Lesson 3 Demo 3 Clustering Column

May 11, 2021

1 Lesson 3 Demo 3: Focus on Clustering Columns

1.0.1 In this demo we are going to walk through the basics of creating a table with a good Primary Key and Clustering Columns in Apache Cassandra, inserting rows of data, and doing a simple SQL query to validate the information.

We will use a python wrapper/ python driver called cassandra to run the Apache Cassandra queries. This library should be preinstalled but in the future to install this library you can run this command in a notebook to install locally: ! pip install cassandra-driver #### More documentation can be found here: https://datastax.github.io/python-driver/

Import Apache Cassandra python package

```
In [1]: import cassandra
```

1.0.2 First let's create a connection to the database

```
In [2]: from cassandra.cluster import Cluster
        try:
            cluster = Cluster(['127.0.0.1']) #If you have a locally installed Apache Cassandra i
            session = cluster.connect()
        except Exception as e:
            print(e)
```

1.0.3 Let's create a keyspace to do our work in

```
In [3]: try:
            session.execute("""
            CREATE KEYSPACE IF NOT EXISTS udacity
            WITH REPLICATION =
            { 'class' : 'SimpleStrategy', 'replication_factor' : 1 }""")
        )

        except Exception as e:
            print(e)
```

Connect to our Keyspace. Compare this to how we had to create a new session in PostgreSQL.

```
In [4]: try:
            session.set_keyspace('udacity')
        except Exception as e:
            print(e)
```

1.0.4 Let's imagine we would like to start creating a new Music Library of albums.

1.0.5 We want to ask 1 question of our data

1. Give me every album in my music library that was released by an Artist with Albumn Name in `DESC` Order and City In `DESC` Order `select * from music_library WHERE ARTIST_NAME="The Beatles"`

1.0.6 Here is our Collection of Data

1.0.7 How should we model this data? What should be our Primary Key and Partition Key? Since our data is looking for the `ARTIST_NAME` let's start with that. From there we will need to add other elements to make sure the Key is unique. We also need to add the `CITY` and `ALBUM_NAME` as Clustering Columns to sort the data. That should be enough to make the row key unique

`Table Name: music_library column 1: Year column 2: Artist Name column 3: Album Name Column 4: City PRIMARY KEY(artist name, album name, city)`

```
In [5]: query = "CREATE TABLE IF NOT EXISTS music_library "
        query = query + "(year int, artist_name text, album_name text, city text, PRIMARY KEY (a
        try:
            session.execute(query)
        except Exception as e:
            print(e)
```

1.0.8 Let's insert our data into of table

```
In [6]: query = "INSERT INTO music_library (year, artist_name, album_name, city)"
        query = query + " VALUES (%s, %s, %s, %s)"

        try:
            session.execute(query, (1970, "The Beatles", "Let it Be", "Liverpool"))
        except Exception as e:
            print(e)

        try:
            session.execute(query, (1965, "The Beatles", "Rubber Soul", "Oxford"))
        except Exception as e:
            print(e)

        try:
            session.execute(query, (1964, "The Beatles", "Beatles For Sale", "London"))
```

```
        except Exception as e:
            print(e)

        try:
            session.execute(query, (1966, "The Monkees", "The Monkees", "Los Angeles"))
        except Exception as e:
            print(e)

        try:
            session.execute(query, (1970, "The Carpenters", "Close To You", "San Diego"))
        except Exception as e:
            print(e)
```

1.0.9 Let's Validate our Data Model -- Did it work?? If we look for Albums from The Beatles we should expect to see 3 rows.

```
select * from music_library WHERE ARTIST_NAME="The Beatles"
```

```
In [7]: query = "select * from music_library WHERE ARTIST_NAME='The Beatles'"
        try:
            rows = session.execute(query)
        except Exception as e:
            print(e)

        for row in rows:
            print (row.artist_name, row.album_name, row.city, row.year)
```

```
The Beatles Beatles For Sale London 1964
The Beatles Let it Be Liverpool 1970
The Beatles Rubber Soul Oxford 1965
```

1.0.10 Success it worked! We created a unique Primary key that evenly distributed our data, with clustering columns that sorted our data.

1.0.11 For the sake of the demo, I will drop the table.

```
In [8]: query = "drop table music_library"
        try:
            rows = session.execute(query)
        except Exception as e:
            print(e)
```

1.0.12 And Finally close the session and cluster connection

```
In [9]: session.shutdown()
        cluster.shutdown()
```

```
In [ ]:
```