

Name: Deep Pawar(A20545137)
Professor: Joseph Rosen
Institute: Illinois Institute of Technology

CSP 554: Big Data Technologies

Fall 2024 - Assignment 12

- **Questions and Answers:**

Exercise 1) (4 points)

Read the article “A Big Data Modeling Methodology for Apache Cassandra” available on the blackboard in the ‘Articles’ section.

Provide a ½ page summary including your comments and impressions.

Ans:

- **Summary of "A Big Data Modeling Methodology for Apache Cassandra"**

In contrast to conventional relational database modeling techniques, this paper presents a query-driven data modeling methodology for Apache Cassandra. The creation of an automated data modeling tool called KDM, the introduction of Chebotko Diagrams for schema visualization, and the principles, mapping rules, and patterns for logical data modeling are some of the major contributions.

While Cassandra promotes efficiency and necessitates denormalization and data duplication, traditional relational design concentrates on eliminating redundancy and utilizing SQL's flexibility. The approach places a strong emphasis on identifying queries that guide table schema design by beginning with an application workflow. To maximize query performance, the method makes use of four fundamental ideas: comprehending data, prioritizing queries, employing data nesting, and embracing data duplication.

While mapping patterns expedite the building of schemas, mapping rules facilitate the shift from conceptual (such as entity-relationship diagrams) to logical models customized to Cassandra's restrictions. Schema is further refined by physical modeling to accommodate real-world limitations such as cluster resources and data partitioning. This procedure can be better understood and automated with the help of the visualization tool Chebotko Diagrams. Schema generation is automated by the suggested KDM tool, which benefits both inexperienced and seasoned users.

- **My Comments and Impressions**

This well-organized paper fills a major adoption gap for Cassandra by offering a methodical approach to its modeling requirements. For developers working with relational systems, the emphasis on query-first architecture is a crucial change. One noteworthy innovation that enhances team communication and clarity is the Chebotko Diagrams. The KDM tool has the potential to simplify schema design and cut down on errors. All things considered, this work is an invaluable tool for big data solution researchers and practitioners.

Exercise 2) (3 points)

Step A – Start an EMR cluster

Create cluster > EMR on EC2: Clusters > Create cluster

Create cluster [Info](#)

Name and applications - required [Info](#)
Name your cluster and choose the applications that you want to install on your cluster.

Name
Mycluster_Assignment12

Amazon EMR release [Info](#)
A release contains a set of applications which can be installed on your cluster.
emr-7.5.0

Application bundle

Spark Interactive	Core Hadoop	Flink	HBase	Presto	Trino	Custom
<input type="checkbox"/> AmazonCloudWatchAgent 1.300032.2	<input type="checkbox"/> HCatalog 3.1.3	<input type="checkbox"/> Flink 1.19.1	<input type="checkbox"/> HBase 2.5.10	<input type="checkbox"/> Oozie 5.2.1	<input type="checkbox"/> JupyterHub 1.5.0	
<input checked="" type="checkbox"/> Hue 4.11.0	<input checked="" type="checkbox"/> Livy 0.8.0	<input checked="" type="checkbox"/> Hadoop 3.4.0	<input checked="" type="checkbox"/> JupyterEnterpriseGateway 2.6.0	<input checked="" type="checkbox"/> Phoenix 5.2.0	<input checked="" type="checkbox"/> Spark 3.5.2	
<input type="checkbox"/> Pig 0.17.0	<input type="checkbox"/> Tez 0.10.2	<input type="checkbox"/> Oozie 5.2.1	<input type="checkbox"/> Presto 0.287	<input checked="" type="checkbox"/> Spark 3.5.2	<input type="checkbox"/> Trino 446	
<input type="checkbox"/> TensorFlow 2.16.1	<input type="checkbox"/> Zeppelin 0.11.1	<input type="checkbox"/> ZooKeeper 3.9.2				

Summary [Info](#)

Name and applications - required

Name
Mycluster_Assignment12

Amazon EMR release
emr-7.5.0

Application bundle
Spark Interactive (Hadoop 3.4.0, Hive 3.1.3, JupyterEnterpriseGateway 2.6.0, Livy 0.8.0, Spark 3.5...)

Cluster configuration - required

Uniform instance groups
Primary (m4.xlarge), Core (m4.xlarge), Task (m5.xlarge)

Cluster scaling and provisioning - required

Provisioning configuration
Core size: 1 instance

Properties > Mycluster_Assignm... > Mycluster_Assignment12

us-east-2.console.aws.amazon.com/emr/home?region=us-east-2#/clusterDetails/j-2VI42ROGKHUYQ

Your cluster "Mycluster_Assignment12" has been successfully created.

Mycluster_Assignment12

Updated less than a minute ago [Terminate](#) [Clone in AWS CLI](#) [Clone](#)

Summary

Cluster info	Applications	Cluster management	Status and time
Cluster ID j-2VI42ROGKHUYQ	Amazon EMR version emr-7.5.0	Log destination in Amazon S3 aws-logs-145023105604-us-east-2/elasticmapreduce	Status Waiting
Cluster configuration Instance groups	Installed applications Hadoop 3.4.0, Hive 3.1.3, JupyterEnterpriseGateway 2.6.0, Livy 0.8.0, Spark 3.5.2	Persistent application UIs Spark History Server YARN timeline server Tez UI	Creation time November 26, 2024, 17:11 (UTC-05:00)
Capacity 1 Primary 1 Core 0 Task		Primary node public DNS ec2-18-220-246-72.us-east-2.compute.amazonaws.com Connect to the Primary node using SSH Connect to the Primary node using SSM	Elapsed time 6 minutes, 13 seconds

Properties | Bootstrap actions | Instances (Hardware) | Steps | Applications | Configurations | Monitoring | Events | Tags (1)

Operating system [Info](#)

Cluster logs [Info](#)

Cluster termination and node replacement [Info](#)

Open up a terminal connection to your EMR primary node. Over the course of this exercise, you will need to open up three separate terminal connections to your EMR primary node. This is the first, which we will call Cass-Term.

