Extras/Working with Oracle Database

xtra Tutorial: Working with Oracle Database

READY

This tutorial was built for BDCS-CE version 17.3.3-20 as part of the New Data Lake User Journey: here (https://github.com/oracle/learning-library/tree/master/workshops/journey2-new-data-lake). Questions and feedback about the tutorial: david.bayard@oracle.com (mailto:david.bayard@oracle.com)

Contents

Identifying your Oracle Database Cloud Service Connection Details

READY

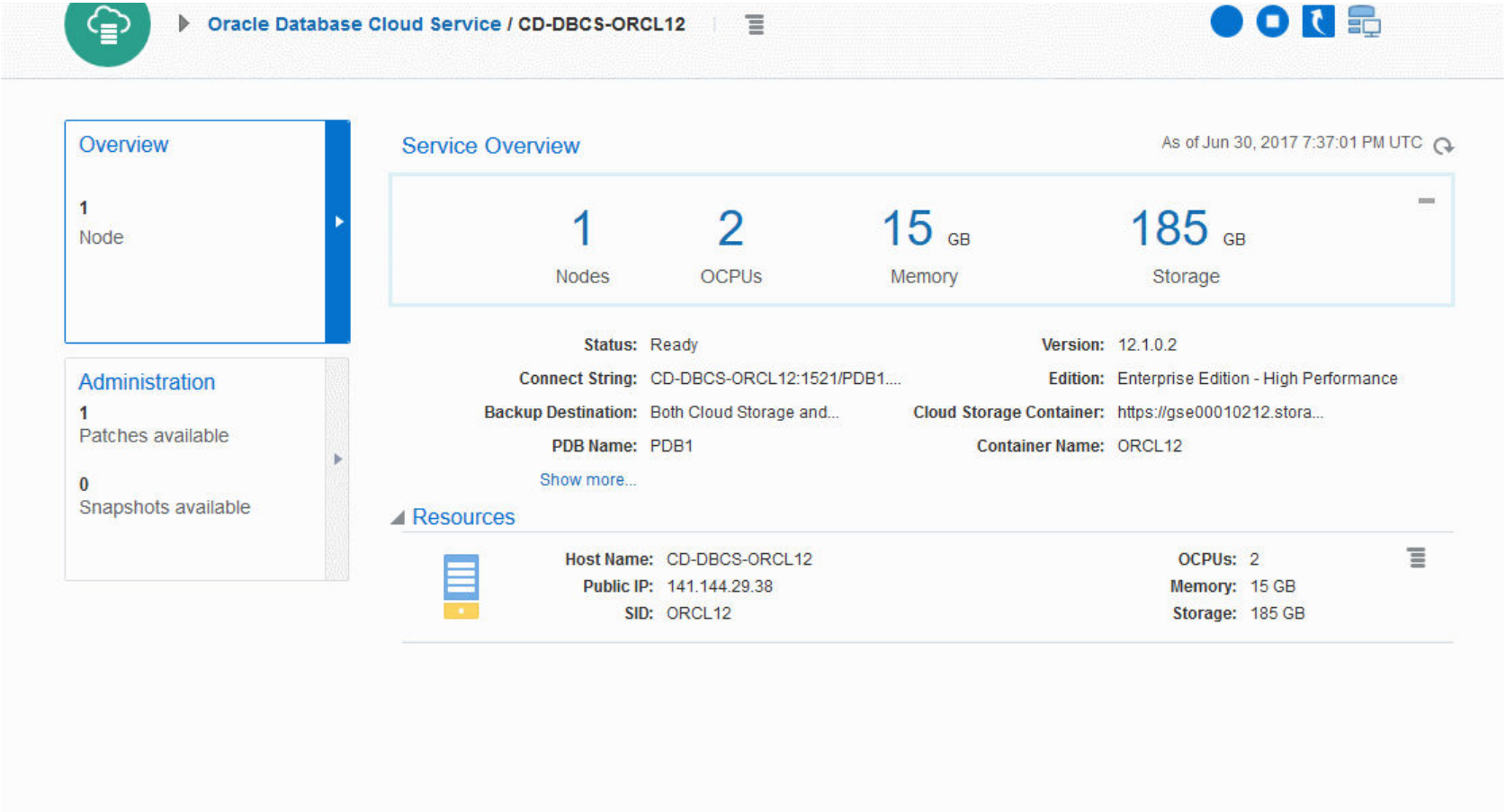
This tutorial will assume you are using an Oracle Database running in the Oracle Database Cloud Service. To connect from BDCS-CE, we need to:

