**INSTALL CASSANDRA**

1. Download cassandra from apache.cassandra.Add environment variable
2. Start cassandra server

**cassandra.bat -f**

1. Under C:\apache-cassandra-3.11.6-bin\apache-cassandra-3.11.6\bin

Type:

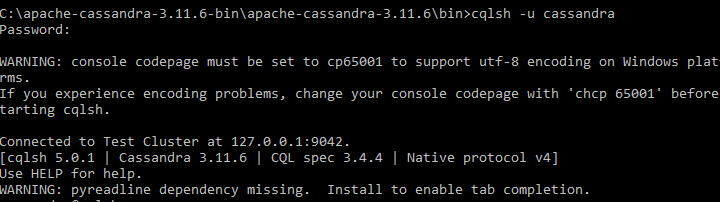
**cqlsh**

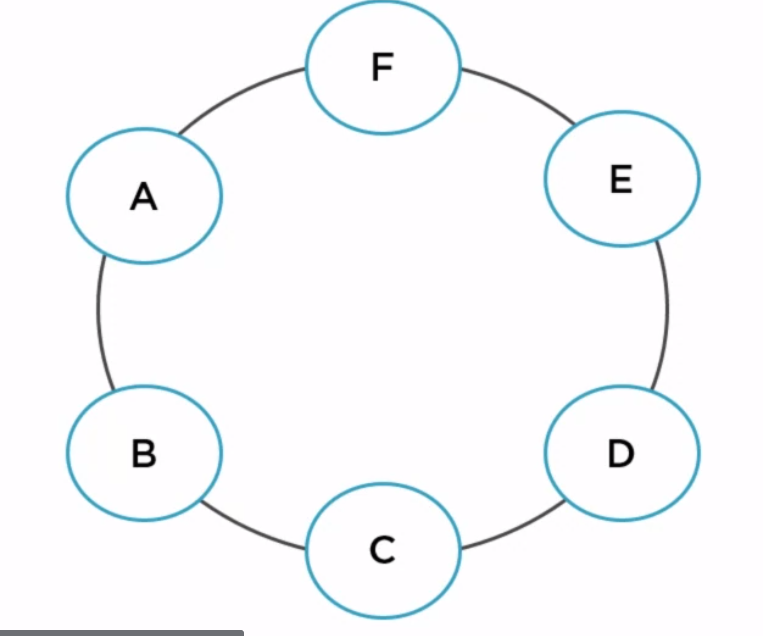
Note: Cassandra needs python2 version else it would throw error

