Tutorial Work 3
Apache Cassandra

The main objectives of this tutorial are:

- To be able to do CRUD operations using Apache Cassandra.
- To be familiar with how Cassandra processes user queries.
- To be able to use secondary index.
- To be able to use collections in Apache Cassandra.
- To be able to user User-Defined Type (UDT).

Important Reminder:

- The unit consists of the Tutorial Work assessment (weighing 5%) conducted during week 4, 5, 7, 9, 10.
- This tutorial work (weighing 1% out of 5%) is the first among the assessed tutorials works. It will be assessed during your tutorial time.

Marking Rubric:

Mark	Description		
	Level of understanding	Preparation	Tutorial Task and Interview Questions
1	Exceptional	Student has prepared prior to the tutorial	Completed all of the tutorial tasks and has answered all interview questions without mistakes/errors
0.75	Competent	Student has prepared prior to the tutorial	either completed all of the tutorial tasks and has answered at least 50% of the interview questions without mistakes/errors or: completed at least 50% of the tutorial tasks and has answered all interview questions without mistakes/errors
0.5	Moderate	Student has prepared prior to the tutorial	either: completed more than 50% of the tutorial tasks and answered less than 50% of the interview questions without mistakes/errors or: completed less than 50% of the tutorial tasks and answered more than 50% of the interview questions without mistakes/errors or: completed 50% of the tutorial tasks and answered 50% of the interview questions without mistakes/errors
0.25	Inadequate	Student has prepared prior to the tutorial	completed less than 50% of the tutorial tasks and answered less than 50% of the interview questions without mistakes/errors
0	No understanding	Student did not prepare prior to the tutorial	did not complete any of the tutorial tasks and did not answer any interview questions without mistakes/errors

File Template for Answer

A file template (`Week7_CassandraExercise.cql`) is provided for you on Moodle (under the Week 7 resources) to answer the tutorial work. You can use any word editor application (e.g. Atom, Notepad++) to edit your answer in the provided file template.

Connecting to Cassandra on Windows

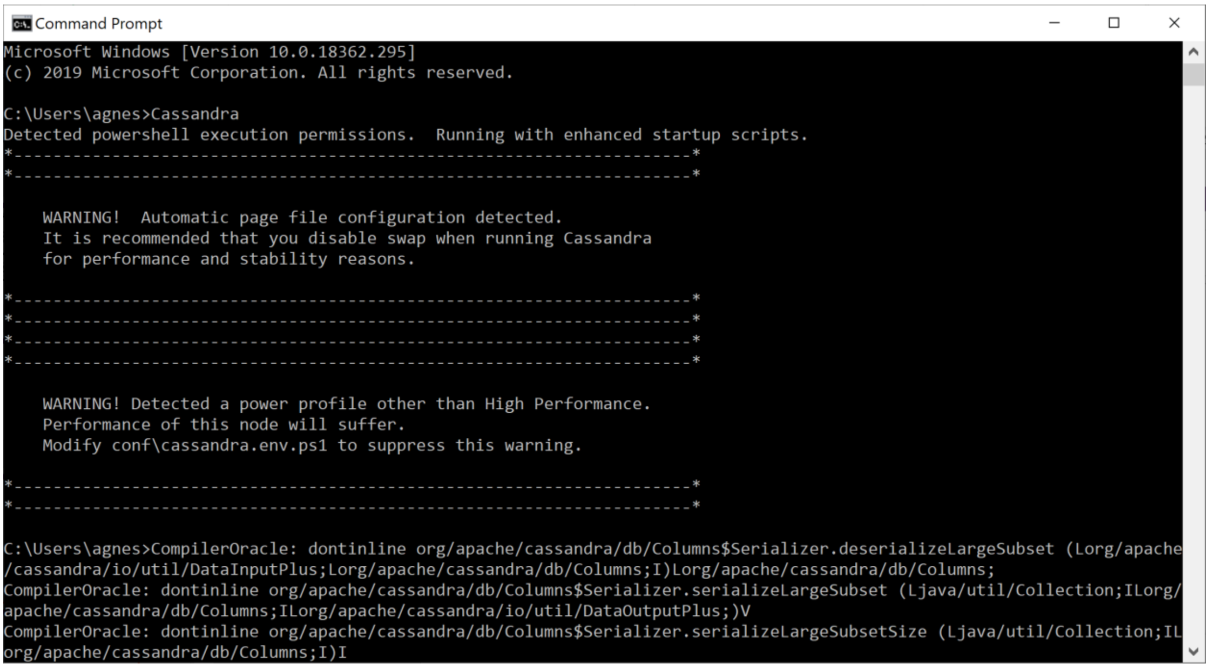
1. Press the **Windows Key** → type **Run** in the search bar → **Enter**.