- Identify the database connect string (which embeds the database hostname and service name)
- Ensure that an Access Rule allows network traffic from BDCS-CE to the Database server

In general, a very useful resource will be the “Oracle Database Cloud - Database as a Service Quick Start” which can be found here (http://www.oracle.com/webfolder/technetwork/tutorials/obe/cloud/dbaas/obe_dbaas_QS/oracle_database_cloud_service_dbaas_quick_start.html). In particular, the topic “Finding the Connection Details for your Database Instance” provides the necessary details. For simplicity, we will repeat the steps here:

Follow these steps:

- Navigate to the Oracle Database Cloud Service page for your DBCS instance.
- Click on the Connect String to see the full value, highlight the full connect string, copy it into the clipboard, and save it somewhere for later use.



Shell command to identify the private IP address of your BDCS-CE instance (used by next paragraph)

FINISHED

```
%sh
ifconfig eth0
```

Took 0 sec. Last updated by anonymous at November 21 2017, 2:37:03 PM. (outdated)

Setting up an Access Rule for DBCS to allow BDCS-CE to connect

READY

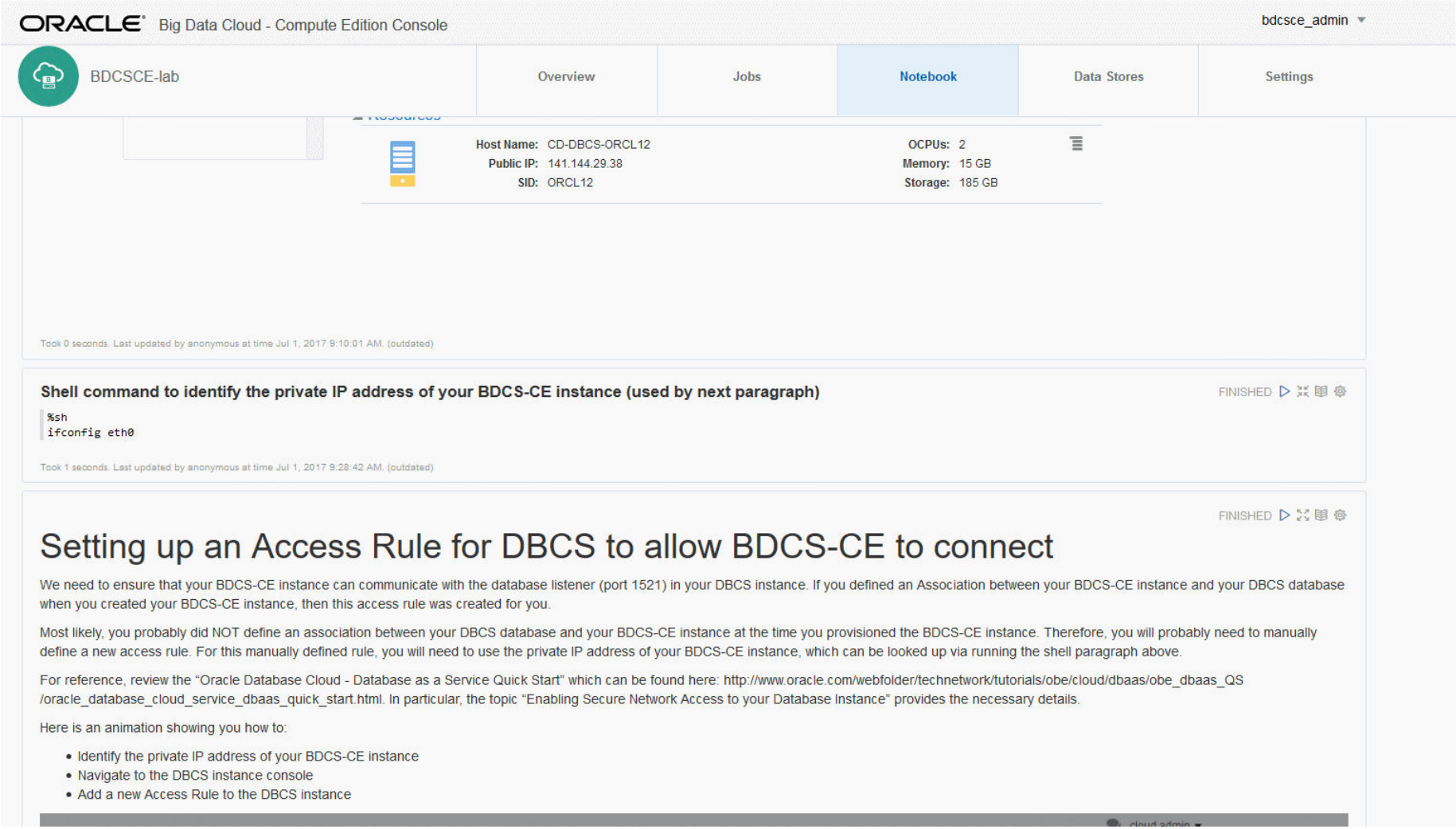
We need to ensure that your BDCS-CE instance can communicate with the database listener (port 1521) in your DBCS instance. If you defined an Association between your BDCS-CE instance and your DBCS database when you created your BDCS-CE instance, then this access rule was created for you.