[illegible]

```
wget https://archive.apache.org/dist/cassandra/4.1.4/apache-cassandra-4.1.4-bin.tar.gz
```

```
hadoop@ip-172-31-28-199:~$ wget https://archive.apache.org/dist/cassandra/4.1.4/apache-cassandra-4.1.4-bin.tar.gz
--2024-11-26 22:19:18-- https://archive.apache.org/dist/cassandra/4.1.4/apache-cassandra-4.1.4-bin.tar.gz
Resolving archive.apache.org (archive.apache.org)... 65.108.204.189, 2a01:4f9:1a:a084::2
Connecting to archive.apache.org (archive.apache.org)|65.108.204.189|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 50342501 (48M) [application/x-gzip]
Saving to: 'apache-cassandra-4.1.4-bin.tar.gz'

apache-cassandra-4.1. 100%[=====>] 48.01M 16.3MB/s in 2.9s

2024-11-26 22:19:22 (16.3 MB/s) - 'apache-cassandra-4.1.4-bin.tar.gz' saved [50342501/50342501]

hadoop@ip-172-31-28-199:~$ tar -xzvf apache-cassandra-4.1.4-bin.tar.gz
apache-cassandra-4.1.4/bin/
apache-cassandra-4.1.4/conf/
apache-cassandra-4.1.4/conf/triggers/
apache-cassandra-4.1.4/doc/
apache-cassandra-4.1.4/doc/cql3/
apache-cassandra-4.1.4/lib/
apache-cassandra-4.1.4/lib/sigar-bin/
apache-cassandra-4.1.4/pylib/
apache-cassandra-4.1.4/pylib/cqlshlib/
apache-cassandra-4.1.4/pylib/cqlshlib/test/
apache-cassandra-4.1.4/pylib/cqlshlib/test/test_authproviderhandling_config/
apache-cassandra-4.1.4/tools/
apache-cassandra-4.1.4/tools/bin/
```

```
hadoop@ip-172-31-28-199:~
apache-cassandra-4.1.4/tools/cqlstress-lwt-example.yaml
apache-cassandra-4.1.4/tools/lib/fqltool.jar
apache-cassandra-4.1.4/tools/lib/stress.jar
apache-cassandra-4.1.4/bin/cassandra.in.sh
apache-cassandra-4.1.4/tools/bin/cassandra.in.sh
apache-cassandra-4.1.4/bin/cassandra
apache-cassandra-4.1.4/bin/cqlsh
apache-cassandra-4.1.4/bin/cqlsh.py
apache-cassandra-4.1.4/bin/debug-cql
apache-cassandra-4.1.4/bin/nodetool
apache-cassandra-4.1.4/bin/sstableloader
apache-cassandra-4.1.4/bin/sstablescrub
apache-cassandra-4.1.4/bin/sstableupgrade
apache-cassandra-4.1.4/bin/sstableutil
apache-cassandra-4.1.4/bin/sstableverify
apache-cassandra-4.1.4/bin/stop-server
apache-cassandra-4.1.4/tools/bin/auditlogviewer
apache-cassandra-4.1.4/tools/bin/cassandra-stress
apache-cassandra-4.1.4/tools/bin/compaction-stress
apache-cassandra-4.1.4/tools/bin/fqltool
apache-cassandra-4.1.4/tools/bin/generatetokens
apache-cassandra-4.1.4/tools/bin/hash_password
apache-cassandra-4.1.4/tools/bin/jmxtool
apache-cassandra-4.1.4/tools/bin/sstabledump
apache-cassandra-4.1.4/tools/bin/sstableexpiredblockers
apache-cassandra-4.1.4/tools/bin/sstablelevelreset
apache-cassandra-4.1.4/tools/bin/sstablemetadata
apache-cassandra-4.1.4/tools/bin/sstableofflinerelevel
apache-cassandra-4.1.4/tools/bin/sstablerepairedset
apache-cassandra-4.1.4/tools/bin/sstablesplit
[hadoop@ip-172-31-28-199 ~]$
```

Then enter this command to start Cassandra (lots of diagnostic messages will appear):

