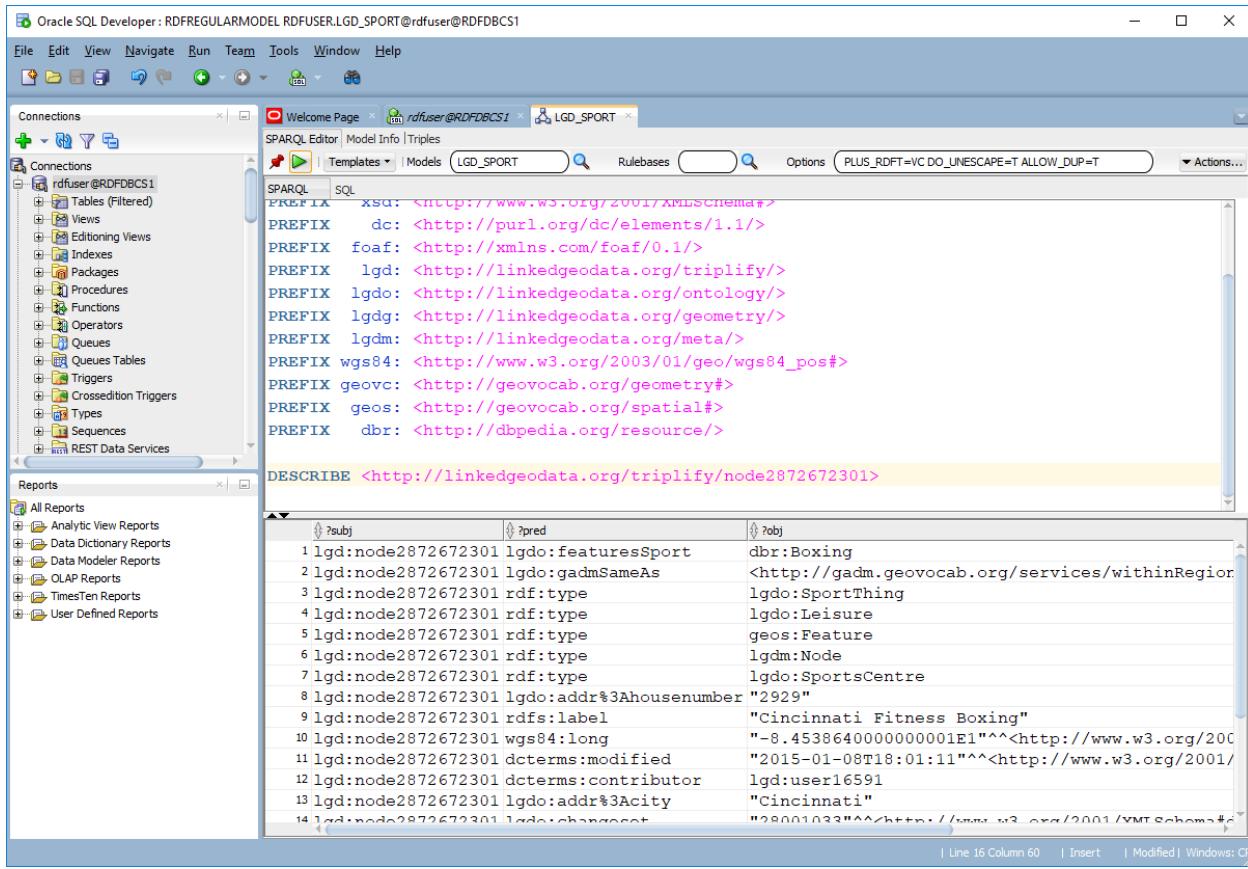
```

- Bottom Status Bar:** Line 9 Column 19 | Insert | Modified | Windows: C:\

For the next query, we have modified the distinct predicates query to get the top 10 properties and their triple counts.



In addition to SPARQL SELECT queries, CONSTRUCT and DESCRIBE queries are also supported. The query below describes a particular resource in the LGD_SPORT model. Note that any namespace PREFIXes used in the query will also be used to simplify values in the query result. Here we have added several more PREFIXes.



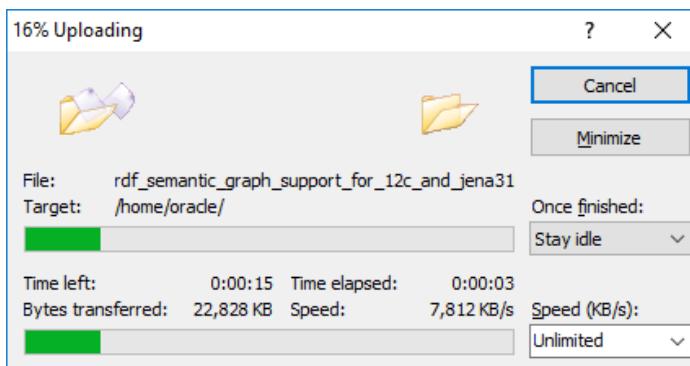
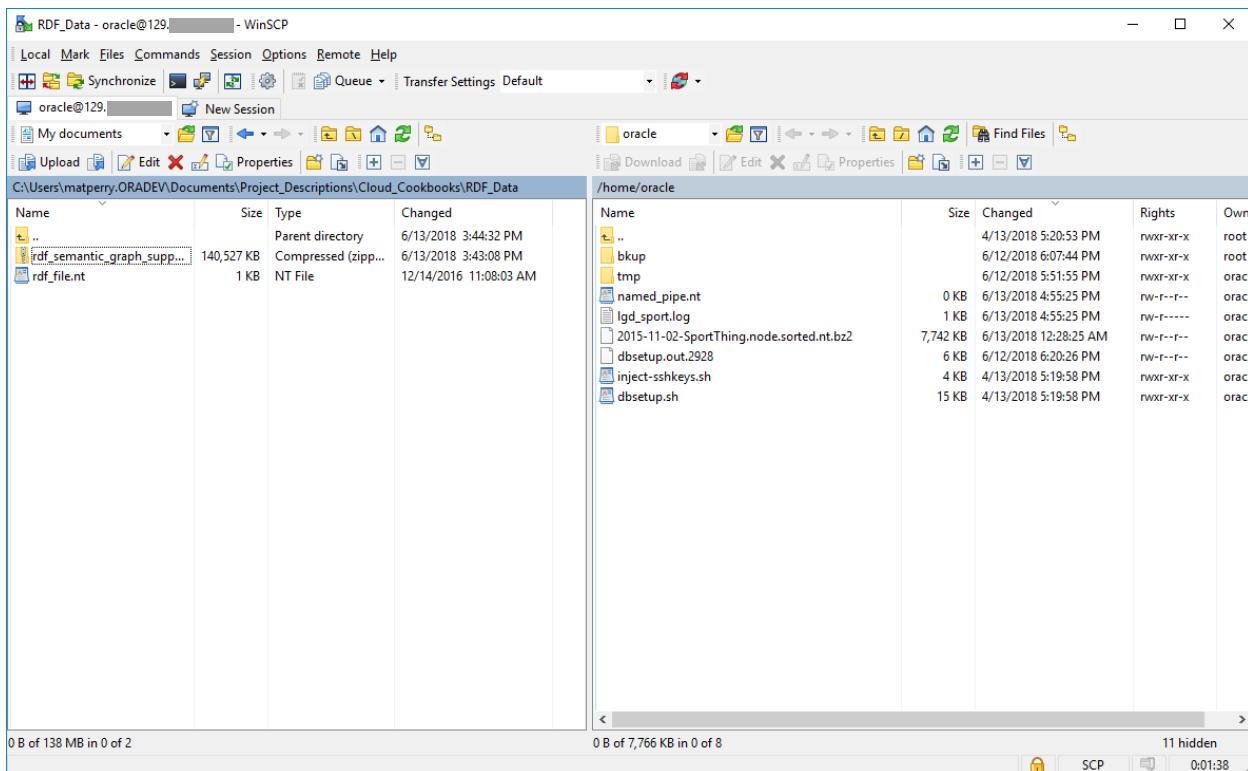
6 SETTING UP A FUSEKI ENDPOINT

The W3C defines several standard REST APIs for querying and updating RDF data. Oracle RDF Semantic graph leverages Apache Jena Fuseki to provide an implementation of those interfaces. Oracle Support for Apache Jena provides a tight integration between Apache Jena and Oracle RDF Semantic Graph through Oracle-specific implementations of Apache Jena interfaces.

In this Section, we will show how to setup and run Apache Jena Fuseki on our DBCS instance. Fuseki can run as a standalone server or a Java web application. In this case, we will run Fuseki as a standalone server on our DBCS instance. You could also setup an Oracle Java Cloud Service instance and deploy the Fuseki Java web application into the included Web Logic Server instance.

The first step is to download the latest Oracle Support for Apache Jena from OTN. Open a web browser to the Spatial and Graph downloads [page](#). Choose: Download Oracle Database 12c Release 12.1.0.2 Support for Apache Jena 3.1, Apache Jena Fuseki 2.4, and Protégé Desktop 5.0. Note that this download works with Oracle Database version 12.1.0.1 and later.

After the download completes, follow the instructions from earlier to connect WinSCP to your DBCS instance. Then copy the downloaded Oracle Support for Apache Jena file to your DBCS instance. In this example, we copied the file to /home/oracle.



Follow the instructions from earlier to open an SSH connection to your DBCS instance. Create a directory named Jena in /home/oracle. Copy `rdf_semantic_graph_support_for_12c_and_jena310_protege_5.0_2017_01_19.zip` to the newly created Jena directory and unzip the file.