Most likely, you probably did NOT define an association between your DBCS database and your BDCS-CE instance at the time you provisioned the BDCS-CE instance. Therefore, you will probably need to manually define a new access rule. For this manually defined rule, you will need to use the private IP address of your BDCS-CE instance, which can be looked up via running the shell paragraph above.

For reference, review the “Oracle Database Cloud - Database as a Service Quick Start” which can be found here (http://www.oracle.com/webfolder/technetwork/tutorials/obe/cloud/dbaas/obe_dbaas_QS/oracle_database_cloud_service_dbaas_quick_start.html). In particular, the topic “Enabling Secure Network Access to your Database Instance” provides the necessary details.

Here is an animation showing you how to:

- Identify the private IP address of your BDCS-CE instance
- Navigate to the DBCS instance console
- Add a new Access Rule to the DBCS instance



Script to patch the Zeppelin JDBC interpreter jar file (ZEPPELIN-1962)

FINISHED

```
%sh
```

Took 1 sec. Last updated by anonymous at November 21 2017, 3:04:35 PM.

Script to setup the Zeppelin JDBC settings for Oracle database connection (edit user, password, connectstring in script)

FINISHED

```
%sh

echo "Set the User, Password, and ConnectString parameters in the script as appropriate"
User=system
Password=Welcome1#
ConnectString="DCB-db-nov17:1521/PDB1.gse00002281.oraclecloud.internal"
#echo $ConnectString

echo "Script to set some zeppelin jdbc parameters."

cat <<EOF > /tmp/jdbc_settings.py
#!/usr/local/bin/python
#based on https://community.hortonworks.com/articles/36031/sample-code-to-automate-interacting-with-zeppelin.html by Ali Bajwa
def post_request(url, body):
  import json, urllib2
  encoded_body = json.dumps(body)
  req = urllib2.Request(str(url), encoded_body)
  req.get_method = lambda: 'PUT'
  try:
    response = urllib2.urlopen(req, encoded_body).read()
  except urllib2.HTTPError, error:
    print 'Exception: ' + error.read()
  jsonresp = json.loads(response.decode('utf-8'))
  print jsonresp['status']


import json, urllib2
zeppelin_int_url = 'http://127.0.0.1:9995/api/interpreter/setting/'
data = json.load(urllib2.urlopen(zeppelin_int_url))
for body in data['body']:
  if body['group'] == 'jdbc':
    jdbcbody = body
  elif body['group'] == 'spark':
    sparkbody = body
```

Took 4 sec. Last updated by anonymous at November 21 2017, 3:10:45 PM. (outdated)

Example of JDBC (change system to your database username)

FINISHED

```
%jdbc(orcl-system)
select * from user_tablespaces
```

TABLESPACE_NAME	BLOCK_SIZE	INITIAL_EXTENT	NEXT_EXTENT	MIN_EXTENTS	MAX_EXTENTS	MAX_SIZE	PCT_INCREASE	MIN_EXTLEN	STATUS	CO
SYSTEM	8192	65536	null	1	2147483645	2147483645	null	65536	ONLINE	PE
SYSAUX	8192	65536	null	1	2147483645	2147483645	null	65536	ONLINE	PE
TEMP	8192	1048576	1048576	1	null	2147483645	0	1048576	ONLINE	TE
USERS	8192	65536	null	1	2147483645	2147483645	null	65536	ONLINE	PE

Took 1 sec. Last updated by anonymous at November 21 2017, 3:16:40 PM.