2. Type **cmd** to open the **Command Prompt** → **Enter**.

3. Go to **apache-cassandra-3.11.7\bin** and copy the directory.

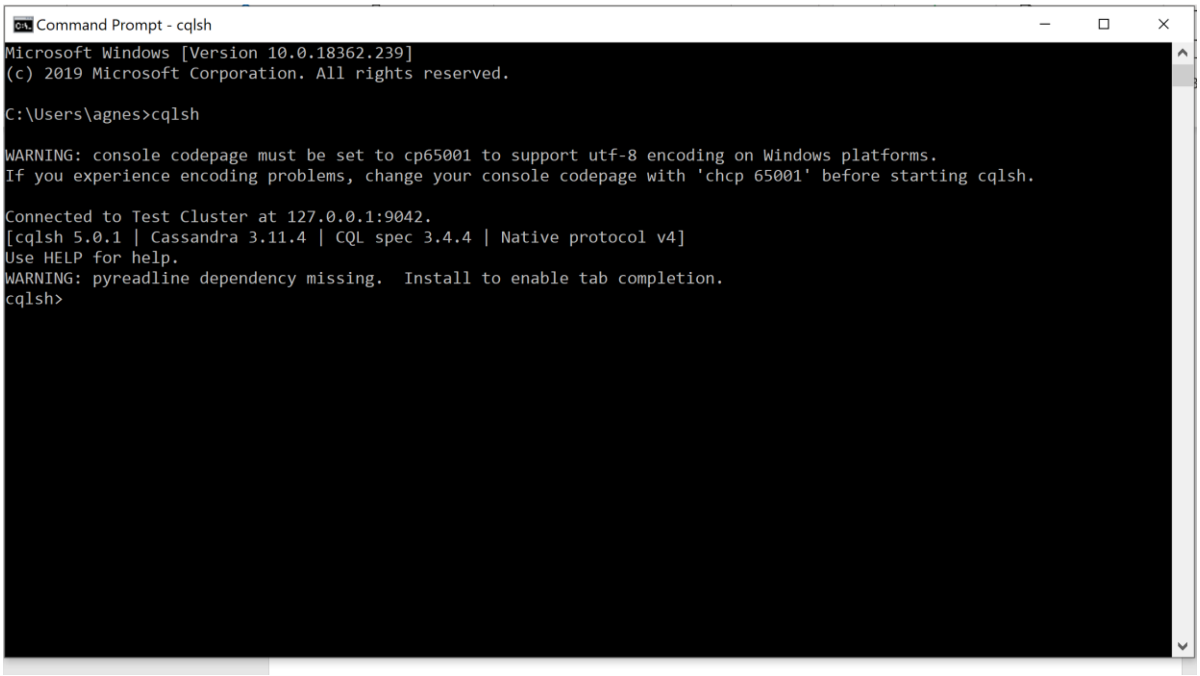
4. Type "**cd [directory]**".

