

cassandra

Krzysztof Danilewicz

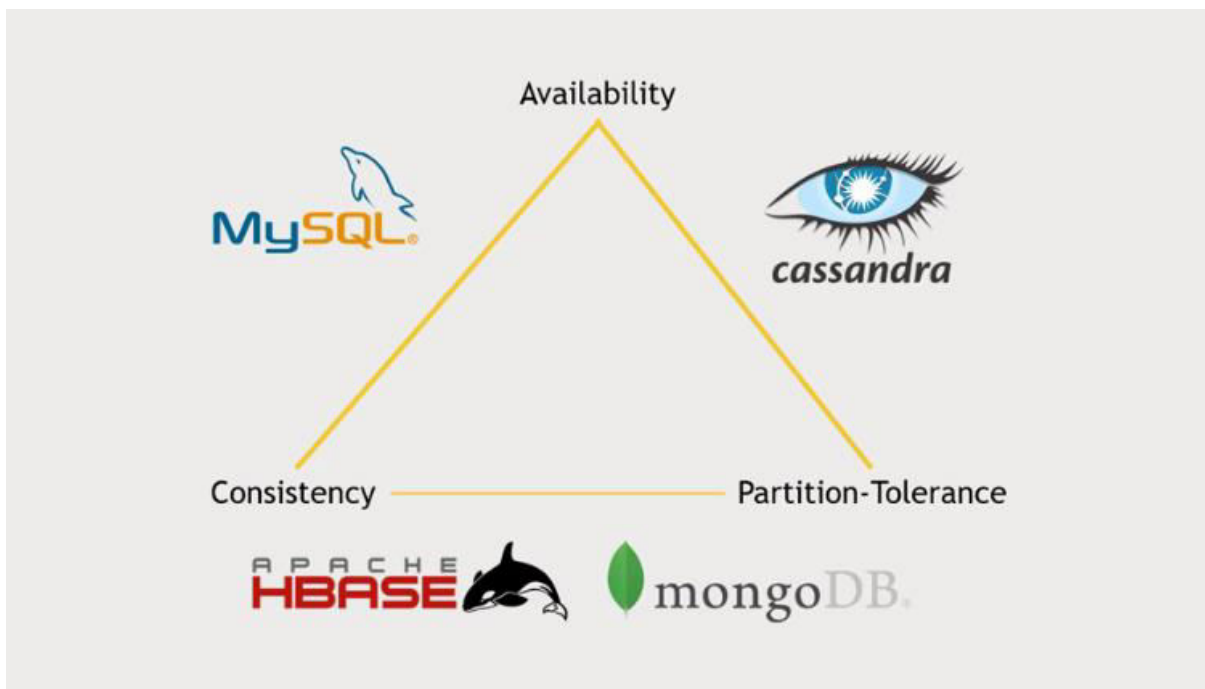
Cassandra to rozproszona baza danych Apache, która jest wysoce skalowalna i zaprojektowana do zarządzania bardzo dużymi ilościami danych strukturalnych. Zapewnia wysoką dostępność bez pojedynczego punktu awarii.

Twierdzenie CAP (Consistency, Availability, Partition):

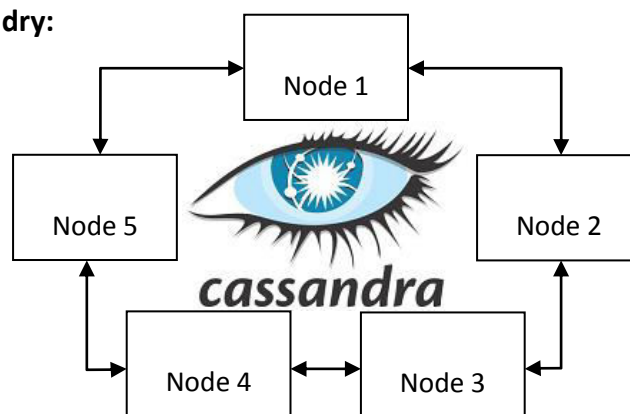
Twierdzenie postawione w 2000r. przez Erica Brewera z Uniwersytetu Berkeley, udowodnione w 2012r. przez Setha Gilberta oraz Nancy Lynch mówiące iż w rozproszonych systemach przetwarzania danych nie jest możliwe jednoczesne utrzymanie trzech właściwości:

- **spójności** (*consistency*) oznaczającej że wszystkie węzły mają jednoczesny dostęp do jednakowych danych;
- **osiągalności** (*availability*) czyli, że każde żądanie doczeka się odpowiedzi;
- **partycjonowanie** (*partition tolerance*) system potrafi działać pomimo utraty części komunikatów lub uszkodzenia niektórych węzłów.

Ponieważ big data nie ma sensu bez partycjonowania, więc jedyny wybór jaki nam pozostaje to spójność czy osiągalność.



Architektura Cassandra:



Mechanizm przechowywania danych w Casandrze:

Keyspace to najbardziej zewnętrzny kontener danych w Cassandrze. Podstawowe atrybuty keyspace w Cassandrze to:

- **Współczynnik replikacji** (*replication factor*) - jest to liczba komputerów w klastrze, które otrzymają kopie tych samych danych.
- **Strategia umieszczania replik** (*replica placement strategy*) - to nic innego jak strategia umieszczania replik w ringu. Mamy strategie, takie jak **prosta strategia** (*simple strategy*), **stara strategia topologii sieci** (*rack-aware strategy*, strategia uwzględniająca szafę serwerową) oraz **strategia topologii sieci** (*network topology strategy*, strategia współdzielona z centrum danych).
- **Rodzina kolumn** (*column families*) – keyspace to kontener dla jednej lub więcej rodziny kolumn. Rodzina kolumn z kolei jest kontenerem zbioru wierszy. Każdy wiersz zawiera uporządkowane kolumny. Rodziny kolumn reprezentują strukturę danych.

Keyspace1

Column Family 1

Key1	ColumnName1	ColumnName2	ColumnName3	ColumnName4
	Value1	Value2	Value3	Value4
Key2	ColumnName1	ColumnName2	ColumnName3	ColumnName4
	Value1	Value2	Value3	Value4

Column Family 2

Key1	ColumnName1	ColumnName2	ColumnName3
	Value1	Value2	Value3
Key2	ColumnName1	ColumnName2	ColumnName3
	Value1	Value2	Value3

CQLSH (Cassandra Query Language Shell) – jest to powłoka za pomocą której można komunikować się z Casandrą.

Instalacja Cassandra:

- Zaloguj się na maszynę korzystając z loginu i hasła: **maria_dev**
- Przełącz się na superuser'a. Wpisz: **su root** i podaj hasło **danilewicz**
- Przełącz się na Python 2.7 wpisując: **scl enable python27 bash**
- Sprawdź czy aktualna wersja pythona jest większa od 2.7. Wpisz: **python --version**

```
maria_dev@sandbox-hdp:/home/maria_dev
login as: maria_dev
maria_dev@127.0.0.1's password:
Last login: Sun May 12 13:04:19 2019 from 10.0.2.2
[maria_dev@sandbox-hdp ~]$ su root
Password:
[root@sandbox-hdp maria_dev]# scl enable python27 bash
[root@sandbox-hdp maria_dev]# python --version
Python 2.7.13
[root@sandbox-hdp maria_dev]#
```

- Wejdź do folderu /etc/yum.repos.d. W tym celu wpisz: **cd /etc/yum.repos.d**
- Utwórz nowy plik tekstowy wpisując: **nano datastax.repo**
- Wpisz do pliku poniższy tekst:

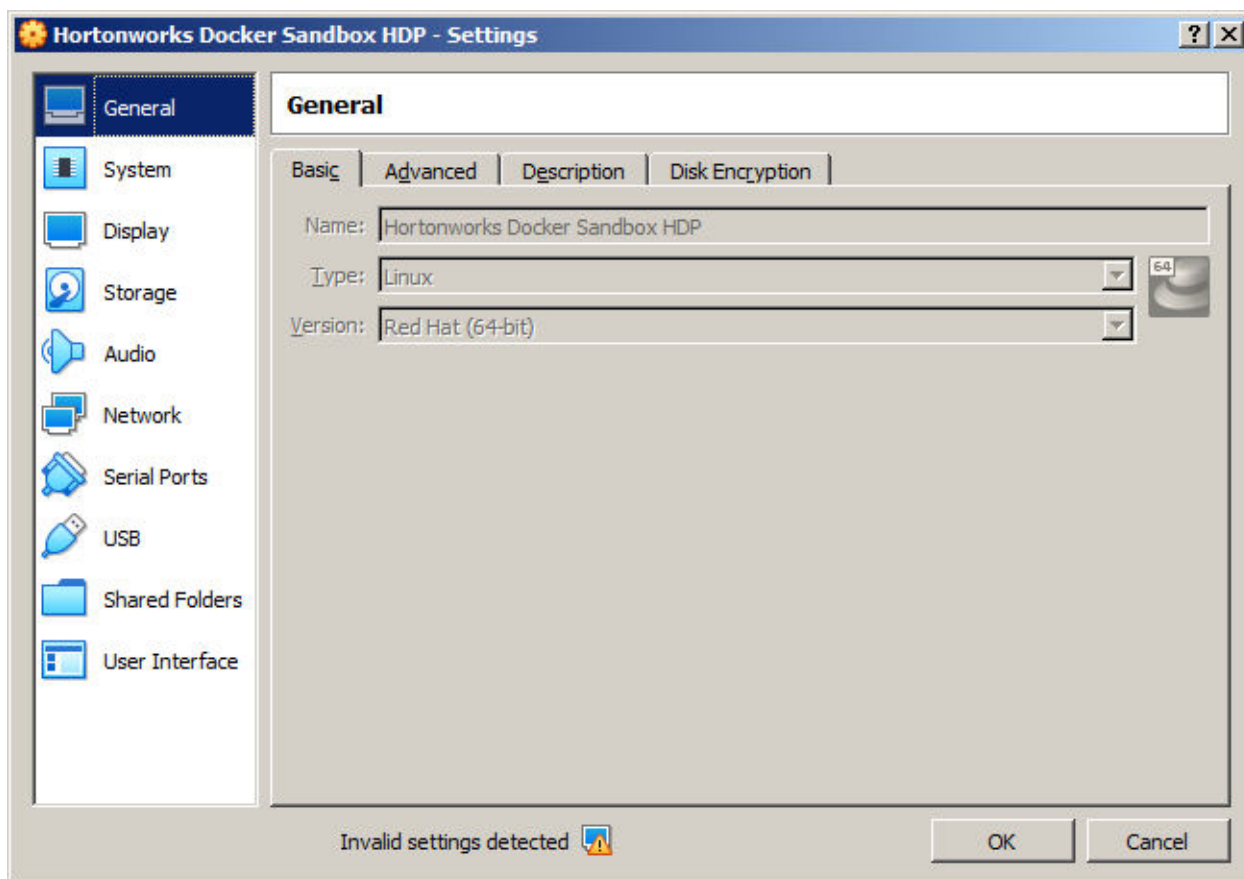
```
[datastax]
name = DataStax Repo for Apache Cassandra
baseurl = http://rpm.datastax.com/community
enabled = 1
gpgcheck = 0
```

- Zapisz zmiany wciskając Ctrl+X, następnie Y i zatwierdź Enterem.
- Teraz możemy zainstalować pakiet Cassandra. Wpisz: **yum install dsc30**
- Potwierdź wpisując **y**

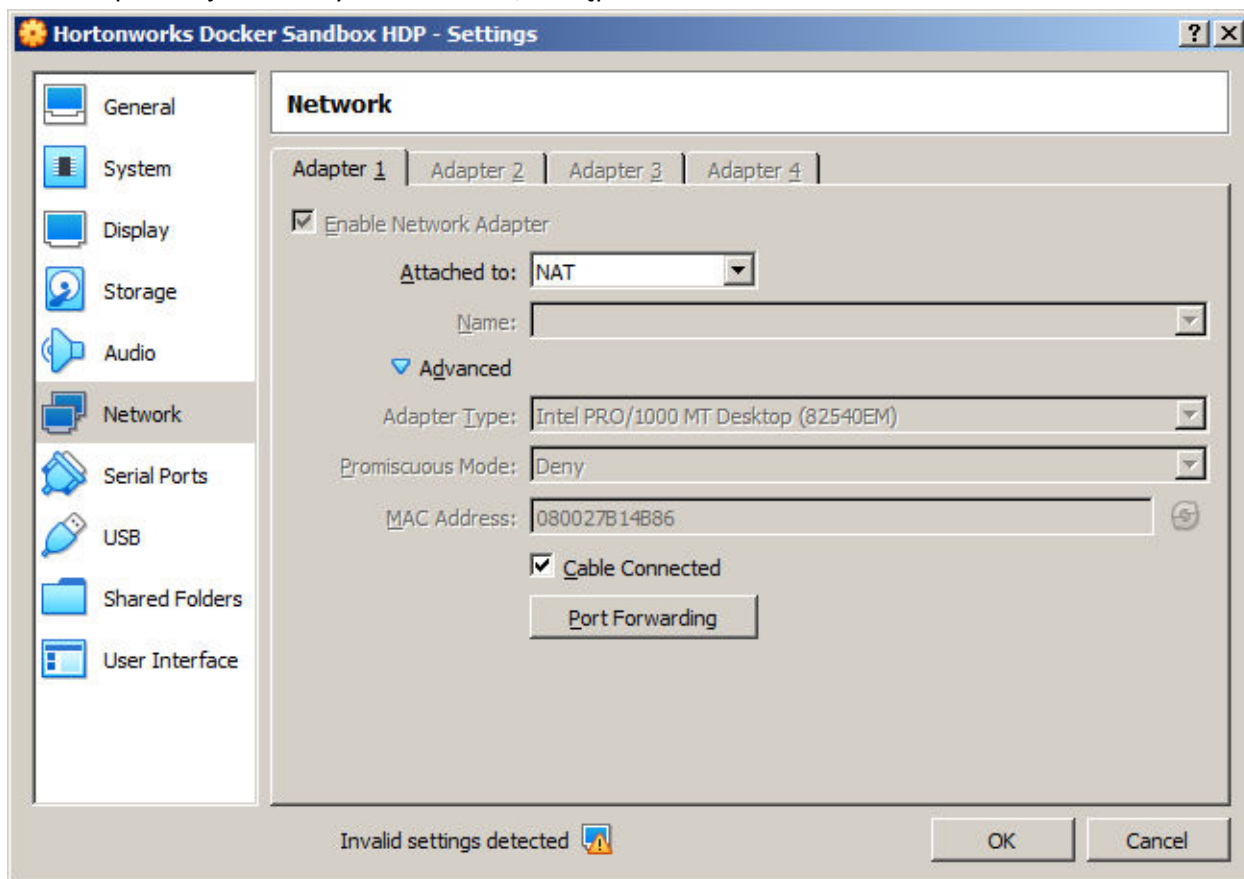
```
=====
Package                Arch          Version        Repository      Size
=====
Installing:
dsc30                  noarch        3.0.9-1        datastax        1.9 k
Installing for dependencies:
cassandra30            noarch        3.0.9-1        datastax        24 M
Transaction Summary
=====
Install                2 Package(s)

Total download size: 24 M
Installed size: 31 M
Is this ok [y/N]: y
```

- Zainstaluj Cassandra Query Language Shell. Wpisz: **pip install cqsh**
- Następnym krokiem jest otwarcie portu 9042:
 - W programie Oracle VM VirtualBox Manager zaznacz wirtualną maszynę i wejdź w jej ustawienia:



- Z menu po lewej stronie wybierz Network, następnie rozwiń menu Advanced:



- Kliknij guzik 'Port Forwarding':

Name	Protocol	Host IP	Host Port	Guest IP	Guest Port
Accumulo	TCP	127.0.0.1	50095		50095
AmbariInfra	TCP	127.0.0.1	8886		8886
AmbariShell	TCP	127.0.0.1	4200		4200
Atlas	TCP	127.0.0.1	21000		21000
Custom1	TCP	127.0.0.1	60000		60000
Custom10	TCP	127.0.0.1	10015		10015
Custom11	TCP	127.0.0.1	10016		10016
Custom12	TCP	127.0.0.1	10502		10502
Custom13	TCP	127.0.0.1	33553		33553
Custom14	TCP	127.0.0.1	39419		39419
Custom15	TCP	127.0.0.1	15002		15002
Custom16	TCP	127.0.0.1	111		111

OK Cancel

- W prawym górnym rogu kliknij plusik, aby dodać nowy wpis dla Cassandra:

nfs	TCP	127.0.0.1	42111		42111
nodemanager	TCP	127.0.0.1	8040		8040
Cassandra	TCP	127.0.0.1	9042		9042

OK Cancel

- Zatwierdź zmiany klikając OK dwa razy.

- Uruchom Cassandre wpisując: **service cassandra start**

```

maria_dev@sandbox-hdp:/etc/yum.repos.d
[root@sandbox-hdp yum.repos.d]# service cassandra start
Starting Cassandra: OK
[root@sandbox-hdp yum.repos.d]#

```

Praca z Cassandra przy użyciu CQLSH.

Uruchom cqlsh wpisując: `cqlsh --cqlversion="3.4.0"`

```
maria_dev@sandbox-hdp:/etc/yum.repos.d
[root@sandbox-hdp yum.repos.d]# cqlsh --cqlversion="3.4.0"
Connected to Test Cluster at 127.0.0.1:9042.
[cqlsh 5.0.1 | Cassandra 3.0.9 | CQL spec 3.4.0 | Native protocol v4]
Use HELP for help.
cqlsh>
```

Tworzenie Keyspace przy użyciu Cqlsh:

Składnia:

`CREATE KEYSPACE nazwa_dla_keyspace WITH replication = {'class': 'nazwa_strategi', 'replication_factor': 'liczba_replikacji'};`

Przykład:

`CREATE KEYSPACE filmy WITH replication = {'class': 'SimpleStrategy', 'replication_factor': '1'};`

Wybranie Keyspace przy użyciu Cqlsh:

Składnia:

`USE nazwa_keyspace;`

Przykład:

`USE filmy;`

```
maria_dev@sandbox-hdp:/etc/yum.repos.d
cqlsh> USE filmy;
cqlsh:filmy>
```

Zmiana właściwości Keyspace przy użyciu Cqlsh:

Składnia:

`ALTER KEYSPACE nazwa_keyspace WITH replication = {'class': 'nazwa_strategi', 'replication_factor': 'liczba_replikacji'};`

Przykład:

`ALTER KEYSPACE filmy WITH replication = {'class': 'SimpleStrategy', 'replication_factor': '2'};`

Wyświetlanie listy Keyspace przy użyciu Cqlsh:

Składnia:

`DESCRIBE keyspaces;`

Przykład:

```
maria_dev@sandbox-hdp:/etc/yum.repos.d
cqlsh:filmy> DESCRIBE keyspaces;

system_schema  system_auth  system  filmy  system_distributed  system_traces

cqlsh:filmy> █
```

Usuwanie Keyspace przy użyciu Cqlsh:

Składnia:

DROP KEYSPACE nazwa_keyspace;

Przykład:

DROP KEYSPACE filmy;

Tworzenie tabeli przy pomocy Cqlsh:

Składnia:

CREATE TABLE nazwa_tabeli (nazwa_kolumny1 rodzaj_danych1, nazwa_kolumny2 rodzaj_danych2);

Przykład:

CREATE TABLE komedia (tytul text PRIMARY KEY, rok int, czas_trwania int, tytul_orginalny text);

Weryfikacja: select * from komedia;

```
maria_dev@sandbox-hdp:/etc/yum.repos.d
cqlsh:filmy> select * from komedia;

  tytul | czas_trwania | rok | tytul_orginalny
-----+-----+---+-----
(0 rows)
cqlsh:filmy> █
```

Dodawanie kolumny do tabeli przy użyciu Cqlsh:

Składnia:

ALTER TABLE nazwa_tabeli ADD nazwa_nowej_kolumny typ_danych;

Przykład:

ALTER TABLE komedia ADD rezyser text;

Usuwanie kolumny z tabeli przy użyciu Cqlsh:

Składnia:

ALTER TABLE nazwa_tabeli DROP nazwa_kolmny;

Przykład:

ALTER TABLE komedia DROP rezyser;

Usuwanie tabeli przy użyciu Cqlsh:

Składnia:

DROP TABLE nazwa_tabeli;

Przykład:

DROP TABLE komedia;

Wstawianie danych przy użyciu Cqlsh:

Składnia:

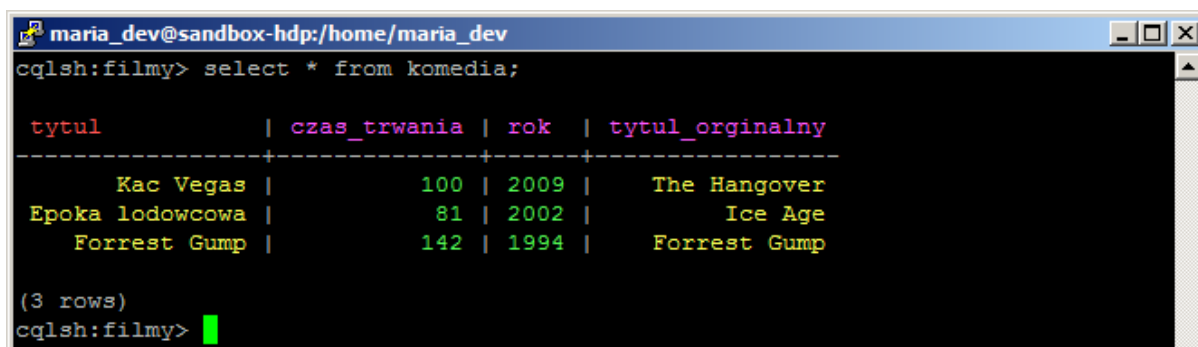
INSERT INTO nazwa_tabeli (nazwa_kolumny1, nazwa_kolumny2,...) VALUES (wartość_kolmny1, wartość_kolumny2, ...);

Przykład:

INSERT INTO komedia (tytul, czas_trwania, rok, tytul_oryginalny) VALUES ('Epoka lodowcowa', 81, 2002, 'Ice Age');

INSERT INTO komedia (tytul, czas_trwania, rok, tytul_oryginalny) VALUES ('Kac Vegas', 100, 2009, 'The Hangover');

INSERT INTO komedia (tytul, czas_trwania, rok, tytul_oryginalny) VALUES ('Forrest Gump', 142, 1994, 'Forrest Gump');



```
maria_dev@sandbox-hdp:/home/maria_dev
cqlsh:filmy> select * from komedia;

  tytul          | czas_trwania | rok  | tytul_oryginalny
-----+-----+-----+-----
      Kac Vegas |          100 | 2009 | The Hangover
  Epoka lodowcowa |           81 | 2002 | Ice Age
    Forrest Gump |          142 | 1994 | Forrest Gump

(3 rows)
cqlsh:filmy>
```

Aktualizacja danych przy użyciu Cqlsh:

Składnia:

UPDATE nazwa_tabeli SET nazwa_kolmny1 = nowa_wartość1, nazwa_kolmny2 = nowa_wartość2 WHERE warunek;

Przykład:

```
UPDATE komedia SET czas_trwania = 85 WHERE tytuł = 'Epoka lodowcowa';
```

Odczyt danych przy użyciu Cqlsh:

Przykład:

```
SELECT nazwa_kolumny1, nazwa_kolumny2 FROM nazwa_tabeli WHERE warunek;
```

Przykłady:

```
SELECT tytuł_oryginalny FROM komedia;
```

```
SELECT * FROM komedia WHERE tytuł = 'Forrest Gump';
```

```
SELECT * FROM komedia WHERE rok = 2002;
```

Tworzenie indeksu przy użyciu Cqlsh:

Składnia:

```
CREATE INDEX nazwa_indeksu ON nazwa_tabeli (nazwa_kolumny);
```

Przykład:

```
CREATE INDEX rok ON komedia (rok);
```

```
SELECT * FROM komedia WHERE rok = 2002;
```

```
SELECT * FROM komedia WHERE rok > 2000;
```

```
SELECT * FROM komedia WHERE rok > 2000 ALLOW FILTERING;
```

Usuwanie indeksu przy użyciu Cqlsh:

Składnia:

```
DROP INDEX nazwa_indeksu;
```

Przykład:

```
DROP INDEX rok;
```

```
SELECT * FROM komedia WHERE rok = 2002;
```

Usuwanie określonej komórki przy użyciu Cqlsh:

Składnia:

```
DELETE nazwa_kolumny FROM nazwa_tabeli WHERE warunek;
```

Przykład:

DELETE tytuł_oryginalny FROM komedia WHERE tytuł = 'Forrest Gump';

```
maria_dev@sandbox-hdp:/home/maria_dev
cqlsh:filmy> DELETE tytuł_oryginalny FROM komedia WHERE tytuł = 'Forrest Gump';
cqlsh:filmy> SELECT * FROM komedia;

tytuł          | czas_trwania | rok  | tytuł_oryginalny
-----+-----+-----+-----
Kac Vegas      | 100          | 2009 | The Hangover
Epoka lodowcowa | 85           | 2002 | Ice Age
Forrest Gump   | 142          | 1994 | null

(3 rows)
cqlsh:filmy>
```

Usuwanie wiersza przy użyciu Cqlsh:

Składnia:

DELETE FROM nazwa_tabeli WHERE warunek;

Przykład:

DELETE FROM komedia WHERE tytuł = 'Forrest Gump';

Jednoczesne wykonywanie operacji przy użyciu Cqlsh:

Składnia:

BEGIN BATCH

polecenie1

polecenie2

...

APPLY BATCH;

Przykład:

BEGIN BATCH

INSERT INTO komedia (tytuł, czas_trwania, rok, tytuł_oryginalny) VALUES ('Shrek', 90, 2001, 'Shrek');

UPDATE komedia SET czas_trwania = 81 WHERE tytuł = 'Epoka lodowcowa';

DELETE rok FROM komedia WHERE tytuł = 'Epoka lodowcowa';

APPLY BATCH;

```
maria_dev@sandbox-hdp:/home/maria_dev
cqlsh:filmy> BEGIN BATCH
... INSERT INTO komedia (tytuł, czas_trwania, rok, tytuł_oryginalny) VALUES ('Shrek', 90, 2001, 'Shrek');
... UPDATE komedia SET czas_trwania = 81 WHERE tytuł = 'Epoka lodowcowa';
... DELETE rok FROM komedia WHERE tytuł = 'Epoka lodowcowa';
... APPLY BATCH;
cqlsh:filmy> SELECT * FROM komedia;

tytuł          | czas_trwania | rok  | tytuł_oryginalny
-----+-----+-----+-----
Kac Vegas      | 100          | 2009 | The Hangover
Shrek          | 90           | 2001 | Shrek
Epoka lodowcowa | 81           | null | Ice Age

(3 rows)
cqlsh:filmy>
```

Złożone typy danych w Cassandraze:

- list – lista, która nie jest sortowana i może zawierać duplikaty.
- set – zbiór, który jest sortowany i nie zawiera duplikatów.
- map – para klucz-wartość.