apache-cassandra-4.1.4/bin/cassandra &

```
hadoop@ip-172-31-28-199:~
apache-cassandra-4.1.4/tools/bin/sstablesplit
[hadoop@ip-172-31-28-199 ~]$ apache-cassandra-4.1.4/bin/cassandra &
[1] 18138
[hadoop@ip-172-31-28-199 ~]$ CompilerOracle: dontinline org/apache/cassandra/db/Column
s$Serializer.deserializeLargeSubset (Lorg/apache/cassandra/io/util/DataInputPlus;Lor
g/apache/cassandra/db/Columns;I)Lorg/apache/cassandra/db/Columns;
CompilerOracle: dontinline org/apache/cassandra/db/Columns$Serializer.serializeLargeS
ubset (Ljava/util/Collection;ILorg/apache/cassandra/db/Columns;ILorg/apache/cassandra
/io/util/DataOutputPlus;)V
CompilerOracle: dontinline org/apache/cassandra/db/Columns$Serializer.serializeLargeS
ubsetSize (Ljava/util/Collection;ILorg/apache/cassandra/db/Columns;I)I
CompilerOracle: dontinline org/apache/cassandra/db/commitlog/AbstractCommitLogSegment
Manager.advanceAllocatingFrom (Lorg/apache/cassandra/db/commitlog/CommitLogSegment;)V
CompilerOracle: dontinline org/apache/cassandra/db/transform/BaseIterator.tryGetMoreC
ontents ()Z
CompilerOracle: dontinline org/apache/cassandra/db/transform/StoppingTransformation.s
top ()V
CompilerOracle: dontinline org/apache/cassandra/db/transform/StoppingTransformation.s
topInPartition ()V
CompilerOracle: dontinline org/apache/cassandra/io/util/BufferedDataOutputStreamPlus.
doFlush (I)V
CompilerOracle: dontinline org/apache/cassandra/io/util/BufferedDataOutputStreamPlus.
writeSlow (JI)V
CompilerOracle: dontinline org/apache/cassandra/io/util/RebufferingInputStream.readPr
imitiveSlowly (I)J
CompilerOracle: exclude org/apache/cassandra/utils/JVMStabilityInspector.forceHeapSpa
ceOomMaybe (Ljava/lang/OutOfMemoryError;)V
CompilerOracle: inline org/apache/cassandra/db/rows/UnfilteredSerializer.serializeRow
Body (Lorg/apache/cassandra/db/rows/Row;ILorg/apache/cassandra/db/rows/SerializationH
elper;Lorg/apache/cassandra/io/util/DataOutputPlus;)V
CompilerOracle: inline org/apache/cassandra/io/util/Memory.checkBounds (JJ)V
CompilerOracle: inline org/apache/cassandra/io/util/SafeMemory.checkBounds (JJ)V
CompilerOracle: inline org/apache/cassandra/io/util/TrackedDataInputPlus.checkCanRead
(I)V
CompilerOracle: inline org/apache/cassandra/net/FrameDecoderWith8bHeader.decode (Ljav
a/util/Collection;Lorg/apache/cassandra/net/ShareableBytes;I)V
CompilerOracle: inline org/apache/cassandra/service/reads/repair/RowIteratorMergeList
ener.applyToPartition (ILjava/util/function/Consumer;)V
CompilerOracle: inline org/apache/cassandra/utils/AsymmetricOrdering.selectBoundary (
Lorg/apache/cassandra/utils/AsymmetricOrdering;Op;II)I
CompilerOracle: inline org/apache/cassandra/utils/AsymmetricOrdering.strictnessOfLess
Than (Lorg/apache/cassandra/utils/AsymmetricOrdering;Op;)I
```

Step C – Run the Cassandra interactive command line interface

Open a second terminal connection to the EMR primary node. Going forward we will call this terminal connection: Cqlsh-Term. Note, if you are using the git bash shell on your PC, open a new terminal window by right clicking on the title bar of the program and select ‘New Window’ or enter Alt+F2.

- **Cqlsh-Term:**

[illegible]

Enter the following into this terminal to start the command line interface `csqlsh`:

apache-cassandra-4.1.4/bin/cqlsh

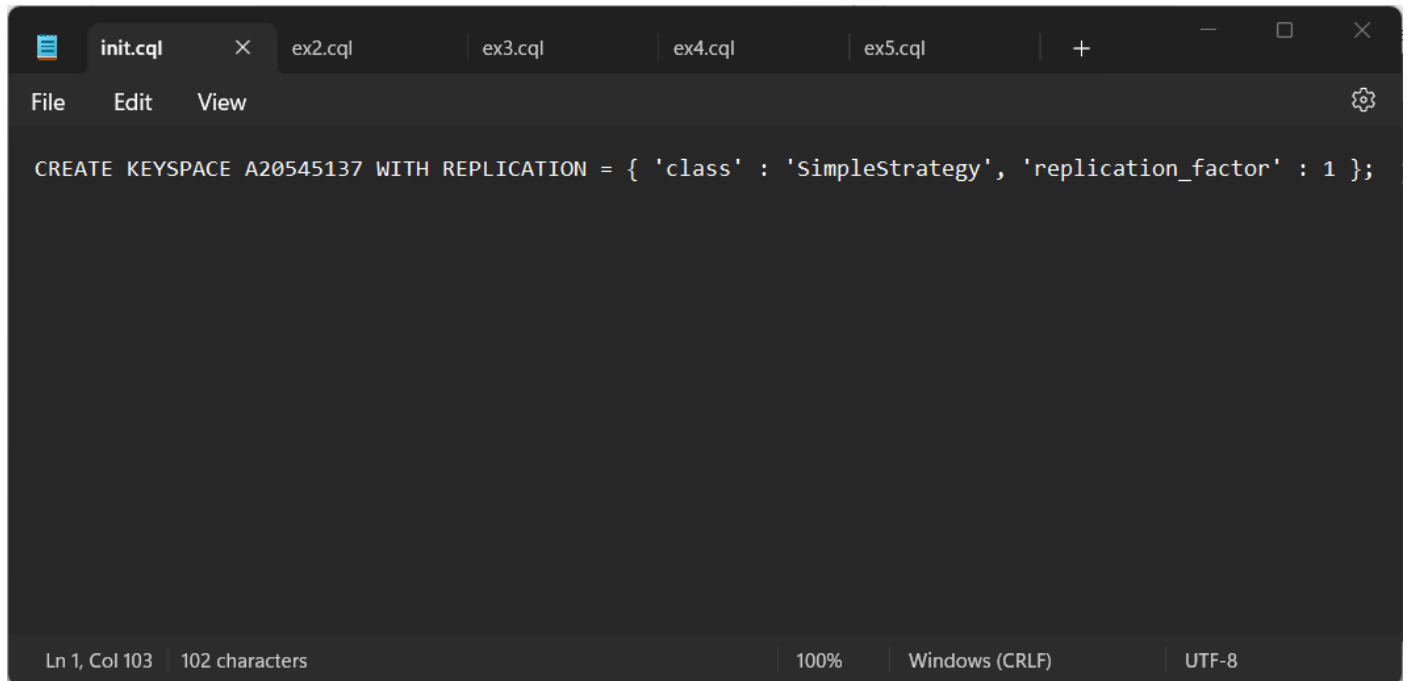
```
hadoop@ip-172-31-28-199:~  
~\#####\  
~\###|  
~\#/  
~V~' ->  
~. -  
~/_/m/' -  
Last login: Tue Nov 26 22:19:51 2024  
  
EEEEEEEEEEEEEEEEEEEE MMMMMMMM MMMMMMMM RRRRRRRRRRRRRRRR  
E::::::::::::::::::E M::::::::M M::::::::M R:::::::::::::R  
EE::::::::::::::::::E M::::::::M M::::::::M R::::::::RRRRRR:::::R  
E:::::E EEEEE M::::::::M M::::::::M RR::::R R:::::R  
E:::::E M::::::::M M::M M::M M::M R::R R:::R  
E:::::EEEEEEEEEE M::::M M::M M::M M::M R::RRRRRR:::::R  
E::::::::::::::::::E M::::M M::M M::M M::M R::::::::RRR  
E:::::EEEEEEEEEE M::::M M::M M::M M::M R::RRRRRR:::::R  
E:::::E M::::M M::M M::M M::M R::R R:::R  
E:::::E EEEEE M::::M MMM M::::M R::R R:::R  
EE:::::EEEEEEEE::::E M::::M M::M M::M R::R R:::R  
E::::::::::::::::::E M::::M M::M M::M RR::::R R:::::R  
EEEEEEEEEEEEEEEEEEEE MMMMMMMM MMMMMMMM RRRRRRR RRRRRR  
  
[hadoop@ip-172-31-28-199 ~]$ apache-cassandra-4.1.4/bin/cqlsh  
Connected to Test Cluster at 127.0.0.1:9042  
[cqlsh 6.1.0 | Cassandra 4.1.4 | CQL spec 3.4.6 | Native protocol v5]  
Use HELP for help.  
cqlsh>
```

