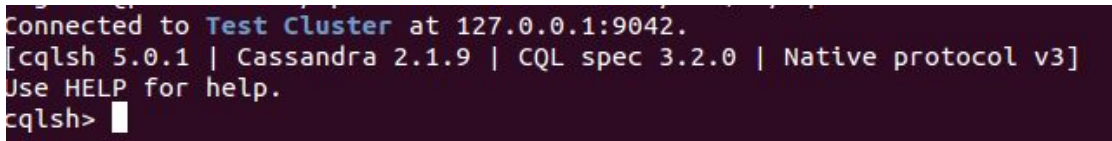


1.	Start Cassandra
1.1	make sure vagrant is installed: http://www.vagrantup.com/downloads
1.2	install virtualbox: https://www.virtualbox.org/wiki/Downloads
1.3	install git: https://git-scm.com/book/en/v2/Getting-Started-Installing-Git
1.4	download the project <ul style="list-style-type: none"> - go to a folder of your choice via command line window - execute the command: <code>git clone https://github.com/tomvdbulck/training-no-sql.git</code>
1.5	start up the vagrant box <ul style="list-style-type: none"> - navigate to the redis/vagrant folder: <code>cd cassandra/vagrant</code> - enter the command: <code>vagrant up</code>
This will now download and install the vagrant box, and download ,install and start the cassandra server. This might take a while	
1.6	The cassandra server will start automatically, we will connect to it with the CQL-shell <ul style="list-style-type: none"> - enter the command: <code>vagrant ssh</code> - you will now go into the vagrant box - navigate to : <code>apache-cassandra-2.1.9/bin/</code> - start the CQL-shell by typing : <code>./cqlsh</code> You should see something like this : 
2	CRUD (Create Read Update Delete)
2.1	Create a keyspace
CREATE KEYSPACE demo WITH replication = {'class':'SimpleStrategy', 'replication_factor':3};	
	Verify the Keyspace was created
DESCRIBE KEYSPACES;	
	you should see something like this

system_traces system demo	
	switch to the "demo" keyspace
USE demo;	
	see which tables are in the keyspace
DESCRIBE TABLES;	
	you should see something like this
<empty>	
2.2	Create a table
	CREATE TABLE users (user_id int PRIMARY KEY, fname text, lname text);
	Verify the table was created
	DESCRIBE TABLE users;
2.3	Inserting and reading data
	INSERT INTO users (user_id, lname) VALUES (3333, 'De Bruyne');
	Let's see what's in the table : - select * from users ;
	<p>you should see something like this :</p> <pre> user_id fname lname -----+-----+----- 3333 null De Bruyne </pre>
	<p>Let's insert a second user :</p> <ul style="list-style-type: none"> INSERT INTO users (user_id, fname, lname) VALUES (4444, 'Tom', 'Van Den Bulck'); <p>And check what's in the table</p> <ul style="list-style-type: none"> select * from users ;
	<p>you should see something like this :</p> <pre> user_id fname lname -----+-----+----- 3333 null De Bruyne 4444 Tom Van Den Bulck </pre>

Lets do an upsert :

- INSERT INTO users (user_id, fname, lname) VALUES (3333, 'Chris', 'De Bruyne');

and check the outcome :

- select * from users;

you should see something like this :

user_id	fname	lname
3333	Chris	De Bruyne
4444	Tom	Van Den Bulck

An insert automatically becomes an update if the record already exists.

Let's insert a second user :

- INSERT INTO users (user_id, fname, lname) VALUES (4445, 'Tom', 'Time To Live')
USING TTL 5;

(4 rows)

```
cqlsh:demo> select * from users;
```

user_id	fname	lname
3333	null	De Bruyne
4445	Tom	Time To Live
4444	Tom	Van Den Bulck

After 5 seconds the row with time to live will be gone.

