4.Working with nonrelational data

-NoSQL databases ((<https://aws.amazon.com/nosql/>). Spring Data covers for many NoSQL databases: MongoDB, Cassandra, Couchbase, Neo4j, Redis… The programming model is nearly identical.

# 4.1 Working with Cassandra repositories

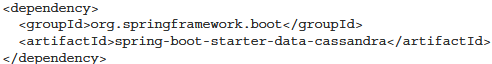
-Cassandra deals in rows of data written to tables, which are partitioned across one-to-many distributed nodes. No single node carries all data, but any given row may be replicated across multiple nodes.

-Spring Data Cassandra provides automatic repository support for Cassandra database that’s quite similar to SD JPA. Read http://cassandra.apache.org/doc/latest/

## 4.1.1 Enabling Spring Data Cassandra

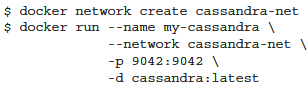
-Add SB starter dependency. There are 2 separate SD Cassandra starter dependencies: one for **reactive** data persistence and one for standard, **nonreactive** persistence.

-Use **nonreactive starter**:



+Remove JPA or JDBC and any relational database.

-Cassandra operates as a cluster of nodes that together act as a complete database system. If you don’t already have one, start a single-node cluster for development purpose using **Docker**:

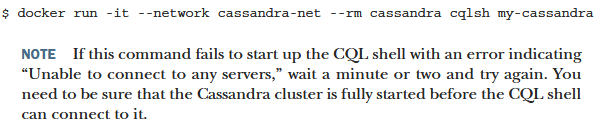


+This starts single-node cluster and exposes the node’s port on host machine.

-You need to configure the name of a **keyspace**: a grouping of tables in Cassandra node. It’s analogous to how tables, views and constraints are grouped in a relational database.

-You can manually create keyspace by Cassandra CQL shell:

+Start CQL shell by Docker:



+create keyspace command:



## 4.1.2 Understanding Cassandra data modeling

## 4.1.3 Mapping domain types for Cassandra persistence

## 4.1.4 Writing Cassandra repositories

# 4.2 Writing MongoDB repositories

## 4.2.1 Enabling Spring Data MongoDB

## 4.2.2 Mapping domain types to documents

## 4.2.3 Writing MongoDB repository interfaces

# -Summary