Step D – Prepare to edit your Cassandra code

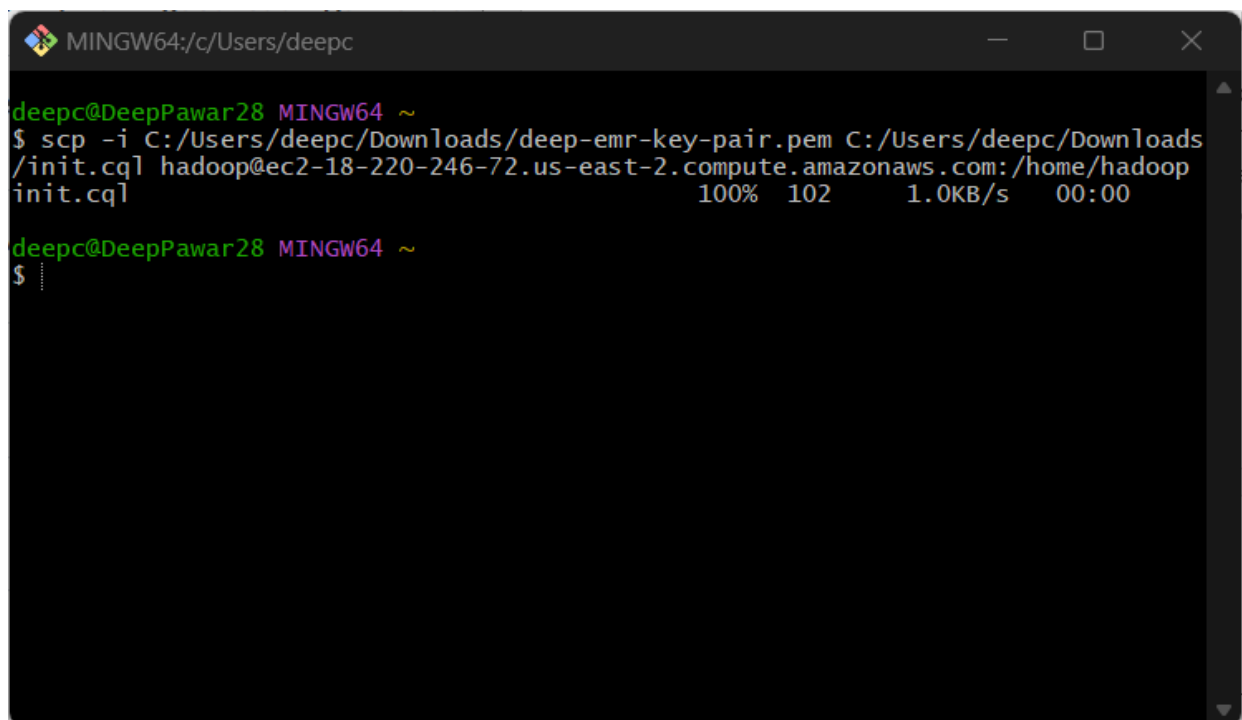
- a) Create a file in your working (home) directory on the primary EMR node called `init.cql` using your Edit-term (or using your PC/MAC and then scp it to the EMR primary node) and enter the following command. Use your IIT id as the name of your keyspace... For example, if your id is A1234567, then replace <IIT id> below with that value:

```
CREATE KEYSPACE <IIT id> WITH REPLICATION = { 'class' : 'SimpleStrategy', 'replication_factor' : 1 };
```

- **init.cql**

A screenshot of a code editor window with a dark theme. The title bar shows several tabs: 'init.cql' (active), 'ex2.cql', 'ex3.cql', 'ex4.cql', and 'ex5.cql'. The 'init.cql' tab is selected. The editor contains the following text:

```
CREATE KEYSPACE A20545137 WITH REPLICATION = { 'class' : 'SimpleStrategy', 'replication_factor' : 1 };
```