READY

Using Spark to query the Oracle Database

In this section of the tutorial, we will show you working code examples that define a Spark Dataframe as an Oracle SQL query. We then define a Spark SQL temporary table against that Dataframe to show you how you can futher use Spark SQL to filter and manipulate your selected data.

To run the spark code, you should **first edit the code and insert your specific database connect string, username, and password**.

To learn more about how Spark Data Frames work with JDBC data sources, check out here (https://spark.apache.org/docs/2.1.0/sql-programming-guide.html#jdbc-to-other-databases).

Spark code to query the Oracle Database and define results and Spark SQL tables

FINISHED

```
%spark

// BEFORE RUNNING THIS, YOU WILL NEED TO EDIT THIS
//  1.Insert your database Connect String
//  2.Insert your database user name
//  3.Insert your database password


//define URL for the Oracle JDBC driver
println(">>>>>>>Defining url for Oracle JDBC")

val url="jdbc:oracle:thin:@//" + "DCB-db-nov17:1521/PDB1.gse00002281.oraclecloud.internal"

//define the username and password as properties
println(">>>>>>>Defining Oracle JDBC username and password")
val prop = new java.util.Properties


prop.setProperty("user","system")
prop.setProperty("password","Welcome1#")


prop.setProperty("driver","oracle.jdbc.OracleDriver") //the driver is needed to be defined with Spark 1.6.1 due to https://issues.apache.org/jira/browse/SPARK-14204

//now you can use JDBC commands like: val movies = sqlContext.read.jdbc(url,"movie",prop)
val utables = sqlContext.read.jdbc(url,"user_tables",prop)
//utables.explain()
utables.printSchema()
//utables.show()

//register the emp dataframe as a SparkSQL table
utables.createOrReplaceTempView("utables_sparksql")

//we can also do specific queries like the following (note that we write our query as if it was a subquery in the FROM section of a select statement)
val ora_query = sqlContext.read.jdbc(url, "(select u.tablespace_name, ts.status, count(*) tcount from user_tables u, user_tablespaces ts where u.tablespace_name=ts.tablespace_name grou
    .tablespace_name, ts.status) eq", prop)
ora_query.show()
//emp_query.explain()

println("done")
```

```
>>>>>>>Defining url for Oracle JDBC
url: String = jdbc:oracle:thin:@//DCB-db-nov17:1521/PDB1.gse00002281.oraclecloud.internal
>>>>>>>Defining Oracle JDBC username and password
```

```
prop: java.util.Properties = {}
res12: Object = null
res13: Object = null
res14: Object = null
utables: org.apache.spark.sql.DataFrame = [TABLE_NAME: string, TABLESPACE_NAME: string ... 62 more fields]
root
 |-- TABLE_NAME: string (nullable = false)
 |-- TABLESPACE_NAME: string (nullable = true)
 |-- CLUSTER_NAME: string (nullable = true)
 |-- IOT_NAME: string (nullable = true)
 |-- STATUS: string (nullable = true)
 |-- PCT_FREE: decimal(38,10) (nullable = true)
 |-- PCT_USED: decimal(38,10) (nullable = true)
 |-- INI_TRANS: decimal(38,10) (nullable = true)
 |-- MAX_TRANS: decimal(38,10) (nullable = true)
 |-- INITIAL_EXTENT: decimal(38,10) (nullable = true)
 |-- NEXT_EXTENT: decimal(38,10) (nullable = true)
 |-- MIN_EXTENTS: decimal(38,10) (nullable = true)
 |-- MAX_EXTENTS: decimal(38,10) (nullable = true)
 |-- PCT_INCREASE: decimal(38,10) (nullable = true)
 |-- FREELISTS: decimal(38,10) (nullable = true)
 |-- FREELIST_GROUPS: decimal(38,10) (nullable = true)
 |-- LOGGING: string (nullable = true)
 |-- BACKED_UP: string (nullable = true)
 |-- NUM_ROWS: decimal(38,10) (nullable = true)
 |-- BLOCKS: decimal(38,10) (nullable = true)
 |-- EMPTY_BLOCKS: decimal(38,10) (nullable = true)
 |-- AVG_SPACE: decimal(38,10) (nullable = true)
 |-- CHAIN_CNT: decimal(38,10) (nullable = true)
 |-- AVG_ROW_LEN: decimal(38,10) (nullable = true)
 |-- AVG_SPACE_FREELIST_BLOCKS: decimal(38,10) (nullable = true)
 |-- NUM_FREELIST_BLOCKS: decimal(38,10) (nullable = true)
 |-- DEGREE: string (nullable = true)
 |-- INSTANCES: string (nullable = true)
 |-- CACHE: string (nullable = true)
 |-- TABLE_LOCK: string (nullable = true)
 |-- SAMPLE_SIZE: decimal(38,10) (nullable = true)
 |-- LAST_ANALYZED: timestamp (nullable = true)
 |-- PARTITIONED: string (nullable = true)
 |-- IOT_TYPE: string (nullable = true)
 |-- TEMPORARY: string (nullable = true)
 |-- SECONDARY: string (nullable = true)
 |-- NESTED: string (nullable = true)
 |-- BUFFER_POOL: string (nullable = true)
 |-- FLASH_CACHE: string (nullable = true)
 |-- CELL_FLASH_CACHE: string (nullable = true)
 |-- ROW_MOVEMENT: string (nullable = true)
 |-- GLOBAL_STATS: string (nullable = true)
 |-- USER_STATS: string (nullable = true)
 |-- DURATION: string (nullable = true)
 |-- SKIP_CORRUPT: string (nullable = true)
 |-- MONITORING: string (nullable = true)
 |-- CLUSTER_OWNER: string (nullable = true)
 |-- DEPENDENCIES: string (nullable = true)
 |-- COMPRESSION: string (nullable = true)
```

```
 |-- COMPRESS_FOR: string (nullable = true)
 |-- DROPPED: string (nullable = true)
 |-- READ_ONLY: string (nullable = true)
 |-- SEGMENT_CREATED: string (nullable = true)
 |-- RESULT_CACHE: string (nullable = true)
 |-- CLUSTERING: string (nullable = true)
 |-- ACTIVITY_TRACKING: string (nullable = true)
 |-- DML_TIMESTAMP: string (nullable = true)
 |-- HAS_IDENTITY: string (nullable = true)
 |-- CONTAINER_DATA: string (nullable = true)
 |-- INMEMORY: string (nullable = true)
 |-- INMEMORY_PRIORITY: string (nullable = true)
 |-- INMEMORY_DISTRIBUTE: string (nullable = true)
 |-- INMEMORY_COMPRESSION: string (nullable = true)
 |-- INMEMORY_DUPLICATE: string (nullable = true)
ora_query: org.apache.spark.sql.DataFrame = [TABLESPACE_NAME: string, STATUS: string ... 1 more field]
+---------------+------+-------------+
```

