Table of Contents

Cassandra / ScyllaDB Basics

CLUSTER → KEYSPACE → TABLE → DATA
Minimum number of running nodes are 2 (otherwise writes are not possible), so **3 or more nodes total** is absolutely minimal deployment.

“Cassandra timeout during read query at consistency TWO (2 responses were required but only 1 replica responded)”

Data Keys

```
PRIMARY KEY (a): The partition key is a.
PRIMARY KEY (a, b): The partition key is a, the clustering key is b.
PRIMARY KEY ((a, b)): The composite partition key is (a, b).
PRIMARY KEY (a, b, c): The partition key is a, the composite clustering key
is (b, c).
PRIMARY KEY ((a, b), c): The composite partition key is (a, b), the
clustering key is c.
PRIMARY KEY ((a, b), c, d): The composite partition key is (a, b), the
composite clustering key is (c, d).
```

In a situation of COMPOSITE primary key, the “a” of the key is called PARTITION KEY (in this example a is the partition key) and the second part of the key is the CLUSTERING KEY (b)

- The Partition Key is responsible for data distribution across your nodes.
- The Clustering Key is responsible for data sorting within the partition.
  - The Primary Key is equivalent to the Partition Key in a single-field-key table (i.e. Simple).
  - The Composite/Compound Key is just a multiple-columns key

Aprox. recommended capacity

5TB on one node
commitlog file on another disk

Shell

```
cqlsh ip 9042
```

Create Keyspace

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

CREATE KEYSPACE Excalibur
           WITH replication = {'class': 'NetworkTopologyStrategy', 'DC1' :
1, 'DC2' : 3}
            AND durable_writes = false;
```

Change number of replicas

```
ALTER KEYSPACE Excelsior
          WITH replication = {'class': 'SimpleStrategy',
'replication_factor' : 4};
```

Describe Keyspaces

```
describe keyspaces;
```

How to list all the tables

```
use {keyspace_name};
describe tables;
```

Info about table

```
use {keyspace_name};
describe table {table_name};
```

Create Table

```
CREATE TABLE positions (
    bucket int,
```

```
    idSap ascii,
    kodSlozka ascii,
    VolaciZnak varchar,
    CasAktualizace int,
    rychlost int,
    lat double,
    lon double,
    PRIMARY KEY (bucket,CasAktualizace,idSap)
) WITH CLUSTERING ORDER BY (CasAktualizace ASC, idSap ASC);
```

bucket = year + week

Create Secondary Index

```
CREATE INDEX idSap ON nis.positions (idSap);
```

experimental: true → scylla.yaml

WSO2 EI

Turn off consistency check

```
<property name="consistencyLevel">ANY</property> <-- write
<property name="consistencyLevel">ONE</property> <-- read
```

```
<data enableBatchRequests="true" name="cassandra"
serviceNamespace="positions" transports="http">
   <config enableOData="false" id="cassandra">
      <property name="cassandraServers">10.160.149.35</property>
      <property name="keyspace">nis</property>
      <property name="clusterName">hzspk</property>
      <property name="consistencyLevel">ONE</property>
   </config>
   <query id="status" useConfig="cassandra">
      <expression>SELECT COUNT(1) as count FROM nis.positions;</expression>
      <result element="status" rowName="positions">
         <element column="count" name="count" xsdType="integer"/>
      </result>
   </query>
   <query id="read" useConfig="cassandra">
      <expression>SELECT
idsap,kodslozka,volaciznak,casaktualizace,rychlost,lat,lon FROM
nis.positions WHERE casaktualizace &gt; :from AND casaktualizace &lt; :to
LIMIT 1000 ALLOW FILTERING;</expression>
      <result element="status" rowName="sap">
```