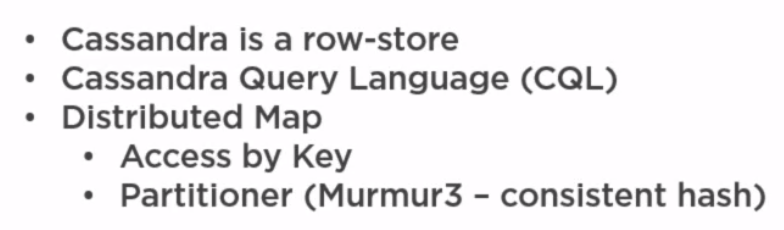
Version of kafka

**cqlsh -u cassandra**

**pwd :temp1234**





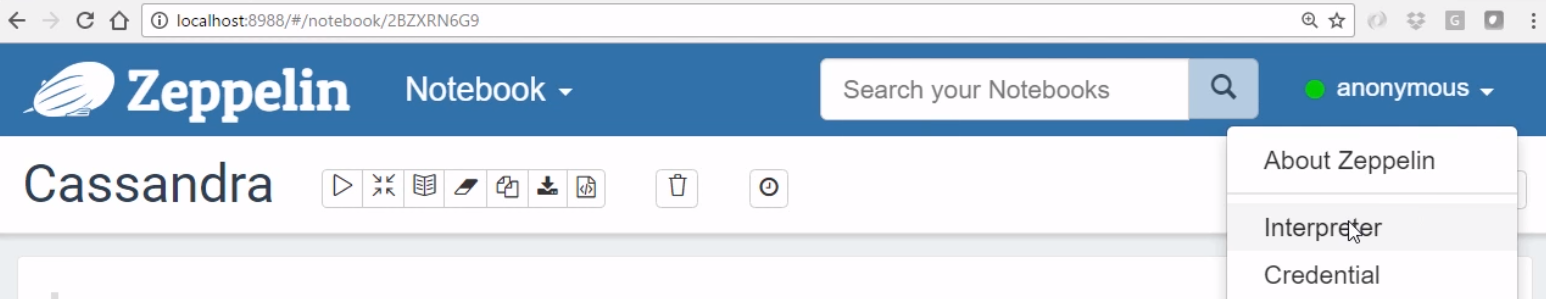


**STEPS TO USE CASSANDRA USING ZEPPLIN**

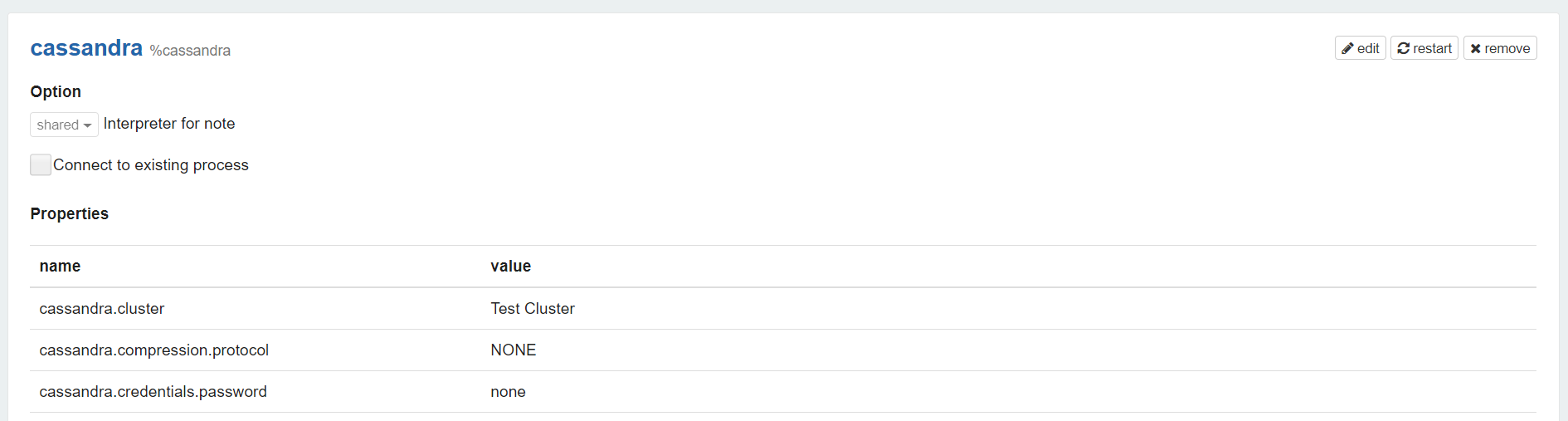
Note: Vm should be up and running.Instructions for these has been provided in lambda architecture repo

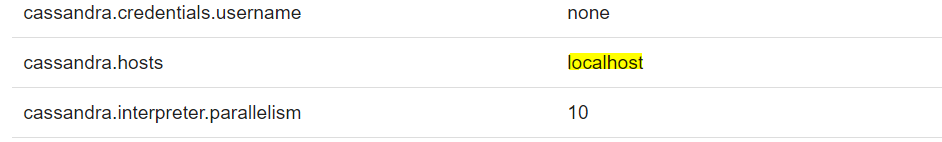
1.Run the vm and open localhost with port for zepplin