Took 4 sec. Last updated by anonymous at November 21 2017, 3:19:24 PM.

SparkSQL Example against our Oracle Database-based Data Frame

FINISHED

```
%sql
select table_name, num_rows, last_analyzed from utables_sparksql
where num_rows > 0
order by num_rows desc
```

table_name	num_rows	last_analyzed
HELP	938.0000000000	2014-07-07 06:08:56.0
LOGSTDBY$SKIP_SUPPORT	351.0000000000	2014-07-07 06:04:31.0
MVIEW$_ADV_PARAMETERS	40.0000000000	2014-07-07 06:04:32.0
REPCAT$_OBJECT_TYPES	28.0000000000	2014-07-07 06:04:33.0
AQ$_QUEUES	21.0000000000	2014-07-07 06:04:27.0
REPCAT$_RESOLUTION_METHOD	19.0000000000	2014-07-07 06:04:33.0
AQ$_QUEUE_TABLES	13.0000000000	2017-11-18 06:00:46.0
AQ$_INTERNET_AGENTS	5.0000000000	2014-07-07 06:04:27.0
REPCAT$_TEMPLATE_STATUS	3.0000000000	2014-07-07 06:04:33.0

Took 1 sec. Last updated by anonymous at November 21 2017, 3:21:51 PM.

Using Spark to write to the Oracle Database

READY

Now we will show a working example of writing back to the Oracle Database from Spark. If you observed above, we created a Spark dataframe called emp_query. In the following example, we will

write this dataframe back to the Oracle Database as a new table called emp_query.