 The status bar at the bottom indicates 'Ln 1, Col 103 | 102 characters', '100%', 'Windows (CRLF)', and 'UTF-8'.A screenshot of a terminal window titled 'MINGW64:/c/Users/deepc'. The prompt is 'deepc@DeepPawar28 MINGW64 ~'. The user enters the command:

```
$ scp -i C:/Users/deepc/Downloads/deep-emr-key-pair.pem C:/Users/deepc/Downloads/init.cql hadoop@ec2-18-220-246-72.us-east-2.compute.amazonaws.com:/home/hadoop
```

 The output shows the file being transferred:

```
init.cql 100% 102 1.0KB/s 00:00
```

 The prompt returns to '\$'.

b) Then execute this file in the CQL shell using the Cqlsh-Term as follows...

source './init.cql';

```
hadoop@ip-172-31-28-199:~$
EEEEEEEEEEEEEEEEEEEE MMMMMMMM MMMMMMMM RRRRRRRRRRRRRRRR
E::::::::::::::::::::E M::::::::M M::::::::M R::::::::::::R
EE::::::::EEEEEEEEEE M::::::::M M::::::::M R::::RRRRRR::::R
E::::E EEEEE M::::::::M M::::::::M RR::::R R::::R
E::::E M::::::::M M::M M::M M::M R::R R::R
E::::EEEEEEEEEE M::M M::M M::M M::M R::RRRRRR::::R
E::::::::::::E M::M M::M M::M M::M R::::::::RR
E::::EEEEEEEEEE M::M M::M M::M M::M R::RRRRRR::::R
E::::E M::M M::M M::M M::M R::R R::R
E::::E EEEEE M::M M M M::M M::M R::R R::R
EE::::::::EEEEEEEEEE M::M M::M M::M R::R R::R
E::::::::::::E M::M M::M RR::R R::R
EEEEEEEEEEEEEEEEEEEE MMMMMMMM MMMMMMMM RRRRRRR RRRRRR

[hadoop@ip-172-31-28-199 ~]$ apache-cassandra-4.1.4/bin/cqlsh
Connected to Test Cluster at 127.0.0.1:9042
[cqlsh 6.1.0 | Cassandra 4.1.4 | CQL spec 3.4.6 | Native protocol v5]
Use HELP for help.
cqlsh> source './init.cql';
```

c) To check if your script file has created a keyspace execute the following in the CQL shell:

describe keyspaces;

```
hadoop@ip-172-31-28-199:~$
EEEEEEEEEEEEEEEEEEEE MMMMMMMM MMMMMMMM RRRRRRRRRRRRRRRR
E::::::::::::::::::::E M::::::::M M::::::::M R::::::::::::R
EE::::::::EEEEEEEEEE M::::::::M M::::::::M R::::RRRRRR::::R
E::::E EEEEE M::::::::M M::::::::M RR::::R R::::R
E::::E M::::::::M M::M M::M M::M R::R R::R
E::::EEEEEEEEEE M::M M::M M::M M::M R::RRRRRR::::R
E::::::::::::E M::M M::M M::M M::M R::::::::RR
E::::EEEEEEEEEE M::M M::M M::M M::M R::RRRRRR::::R
E::::E M::M M::M M::M M::M R::R R::R
E::::E EEEEE M::M M M M::M M::M R::R R::R
EE::::::::EEEEEEEEEE M::M M::M M::M R::R R::R
E::::::::::::E M::M M::M RR::R R::R
EEEEEEEEEEEEEEEEEEEE MMMMMMMM MMMMMMMM RRRRRRR RRRRRR

[hadoop@ip-172-31-28-199 ~]$ apache-cassandra-4.1.4/bin/cqlsh
Connected to Test Cluster at 127.0.0.1:9042
[cqlsh 6.1.0 | Cassandra 4.1.4 | CQL spec 3.4.6 | Native protocol v5]
Use HELP for help.
cqlsh> source './init.cql';
Invalid syntax at char 8
source './init.cql';
^
cqlsh> source './init.cql';
cqlsh> describe keyspaces;

a20545137 system_auth system_schema system_views
system system_distributed system_traces system_virtual_schema

cqlsh>
```

- d) At this point you have created a keyspace unique to you. So, make that keyspace the default by entering the following into the CQL shell:

USE <IIT id>;

```
hadoop@ip-172-31-28-199:~
E::::::::::::::::::::E M::::::::M M::::::::M R::::::::::::R
EE::::::::EEEEEEEEEE M::::::::M M::::::::M R::::::::RRRRRR::::R
E::::E EEEEE M::::::::M M::::::::M RR::::R R::::R
E::::E M::::::::M:M M::M:M:M M::::::::M R:::R R::::R
E::::EEEEEEEEEE M::::M M::M M::M M::::::::M R::RRRRRR::::R
E::::::::::::E M::::M M::M:M:M M::::::::M R:::::::::RR
E::::EEEEEEEEEE M::::M M::M:M M::::::::M R::RRRRRR::::R
E::::E M::::M M::M M::::::::M R:::R R::::R
E::::E EEEEE M::::M MMM M::::::::M R:::R R::::R
EE::::::::EEEEEEEEEE M::::M M::::::::M R:::R R::::R
E::::::::::::E M::::M M::::::::M RR::::R R::::R
EEEEEEEEEEEEEEEEEEEE MMMMMM MMMMMMM RRRRRRR RRRRRR

[hadoop@ip-172-31-28-199 ~]$ apache-cassandra-4.1.4/bin/cqlsh
Connected to Test Cluster at 127.0.0.1:9042
[cqlsh 6.1.0 | Cassandra 4.1.4 | CQL spec 3.4.6 | Native protocol v5]
Use HELP for help.
cqlsh> source './init.cql';
Invalid syntax at char 8
source './init.cql';
^
cqlsh> source './init.cql';
cqlsh> describe keyspaces;

a20545137 system_auth system_schema system_views
system system_distributed system_traces system_virtual_schema

cqlsh> USE A20545137;
cqlsh:a20545137>
```