```
oracle@RDFDBCS1:~/Jena
login as: oracle
Authenticating with public key "rsa-key-20180612"
Passphrase for key "rsa-key-20180612":
[oracle@RDFDBCS1 ~]$ ls
2015-11-02-SportThing.node.sorted.nt.bz2
bkup
dbsetup.out.2928
dbsetup.sh
inject-sshkeys.sh
lgd_sport.log
named_pipe.nt
rdf_semantic_graph_support_for_12c_and_jena310_protege_5.0_2017_01_19.zip
tmp
[oracle@RDFDBCS1 ~]$ mkdir Jena
[oracle@RDFDBCS1 ~]$ mv rdf_semantic_graph_support_for_12c_and_jena310_protege_5
.0_2017_01_19.zip Jena/
[oracle@RDFDBCS1 ~]$ cd Jena
[oracle@RDFDBCS1 Jena]$ ls
rdf_semantic_graph_support_for_12c_and_jena310_protege_5.0_2017_01_19.zip
[oracle@RDFDBCS1 Jena]$ unzip rdf_semantic_graph_support_for_12c_and_jena310_pro
tege_5.0_2017_01_19.zip
```

```
oracle@RDFDBCS1:~/Jena
inflating: sparqlgateway/Scripts/util.js
inflating: sparqlgateway/application.xml
inflating: sparqlgateway/qbl.sparql
creating: sparqlgateway/admin/
inflating: sparqlgateway/admin/sparql.jsp
inflating: sparqlgateway/admin/xslt.jsp
extracting: sparqlgateway/noop.xslt
creating: sparqlgateway/WEB-INF/
inflating: sparqlgateway/WEB-INF/web.xml
inflating: sparqlgateway/WEB-INF/weblogic.xml
creating: sparqlgateway/WEB-INF/lib/
creating: sparqlgateway/WEB-INF/classes/
inflating: sparqlgateway/default.xslt
inflating: sparqlgateway/index.html
creating: sparqlgateway/StyleSheets/
inflating: sparqlgateway/StyleSheets/sg.css
inflating: sparqlgateway/StyleSheets/sgmin.css
inflating: sparqlgateway/StyleSheets/navtable.css
inflating: sparqlgateway/StyleSheets/paginator.css
creating: sparqlgateway_web_app/
inflating: sparqlgateway_web_app/sparqlgateway.war
creating: web/
inflating: web/web.xml
[oracle@RDFDBCS1 Jena]$
```

After the unzip command has finished, you will see several directories and a README file.