5. Type **Cassandra** to start Cassandra server. The Command Prompt will load Cassandra. Wait until it stops loading.



```
Microsoft Windows [Version 10.0.18362.295]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\agnes>Cassandra
Detected powershell execution permissions.  Running with enhanced startup scripts.
*---------------------------------------------------------------------*
*---------------------------------------------------------------------*

    WARNING!  Automatic page file configuration detected.
    It is recommended that you disable swap when running Cassandra
    for performance and stability reasons.

*---------------------------------------------------------------------*
*---------------------------------------------------------------------*
*---------------------------------------------------------------------*
*---------------------------------------------------------------------*

    WARNING! Detected a power profile other than High Performance.
    Performance of this node will suffer.
    Modify conf\cassandra.env.ps1 to suppress this warning.

*---------------------------------------------------------------------*
*---------------------------------------------------------------------*

C:\Users\agnes>CompilerOracle: dontinline org/apache/cassandra/db/Columns$Serializer.deserializeLargeSubset (Lorg/apache
/cassandra/io/util/DataInputPlus;Lorg/apache/cassandra/db/Columns;I)Lorg/apache/cassandra/db/Columns;
CompilerOracle: dontinline org/apache/cassandra/db/Columns$Serializer.serializeLargeSubset (Ljava/util/Collection;ILorg/
apache/cassandra/db/Columns;ILorg/apache/cassandra/io/util/DataOutputPlus;)V
CompilerOracle: dontinline org/apache/cassandra/db/Columns$Serializer.serializeLargeSubsetSize (Ljava/util/Collection;IL
org/apache/cassandra/db/Columns;I)I
```

6. Open another Command Prompt with the same Cassandra directory, then type **cqlsh**. The figure below indicates that you have successfully load Cassandra.

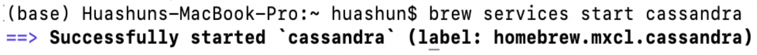


```
Microsoft Windows [Version 10.0.18362.239]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\agnes>cqlsh

WARNING: console codepage must be set to cp65001 to support utf-8 encoding on Windows platforms.
If you experience encoding problems, change your console codepage with 'chcp 65001' before starting cqlsh.

Connected to Test Cluster at 127.0.0.1:9042.
[cqlsh 5.0.1 | Cassandra 3.11.4 | CQL spec 3.4.4 | Native protocol v4]
Use HELP for help.
WARNING: pyreadline dependency missing.  Install to enable tab completion.
cqlsh>
```

Connecting to Cassandra on Mac

1. Start Cassandra.

   $ brew services start cassandra



```
(base) Huashuns-MacBook-Pro:~ huashun$ brew services start cassandra
==> Successfully started `cassandra` (label: homebrew.mxcl.cassandra)
```

2. Go to /usr/local/Cellar/cassandra/3.11.7/bin.

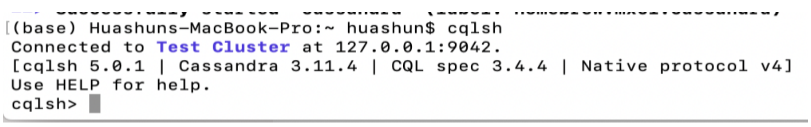
   $ cd /usr/local/Cellar/cassandra/3.11.7/bin

3. Initialize cassandra

   $ cassandra

4. Open another terminal, then start CQL

   $ cqlsh



```
[(base) Huashuns-MacBook-Pro:~ huashun$ cqlsh
Connected to Test Cluster at 127.0.0.1:9042.
[cqlsh 5.0.1 | Cassandra 3.11.4 | CQL spec 3.4.4 | Native protocol v4]
Use HELP for help.
cqlsh>
```

Cassandra

Keyspace

a. Create a new keyspace called `books_keyspace` in one cluster with a total of 3 nodes.

```
CREATE KEYSPACE books_keyspace
WITH      replication      =      {'class':'SimpleStrategy',
'replication_factor':3};
```

b. Switch to the newly created keyspace.

```
USE books_keyspace;
```

c. Alter the keyspace and update it to have a total node of 2 rather than 3.

```
ALTER KEYSPACE books_keyspace WITH replication = {'class':
'SimpleStrategy’, 'replication_factor' : 2};
```