2.5

updating data

UPDATE users USING TTL 5

SET fname = 'null'

WHERE user_id = 3333;

user_id	fname	lname
3333	null	De Bruyne
4445	Tom	Time To Live
4444	Tom	Van Den Bulck

After 5 seconds the value revert back to null;

Please note: after TTL the column will be emptied => not reset to the previous value

2.5	deleting data
	<p>Let's delete a user :</p> <ul style="list-style-type: none"> - DELETE FROM users WHERE user_id = 4444; <p>and verify :</p> <ul style="list-style-type: none"> - select * from users; <p>you should see something like this :</p> <pre> user_id fname lname -----+-----+----- 3333 Chris De Bruyne </pre>
	<p>You can also delete all data from a table</p> <ul style="list-style-type: none"> - TRUNCATE users ;
	<p>Please note that a delete is not immediately a delete. Cassandra places a tombstone (marker) to indicate a column has been deleted. When retrieving the values - all columns / rows indicated by a tombstone is ignored and the values with the most recent timestamps are shown.</p> <p>Tombstones are kept by default for 10 days.</p> <p>They allow for propagation of updates to the other nodes.</p>
3.0	Compound primary keys
	<pre>CREATE KEYSPACE stockwatcher WITH replication = {'class':'SimpleStrategy', 'replication_factor':3};</pre>
3.1	<p>Create a table with a compound primary key</p> <ul style="list-style-type: none"> - CREATE TABLE stockwatcher.WatchListItem (watchlist_id TIMEUUID, stock_symbol VARCHAR, start_price DECIMAL, created TIMESTAMP, PRIMARY KEY (watchlist_id, stock_symbol)); <p>and verify the creation :</p> <ul style="list-style-type: none"> - SELECT * FROM stockwatcher.watchlistitem ; - you should see something like this <pre> watchlist_id stock_symbol created start_price -----+-----+-----+----- </pre>

