Cassandra Query Language

Cassandra Query Language (CQL)

query language for communicating
with the Cassandra database

commands are intentionally similar
SQL commands so it's easy to pick up

Cassandra Query Language (CQL)

case insensitive

internally keyspace, columnfamily and columnnames are stored in lowercase

But you can force cql to preserve case by enclosing the name in double quotation marks

Cassandra Query Language (CQL)

We can interact with Cassandra using cqlsh

cqlsh is a python based command line client

it comes bundled with the cassandra installation

Example 3:
Using cqlsh

Starting cqlsh

Go to the cassandra installation directory on your machine

Start cqlsh by running the following command:

```
>bin/cqlsh
```

if you have used `-pwd-auth` option while creating your cluster, then provide user/password

By default cqlsh uses localhost and port 9042 to connect to cassandra

Starting cqlsh

Go to the cassandra installation directory on your machine

Start cqlsh by running the
following command:
>bin/cqlsh

by default cqlsh uses localhost and
port 9042 to connect to cassandra

But if you want to use a different host or a port number,

you can set them in environment variables CQLSH_HOST and
CQLSH_PORT respectively

Login to cqlsh

cassandra is the default username and password

```
>cqlsh -u cassandra -p cassandra
```

Since we used --pwd-auth (password authenticator) while creating a cluster we need credentials to login

Example 4:
cqlsh
Creating a keyspace

Creating a keyspace

```
cassandra@cqlsh> CREATE KEYSPACE catalog WITH replication={'class':'SimpleStrategy',
                                                          'replication_factor':'3'};
```

For keyspace names which have upper and lower cases,

```
cassandra@cqlsh:catalog> create keyspace "MyCatalog" WITH replication={'class':'Simp
leStrategy', 'replication_factor':3};
```

enclose the name in double quotes

Creating a keyspace

cassandra@cqlsh> CREATE KEYSPACE catalog WITH replication={'class':'SimpleStrategy',
                                                            'replication_factor':'3'};

Command to create a keyspace

Creating a keyspace

cassandra@cqlsh> CREATE KEYSPACE catalog WITH replication={'class':'SimpleStrategy', 'replication_factor':'3'};

Name of the keyspace

Creating a keyspace

```
cassandra@cqlsh> CREATE KEYSPACE catalog WITH replication={'class':'SimpleStrategy',
                                                            'replication_factor':'3'};
```

Properties of the keyspace

Let's go through the properties

Creating a keyspace

```
cassandra@cqlsh> CREATE KEYSPACE catalog WITH replication={'class':'SimpleStrategy',
                                                            'replication_factor':'3'};
```

The only property we configure here is those for replication

Creating a keyspace

```
cassandra@cqlsh> CREATE KEYSPACE catalog WITH replication={'class':'SimpleStrategy',
                                                            'replication_factor':'3'};
```

REPLICATION

Multiple copies of data are created and stored on different nodes

Each copy is called a replica

Fault- tolerant

Creating a keyspace

REPLICATION

```
cassandra@cqlsh> CREATE KEYSPACE catalog WITH replication={'class':'SimpleStrategy',
replication_factor : 3 };
```

Replica Placement Strategy

Algorithm to determine which nodes hold the replicas for this keyspace

Creating a keyspace

REPLICATION

```
cassandra@cqlsh> CREATE KEYSPACE catalog WITH replication= 'class':'SimpleStrategy'
                                                            replication_factor : 3 };
```

Replica Placement Strategy

SIMPLE STRATEGY

NETWORK TOPOLOGY STRATEGY

For only <u>one</u> datacenter

For multiple datacenters

Creating a keyspace

REPLICATION

```
cassandra@cqlsh> CREATE KEYSPACE catalog WITH replication={'class':'SimpleStrategy',
                                                            'replication_factor':'3'};
```

Number of replicas for our data

We store 3 copies of our data

Creating a keyspace

```
cassandra@cqlsh> CREATE KEYSPACE catalog WITH replication={'class':'SimpleStrategy',
                                                            'replication_factor':'3'};
```

Lets check if our keyspace was created

Creating a keyspace

```
cassandra@cqlsh> CREATE KEYSPACE catalog WITH replication={'class':'SimpleStrategy',
                                                            'replication_factor':'3'};
```

Lets check if our keyspace was created

```
cassandra@cqlsh> describe keyspaces;
```
command to list all the keyspaces

```
system_schema   system_auth   system  catalog  system_distr
ibuted   system_traces
```

The remaining keyspaces are system generated
keyspaces. We will come back to them later