Create Operations

a. Create a new column family called `books_by_author` that consists of the author name, book’s publish year, book ID, book name, and rating. Make the author name, publish year, and book ID as the primary key. You should use UUID for the book ID.

```
CREATE TABLE books_by_author (
  author_name TEXT,
  publish_year INT,
  book_id UUID,
  book_name TEXT,
  rating FLOAT,
  PRIMARY KEY((author_name), publish_year, book_id)
);
```

b. Insert the following data to the column family:

Author Name	Publish Year	Book Name	Rating
James Patterson	2008	Cross	3
Adam Liaw	2016	The Zen Kitchen	4.5
Rob Galea	2018	Breakthrough	4

Use UUID to create the book ID.

```
INSERT INTO books_by_author (author_name, publish_year,
book_id, book_name,
rating)
```

```
VALUES('James Patterson',2008, uuid(),'Cross',3);

INSERT INTO books_by_author (author_name, publish_year,
book_id, book_name,
rating)
VALUES('Adam Liaw',2016, uuid(),'The Zen Kitchen',4.5);

INSERT INTO books_by_author (author_name, publish_year,
book_id, book_name,
rating)
VALUES('Rob Galea',2018, uuid(),'Breakthrough',4);
```

Read Operations

a. Find books that were written by James Patterson before 2009.

```
SELECT * FROM books_by_author where author_name='James
Patterson' and publish_year < 2009;
```

b. Find books with rating of 4.5.

```
SELECT * FROM books_by_author where rating = 4.5;
```

Can you find any books? Write down your observation.

Unable to find books with rating 4.5 since rating is not part of the indexes.

Secondary Index

a. Create a secondary index for rating.

```
CREATE INDEX ON books_by_author ( rating );
```

Can you find books with rating of 4.5 now?

Timestamp

a. Show the timestamp of book name.

```
SELECT author_name, book_name, writetime(book_name) FROM
books_by_author;
```

b. Show the timestamp of author's name.

```
SELECT author_name, book_name,
writetime(author_name) FROM books_by_author;
```

Write down your observation.

Printing out timestamp is only allowed for non-primary key columns.

c. Add a new row with the following details:
Author's name : James Patterson
Publish year : 2017
Book name : Manning
Rating : 4.0

```
INSERT  INTO  books_by_author  (author_name,  publish_year,
book_id, book_name, rating)
VALUES('James Patterson',2017, uuid(),'Manning',4.0);
```

d. Find all books written by James Patterson.

```
SELECT * FROM books_by_author
WHERE author_name= 'James Patterson';
```

Collections – Set

a. Create a new column for emails that can store a set of email values.

```
ALTER TABLE books_by_author
ADD emails set<text>;
```

b. Add James Patterson's email address: `james_patter@gmail.com`. Can you update all records of James Patterson with a single update command?

```
UPDATE books_by_author SET emails =
{'james_patter@gmail.com'}
WHERE author_name= 'James Patterson';
```

Having the above command will give this following error message:

```
InvalidRequest: Error from server: code=2200 [Invalid query]
message="Some  clustering  keys  are  missing:  publish_year,
book_id"
```

This is due to the fact that we have multiple primary keys and Cassandra requires a complete primary key field in order to do the update operation.

