
DATASTAX

DATASTAX

DATASTAX ACADEMY

DataStax Enterprise

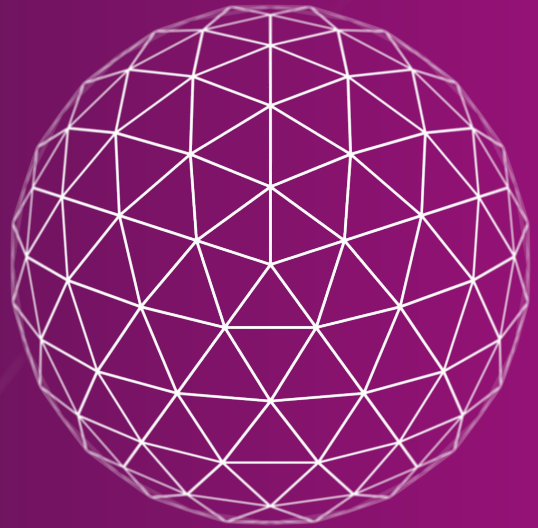
Foundations of Apache Cassandra™

DS201 - sections 3 - 5

July 27th, 2023

DATASTAX ACADEMY

#3 - Partitions



Primary Keys

DATASTAX ACADEMY

What is a Primary Key?

A table's primary key defines a unique path to access a single row of data.

Primary Keys

DATASTAX ACADEMY

Cassandra Primary Key Definition:

```
PRIMARY KEY ((partitionKey1, partitionKeyN),
clusteringKey1, cluseringKeyN)
```

Partition Keys

DATASTAX ACADEMY

What is a Partition Key?

A partition key is the part of the Primary Key definition which tells Cassandra where to place the data in the cluster.

Partition Keys

DATASTAX ACADEMY

Cassandra Partition Key Definition:

```
PRIMARY KEY ((partitionKey1, partitionKeyN),
clusteringKey1, cluseringKeyN)
```

Table w/ a simple key

DATASTAX ACADEMY

PRIMARY KEY (user_id)

```
> INSERT INTO users VALUES (1, 'Dev Awesome', 'TX');
> INSERT INTO users VALUES (2, 'Mrs. Reliable', 'CA');
> INSERT INTO users VALUES (3, 'Epic Dev', 'NY');
> INSERT INTO users VALUES (4, 'IgotUr Data', 'TX');
> INSERT INTO users VALUES (5, 'Always Onomnom', 'TX');
> INSERT INTO users VALUES (6, 'Store Dat Data', 'NY');
> INSERT INTO users VALUES (7, 'Lolo Latency', 'CA');
> INSERT INTO users VALUES (8, 'Lovin Ur Bytes', 'TX');
> INSERT INTO users VALUES (9, 'Model De Tables', 'NY');
```

user_id	name	state
1	Dev Awesome	TX
2	Mrs. Reliable	CA
3	Epic Dev	NY
4	IgotUr Data	TX
5	Always Onomnom	TX
6	Store Dat Data	NY
7	Lolo Latency	CA
8	Luvin Ur Bytes	TX
9	Model Dat Data	NY

Table w/ a simple key

DATASTAX ACADEMY

PRIMARY KEY (user_id)

```
> SELECT user_id, state, name
  FROM users WHERE user_id=1;

 user_id | state | name
---------+-------+----------------
       1 |    TX |    Dev Awesome
```

user_id	name	state
1	Dev Awesome	TX
2	Mrs. Reliable	CA
3	Epic Dev	NY
4	IgotUr Data	TX
5	Always Onomnom	TX
6	Store Dat Data	NY
7	Lolo Latency	CA
8	Luvin Ur Bytes	TX
9	Model Dat Data	NY

However...

DATASTAX ACADEMY

What if we want to query by something other than user_id?

<shrug/>

Maybe we want to query by state.