```
oracle@RDFDBCS1:~/Jena
inflating: sparqlgateway/StyleSheets/sgmin.css
inflating: sparqlgateway/StyleSheets/navtable.css
inflating: sparqlgateway/StyleSheets/paginator.css
creating: sparqlgateway_web_app/
inflating: sparqlgateway_web_app/sparqlgateway.war
creating: web/
inflating: web/web.xml
[oracle@RDFDBCS1 Jena]$ ls
examples
fuseki
fuseki_web_app
jar
javadoc
joseki
joseki_web_app
META-INF
protege_plugin
rdf_semantic_graph_support_for_12c_and_jena310_protege_5.0_2017_01_19.zip
README
server
sparqlgateway
sparqlgateway_web_app
web
[oracle@RDFDBCS1 Jena]$
```

Now, we will configure Fuseki to access the LGD_SPORT semantic model that we created earlier. Change directory to /fuseki and edit the config-oracle.ttl file.

```
oracle@RDFDBCS1:~/Jena/fuseki
inflating: sparqlgateway_web_app/sparqlgateway.war
creating: web/
inflating: web/web.xml
[oracle@RDFDBCS1 Jena]$ ls
examples
fuseki
fuseki_web_app
jar
javadoc
joseki
joseki_web_app
META-INF
protege_plugin
rdf_semantic_graph_support_for_12c_and_jena310_protege_5.0_2017_01_19.zip
README
server
sparqlgateway
sparqlgateway_web_app
web
[oracle@RDFDBCS1 Jena]$ cd fuseki
[oracle@RDFDBCS1 fuseki]$ ls
bin          fuseki      fuseki-server.bat  LICENSE  webapp
config-oracle.ttl  fuseki-server  fuseki-server.jar  NOTICE
[oracle@RDFDBCS1 fuseki]$
```

Change the following default <#oracle> dataset specification from

```
<#oracle> rdf:type oracle:Dataset;
          oracle:connection
          [ a oracle:OracleConnection ;
            oracle:jdbcURL
"jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP) (HOST=localhost) (PORT=
1521)) (CONNECT_DATA=(SERVER=DEDICATED) (SERVICE_NAME=orcl)))";
```

```

        oracle:User "rdfuser" ;
        oracle:Password "rdfuser"
];
oracle:allGraphs [ oracle:firstModel "TEST_MODEL" ] .

to

<#oracle> rdf:type oracle:Dataset;
    oracle:connection
    [ a oracle:OracleConnection ;
    oracle:jdbcURL
"jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP) (HOST=localhost) (PORT=
1521)) (CONNECT_DATA=(SERVER=DEDICATED) (SERVICE_NAME=PDB1.uspm020.oraclecloud.
internal))";
    oracle:User "rdfuser" ;
    oracle:Password "rdfuser"
];
oracle:allGraphs [ oracle:firstModel "LGD_SPORT" ] .

```

```

oracle@RDFDBCS1:~/Jena/fuseki - □ ×
fuseki:services (
  <#service1>
) .

# Custom code.
[] ja:loadClass "oracle.spatial.rdf.client.jena.assembler.OracleAssemblerVocab"
.
oracle:Dataset rdfs:subClassOf ja:RDFDataset .

<#oracle> rdf:type oracle:Dataset;
    oracle:connection
    [ a oracle:OracleConnection ;
    oracle:jdbcURL "jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP) (HOST=localhost) (PORT=1521)) (CONNECT_DATA=(SERVER=DEDICATED) (SERVICE_NAME=PDB1.uspm020.oraclecloud.internal))";
    oracle:User "rdfuser" ;
    oracle:Password "rdfuser"
];
oracle:allGraphs [ oracle:firstModel "LGD_SPORT" ] .

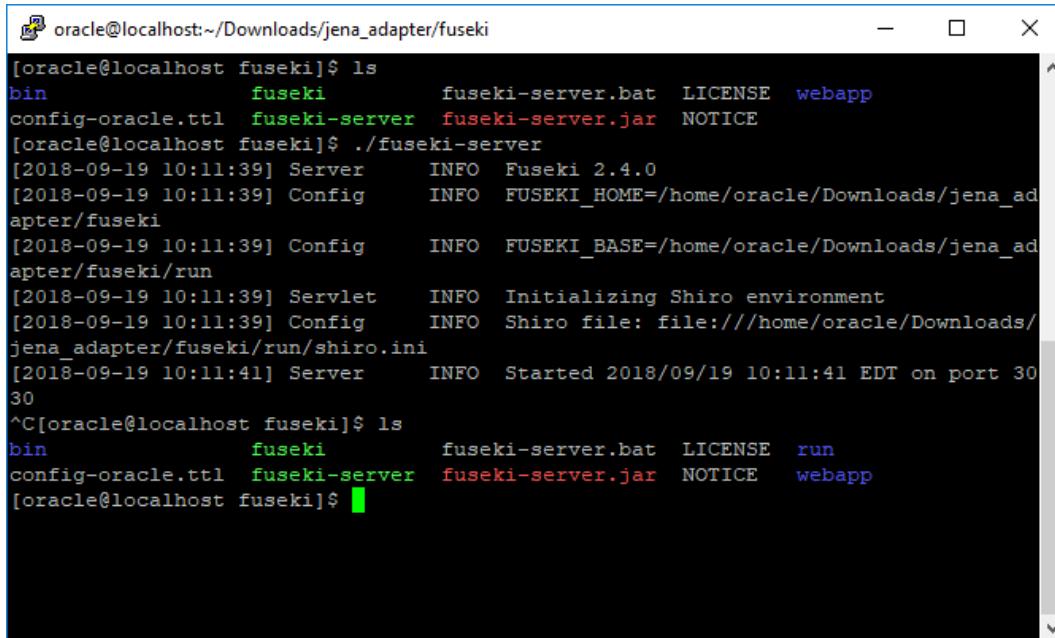
## # Custom code.
## [] ja:loadClass "com.hp.hpl.jena.tdb.TDB" .
-- INSERT --
39,53      37% ↓

```

Next, we will change the default shiro.ini configuration to allow non-localhost connections. First, we need to startup Fuseki to create a /run directory. Simply execute the following command in the current /fuseki directory.

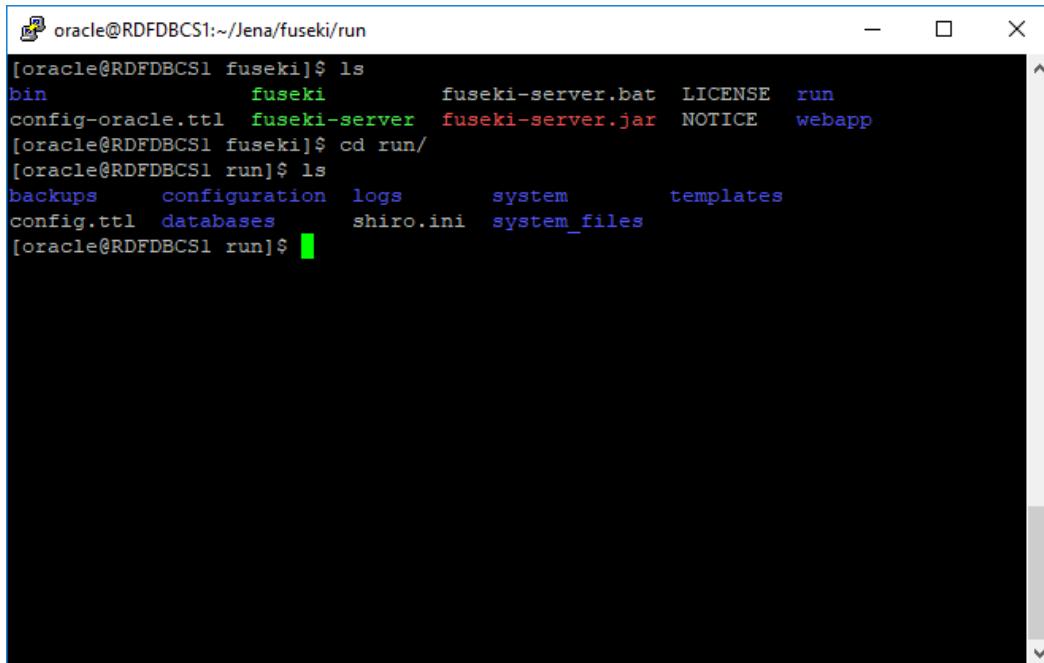
```
./fuseki-server
```

Once you see the message that Fuseki has started on port 3030, kill the process with Ctrl-C.