c. If you cannot insert the email address from the previous question, insert the email address for each record of James Patterson.

```
UPDATE books_by_author SET emails =
{'james_patter@gmail.com'}
```

```
WHERE author_name= 'James Patterson' and publish_year=2008
and book_ID=<book_ID>;

UPDATE books_by_author SET emails =
{'james_patter@gmail.com'}
WHERE author_name= 'James Patterson' and publish_year=2017
and book_ID=<book_ID>;
```

d. Add an email address for Adam Liaw: `adam_liaw@gmail.com`.

```
UPDATE books_by_author SET emails = {'adam_liaw@gmail.com'}
WHERE  author_name=  'Adam  Liaw'  and  publish_year=2016  and
book_ID=8d57f6e7-e57d-497b-9e69-0649c63dbfd5;
```

e. Add another email address of Adam Liaw: `adamliaw@gmail.com`.

```
UPDATE books_by_author SET emails = emails +
{'adamliaw@gmail.com'}
WHERE  author_name=  'Adam  Liaw'  and  publish_year=2016  and
book_ID=8d57f6e7-e57d-497b-9e69-0649c63dbfd5;
```

f. Delete Adam Liaw's email that we inserted first (`adam_liaw@gmail.com`).

```
UPDATE books_by_author SET emails = emails -
{'adam_liaw@gmail.com'}
WHERE  author_name=  'Adam  Liaw'  and  publish_year=2016  and
book_ID=8d57f6e7-e57d-497b-9e69-0649c63dbfd5;
```

Collections – List

a. Add a new column called series to store the list of other book titles of the series.

```
ALTER TABLE books_by_author
ADD series list<text>;
```

b. Add "Along Came a Spider" as one of the series of James Patterson's book titled Cross.

```
UPDATE books_by_author SET series = ['Along Came a Spider']
WHERE  author_name=  'James  Patterson'  and  publish_year=2008
and book_ID=ba42d525-02ee-4053-89e8-8e01998eb22e;
```

c. Add another series of James Patterson's Cross, which is "Jack and Jill" and "Cat and Mouse".

```
UPDATE books_by_author SET series = series + ['Jack and
Jill', 'Cat and Mouse']
WHERE author_name= 'James Patterson' and publish_year=2008
and book_ID=ba42d525-02ee-4053-89e8-8e01998eb22e;
```

d. Delete “Jack and Jill” from the list.

```
DELETE series[1]
FROM books_by_author
WHERE author_name= 'James Patterson' and publish_year=2008
and book_ID=ba42d525-02ee-4053-89e8-8e01998eb22e;
```

Collections – Map

a. Add a new column called ISBN and store it as a Map collection:

```
ALTER TABLE books_by_author
ADD isbn map<text,text>;
```

b. Add a new column to store this information for Adam Liaw’s The Zen Kitchen:
ISBN-10: 0733634311
ISBN-13: 978-0733634314

```
UPDATE books_by_author SET isbn = { 'isbn-10': '0733634311',
'isbn-13': '978-0733634314'}
WHERE author_name= 'Adam Liaw' and publish_year=2016 and
book_ID=8d57f6e7-e57d-497b-9e69-0649c63dbfd5;
```

c. Delete the ISBN-13 from Adam Liaw’s The Zen Kitchen.

```
UPDATE books_by_author SET isbn = isbn - {'isbn-13'}
WHERE author_name= 'Adam Liaw' and publish_year=2016 and
book_ID=8d57f6e7-e57d-497b-9e69-0649c63dbfd5;
```

User-Defined Types (UDT)

a. Create a UDT to store the product details. In this case, we want to store the number of pages, cover type (hardcover/softcover), book dimension length, width, height, and unit of measurement.

```
CREATE TYPE product_details (
  pages int,
  cover_type text,
  dim_length int,
  dim_width int,
  dim_height int,
  dim_oum text
);
```

b. Add this detail to Adam Liaw's The Zen Kitchen:

Pages	: 240
Cover type	: Hardcover
Length	: 20.8
Width	: 2.6
Height	: 25.6
UOM	: cm

```
ALTER TABLE books_by_author ADD product_details frozen<
product_details>;

UPDATE books_by_author SET product_details = { pages:240,
cover_type:'Hardcover', dim_length:20.8, dim_width:2.6,
dim_height:25.6, dim_uom:'cm' }
WHERE author_name= 'Adam Liaw' and publish_year=2016 and
book_ID=8d57f6e7-e57d-497b-9e69-0649c63dbfd5;
```

The End