2.Select interpretor ->

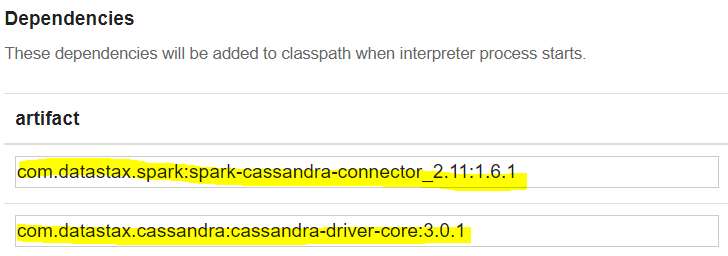


3. Here we would see the Cassandra interpreter

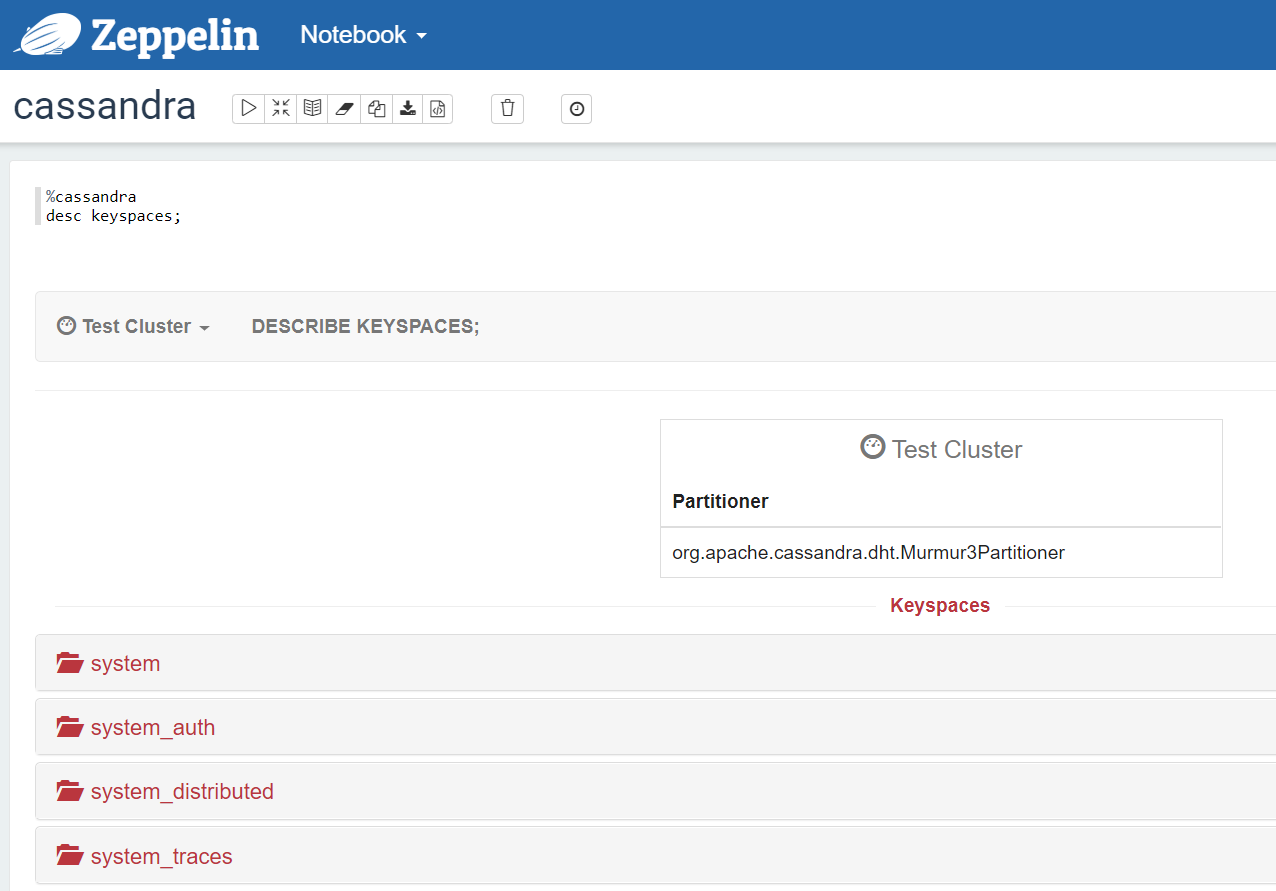




4. Set dependencies for spark



5.Look for keyspaces



6.Create keyspace



**%cassandra**

**CREATE KEYSPACE lambda WITH replication ={'class':'SimpleStrategy','replication\_factor':1};**

7.Create table

Note:

1.Primary key consists of 2 columns (composite key)

Product, timestamp\_hour

* *Product* key is used to partition the data.
* Data is further grouped by and sorted by using the remaining columns known as clustering keys *(timestamp\_hour*)