For this example, we will use Spark to read in some Citibike data and write that data into an Oracle table.

Spark Scala to read CSV and register as Spark SQL temporary table

FINISHED

```
%spark

//a previous tutorial placed the csv file into /var/lib/zeppelin/bikes/201612-citibike-tripdata.csv

val df = sqlContext.read.format("com.databricks.spark.csv").option("header", "true").load("file:/var/lib/zeppelin/citibike/201612-citibike-tripdata.csv")

//cache the data frame for performance
df.cache()


println("Here is the schema detected from the CSV")
df.printSchema()
println("..")

println("# of rows: %s".format(
  df.count()
))
println("..")

df.createOrReplaceTempView("bike_trips_csvtemp")
println("done")
```

```
df: org.apache.spark.sql.DataFrame = [Trip Duration: string, Start Time: string ... 13 more fields]
res20: df.type = [Trip Duration: string, Start Time: string ... 13 more fields]
Here is the schema detected from the CSV
root
 |-- Trip Duration: string (nullable = true)
 |-- Start Time: string (nullable = true)
 |-- Stop Time: string (nullable = true)
 |-- Start Station ID: string (nullable = true)
 |-- Start Station Name: string (nullable = true)
 |-- Start Station Latitude: string (nullable = true)
 |-- Start Station Longitude: string (nullable = true)
 |-- End Station ID: string (nullable = true)
 |-- End Station Name: string (nullable = true)
 |-- End Station Latitude: string (nullable = true)
 |-- End Station Longitude: string (nullable = true)
 |-- Bike ID: string (nullable = true)
 |-- User Type: string (nullable = true)
```

Took 13 sec. Last updated by anonymous at November 21 2017, 3:22:48 PM.