```
[oracle@localhost fuseki]$ ls
bin          fuseki      fuseki-server.bat  LICENSE  webapp
config-oracle.ttl  fuseki-server  fuseki-server.jar  NOTICE
[oracle@localhost fuseki]$ ./fuseki-server
[2018-09-19 10:11:39] Server      INFO  Fuseki 2.4.0
[2018-09-19 10:11:39] Config      INFO  FUSEKI_HOME=/home/oracle/Downloads/jena_adapter/fuseki
[2018-09-19 10:11:39] Config      INFO  FUSEKI_BASE=/home/oracle/Downloads/jena_adapter/fuseki/run
[2018-09-19 10:11:39] Servlet     INFO  Initializing Shiro environment
[2018-09-19 10:11:39] Config      INFO  Shiro file: file:///home/oracle/Downloads/jena_adapter/fuseki/run/shiro.ini
[2018-09-19 10:11:41] Server      INFO  Started 2018/09/19 10:11:41 EDT on port 3030
^C[oracle@localhost fuseki]$ ls
bin          fuseki      fuseki-server.bat  LICENSE  run
config-oracle.ttl  fuseki-server  fuseki-server.jar  NOTICE  webapp
[oracle@localhost fuseki]$
```

Now the /run directory should be created. Change directory to /run and edit shiro.ini.



```
[oracle@RDFDBCS1:~/Jena/fuseki/run
[oracle@RDFDBCS1 fuseki]$ ls
bin          fuseki      fuseki-server.bat  LICENSE  run
config-oracle.ttl  fuseki-server  fuseki-server.jar  NOTICE  webapp
[oracle@RDFDBCS1 fuseki]$ cd run/
[oracle@RDFDBCS1 run]$ ls
backups      configuration  logs      system      templates
config.ttl  databases    shiro.ini  system_files
[oracle@RDFDBCS1 run]$
```

Replace

```
/$/** = localhostFilter
with
/$/server = anon
$/** = localhostFilter
```

```
oracle@RDFDBCS1:~/Jena/fuseki/run

[users]
# Implicitly adds "iniRealm = org.apache.shiro.realm.text.IniRealm"
admin=pw

[roles]

[url]
## Control functions open to anyone
/$/status = anon
/$/ping = anon

## and the rest are restricted to localhost.
/$/** = localhostFilter

## If you want simple, basic authentication user/password
## on the operations,
##   1 - set a better password in [users] above.
##   2 - comment out the "/$/** = localhost" line and use:
##   "/$/** = authcBasic,user[admin]"

## or to allow any access.
##$/** = anon
24,1          78% ▼
```

```
oracle@RDFDBCS1:~/Jena/fuseki/run

[users]
# Implicitly adds "iniRealm = org.apache.shiro.realm.text.IniRealm"
admin=pw

[roles]