Partition by State

DATASTAX ACADEMY

PRIMARY KEY (state, user_id)

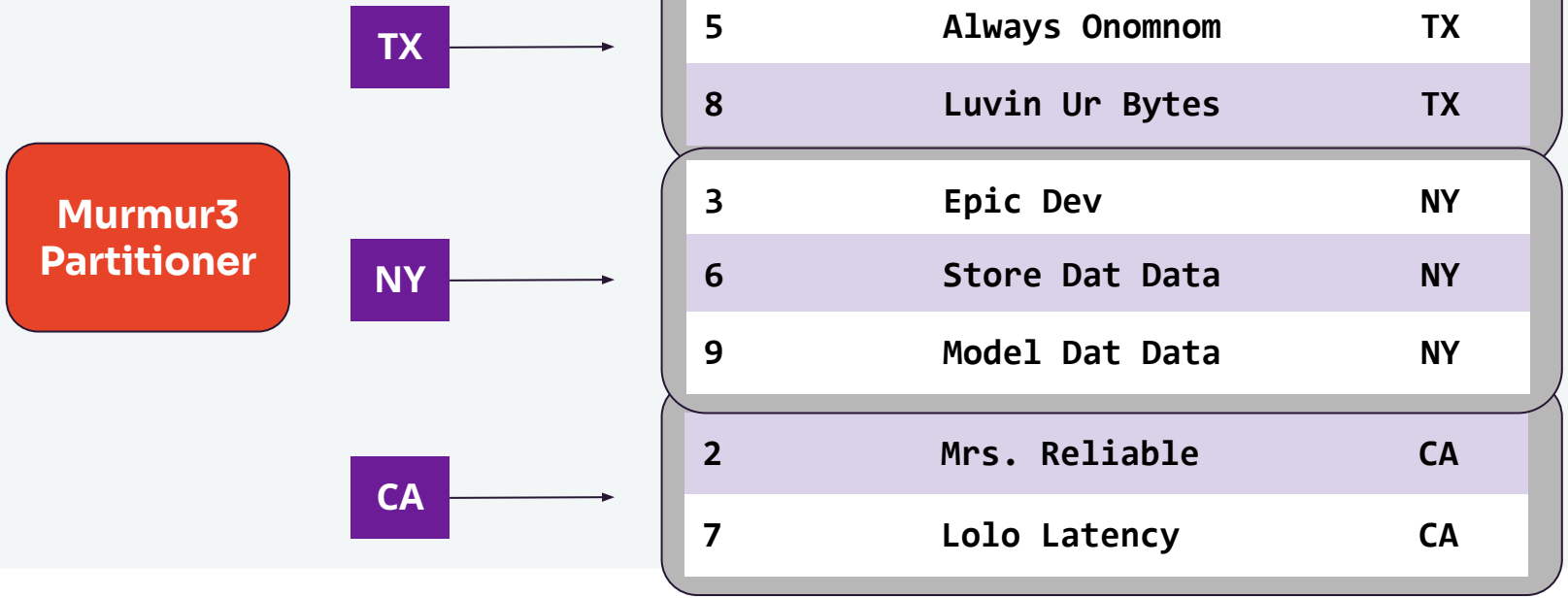
1	Dev Awesome	TX
2	Mrs. Reliable	CA
3	Epic Dev	NY
4	IgotUr Data	TX
5	Always Onomnom	TX
6	Store Dat Data	NY
7	Lolo Latency	CA
8	Luvin Ur Bytes	TX
9	Model Dat Data	NY

Partition by State

DATASTAX ACADEMY

PRIMARY KEY (state, user_id)

Murmur3 Partitioner



Murmur3 Partitioner

DATASTAX ACADEMY

PRIMARY KEY (state, user_id)

-2594951604484898973

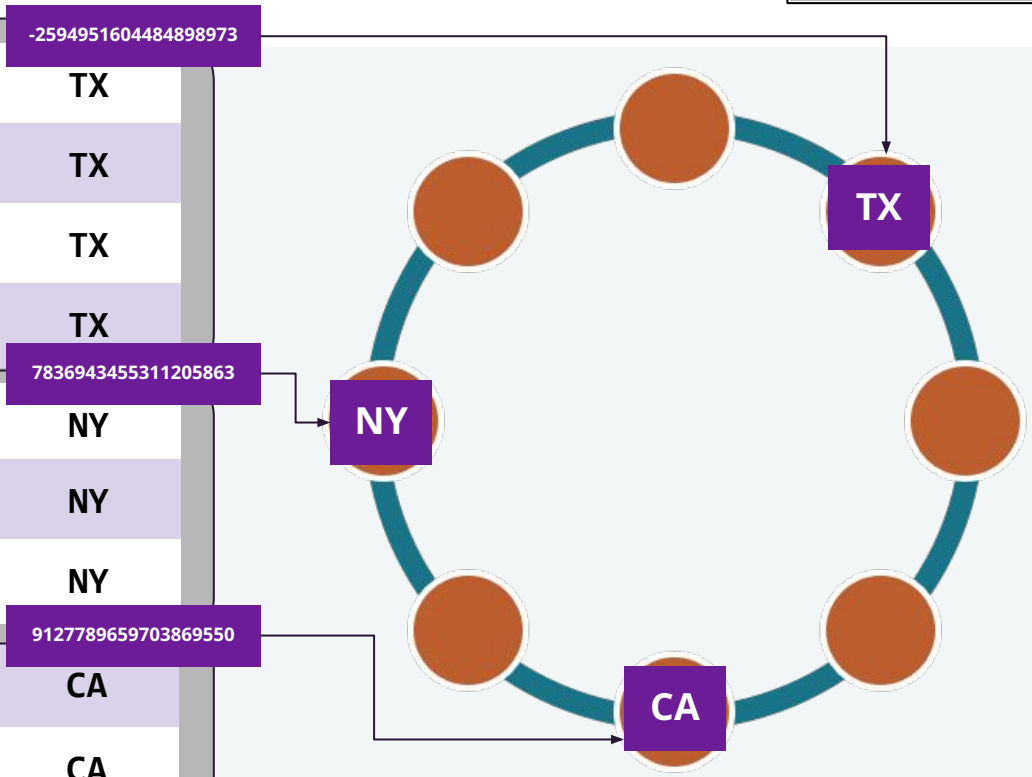
7836943455311205863

9127789659703869550

1	Dev Awesome	TX
4	IgotUr Data	TX
5	Always Onomnom	TX
8	Luvin Ur Bytes	TX
3	Epic Dev	NY
6	Store Dat Data	NY
9	Model Dat Data	NY
2	Mrs. Reliable	CA
7	Lolo Latency	CA

Data is placed in the cluster

1	Dev Awesome	TX
4	IgotUr Data	TX
5	Always Onomnom	TX
8	Luvin Ur Bytes	TX
3	Epic Dev	NY
6	Store Dat Data	NY
9	Model Dat Data	NY
2	Mrs. Reliable	CA
7	Lolo Latency	CA



Exercise #3

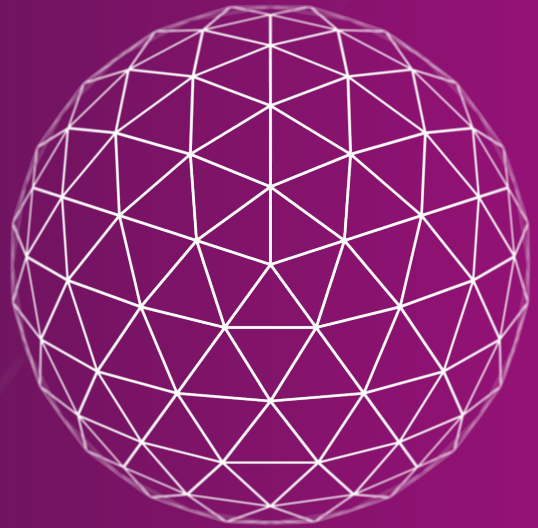
DATASTAX ACADEMY

Hands-on Exercise #3

- Create tables with different partition keys
- Execute CQL queries on tables with different partition keys

DATASTAX ACADEMY

#4 - Clustering Columns



Clustering Columns

DATASTAX ACADEMY

What is a Clustering Column?

A clustering column is the part of the Primary Key definition which defines the on-disk sort order within a partition.

Primary Keys

DATASTAX ACADEMY

Cassandra Clustering Column Definition:

```
PRIMARY KEY ((partitionKey1, partitionKeyN),
clusteringKey1, cluseringKeyN)
```

Effects of different PK definitions

DATASTAX ACADEMY

PRIMARY KEY (user_id)

5	Always Onomnom	TX	Dallas
1	Dev Awesome	TX	Houston
8	Luvin Ur Bytes	TX	Santa Fe
2	ComeTo Learnin	TX	Dallas
4	IgotUr Data	TX	Austin
7	Data Rowman	TX	Austin
6	Lone Star	TX	El Paso
9	Compact One	TX	Houston
3	Lone Node	TX	Snyder

Assume the rows are written in the order of their user_id.

Effects of different PK definitions

DATASTAX ACADEMY

PRIMARY KEY ((state), city)

~~4~~	~~IgotUr Data~~	~~TX~~	~~Austin~~
7	Data Rowman	TX	Austin
~~2~~	~~ComeTo Learnin~~	~~TX~~	~~Dallas~~
5	Always Onomnom	TX	Dallas
6	Lone Star	TX	El Paso
~~1~~	~~Dev Awesome~~	~~TX~~	~~Houston~~
9	Compact One	TX	Houston
8	Luvin Ur Bytes	TX	Santa Fe
3	Lone Node	TX	Snyder

Assume the rows are written in the order of their user_id.

DATASTAX ACADEMY

Tips

› From the Trenches

Cassandra Primary Keys are unique!

For multiple writes to the same PK, *the last write wins!*

Effects of different PK definitions

DATASTAX ACADEMY

PRIMARY KEY ((state), city, user_id))

-2594951604484898973

4	IgotUr Data	TX	Austin
7	Data Rowman	TX	Austin
2	ComeTo Learnin	TX	Dallas
5	Always Onomnom	TX	Dallas
6	Lone Star	TX	El Paso
1	Dev Awesome	TX	Houston
9	Compact One	TX	Houston
8	Luvin Ur Bytes	TX	Santa Fe
3	Lone Node	TX	Snyder

Clustering Columns

DATASTAX ACADEMY

Querying

- You must first provide the (complete) partition key.
- Clustering columns must be specified *in order*.
- You can perform equality (=) or range queries (< , <) on clustering columns.
- Equality comparisons must be performed before range comparisons.
- Since data is sorted on disk, range queries are a binary search followed by a sequential read.

Clustering Column queries

DATASTAX ACADEMY

Query:

```
> SELECT state, city, user_id, name
  FROM users_by_city_by_state
  WHERE state= 'TX' AND city='Austin';

 state | city   | user_id | name
-------+--------+---------+-------------
    TX | Austin |       4 | IgotUr Data
    TX | Austin |       7 | Data Rowman
```

Clustering Column queries

DATASTAX ACADEMY

Query:

```
> SELECT state, city, user_id, name
  FROM users_by_city_by_state
  WHERE state= 'TX' AND city='Austin'
  AND user_id > 5;

 state | city   | user_id | name
-------+--------+---------+-------------
    TX | Austin |       7 | Data Rowman
```

Clustering Column queries

DATASTAX ACADEMY

Query:

```
> SELECT state, city, user_id, name
  FROM users_by_city_by_state
  WHERE state= 'TX' AND user_id > 5;


 [Invalid query] message="PRIMARY KEY column
 "user_id" cannot be restricted as preceding
 column "city" is not restricted"
```

Changing Default Ordering

DATASTAX ACADEMY

Clustering Column definitions

- Default to ASCending sort direction.
- Sort direction can be influenced by the **`WITH CLUSTERING ORDER BY`** clause.
- Must include up to and including the columns you wish to order DESCending.

DATASTAX ACADEMY

Tips

› From the Trenches

Queries filtered by clustering columns *perform faster* if you do not switch the table's sort order!

Sometimes you may want to duplicate a table with a different sort order.

Clustering Column queries

DATASTAX ACADEMY

Error:

```
[Invalid query] message="Cannot execute this
query as it might involve data filtering and thus
may have unpredictable performance. If you want
to execute this query despite the performance
unpredictability, use ALLOW FILTERING.
```

Clustering Column queries

DATASTAX ACADEMY

Allow Filtering

- Relaxes the query requirements on partition keys.
- You can then query on just clustering columns.
- Causes Cassandra to scan all partitions in a table.
- DO NOT USE IT:
  - Might be ok on small data sets or small clusters.
  - But seriously just don't.