2.clustering order makes sure that we get the latest records

**%cassandra**

**CREATE TABLE lambda.stream\_activity\_by\_product (**

**product text,**

**timestamp\_hour bigint,**

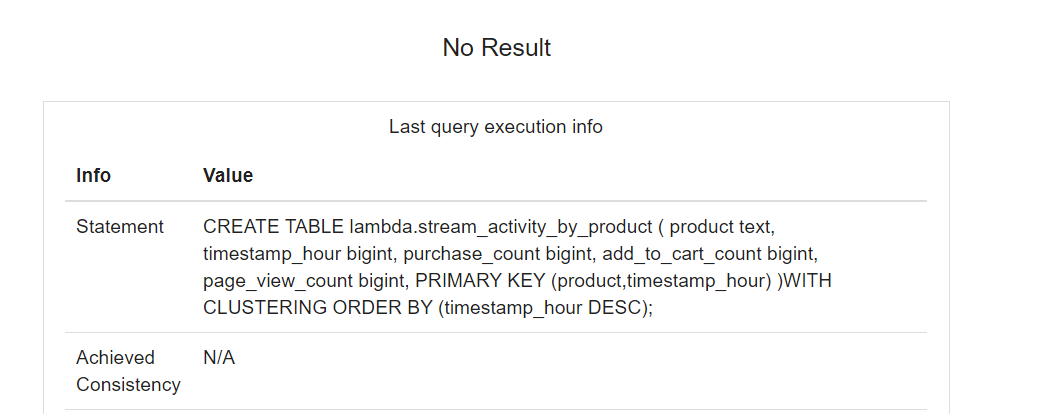
**purchase\_count bigint,**

**add\_to\_cart\_count bigint,**

**page\_view\_count bigint,**

**PRIMARY KEY (product,timestamp\_hour)**

**) WITH CLUSTERING ORDER BY (timestamp\_hour DESC);**



**%cassandra**

**CREATE TABLE lambda.stream\_visitors\_by\_product (**

**product text,**

**timestamp\_hour bigint,**

**unique\_visitors bigint,**

**PRIMARY KEY (product, timestamp\_hour)**

**) WITH CLUSTERING ORDER BY (timestamp\_hour DESC);**

**CREATE TABLE lambda.batch\_activity\_by\_product (**

**product text,**

**timestamp\_hour bigint,**

**purchase\_count bigint,**

**add\_to\_cart\_count bigint,**

**page\_view\_count bigint,**

**PRIMARY KEY (product, timestamp\_hour)**

**) WITH CLUSTERING ORDER BY (timestamp\_hour DESC);**

**CREATE TABLE lambda.batch\_visitors\_by\_product (**

**product text,**

**timestamp\_hour bigint,**

**unique\_visitors bigint,**

**PRIMARY KEY (product, timestamp\_hour)**

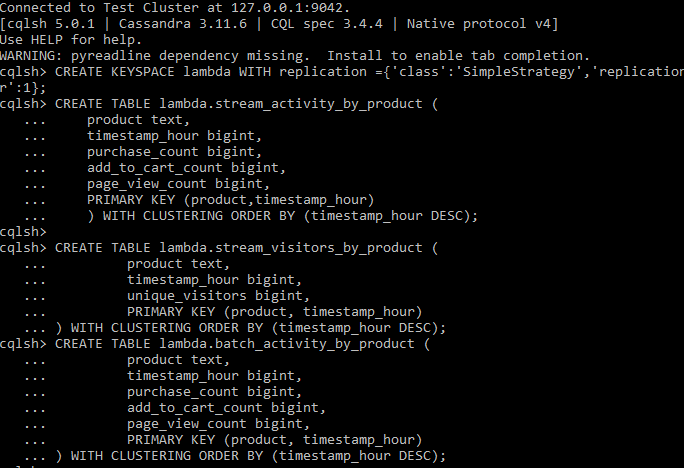
**) WITH CLUSTERING ORDER BY (timestamp\_hour DESC);**