```
         <element column="idsap" name="idSap" xsdType="string"/>
         <element column="kodslozka" name="kodSlozka" xsdType="string"/>
         <element column="volaciznak" name="VolaciZnak" xsdType="string"/>
         <element column="casaktualizace" name="CasAktualizace"
xsdType="string"/>
         <element column="rychlost" name="rychlost" xsdType="string"/>
         <element column="lat" name="lat" xsdType="string"/>
         <element column="lon" name="lon" xsdType="string"/>
      </result>
      <param name="from" sqlType="INTEGER"/>
      <param name="to" sqlType="INTEGER"/>
   </query>
   <query id="save" useConfig="cassandra">
      <expression>INSERT INTO nis.positions
(idsap,typsap,kodslozka,volaciznak,casaktualizace,rychlost,lat,lon) VALUES
(:idSap,:typSap,:kodSlozka,:volaciZnak,:casAktualizace,:rychlost,:lat,:lon);
</expression>
      <param name="idSap" sqlType="STRING"/>
      <param name="typSap" sqlType="STRING"/>
      <param name="kodSlozka" sqlType="STRING"/>
      <param name="rychlost" sqlType="INTEGER"/>
      <param name="volaciZnak" sqlType="STRING"/>
      <param name="casAktualizace" sqlType="INTEGER"/>
      <param name="lat" sqlType="DOUBLE"/>
      <param name="lon" sqlType="DOUBLE"/>
   </query>
   <operation name="get" returnRequestStatus="true">
      <call-query href="read">
         <with-param name="from" query-param="from"/>
         <with-param name="to" query-param="to"/>
      </call-query>
   </operation>
   <operation name="status" returnRequestStatus="true">
      <call-query href="status"/>
   </operation>
   <operation name="savebatch" returnRequestStatus="true">
      <call-query href="save">
         <with-param name="idSap" query-param="idSap"/>
         <with-param name="typSap" query-param="typSap"/>
         <with-param name="kodSlozka" query-param="kodSlozka"/>
         <with-param name="rychlost" query-param="rychlost"/>
         <with-param name="volaciZnak" query-param="volaciZnak"/>
         <with-param name="casAktualizace" query-param="casAktualizace"/>
         <with-param name="lat" query-param="lat"/>
         <with-param name="lon" query-param="lon"/>
      </call-query>
   </operation>
   <resource method="GET" path="/status">
      <call-query href="status"/>
   </resource>
   <resource method="GET" path="/read">
```

```
      <call-query href="read">
         <with-param name="from" query-param="from"/>
         <with-param name="to" query-param="to"/>
      </call-query>
   </resource>
   <resource method="POST" path="/">
      <call-query href="save">
         <with-param name="idSap" query-param="idSap"/>
         <with-param name="typSap" query-param="typSap"/>
         <with-param name="kodSlozka" query-param="kodSlozka"/>
         <with-param name="rychlost" query-param="rychlost"/>
         <with-param name="volaciZnak" query-param="volaciZnak"/>
         <with-param name="casAktualizace" query-param="casAktualizace"/>
         <with-param name="lat" query-param="lat"/>
         <with-param name="lon" query-param="lon"/>
      </call-query>
   </resource>
</data>
```

Number of Rows

```
nodetool tablestats <keyspace.table>
SELECT COUNT(1) FROM table;
```

Convert time to timestamp

```
unixTimestampOf(minTimeuuid(casAktualizace))
```

Send data

```
curl -v -d
'<save><idSap>1</idSap><typSap>2</typSap><kodSlozka>3</kodSlozka><rychlost>4
</rychlost><volaciZnak>5</volaciZnak><casAktualizace>2017-08-04T17:46:55</ca
sAktualizace><lat>1</lat><lon>2</lon></save>' -H Content-Type:"text/xml"
http://endpoint
```

Export CSV

```
$ COPY nis.positions (lat, lon) FROM 'positions.csv' WITH DELIMITER='|' AND
HEADER=TRUE;
```

```
$ COPY nis.positions to 'positions-20170101.csv';
```