Tutorial 7
Python Driver for Apache Cassandra

The main objective of this tutorial is to understand how Cassandra Driver works, particularly by using the Python Driver.

Cassandra Driver

1. Install cassandra driver in the command prompt/terminal
   If python is installed then the following can be used:
   ```
   python -m pip install cassandra-driver
   ```

   [Note: If there is an error saying : no module named pip then it could be that you are using an anaconda python installation and therefore the following can be used:
   ```
   conda install -c conda-forge cassandra-driver
   ```

2. Then run python in the terminal to start the python shell
   ```
   python
   ```

   Meanwhile, in a separate command prompt/terminal window please run the cassandra server using
   ```
   cassandra
   ```

3. The cassandra driver can now be imported but we will import only the cluster module in the cassandra driver using
   ```
   from cassandra.cluster import Cluster
   ```

4. Now we can create a connection to a Cassandra instance on your local host
   ```
   cluster = Cluster(['localhost'])
   ```

   To establish connections and begin executing queries we need a Session,
   # which is created by calling Cluster.connect():
   ```
   session = cluster.connect()
   ```

   Drop existing keyspace of the same name
   ```
   session.execute("DROP KEYSPACE stocks")
   ```

5. Once connected we can now create a keyspace
   ```
   session.execute("CREATE KEYSPACE stocks WITH replication =
   {'class':'SimpleStrategy', 'replication_factor':1}")
   ```

6. Next switch to the newly created keyspace we can use:
   ```
   session.set_keyspace('stocks')
   ```

7. Now we can create the following tables using the execute command again:
   Company Table

```
session.execute("""
                  CREATE TABLE company (
                        company_id text,
                        name_latest text,
                        names_previous text,
                        PRIMARY KEY (company_id)
                   )
                  """)
```

   Drop the table of same name

```
session.execute("""
                DROP TABLE company
                """)
```

   Indicator Table

```
session.execute("""
                  CREATE TABLE indicator_by_company (
                        company_id text,
                        indicator_id text,
                        yr_2010 bigint,
                        yr_2011 bigint,
                        yr_2012 bigint,
                        yr_2013 bigint,
                        yr_2014 bigint,
                        yr_2015 bigint,
                        yr_2016 bigint,
                        PRIMARY KEY (company_id, indicator_id)
                   )
                  """)
```

   Drop the table of same name

```
session.execute("""
                DROP TABLE indicator_by_company
                """)
```

   NOTE the indicator table is highly denormalized (following the typical cassandra tables i.e. wide column normalisation)
   Bigint data types were used because the values for yr_2010 to yr_2016 are large.

8. Now that the preparation stage is over we can start importing the data.
   For JSON data we will need to import the JSON library in python

```
import json
```

9. Use the following command to open command to open the JSON file

```
with open('companies.json') as f_in:
      companies = json.load(f_in)
```

   Note:
   You may need to click on the enter key twice to end the command

   To Find out the type of the variable companies

```
type(companies)
```

To Find out the type of an element in the imported companies

```
type(companies[0])
```

10. For inserting into the table we can use:

```
session.execute("""
      INSERT INTO company (company_id, name_latest, names_previous)
      VALUES (%s, %s, %s)
            """,
      ("1000045","Nicholas Financial Inc",""))
```

11. For inserting into the table use the file:

```
for company in companies:
    try:
        session.execute("""
                        INSERT INTO company (company_id, name_latest,
name_previous)
                        VALUES (%s, %s, %s)
                        """,
                        (company['company_id'],
company['name_latest'], company['names_previous']))
    except:
        pass
```

To display the inserted data

```
rows = session.execute('SELECT * FROM company')

for company_row in rows:
    print company_row.company_id, company_row.name_latest,
company_row.names_previous
```

References:
https://docs.datastax.com/en/developer/python-driver/3.25/getting_started/

The End