Now create a file in your working directory called **ex2.cql** using the Edit-Term (or PC/MAC and scp). In this file write the command to create a table named 'Music' with the following characteristics:

Attribute Name	Attribute Type	Primary Key / Cluster Key
artistName	text	Primary Key
albumName	text	Cluster Key
numberSold	int	Non Key Column
cost	int	Non Key Column

- **ex2.cql**

```
init.cql ex2.cql ex3.cql ex4.cql ex5.cql
File Edit View
CREATE TABLE Music (
  artistName text,
  albumName text,
  numberSold int,
  Cost int,
  PRIMARY KEY (artistName, albumName)
);

Ln 7, Col 3 123 characters 100% Windows (CRLF) UTF-8
```



```
MINGW64/c/Users/deepc

deepc@DeepPawar28 MINGW64 ~
$ scp -i C:/Users/deepc/Downloads/deep-emr-key-pair.pem C:/Users/deepc/Downloads/init.cql hadoop@ec2-18-220-246-72.us-east-2.compute.amazonaws.com:/home/hadoop
init.cql 100% 102 1.0KB/s 00:00

deepc@DeepPawar28 MINGW64 ~
$ scp -i C:/Users/deepc/Downloads/deep-emr-key-pair.pem C:/Users/deepc/Downloads/ex2.cql hadoop@ec2-18-220-246-72.us-east-2.compute.amazonaws.com:/home/hadoop
ex2.cql 100% 129 1.2KB/s 00:00

deepc@DeepPawar28 MINGW64 ~
$
```

Execute **ex2.cql** in the CQL shell. Then execute the shell command **‘DESCRIBE TABLE Music’**.

```
hadoop@ip-172-31-28-199:~
cqlsh:a20545137> source './ex2.cql';
cqlsh:a20545137> DESCRIBE TABLE Music;

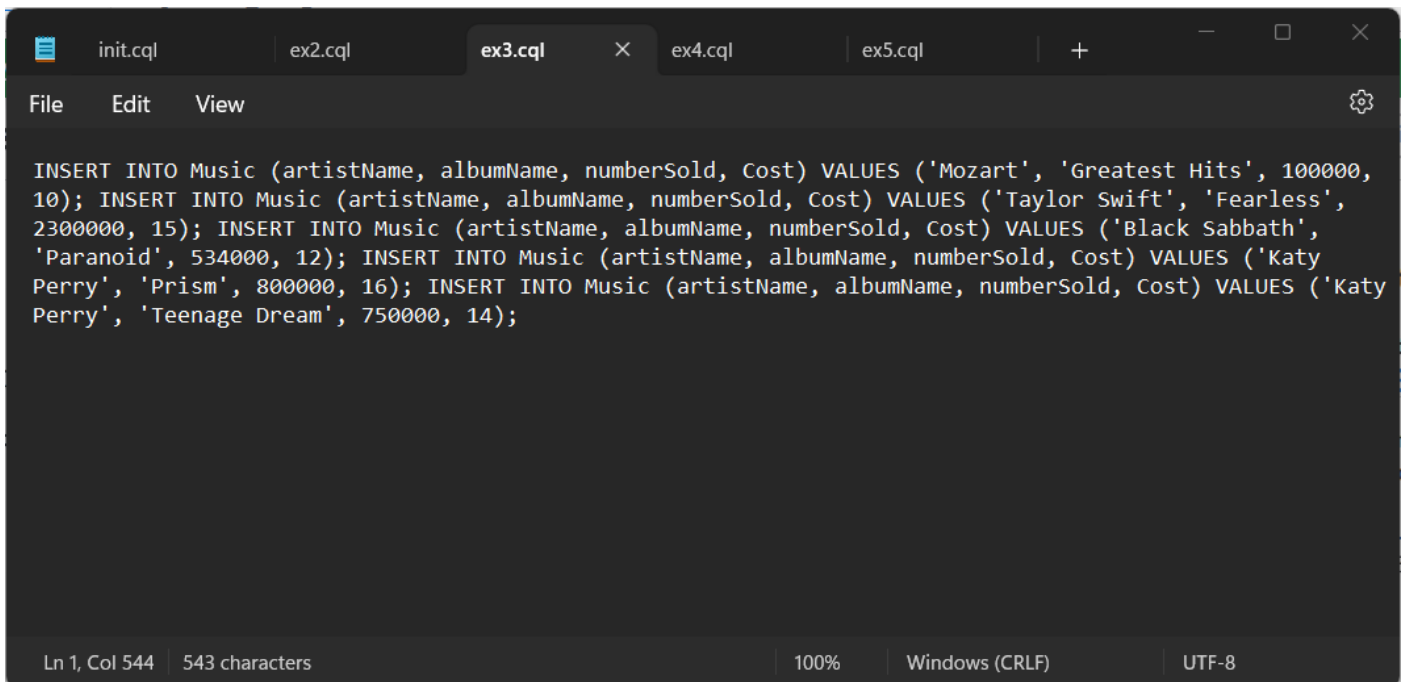
CREATE TABLE a20545137.music (
  artistname text,
  albumname text,
  cost int,
  numbersold int,
  PRIMARY KEY (artistname, albumname)
) WITH CLUSTERING ORDER BY (albumname ASC)
   AND additional_write_policy = '99p'
   AND bloom_filter_fp_chance = 0.01
   AND caching = {'keys': 'ALL', 'rows_per_partition': 'NONE'}
   AND cdc = false
   AND comment = ''
   AND compaction = {'class': 'org.apache.cassandra.db.compaction.SizeTieredCompactionStrategy',
'max_threshold': '32', 'min_threshold': '4'}
   AND compression = {'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
   AND memtable = 'default'
   AND crc_check_chance = 1.0
   AND default_time_to_live = 0
   AND extensions = {}
   AND gc_grace_seconds = 864000
   AND max_index_interval = 2048
   AND memtable_flush_period_in_ms = 0
   AND min_index_interval = 128
   AND read_repair = 'BLOCKING'
   AND speculative_retry = '99p';
cqlsh:a20545137>
```