Spark to write a DataFrame to an Oracle table

FINISHED

```
%spark



// To make sure we have Oracle friendly column names, lets select against our Spark SQL temp table and rename columns
println("Renaming column names via bike_query...")
```

```
Renaming column names via bike_query...
bike_query: org.apache.spark.sql.DataFrame = [TRIPDURATION: string, STARTTIME: string ... 13 more fields]
+------------+-------------------+-------------------+--------------+--------------------+--------------------+---------------------+------------+--------------------+-----------------
-+-------------------+------+----------+---------+------+
|TRIPDURATION|          STARTTIME|           STOPTIME|STARTSTATIONID|    STARTSTATIONNAME|STARTSTATIONLATITUDE|STARTSTATIONLONGITUDE|ENDSTATIONID|      ENDSTATIONNAME|ENDSTATIONLATITUD
E|ENDSTATIONLONGITUDE|BIKEID|  USERTYPE|BIRTHYEAR|GENDER|
+------------+-------------------+-------------------+--------------+--------------------+--------------------+---------------------+------------+--------------------+-----------------
-+-------------------+------+----------+---------+------+
|         528|2016-12-01 00:00:04|2016-12-01 00:08:52|           499|  Broadway & W 60 St|         40.76915505|         -73.98191841|         228|     E 48 St & 3 Ave|     40.754601102
6|      -73.971878855| 26931|Subscriber|     1964|     1|
|         218|2016-12-01 00:00:28|2016-12-01 00:04:06|          3418|Plaza St West & F...|          40.6750207|         -73.97111473|        3358| Garfield Pl & 8 Ave|        40.671197
8|       -73.97484126| 27122|Subscriber|     1955|     1|
|         399|2016-12-01 00:00:39|2016-12-01 00:07:19|           297|     E 15 St & 3 Ave|           40.734232|           -73.986923|         345|     W 13 St & 6 Ave|       40.7364940
3|       -73.99704374| 19352|Subscriber|     1985|     1|
|         254|2016-12-01 00:00:44|2016-12-01 00:04:59|           405|Washington St & G...|           40.739323|           -74.008119|         358|Christopher St & ...|       40.7329155
3|       -74.00711384| 20015|Subscriber|     1982|     1|
|        1805|2016-12-01 00:00:54|2016-12-01 00:31:00|           279|Peck Slip & Front St|           40.707873|            -74.00167|         279|Peck Slip & Front St|         40.70787
3|          -74.00167| 23148|Subscriber|     1989|     1|
|         483|2016-12-01 00:01:13|2016-12-01 00:09:17|           245|Myrtle Ave & St E...|         40.69327018|         -73.97703874|         372|Franklin Ave & My...|         40.69452
8|         -73.958089| 16140|Subscriber|     1986|     1|
|        1114|2016-12-01 00:01:37|2016-12-01 00:20:12|           470|     W 20 St & 8 Ave|         40.74345335|         -74.00004031|         453|     W 22 St & 8 Ave|       40.7447514
```

```
8|       -73.99915362| 19997|Subscriber|     1964|     1|
|        2680|2016-12-01 00:01:50|2016-12-01 00:46:30|          3312|     1 Ave & E 94 St|          40.7817212|            -73.94594|        3325|     E 95 St & 3 Ave|        40.784903
2|         -73.950503| 26105|Subscriber|     null|     0|
|        1967|2016-12-01 00:01:52|2016-12-01 00:34:40|           387|Centre St & Chamb...|         40.71273266|          -74.0046073|         387|Centre St & Chamb...|       40.7127326
6|        -74.0046073| 21348|  Customer|     null|     0|
|         356|2016-12-01 00:01:54|2016-12-01 00:07:50|           496|     E 16 St & 5 Ave|         40.73726186|         -73.99238967|         212|W 16 St & The Hig...|       40.7433493
5|       -74.00681753| 22517|Subscriber|     1954|     1|
|         298|2016-12-01 00:01:54|2016-12-01 00:06:53|           297|     E 15 St & 3 Ave|           40.734232|           -73.986923|         476|     E 31 St & 3 Ave|       40.7439431
4|       -73.97966069| 26676|Subscriber|     1986|     1|
|         315|2016-12-01 00:02:05|2016-12-01 00:07:20|          2004|   6 Ave & Broome St|           40.724399|           -74.004704|         426|West St & Chamber...|       40.7175483
4|       -74.01322069| 22515|Subscriber|     1976|     1|
|         735|2016-12-01 00:02:10|2016-12-01 00:14:26|           390|Duffield St & Wil...|         40.69221589|          -73.9842844|        3060|Willoughby Ave & ...|       40.6942540
3|       -73.94626915| 26945|Subscriber|     1987|     1|
|         361|2016-12-01 00:02:10|2016-12-01 00:08:12|          3164|Columbus Ave & W ...|          40.7770575|         -73.97898475|        3170|W 84 St & Columbu...|       40.7849997
9|       -73.97283406| 22340|Subscriber|     1962|     1|
|        1633|2016-12-01 00:02:18|2016-12-01 00:29:31|           387|Centre St & Chamb...|         40.71273266|          -74.0046073|         387|Centre St & Chamb...|       40.7127326
6|        -74.0046073| 26482|  Customer|     null|     0|
```

Took 34 sec. Last updated by anonymous at November 21 2017, 3:23:52 PM.

```
%jdbc(orcl-system)
select * from CITIBIKE_ORCL
```

FINISHED

TRIPDURATION	STARTTIME	STOPTIME	STARTSTATIONID	STARTSTATIONNAME	STARTSTATIONLATITUDE	STARTSTATIONLONGITUDE	ENDS
528	2016-12-01 00:00:04	2016-12-01 00:08:52	499	Broadway & W 60 St	40.76915505	-73.98191841	228
218	2016-12-01 00:00:28	2016-12-01 00:04:06	3418	Plaza St West & Flatbush Ave	40.6750207	-73.97111473	3358
399	2016-12-01 00:00:39	2016-12-01 00:07:19	297	E 15 St & 3 Ave	40.734232	-73.986923	345
254	2016-12-01 00:00:44	2016-12-01 00:04:59	405	Washington St & Gansevoort St	40.739323	-74.008119	358
1805	2016-12-01 00:00:54	2016-12-01 00:31:00	279	Peck Slip & Front St	40.707873	-74.00167	279
483	2016-12-01 00:01:13	2016-12-01 00:09:17	245	Myrtle Ave & St Edwards St	40.69327018	-73.97703874	372
1114	2016-12-01 00:01:37	2016-12-01 00:20:12	470	W 20 St & 8 Ave	40.74345335	-74.00004031	453
2680	2016-12-01 00:01:50	2016-12-01 00:46:30	3312	1 Ave & E 94 St	40.7817212	-73.94594	3325

Took 1 sec. Last updated by anonymous at November 21 2017, 3:25:08 PM.

READY