[url]
## Control functions open to anyone
/$/status = anon
/$/ping = anon

## and the rest are restricted to localhost.
/$/** = localhostFilter

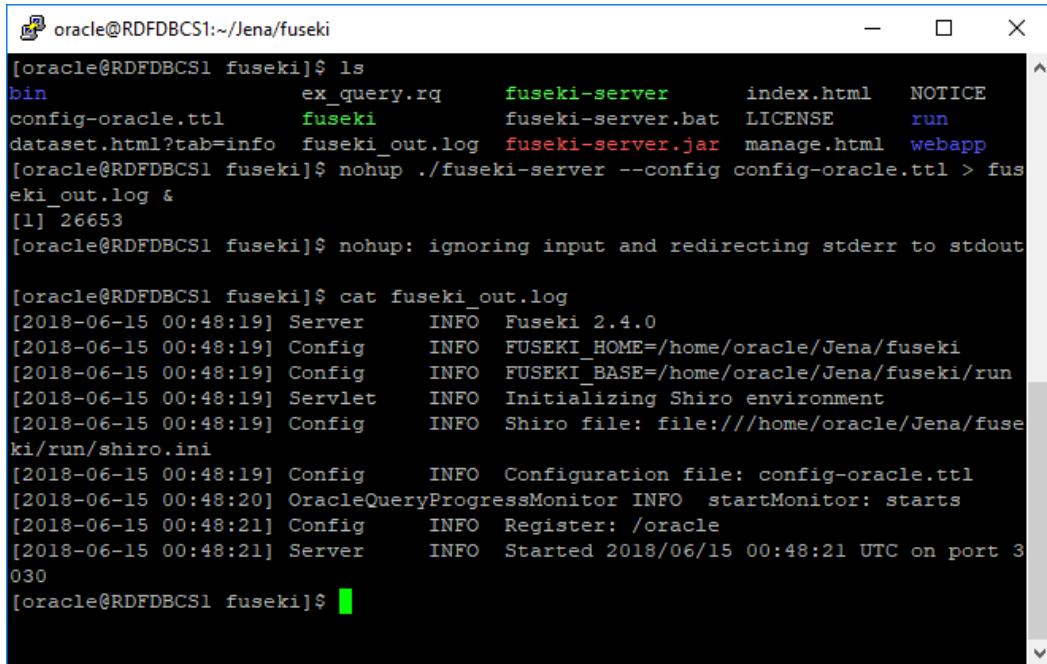
/$/server = anon
$/** = localhostFilter
## If you want simple, basic authentication user/password
## on the operations,
##   1 - set a better password in [users] above.
##   2 - comment out the "/$/** = localhost" line and use:
##   "/$/** = authcBasic,user[admin]"

## or to allow any access.
29,1          75% ▼
```

Change directory back to /fuseki and start the Fuseki service by running the following command:

```
nohup ./fuseki-server --config config-oracle.ttl > fuseki_out.log &
```

Note that we are using nohup to prevent the Fuseki process from terminating if our connection is closed.



```
[oracle@RDFDBCS1:~/Jena/fuseki]
[oracle@RDFDBCS1 fuseki]$ ls
bin          ex_query.rq    fuseki-server      index.html  NOTICE
config-oracle.ttl  fuseki    fuseki-server.bat  LICENSE    run
dataset.html?tab=info  fuseki_out.log  fuseki-server.jar  manage.html webapp
[oracle@RDFDBCS1 fuseki]$ nohup ./fuseki-server --config config-oracle.ttl > fuseki_out.log &
[1] 26653
[oracle@RDFDBCS1 fuseki]$ nohup: ignoring input and redirecting stderr to stdout