Exercise 3) (3 points)

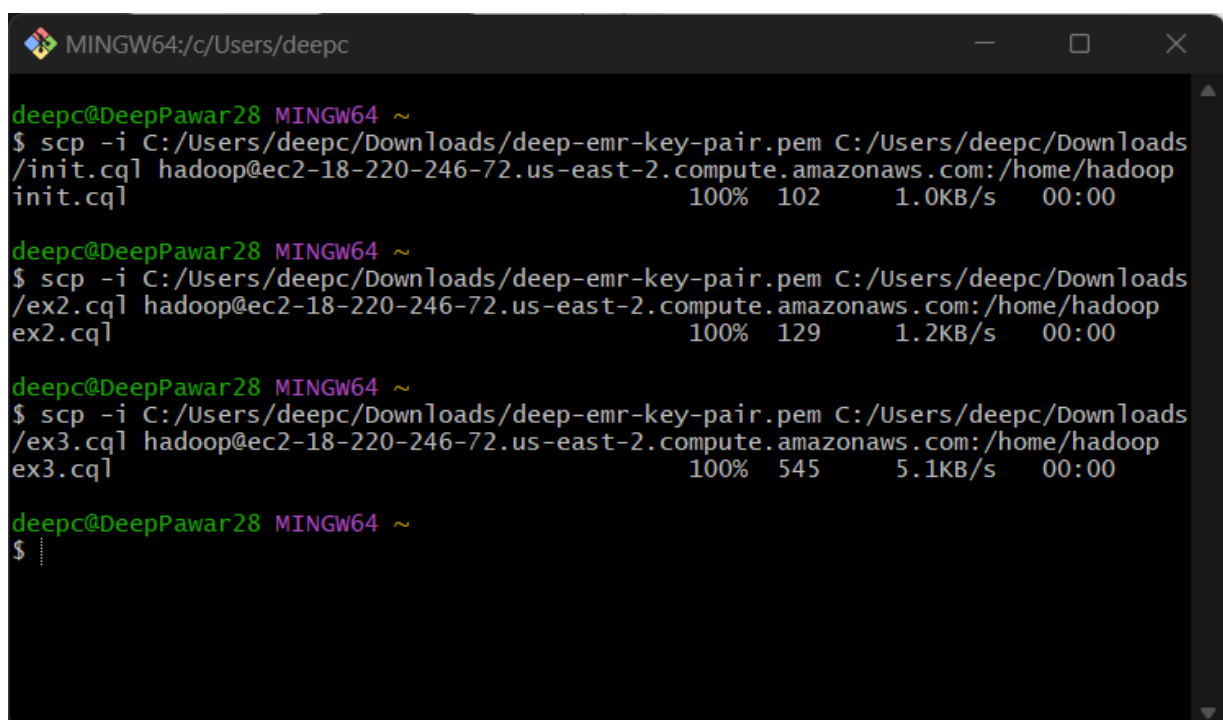
Now create a file in your working directory called **ex3.cql** using the Edit-Term. In this file write the commands to insert the following records into table 'Music'

artistName	albumName	numberSold	cost
Mozart	Greatest Hits	100000	10
Taylor Swift	Fearless	2300000	15
Black Sabbath	Paranoid	534000	12
Katy Perry	Prism	800000	16
Katy Perry	Teenage Dream	750000	14

- **ex3.cql**



```
INSERT INTO Music (artistName, albumName, numberSold, Cost) VALUES ('Mozart', 'Greatest Hits', 100000, 10); INSERT INTO Music (artistName, albumName, numberSold, Cost) VALUES ('Taylor Swift', 'Fearless', 2300000, 15); INSERT INTO Music (artistName, albumName, numberSold, Cost) VALUES ('Black Sabbath', 'Paranoid', 534000, 12); INSERT INTO Music (artistName, albumName, numberSold, Cost) VALUES ('Katy Perry', 'Prism', 800000, 16); INSERT INTO Music (artistName, albumName, numberSold, Cost) VALUES ('Katy Perry', 'Teenage Dream', 750000, 14);
```



```
deepc@DeepPawar28 MINGW64 ~
$ scp -i C:/Users/deepc/Downloads/deep-emr-key-pair.pem C:/Users/deepc/Downloads/init.cql hadoop@ec2-18-220-246-72.us-east-2.compute.amazonaws.com:/home/hadoop
100% 102 1.0KB/s 00:00

deepc@DeepPawar28 MINGW64 ~
$ scp -i C:/Users/deepc/Downloads/deep-emr-key-pair.pem C:/Users/deepc/Downloads/ex2.cql hadoop@ec2-18-220-246-72.us-east-2.compute.amazonaws.com:/home/hadoop
100% 129 1.2KB/s 00:00

deepc@DeepPawar28 MINGW64 ~
$ scp -i C:/Users/deepc/Downloads/deep-emr-key-pair.pem C:/Users/deepc/Downloads/ex3.cql hadoop@ec2-18-220-246-72.us-east-2.compute.amazonaws.com:/home/hadoop
100% 545 5.1KB/s 00:00

deepc@DeepPawar28 MINGW64 ~
$ :
```

Execute **ex3.cql**.