	Take note of the different colors for the primary key (partition key), clustering column and simple columns.																																								
3.2	Ordering in tables and queries																																								
	<p>First load some data. Perform the following command :</p> <ul style="list-style-type: none">- SOURCE '/vagrant/load-table-watchlistitem.cql'; <p>And verify the contents of the table</p> <ul style="list-style-type: none">- SELECT * FROM stockwatcher.watchlistitem ; <table><thead><tr><th>watchlist_id</th><th>stock_symbol</th><th>created</th><th>start_price</th></tr></thead><tbody><tr><td>ca58ee3c-d8a7-11e2-a440-85054b6a0b99</td><td>EC</td><td>2015-09-02 17:04:43+0000</td><td>44.88</td></tr><tr><td>ca58ee3c-d8a7-11e2-a440-85054b6a0b99</td><td>FISI</td><td>2015-09-02 17:04:43+0000</td><td>19.69</td></tr><tr><td>ca58ee3c-d8a7-11e2-a440-85054b6a0b99</td><td>ICE</td><td>2015-09-02 17:04:43+0000</td><td>171.30</td></tr><tr><td>ca58ee34-d8a7-11e2-a440-85054b6a0b99</td><td>ATRI</td><td>2015-09-02 17:04:43+0000</td><td>219.39</td></tr><tr><td>ca58ee34-d8a7-11e2-a440-85054b6a0b99</td><td>BANF</td><td>2015-09-02 17:04:43+0000</td><td>43.50</td></tr><tr><td>ca58ee34-d8a7-11e2-a440-85054b6a0b99</td><td>STEL</td><td>2015-09-02 17:04:43+0000</td><td>15.83</td></tr><tr><td>ca58ee3b-d8a7-11e2-a440-85054b6a0b99</td><td>ADEP</td><td>2015-09-02 17:04:43+0000</td><td>3.66</td></tr><tr><td>ca58ee3b-d8a7-11e2-a440-85054b6a0b99</td><td>PZN</td><td>2015-09-02 17:04:43+0000</td><td>6.75</td></tr><tr><td>ca58ee3b-d8a7-11e2-a440-85054b6a0b99</td><td>UNTK</td><td>2015-09-02 17:04:43+0000</td><td>2.01</td></tr></tbody></table>	watchlist_id	stock_symbol	created	start_price	ca58ee3c-d8a7-11e2-a440-85054b6a0b99	EC	2015-09-02 17:04:43+0000	44.88	ca58ee3c-d8a7-11e2-a440-85054b6a0b99	FISI	2015-09-02 17:04:43+0000	19.69	ca58ee3c-d8a7-11e2-a440-85054b6a0b99	ICE	2015-09-02 17:04:43+0000	171.30	ca58ee34-d8a7-11e2-a440-85054b6a0b99	ATRI	2015-09-02 17:04:43+0000	219.39	ca58ee34-d8a7-11e2-a440-85054b6a0b99	BANF	2015-09-02 17:04:43+0000	43.50	ca58ee34-d8a7-11e2-a440-85054b6a0b99	STEL	2015-09-02 17:04:43+0000	15.83	ca58ee3b-d8a7-11e2-a440-85054b6a0b99	ADEP	2015-09-02 17:04:43+0000	3.66	ca58ee3b-d8a7-11e2-a440-85054b6a0b99	PZN	2015-09-02 17:04:43+0000	6.75	ca58ee3b-d8a7-11e2-a440-85054b6a0b99	UNTK	2015-09-02 17:04:43+0000	2.01
watchlist_id	stock_symbol	created	start_price																																						
ca58ee3c-d8a7-11e2-a440-85054b6a0b99	EC	2015-09-02 17:04:43+0000	44.88																																						
ca58ee3c-d8a7-11e2-a440-85054b6a0b99	FISI	2015-09-02 17:04:43+0000	19.69																																						
ca58ee3c-d8a7-11e2-a440-85054b6a0b99	ICE	2015-09-02 17:04:43+0000	171.30																																						
ca58ee34-d8a7-11e2-a440-85054b6a0b99	ATRI	2015-09-02 17:04:43+0000	219.39																																						
ca58ee34-d8a7-11e2-a440-85054b6a0b99	BANF	2015-09-02 17:04:43+0000	43.50																																						
ca58ee34-d8a7-11e2-a440-85054b6a0b99	STEL	2015-09-02 17:04:43+0000	15.83																																						
ca58ee3b-d8a7-11e2-a440-85054b6a0b99	ADEP	2015-09-02 17:04:43+0000	3.66																																						
ca58ee3b-d8a7-11e2-a440-85054b6a0b99	PZN	2015-09-02 17:04:43+0000	6.75																																						
ca58ee3b-d8a7-11e2-a440-85054b6a0b99	UNTK	2015-09-02 17:04:43+0000	2.01																																						
	<p>Query the table with stock_symbol descending</p> <ul style="list-style-type: none">- SELECT * FROM stockwatcher.watchlistitem WHERE watchlist_id = ca58ee3c-d8a7-11e2-a440-85054b6a0b99 ORDER BY stock_symbol DESC ; <p>You should see something like this:</p> <table><thead><tr><th>watchlist_id</th><th>stock_symbol</th><th>created</th><th>start_price</th></tr></thead><tbody><tr><td>ca58ee3c-d8a7-11e2-a440-85054b6a0b99</td><td>ICE</td><td>2015-09-02 17:04:43+0000</td><td>171.30</td></tr><tr><td>ca58ee3c-d8a7-11e2-a440-85054b6a0b99</td><td>FISI</td><td>2015-09-02 17:04:43+0000</td><td>19.69</td></tr><tr><td>ca58ee3c-d8a7-11e2-a440-85054b6a0b99</td><td>EC</td><td>2015-09-02 17:04:43+0000</td><td>44.88</td></tr></tbody></table> <p>A WHERE clause can only be executed on the PK => and it must be an equals statement.</p> <p>Ordering can only be done on columns in the primary key and after a WHERE or IN statement has been executed on the partition key.</p>	watchlist_id	stock_symbol	created	start_price	ca58ee3c-d8a7-11e2-a440-85054b6a0b99	ICE	2015-09-02 17:04:43+0000	171.30	ca58ee3c-d8a7-11e2-a440-85054b6a0b99	FISI	2015-09-02 17:04:43+0000	19.69	ca58ee3c-d8a7-11e2-a440-85054b6a0b99	EC	2015-09-02 17:04:43+0000	44.88																								
watchlist_id	stock_symbol	created	start_price																																						
ca58ee3c-d8a7-11e2-a440-85054b6a0b99	ICE	2015-09-02 17:04:43+0000	171.30																																						
ca58ee3c-d8a7-11e2-a440-85054b6a0b99	FISI	2015-09-02 17:04:43+0000	19.69																																						
ca58ee3c-d8a7-11e2-a440-85054b6a0b99	EC	2015-09-02 17:04:43+0000	44.88																																						
3.4	<p>Limitations of the WHERE clause => it must be exact values</p> <p>You can use the TOKEN function together with a condition operator. Which only works on the partition key.</p> <ul style="list-style-type: none">- SELECT * from users where TOKEN(user_id) < 4000 <p>Will result in:</p>																																								

	<table><tr><th>user_id</th><th>fname</th><th>lname</th></tr><tr><td>3333</td><td>null</td><td>De Bruyne</td></tr></table>	user_id	fname	lname	3333	null	De Bruyne
user_id	fname	lname					
3333	null	De Bruyne					