[oracle@RDFDBCS1 fuseki]$ cat fuseki_out.log
[2018-06-15 00:48:19] Server      INFO  Fuseki 2.4.0
[2018-06-15 00:48:19] Config      INFO  FUSEKI_HOME=/home/oracle/Jena/fuseki
[2018-06-15 00:48:19] Config      INFO  FUSEKI_BASE=/home/oracle/Jena/fuseki/run
[2018-06-15 00:48:19] Servlet     INFO  Initializing Shiro environment
[2018-06-15 00:48:19] Config      INFO  Shiro file: file:///home/oracle/Jena/fuseki/run/shiro.ini
[2018-06-15 00:48:19] Config      INFO  Configuration file: config-oracle.ttl
[2018-06-15 00:48:20] OracleQueryProgressMonitor INFO  startMonitor: starts
[2018-06-15 00:48:21] Config      INFO  Register: /oracle
[2018-06-15 00:48:21] Server     INFO  Started 2018/06/15 00:48:21 UTC on port 3030
[oracle@RDFDBCS1 fuseki]$
```

Now the Fuseki server is up and running on port 3030 of our DBCS instance. There are two options for connecting:

1. Create an access rule to open up port 3030 of our DBCS instance to the public internet.
2. Create an SSH tunnel to our DBCS instance for port 3030

6.1 METHOD 1: CREATE AN ACCESS RULE

This Section describes how to create an access rule for port 3030 to open up our SPARQL endpoint to the public internet.

First, login to cloud.oracle.com and open the details page for your DBCS instance.

Oracle Database Cloud Service Details - Mozilla Firefox

Oracle Database Cloud Ser... × +

https://dbaas.oraclecloud.com/dbaas/faces/dbRunner.jspx

Search

ORACLE CLOUD My Services

RDFDBCS1

Oracle Database Cloud Service / RDFDBCS1

Overview

1 Node

Instance Overview

1 Nodes 2 OCpus 15 GB Memory 150 GB Storage

Status: Ready Version: 18.0.0.0

Connect String: RDfdbcs1:1521/PDB1/uspm02... Edition: Enterprise Edition - High Performance

Backup Destination: Cloud Storage Only Cloud Storage Container: https://uspm020.storage.o...

PDB Name: PDB1 Container Name: ORCL

Character Set: AL32UTF8 - Unicode Univers... National Character Set: AL16UTF16 - Unicode UTF-1...

SQL *Net Port: 1521 Timezone: Coordinated Universal Time

Administration

1 Patches available

0 Snapshots available

Show less...

Resources

Host Name: RDfdbcs1 OCpus: 2

Public IP: 129.0.0.1 Memory: 15 GB

SID: ORCL Storage: 150 GB

About Oracle | Contact Us | Legal Notices | Terms of Use | Your Privacy Rights
Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Click the menu on the right and select Access Rules.

The screenshot shows the Oracle Database Cloud Service Details page in Mozilla Firefox. The URL is https://dbaas.oraclecloud.com/dbaas/faces/dbRunner.jsp?_adf.ctrl-state=undefined. The main content area displays the Instance Overview for database instance RDFDBCS1. Key details shown include:

- Nodes:** 1 Node
- OCPUs:** 2 OCPUs
- Memory:** 15 GB
- Storage:** 150 GB

The status is Ready, and the edition is Enterprise Edition - High Performance. Other configuration details listed include:

- Connect String: RDfdbcs1:1521/PDB1.uspm02...
- Backup Destination: Cloud Storage Only
- PDB Name: PDB1
- Character Set: AL32UTF8 - Unicode Universal
- SQL *Net Port: 1521
- National Character Set: AL16UTF16 - Unicode UTF-1...
- Container Name: ORCL
- Timezone: Coordinated Universal Time

The sidebar on the left shows the following sections:

- Overview:** 1 Node
- Administration:** 1 Patches available, 0 Snapshots available

A context menu is open on the right side of the screen, listing options such as Open DBaaS Monitor Console, Open Application Express Console, Open EM Console, Start, Stop, Restart, Scale Up/Down, Access Rules, SSH Access, Add Tags, Replace Database using Backup, and View Activity.

At the bottom of the page, there are links for About Oracle, Contact Us, Legal Notices, Terms of Use, Your Privacy Rights, and a note about reserved IP addresses. The URL in the address bar is https://dbaas.oraclecloud.com/dbaas/faces/dbRunner.jsp?_adf.ctrl-state=undefined.

This will bring up the Access Rules page. Click Create Rule to add an access rule for port 3030.

Oracle Database Cloud Service Details - Mozilla Firefox

Oracle Database Cloud Service... +

https://dbaas.oracledb.com/dbaas/faces/dbRunner.jspx?_adf.ctrl-state=undefined

cloud.oracle.com

ORACLE CLOUD My Services

RDFDBCS1

Oracle Database Cloud Service / RDFDBCS1

Access Rules

You can use access rules to control network access to service components. On this page, you can manage your access rules.

Create Rule

Results per page: 10

8 result(s) as of Jun 14, 2018 1:28:51 PM UTC

Status	Rule Name	Source	Destination	Ports	Protocol	Description	Rule Type	Actions
Green	ora_p2_ssh	PUBLIC-INTERNET	DB_1	22	TCP		DEFAULT	
Green	ora_p2_http	PUBLIC-INTERNET	DB_1	80	TCP		DEFAULT	
Green	ora_p2_https	PUBLIC-INTERNET	DB_1	443	TCP		DEFAULT	
Green	ora_p2_dbconsole	PUBLIC-INTERNET	DB_1	1158	TCP		DEFAULT	
Green	ora_p2_dbexpress	PUBLIC-INTERNET	DB_1	5500	TCP		DEFAULT	
Green	ora_p2_dblistener	PUBLIC-INTERNET	DB_1	1521	TCP		DEFAULT	
Green	sys_infra2db_ssh	PAAS-INFRA	DB_1	22	TCP	DO NOT MODIFY: Permit P...	SYSTEM	
Green	ora_trusted_hosts_db...	127.0.0.1/32	DB_1	1521	TCP	DO NOT MODIFY: A securle ...	SYSTEM	

About Oracle | Contact Us | Legal Notices | Terms of Use | Your Privacy Rights
https://dbaas.oracledb.com/dbaas/faces/dbRunner.jspx?_adf.ctrl-state=undefined#

Enter a Rule Name and Description. Choose PUBLIC-INTERNET for Source and DB_1 for Destination. Enter 3030 for Destination Port(s) and choose TCP for Protocol.

Oracle Database Cloud Service Details - Mozilla Firefox

Oracle Database Cloud Service... + https://dbaas.oracledb.com/dbaas/faces/dbRunner.jsp?_adf.ctrl-state=undefined cloud.oracle.com

ORACLE CLOUD My Services

RDFBCS1

Oracle Database Cloud Service / RDFBCS1

Access Rules

Create Access Rule

Rule Name: Fuseki
Description: Port for Fuseki SPARQL endpoint

Source: PUBLIC-INTERNET
Destination: DB_1
Destination Port(s): 3030
Protocol: TCP

This operation may take some time.

Create Cancel

Status Rule Name

Status	Rule Name	Source	Destination	Port	Protocol	Action
Active	ora_p2_					
Active	ora_p2_					
Active	ora_p2_					
Active	ora_p2_					
Active	ora_p2_					
Active	ora_p2_					
Active	ora_p2_					
Active	sys_infra2db_ssh	PAAS-INFRA	DB_1	22	TCP	DO NOT MODIFY: Permit P... SYSTEM
Active	ora_trusted_hosts_db...	127.0.0.1/32	DB_1	1521	TCP	DO NOT MODIFY: A securle ... SYSTEM

About Oracle | Contact Us | Legal Notices | Terms of Use | Your Privacy Rights
Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

The screenshot shows the Oracle Database Cloud Service interface for creating an access rule. The 'Create Access Rule' dialog is open, prompting for a Rule Name ('Fuseki'), Source ('PUBLIC-INTERNET'), Destination ('DB_1'), Destination Port(s) ('3030'), and Protocol ('TCP'). A note indicates the operation may take some time. Below the dialog, a table lists existing access rules, including ones for sys_infra2db_ssh and ora_trusted_hosts_db. At the bottom, a progress bar shows the task is in progress.

After some time it will complete.

Oracle Database Cloud Service Details - Mozilla Firefox

Oracle Database Cloud Service... +

https://dbaas.oracledb.com/dbaas/faces/dbRunner.jsp?_adf.ctrl-state=undefined cloud.oracle.com

ORACLE CLOUD My Services

RDFDBCS1

Oracle Database Cloud Service / RDFDBCS1

Access Rules

You can use access rules to control network access to service components. On this page, you can manage your access rules.

Create Rule

Results per page: 10 9 result(s) as of Jun 14, 2018 1:32:51 PM UTC

Status	Rule Name	Source	Destination	Ports	Protocol	Description	Rule Type	Actions
Green	ora_p2_ssh	PUBLIC-INTERNET	DB_1	22	TCP		DEFAULT	
Green	ora_p2_http	PUBLIC-INTERNET	DB_1	80	TCP		DEFAULT	
Green	ora_p2_httpsl	PUBLIC-INTERNET	DB_1	443	TCP		DEFAULT	
Green	ora_p2_dbconsole	PUBLIC-INTERNET	DB_1	1158	TCP		DEFAULT	
Green	ora_p2_dbexpress	PUBLIC-INTERNET	DB_1	5500	TCP		DEFAULT	
Green	ora_p2_dblistener	PUBLIC-INTERNET	DB_1	1521	TCP		DEFAULT	
Green	sys_infra2db_ssh	PAAS-INFRA	DB_1	22	TCP	DO NOT MODIFY: Permit P...	SYSTEM	
Green	ora_trusted_hosts_db...	127.0.0.1/32	DB_1	1521	TCP	DO NOT MODIFY: A securle ...	SYSTEM	
Green	Fuseki	PUBLIC-INTERNET	DB_1	3030	TCP	Port for Fuseki SPARQL en...	USER	

About Oracle | Contact Us | Legal Notices | Terms of Use | Your Privacy Rights
https://dbaas.oracledb.com/dbaas/faces/dbRunner.jsp?_adf.ctrl-state=undefined#

After rule creation has finished, open a web browser to `http://<your_dbcs_ip>:3030`

The screenshot shows the Apache Jena Fuseki interface running on port 3030. The browser title bar reads "Apache Jena Fuseki - Mozilla Firefox". The address bar shows "129.0.0.1:3030". The page header includes the Apache Jena Fuseki logo, navigation links for "dataset", "manage datasets", and "help", and a "Server status" indicator showing a green circle. The main content area displays the "Apache Jena Fuseki" title, version information ("Version 2.4.0. Uptime: 13m 57s"), and a section titled "Datasets on this server". A table lists one dataset, "/oracle", with actions: "query", "add data", and "info". Below this is a help section with links: "Dataset" (Run queries and modify datasets), "Manage datasets" (Administer datasets), and "Help" (Summary of commands and links to online documentation).

Click query to open the SPARQL query interface.

Apache Jena Fuseki - inspect dataset - Mozilla Firefox

Oracle Database Cloud Service... Apache Jena Fuseki - inspect d... +

129.1.1.1:3030/dataset.html?tab=query&ds=/oracle

Apache Jena Fuseki dataset manage datasets help Server status: green

Dataset: /oracle

query upload files edit info

SPARQL query

To try out some SPARQL queries against the selected dataset, enter your query here.

EXAMPLE QUERIES Selection of triples Selection of classes

PREFIXES rdf rdfs owl xsd

SPARQL ENDPOINT: http://129.1.1.1:3030/oracle/query CONTENT TYPE (SELECT): JSON CONTENT TYPE (GRAPH): Turtle

```
1 prefix owl: <http://www.w3.org/2002/07/owl#>
2 prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
3
4 SELECT DISTINCT ?class ?label ?description
5 WHERE {
6   ?class a owl:Class.
7   OPTIONAL { ?class rdfs:label ?label }
8   OPTIONAL { ?class rdfs:comment ?description }
9 }
10 LIMIT 25
```

◀ ▶

QUERY RESULTS

Click info to see all the available REST endpoints.

The screenshot shows the Apache Jena Fuseki interface for inspecting a dataset named '/oracle'. The top navigation bar includes tabs for 'dataset' and 'manage datasets'. A 'Server status' indicator shows a green circle. The main content area displays the following sections:

- Available services:**

File Upload:	http://129.132.1.129:3030/oracle/upload
Graph Store Protocol:	http://129.132.1.129:3030/oracle/data
Graph Store Protocol (Read):	http://129.132.1.129:3030/oracle/get
HTTP Quads:	http://129.132.1.129:3030/oracle/
SPARQL Query:	http://129.132.1.129:3030/oracle/query
SPARQL Update:	http://129.132.1.129:3030/oracle/sparql
SPARQL Update:	http://129.132.1.129:3030/oracle/update
- Statistics:**

Name	Overall good	Overall bad	Graph Store Protocol (Read)	Graph Store Protocol	SPARQL Update	File Upload	SPARQL Query	SPARQL Query	HTTP Quads
/oracle	3	3	0	0	0	0	2 (0 bad)	1 (0 bad)	0
- Dataset size:**

Note: this may be slow and impose a significant load on large datasets: [count triples in all graphs](#)
- Ongoing operations:**

TBD. Will list any long-lasting operations that are ongoing or recently completed, e.g. backups.

We can also use Cygwin to test the REST SPARQL interface directly with curl. When run from a Cygwin terminal, the following command

```
curl -X POST -data-binary "@test_query.rq" -H "Content-Type: application/sparql-query" -H "Accept: application/sparql-results+json" "http://<your_dbcs_ip>:3030/sparql"
```

will send the SPARQL query in the file test_query.rq to the Fuseki endpoint running on your DBCS instance and print the result to stdout.

```

/cygdrive/c/Documents and Settings/matperry.ORADEV/Documents/Project_Descriptions/Cloud_Cookbooks/RDF_18_1_DBCS
$ cat test_query.rq
SELECT *
WHERE { ?s ?p ?o }
LIMIT 3
matperry@MATPERRY-LAP1 /cygdrive/c/Documents and Settings/matperry.ORADEV/Documents/Project_Descriptions/Cloud_Cookbooks/RDF_18_1_DBCS
$ export http_proxy=http://www-proxy[REDACTED].com:80/

matperry@MATPERRY-LAP1 /cygdrive/c/Documents and Settings/matperry.ORADEV/Documents/Project_Descriptions/Cloud_Cookbooks/RDF_18_1_DBCS
$ curl -X POST --data-binary "@test_query.rq" -H "Content-Type: application/sparql-query" -H "Accept: application/sparql-results+json" "http://129.[REDACTED]:3030/oracle/sparql"
{
  "head": {
    "vars": [ "s" , "p" , "o" ]
  } ,
  "results": {
    "bindings": [
      {
        "s": { "type": "uri" , "value": "http://linkedgeodata.org/triplify/node301033430" } ,
        "p": { "type": "uri" , "value": "http://linkedgeodata.org/ontology/gadmSameAs" } ,
        "o": { "type": "uri" , "value": "http://gadm.geovocab.org/services/withinRegion?lat=45.548901&long=-73.657936#point" }
      } ,
      {
        "s": { "type": "uri" , "value": "http://linkedgeodata.org/triplify/node301033430" } ,
        "p": { "type": "uri" , "value": "http://linkedgeodata.org/ontology/type" } ,
        "o": { "type": "literal" , "value": "petanque" }
      } ,
      {
        "s": { "type": "uri" , "value": "http://linkedgeodata.org/triplify/node301033430" } ,
        "p": { "type": "uri" , "value": "http://linkedgeodata.org/ontology/version" } ,
        "o": { "type": "literal" , "datatype": "http://www.w3.org/2001/XMLSchema#decimal" , "value": "3" }
      }
    ]
  }
}

matperry@MATPERRY-LAP1 /cygdrive/c/Documents and Settings/matperry.ORADEV/Documents/Project_Descriptions/Cloud_Cookbooks/RDF_18_1_DBCS
$
```

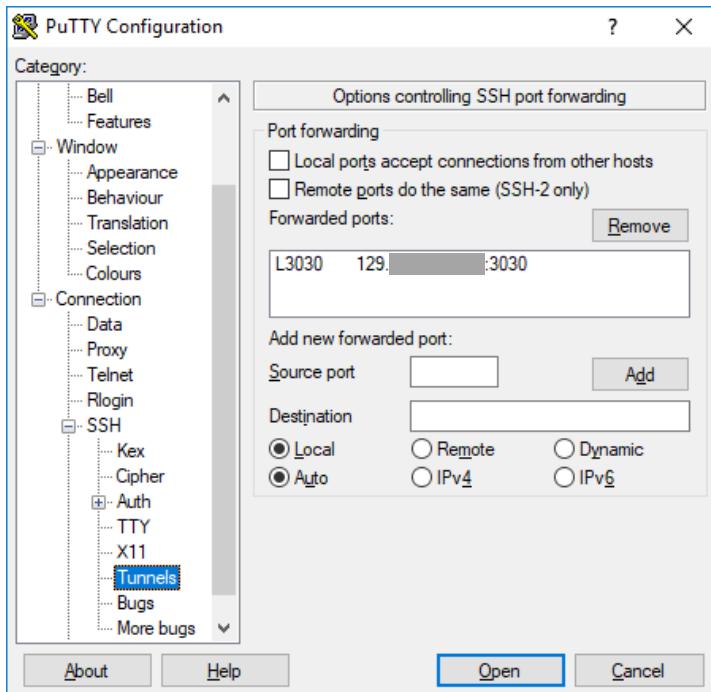
Now we have a W3C standard SPARQL endpoint up and running on our DBCS instance.

6.2 METHOD 2: USE PUTTY TO CREATE AN SSH TUNNEL TO OUR DBCS INSTANCE FOR PORT 3030

In this Section, we will show how to connect to our SPARQL endpoint on port 3030 without opening port 3030 to the public internet. This setup does not require the port 3030 access rule that we created previously. That access rule can be disabled or dropped.

First, use PuTTY to open an SSH connection to your DBCS instance and start fuseki-server on the DBCS instance the same way we did before.

Next, use PuTTY to create an SSH tunnel to forward port 3030 on your local machine to port 3030 on your DBCS instance. The steps are the same as the ones we used earlier to forward port 1521 for SQL Developer, but we are using port 3030 in this case.



Click Open to open the SSH tunnel and then open a web browser to <http://localhost:3030>.

The screenshot shows the Apache Jena Fuseki web interface running on Mozilla Firefox. The URL in the address bar is `localhost:3030`. The page title is "Apache Jena Fuseki". The header includes links for "dataset", "manage datasets", and "help", along with a "Server status" indicator which is green. Below the header, the title "Apache Jena Fuseki" and version information "Version 2.4.0. Uptime: 6m 39s" are displayed. A section titled "Datasets on this server" lists a single dataset named "/oracle". For this dataset, there are three action buttons: "query" (blue), "add data" (white with blue outline), and "info" (white with blue outline). Below this section, a note says "Use the following pages to perform actions or tasks on this server." It provides links to "Dataset" (Run queries and modify datasets hosted by this server), "Manage datasets" (Administer the datasets on this server, including adding datasets, uploading data and performing backups), and "Help" (Summary of commands and links to online documentation). The bottom of the page shows the URL `localhost:3030/dataset.html?tab=query&ds=/oracle`.

Now we have used an SSH tunnel to connect to the SPARQL endpoint running on our DBCS instance.

We can also use Cygwin to directly test the SPARQL REST interface over the SSH tunnel. Enter the same curl command as before but use `http://localhost:3030/sparql` as the endpoint URL this time:

```
curl -X POST -data-binary "@test_query.rq" -H "Content-Type: application/sparql-query" -H "Accept: application/sparql-results+json" "http://localhost:3030/sparql"
```

```
matperry@MATTERRY-LAP1 /cygdrive/c/Documents and Settings/matperry.ORADEV/Documents/Project_Descriptions/Cloud_Cookbooks/RDF_18_1_DBCS
$ cat test_query.rq
SELECT *
WHERE { ?s ?p ?o }
LIMIT 3
matperry@MATTERRY-LAP1 /cygdrive/c/Documents and Settings/matperry.ORADEV/Documents/Project_Descriptions/Cloud_Cookbooks/RDF_18_1_DBCS
$ curl -X POST --data-binary "@test_query.rq" -H "Content-Type: application/sparql-query" -H "Accept: application/sparql-results+json" "http://localhost:3030/oracle/sparql"
{
  "head": {
    "vars": [ "s" , "p" , "o" ]
  } ,
  "results": {
    "bindings": [
      {
        "s": { "type": "uri" , "value": "http://linkedgeodata.org/triplify/node301033430" } ,
        "p": { "type": "uri" , "value": "http://linkedgeodata.org/ontology/gadmSameAs" } ,
        "o": { "type": "uri" , "value": "http://gadm.geovocab.org/services/withinRegion?lat=45.548901&long=-73.657936#point" }
      } ,
      {
        "s": { "type": "uri" , "value": "http://linkedgeodata.org/triplify/node301033430" } ,
        "p": { "type": "uri" , "value": "http://linkedgeodata.org/ontology/type" } ,
        "o": { "type": "literal" , "value": "petanque" }
      } ,
      {
        "s": { "type": "uri" , "value": "http://linkedgeodata.org/triplify/node301033430" } ,
        "p": { "type": "uri" , "value": "http://linkedgeodata.org/ontology/version" } ,
        "o": { "type": "literal" , "datatype": "http://www.w3.org/2001/XMLSchema#decimal" , "value": "3" }
      }
    ]
  }
}

matperry@MATTERRY-LAP1 /cygdrive/c/Documents and Settings/matperry.ORADEV/Documents/Project_Descriptions/Cloud_Cookbooks/RDF_18_1_DBCS
$
```

Now, we have successfully used an SSH tunnel to access a W3C-standard SPARQL REST endpoint running on our DBCS instance.