Provide (a) the content of the **ex3.cql** file as one result of this exercise, and (b) execute the command 'SELECT * FROM Music;' and provide a screenshot of the output of this command as another result of the exercise.

```
hadoop@ip-172-31-28-199:~
cqlsh:a20545137> source './ex3.cql';
cqlsh:a20545137> SELECT * FROM Music;
```

artistname	albumname	cost	numbersold
Mozart	Greatest Hits	10	100000
Black Sabbath	Paranoid	12	534000
Taylor Swift	Fearless	15	2300000
Katy Perry	Prism	16	800000
Katy Perry	Teenage Dream	14	750000

```
(5 rows)
cqlsh:a20545137>
```

Exercise 4) (2 points)

Now create a file in your working directory called **ex4.cql** using the Edit-Term. In this file write the commands to query and output only Katy Perry songs.

- **ex4.cql**

```
init.cql  ex2.cql  ex3.cql  ex4.cql  ex5.cql
File Edit View
```

```
SELECT * FROM Music WHERE artistName = 'Katy Perry';
```

Ln 1, Col 53 | 52 characters | 100% | Windows (CRLF) | UTF-8

```
MINGW64:/c/Users/deepc
deepc@DeepPawar28 MINGW64 ~
$ scp -i C:/Users/deepc/Downloads/deep-emr-key-pair.pem C:/Users/deepc/Downloads/ex3.cql hadoop@ec2-18-220-246-72.us-east-2.compute.amazonaws.com:/home/hadoop
ex3.cql 100% 543 5.2KB/s 00:00
deepc@DeepPawar28 MINGW64 ~
$ scp -i C:/Users/deepc/Downloads/deep-emr-key-pair.pem C:/Users/deepc/Downloads/ex4.cql hadoop@ec2-18-220-246-72.us-east-2.compute.amazonaws.com:/home/hadoop
ex4.cql 100% 52 0.5KB/s 00:00
deepc@DeepPawar28 MINGW64 ~
$
```

Execute **ex4.cql**.

Provide (a) the content of the **ex4.cql** file and (b) a screenshot of the output of executing this file as the result of this exercise.

```
hadoop@ip-172-31-28-199:~
cqlsh:a20545137> source './ex4.cql';

artistname | albumname | cost | numbersold
-----+-----+-----+-----
Katy Perry | Prism    | 16   | 800000
Katy Perry | Teenage Dream | 14   | 750000

(2 rows)
cqlsh:a20545137>
```

Exercise 5) (2 points)

Now create a file in your working directory called **ex5.cql** using the Edit-Term. In this file write the commands to query only albums that have sold 700000 copies or more.

- **ex5.cql**

```
init.cql  ex2.cql  ex3.cql  ex4.cql  ex5.cql  X  +
File Edit View
SELECT * FROM Music WHERE numberSold >= 700000 ALLOW FILTERING;

Ln 1, Col 17  63 characters  100%  Windows (CRLF)  UTF-8
```

```
MINGW64:/c/Users/deepc
deepc@DeepPawar28 MINGW64 ~
$ scp -i C:/Users/deepc/Downloads/deep-emr-key-pair.pem C:/Users/deepc/Downloads/ex4.cql hadoop@ec2-18-220-246-72.us-east-2.compute.amazonaws.com:/home/hadoop/ex4.cql
100% 52 0.5KB/s 00:00

deepc@DeepPawar28 MINGW64 ~
$ scp -i C:/Users/deepc/Downloads/deep-emr-key-pair.pem C:/Users/deepc/Downloads/ex5.cql hadoop@ec2-18-220-246-72.us-east-2.compute.amazonaws.com:/home/hadoop/ex5.cql
100% 63 0.6KB/s 00:00

deepc@DeepPawar28 MINGW64 ~
$
```

Execute **ex5.cql**.

Provide (a) the content of the **ex5.cql** file and (b) a screenshot of the output of executing this file as the result of this exercise.

```
hadoop@ip-172-31-28-199:~
cqlsh:a20545137> source './ex5.cql';

artistname | albumname | cost | numbersold
-----
Taylor Swift | Fearless | 15 | 2300000
Katy Perry | Prism | 16 | 800000
Katy Perry | Teenage Dream | 14 | 750000

(3 rows)
cqlsh:a20545137>
```

Remember to terminate your EMR cluster when you complete this assignment.

The screenshot shows the AWS Management Console for an Amazon EMR cluster named "Mycluster_Assignment12". The cluster is in a "Terminated" state. The console displays various details including cluster ID, configuration, applications, and status.

Summary

- Cluster info**
 - Cluster ID: j-2VI42ROGKHUYQ
 - Cluster configuration: Instance groups
 - Capacity: 1 Primary | 1 Core | 0 Task
- Applications**
 - Amazon EMR version: emr-7.5.0
 - Installed applications: Hadoop 3.4.0, Hive 3.1.3, JupyterEnterpriseGateway 2.6.0, Livy 0.8.0, Spark 3.5.2
- Cluster management**
 - Log destination in Amazon S3: aws-logs-145023105604-us-east-2/elasticmapreduce
 - Persistent application UIs: Spark History Server, YARN timeline server, Tez UI
 - Primary node public DNS: ec2-18-220-246-72.us-east-2.compute.amazonaws.com
 - Connect to the Primary node using SSH
- Status and time**
 - Status: Terminated
 - Creation time: November 26, 2024, 17:11 (UTC-05:00)
 - Elapsed time: 55 minutes, 47 seconds
 - End time: November 26, 2024, 18:07 (UTC-05:00)

Properties | Bootstrap actions | Instances (Hardware) | Steps | Applications | Configurations | Monitoring | Events | Tags (1)

Operating system [Info](#)
Amazon Linux release 2023.6.20241031.0

Cluster logs [Info](#)
Archive log files to Amazon S3
Turned on

Cluster termination and node replacement [Info](#)

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences