

**Autoren:** Nathan Ribinin, Kacper Bohaczyk

**version** 16-01-2024

# 1. - Key-Value-Datenbanken

## Was sind Key value Datenbanken

Key-Value-Datenbanken sind eine Art von NoSQL-Datenbanken, die auf dem Prinzip der Zuordnung von Schlüssel-Wert-Paaren basieren.

## Wo werden Key value Datenbanke verwendet

- *Benutzerverwaltung:*
- *\_Konfigurationsmanagement:*
- Produktdatenbanken

## Umsetzung

Download von redis über docker :

```
docker pull redis
```

Runnen des redis containers

```
``docker run --name my-redis-container -p 6379:6379 -d redis
```

Starten des Redis-Client

```
docker exec -it my-redis-container redis-cli
```

```
``
```

User hinzufügen

```
127.0.0.1:6379> SET user:kacper '{"id": 1, "name": "Kacper Bohaczyk",  
"email": "kacper@gmail.com"}'
```

```
127.0.0.1:6379> SET user:nathan '{"id": 2, "name": "Nathan Ribinin",  
"email": "nathan@gmail.com"}'
```

User können über

```
GET user:kacper
```

oder

```
GET user:nathan
```

abgerufen werden

das ausgabeformat ist dabei

```
{"id": 1, "name": "Kacper Bohaczyk", "email":  
"kacper@gmail.com"}
```

um einen user zu löschen benutzen wir den `DEL` befehl

```
DEL user:kacper
```

---

## 2. Spaltenorientierte Datenbank: Apache Cassandra

### Was ist Apache Cassandra?

Apache Cassandra ist eine Open-Source-Datenbank, die entwickelt wurde, um große Mengen von Daten sicher und hoch verfügbar zu speichern. Sie ist besonders gut für Anwendungen geeignet, die schnelle Datenzugriffe und Skalierbarkeit erfordern, wie zum Beispiel soziale Netzwerke, E-Commerce-Websites und Big-Data-Analysen. Cassandra verteilt Daten über mehrere Server, um Ausfallsicherheit und hohe Leistung zu gewährleisten. Es ermöglicht auch flexible Datenmodelle und ist bekannt für seine Fähigkeit, mit wachsenden Datenmengen umzugehen, indem es horizontal skaliert.

### Umsetzung

Cassandra basiert auf Java, also muss zuerst einmal Java installiert werden. Wichtig zu beachten dass Cassandra nur mit den Java Versionen 8 oder 11 kompatibel ist. Dazu werden folgende Commands benötigt:

```
sudo apt update
```

```
sudo apt-get install openjdk-11-jre
```

Für die Überprüfung sollte dann das auftreten

```
nathan@nathan-vm:~$ java -version
openjdk version "11.0.21" 2023-10-17
OpenJDK Runtime Environment (build 11.0.21+9-post-Ubuntu-0ubuntu123.04)
OpenJDK 64-Bit Server VM (build 11.0.21+9-post-Ubuntu-0ubuntu123.04, mixed
mode, sharing)
```

Nun kommen wir zur Installation von Cassandra:

Das Repository wird hinzugefügt

```
echo "deb https://deb.debian.cassandra.apache.org 41x main" | sudo tee -a
/etc/apt/sources.list.d/cassandra.sources.list
```

Dann werden die Repository Keys hinzugefügt

```
curl https://downloads.apache.org/cassandra/KEYS | sudo apt-key add -
```

Die Repositorys updaten

```
sudo apt-get update
```

Und nun Cassandra installieren

```
sudo apt-get install cassandra
```

Und fertig. Cassandra ist installiert worden.

Nun müssen wir Cassandra aktivieren:

```
sudo systemctl enable cassandra
```

```
sudo systemctl start cassandra
```

Um zu überprüfen ob Cassandra läuft, kann folgender Command ausgeführt werden:

```
nathan@nathan-vm:~$ sudo systemctl status cassandra
• cassandra.service - LSB: distributed storage system for structured data
   Loaded: loaded (/etc/init.d/cassandra; generated)
   Active: active (running) since Thu 2024-01-11 13:02:38 CET; 1min 51s
   ago
```

```
Docs: man:systemd-sysv-generator(8)
Tasks: 44 (limit: 4600)
Memory: 1.3G
CPU: 21.433s
CGroup: /system.slice/cassandra.service
└─5472 /usr/bin/java -ea -da:net.openhft... -
XX:+UseThreadPriorities -XX:+HeapDumpOnOut0
```

Will man in Cassandra hinein, führt man folgendes aus:

```
nathan@nathan-vm:~$ cqlsh
Connected to Test Cluster at 127.0.0.1:9042
[cqlsh 6.1.0 | Cassandra 4.1.3 | CQL spec 3.4.6 | Native protocol v5]
Use HELP for help.
cqlsh>
```

## Beispiele

Der Befehl `CREATE KEYSPACE` in Cassandra wird verwendet, um einen neuen Keyspace zu erstellen. Ein Keyspace ist in Cassandra eine logische Container-Struktur, die dazu dient, Daten in Tabellen zu organisieren und zu isolieren.

```
cqlsh> create keyspace test_db
... WITH REPLICATION = {
... 'class': 'SimpleStrategy',
... 'replication_factor' : 1
... };
```

Um dann die Datenbank zu verwenden wird folgendes ausgeführt:

```
cqlsh> USE test_db;
cqlsh:test_db>
```

Dann kann man eine Tabelle erstellen:

```
cqlsh:test_db> create table test_db
... (
... name TEXT PRIMARY KEY,
... surname TEXT,
... phone FLOAT
... );
```

Dann können weiters Personen in die Tabelle hinzugefügt werden:

```
cqlsh:test_db> INSERT INTO test_db(name, surname, phone) Values('Kacper',  
'Bohaczyk', 456123789);
```

```
cqlsh:test_db> INSERT INTO test_db(name, surname, phone) Values('Nathan',  
'Ribinin', 123456789);
```

Und zu guter Schluss kann die Tabelle auch angezeigt werden lassen:

```
cqlsh:test_db> select * from test_db ;
```

name	phone	surname
Kacper	4.5612e+08	Bohaczyk
Nathan	1.2346e+08	Ribinin

(2 rows)