Exercise 5 – Drivers

In this exercise, you will:

- You will understand what Apache Cassandra™ drivers are and their purpose.
- You will be able to create and read records using a driver.

As we have already seen, we can access Apache Cassandra™ for client applications such as CQLSH. However, we may want to access Apache Cassandra™ directly from within an application we create. Drivers are the mechanisms we use to interact with Apache Cassandra™ from a programming language. In this exercise, we will connect to our Apache Cassandra™ database using the Python driver and read and write some data.

Steps

1) Back in your Terminal window, make sure DataStax Enterprise is still running with `/home/ubuntu/node/bin/dsetool status`. If not, restart DataStax Enterprise.

2) Start your Python interpreter:

```
ubuntu@ds201-node1:~$ python
Python 2.7.12 (default, Nov 20 2017, 18:23:56)
[GCC 5.4.0 20160609] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

3) Let's write Python code to connect to the cluster and open a session:

```
from cassandra.cluster import Cluster
cluster = Cluster(protocol_version = 3)
session = cluster.connect('killrvideo')
```

NOTE: If you have trouble connecting, your python driver may be out of date. Use the following command to update the driver.

```
sudo pip install --upgrade cassandra-driver
```

4) Now write code to retrieve all records in the `videos_by_tag` table. Display your results using Python's `print()` function:

```
for val in session.execute("SELECT * FROM videos_by_tag"):
    print(val[0])
```

5) `session.execute()` returns a sequence of rows (tuples), which you can further index into to get cell values. Try executing the following:

```
print('{0:12} {1:40} {2:5}'.format('Tag', 'ID', 'Title'))
for val in session.execute("select * from videos_by_tag"):
    print('{0:12} {1:40} {2:5}'.format(val[0], val[2], val[3]))
```

We use some simple string formatting to make the output bearable. Notice we index into each tuple using the bracket operator.

6) Write some Python code to insert a new video into the database:

```
session.execute(
    "INSERT INTO videos_by_tag (tag, added_date, video_id, title)" +
    "VALUES ('cassandra', '2013-01-10', uuid(), 'Cassandra Is My Friend')")
```

7) Now run the following code to view your new record:

```
print('{0:12} {1:40} {2:5}'.format('Tag', 'ID', 'Title'))
for val in session.execute("select * from videos_by_tag"):
    print('{0:12} {1:40} {2:5}'.format(val[0], val[2], val[3]))
```

8) Now write Python code to delete your record:

```
session.execute("DELETE FROM videos_by_tag " +
                "WHERE tag = 'cassandra' AND added_date = '2013-01-10' AND " +
                "video_id = INSERT_YOUR_UUID_HERE")
```

Note to instructor - be sure to use the correct UUID for video_id.

9) Now execute the following code to confirm that your record is gone:

```
print('{0:12} {1:40} {2:5}'.format('Tag', 'ID', 'Title'))
for val in session.execute("select * from videos_by_tag"):
  print('{0:12} {1:40} {2:5}'.format(val[0], val[2], val[3]))
```