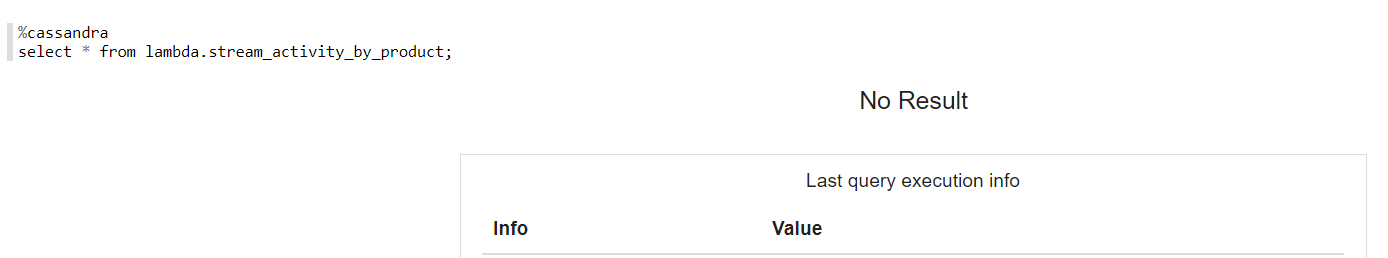
**8. Run the select query**

**%cassandra**

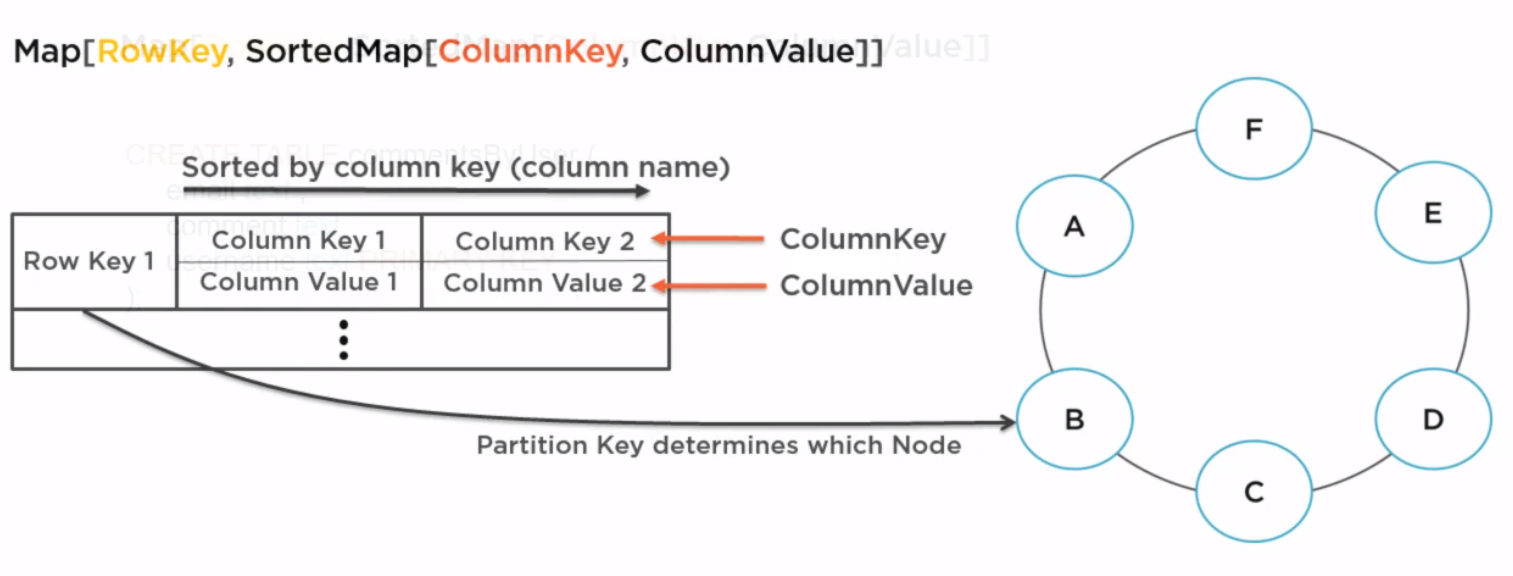
**select \* from lambda.stream\_activity\_by\_product;**