DATASTAX ACADEMY

Pop › Quiz!

When should you use ALLOW FILTERING?

Never

Guardrails in Cassandra 4.1

DATASTAX ACADEMY

cassandra.yaml:

```
# Guardrail to allow/disallow querying
# with ALLOW FILTERING. Defaults to true.

allow_filtering_enabled: false
```

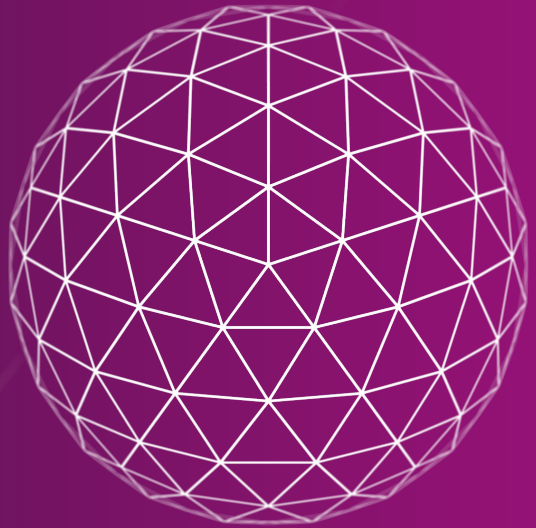
Exercise #4

Hands-on Exercise #4

- Create tables with different clustering columns
- Execute CQL queries on tables with different clustering columns

DATASTAX ACADEMY

#5 - Application Connectivity



Drivers

DATASTAX ACADEMY

There are drivers for most languages:

- Java
- Python
- C#
- C++
- Node.js
- Ruby
- Go

Similar Across Most Languages

DATASTAX ACADEMY

Python:

```
auth = PlainTextAuthProvider(username=user,
                             password=pwd)
cluster = Cluster(["127.0.0.1"], auth_provider=auth)
session = cluster.connect("killrvideo")
result = session.execute("<query>")[0]
```

Similar Across Most Languages

DATASTAX ACADEMY

Java:

```
CqlSession session= CqlSession.builder()
          .withKeyspace("killrvideo")
          .setAuthCredentials(user, pwd)
          .build();
ResultSet results = session.execute("<query>");
```

Similar Across Most Languages

DATASTAX ACADEMY

C#:

```
var cluster = Cluster.Builder()
        .AddContactPoint("127.0.0.1")
        .WithAuthProvider(
            new PlainTextAuthProvider(user,  pwd))
        .Build();
var session = cluster.Connect("killrvideo");
var rs = session.Execute("<query>");
```

Previous Workshop

DATASTAX ACADEMY

Cassandra Application Development

https://github.com/datastaxdevs/

workshop-cassandra-application-development

https://www.youtube.com/watch?v=nPps2oqWhnY

› Setup

DATASTAX ACADEMY

Python

```
$ pip install cassandra-driver
```

```
from cassandra.cluster import Cluster
from cassandra.auth import PlainTextAuthProvider
import sys
```

Parameters

Set defaults

```
hostname="127.0.0.1"
username="cassandra"
password="cassandra"
protocol=4
```

```
if (len(sys.argv) > 1):
    hostname=sys.argv[1]
if (len(sys.argv) > 2):
    username=sys.argv[2]
if (len(sys.argv) > 3):
    password=sys.argv[3]
```

Build connection

```
nodes = []
nodes.append(hostname)
auth_provider =
        PlainTextAuthProvider(username=username,
                              password=password)
cluster = Cluster(nodes,auth_provider=auth_provider,
        protocol_version=protocol)
session = cluster.connect()
```

Query Data

DATASTAX ACADEMY

SELECT

```
rows = session.execute(
    "SELECT broadcast_address, host_id,
     data_center, rack, tokens
     FROM system.local; -- this is a comment\n")
```

Display Results

DATASTAX ACADEMY

Iterate & print

```
for row in rows:
    print("address: " + row[0])
    print("host id: " + str(row[1]))
    print("data center: " + row[2])
    print("rack: " + row[3])
    print("tokens: " + str(len(row[4])))
```

No Exercises for section #5