Note: This can be done in zeppelin or in in command line as shown below

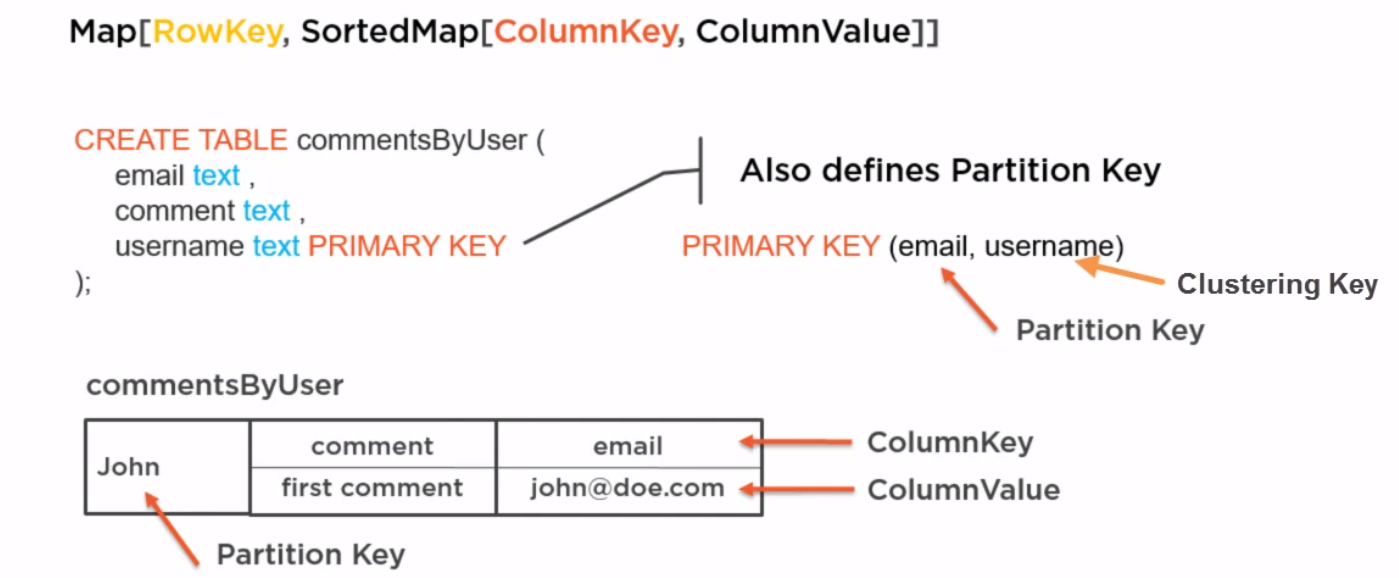




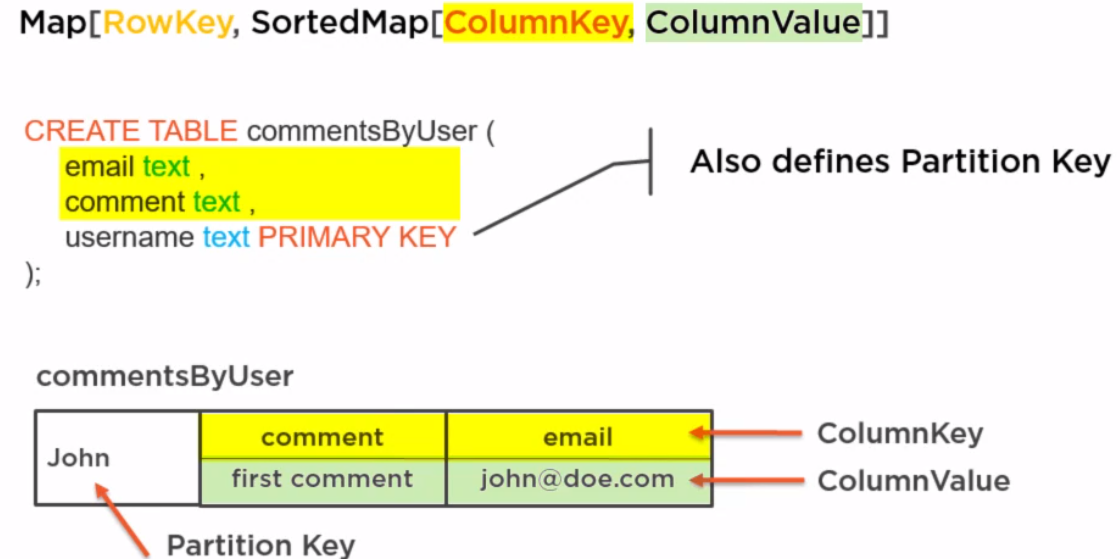
**CASSANDRA DATA MODEL**



**PARTITIONS**

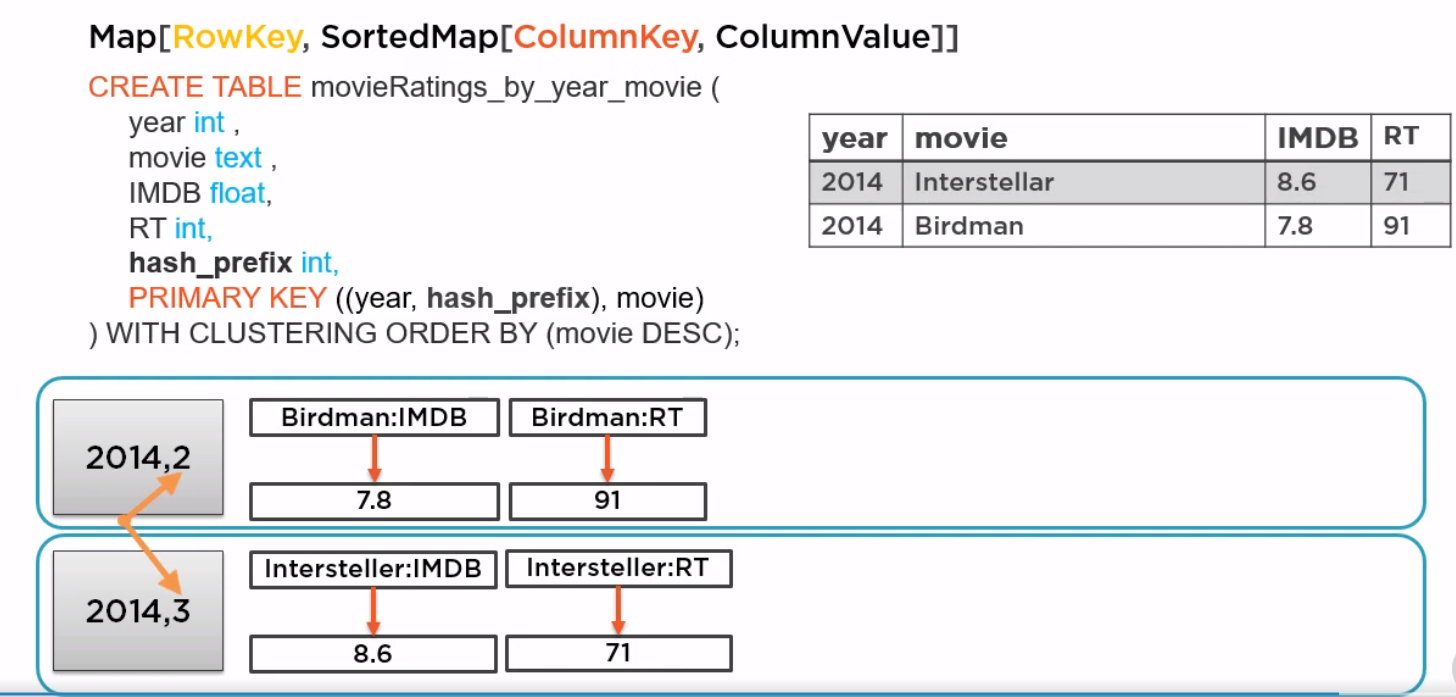


**KEY AND KEY VALUES**

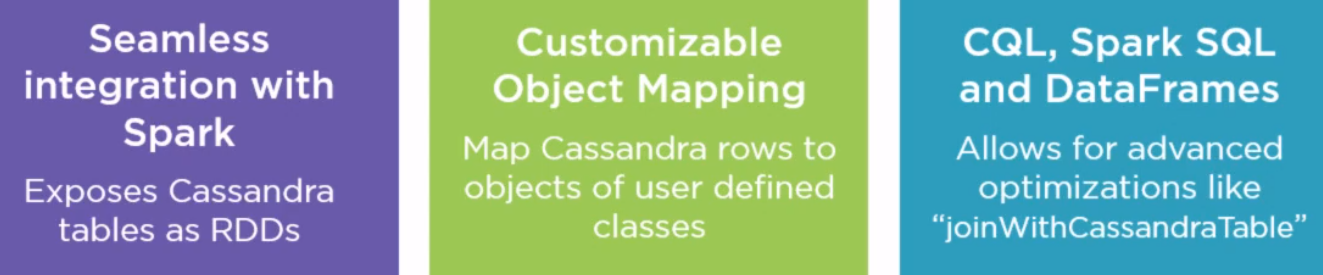


**COMPOSITE PARTITION KEYS WITH HASH PREFIX**

Here we are creating composite partition key based on year and hash



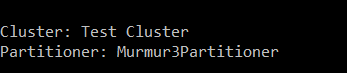
**SPARK CASSANDRA CONNECTOR**



**CQLSH COMMANDS**

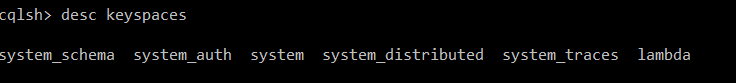
1. Show clusters

**cqlsh> desc cluster**



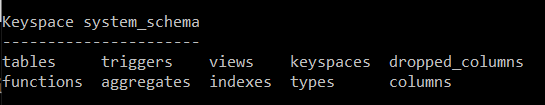
1. Show keyspaces

**cqlsh> desc keyspaces**



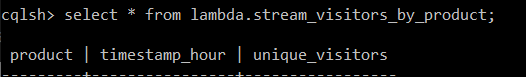
1. Show tables

**cqlsh> desc tables**



1. Run select queries

cqlsh> select \* from lambda.stream\_visitors\_by\_product;



Sample

cqlsh> CREATE KEYSPACE test WITH replication = {'class': 'SimpleStrategy', 'replication\_factor': 1 };

cqlsh> USE "test";

cqlsh:test> CREATE TABLE my\_table(key text PRIMARY KEY, value int);

cqlsh:test> INSERT INTO my\_table(key, value) VALUES ('key1', 1);

cqlsh:test> INSERT INTO my\_table(key, value) VALUES ('key2', 2);

cqlsh:test> SELECT \* from my\_table;

**CQLSH COMMANDS**

1.Create keyspace

**CREATE KEYSPACE lambda WITH replication ={'class':'SimpleStrategy','replication\_factor':1};**

2.Create table

Note:

1.Primary key consists of 2 columns (composite key)

Product, timestamp\_hour

* *Product* key is used to partition the data.
* Data is further grouped by and sorted by using the remaining columns known as clustering keys *(timestamp\_hour*)

2.clustering order makes sure that we get the latest records

**%cassandra**

**CREATE TABLE lambda.stream\_activity\_by\_product (**

**product text,**

**timestamp\_hour bigint,**

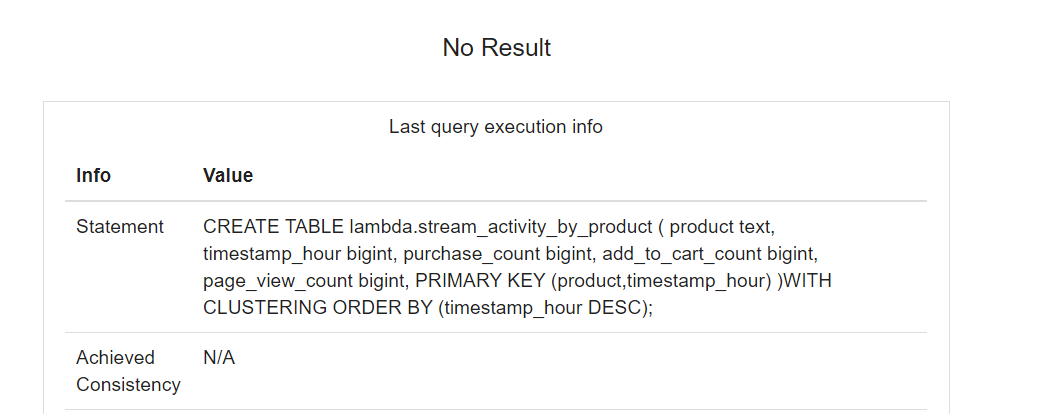
**purchase\_count bigint,**

**add\_to\_cart\_count bigint,**

**page\_view\_count bigint,**

**PRIMARY KEY (product,timestamp\_hour)**

**) WITH CLUSTERING ORDER BY (timestamp\_hour DESC);**



**%cassandra**

**CREATE TABLE lambda.stream\_visitors\_by\_product (**

**product text,**

**timestamp\_hour bigint,**

**unique\_visitors bigint,**

**PRIMARY KEY (product, timestamp\_hour)**

**) WITH CLUSTERING ORDER BY (timestamp\_hour DESC);**

**CREATE TABLE lambda.batch\_activity\_by\_product (**

**product text,**

**timestamp\_hour bigint,**

**purchase\_count bigint,**

**add\_to\_cart\_count bigint,**

**page\_view\_count bigint,**

**PRIMARY KEY (product, timestamp\_hour)**

**) WITH CLUSTERING ORDER BY (timestamp\_hour DESC);**

**CREATE TABLE lambda.batch\_visitors\_by\_product (**

**product text,**

**timestamp\_hour bigint,**

**unique\_visitors bigint,**

**PRIMARY KEY (product, timestamp\_hour)**

**) WITH CLUSTERING ORDER BY (timestamp\_hour DESC);**

2. Add library references

//casssandra  
compile 'com.datastax.cassandra:cassandra-driver-core:3.8.0'  
compile 'com.datastax.spark:spark-cassandra-connector\_2.12:2.4.3'

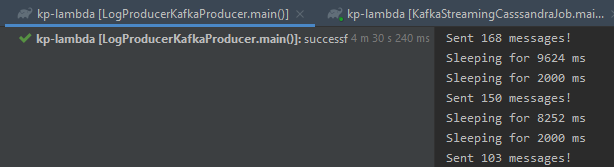
3.Create new object for saving the data in Cassandra tables



4.Specify Cassandra server details

5.start the Cassandra server

6.Run the kafka producer



7. Run spark streaming

8. data saved in Cassandra tables

Note:Zookeeper and kafka brokers should be running.