Do naszej tabeli dodamy kolumnę 'nagrody' typu list:

ALTER TABLE komedia ADD nagrody list<text>;

UPDATE komedia SET nagrody = ['Oscar', 'Złote globy', 'Złote szpule'] WHERE tytuł = 'Forrest Gump';

```

maria_dev@sandbox-hdp:/home/maria_dev
cqlsh:filmy> UPDATE komedia SET nagrody = ['Oskar', 'Złote globy', 'Złote szpule'] WHERE tytuł = 'Forrest Gump';
cqlsh:filmy> SELECT * FROM komedia;

   tytuł   | czas_trwania | nagrody | rok | tytuł_oryginalny
-----+-----+-----+-----+-----
   Kac Vegas |      100 | null | 2009 | The Hangover
   Shrek |      90 | null | 2001 | Shrek
Epoka lodowcowa |      81 | null | null | Ice Age
Forrest Gump |      null | ['Oskar', 'Złote globy', 'Złote szpule'] | null | null

(4 rows)
cqlsh:filmy>

```

Aktualizowanie listy przy użyciu Cqlsh:

Usuniemy 'Złote szpule' z nagród filmu 'Forrest Gump'.

UPDATE komedia SET nagrody = nagrody - ['Złote szpule'] WHERE tytuł = 'Forrest Gump';

```

maria_dev@sandbox-hdp:/home/maria_dev
cqlsh:filmy> UPDATE komedia SET nagrody = nagrody - ['Złote szpule'] WHERE tytuł = 'Forrest Gump';
cqlsh:filmy> SELECT * FROM komedia;

   tytuł   | czas_trwania | nagrody | rok | tytuł_oryginalny
-----+-----+-----+-----+-----
   Kac Vegas |      100 | null | 2009 | The Hangover
   Shrek |      90 | null | 2001 | Shrek
Epoka lodowcowa |      81 | null | null | Ice Age
Forrest Gump |      null | ['Oskar', 'Złote globy'] | null | null

(4 rows)
cqlsh:filmy>

```

Teraz dodamy kolumnę 'obsada' typu set:

ALTER TABLE komedia ADD obsada set<text>;

UPDATE komedia SET obsada = {'Murphy', 'Diaz', 'Myers'} WHERE tytuł = 'Shrek';

SELECT * FROM komedia WHERE tytuł = 'Shrek';

```

maria_dev@sandbox-hdp:/home/maria_dev
cqlsh:filmy> ALTER TABLE komedia ADD obsada set<text>;
cqlsh:filmy> UPDATE komedia SET obsada = {'Murphy', 'Diaz', 'Myers'} WHERE tytuł = 'Shrek';
cqlsh:filmy> SELECT * FROM komedia WHERE tytuł = 'Shrek';

   tytuł | czas_trwania | nagrody | obsada | rok | tytuł_oryginalny
-----+-----+-----+-----+-----+-----
   Shrek |      90 | null | {'Diaz', 'Murphy', 'Myers'} | 2001 | Shrek

(1 rows)
cqlsh:filmy>

```

Aktualizowanie zbioru przy użyciu Cqlsh:

Dodamy Zamachowskiego do obsady Shreka.

```
UPDATE komedia SET obsada = obsada + {'Zamachowski'} WHERE tytuł = 'Shrek';  
SELECT obsada FROM komedia WHERE tytuł = 'Shrek';
```

```
maria_dev@sandbox-hdp:/home/maria_dev  
cqlsh:filmy> UPDATE komedia SET obsada = obsada + {'Zamachowski'} WHERE tytuł = 'Shrek';  
cqlsh:filmy> SELECT obsada FROM komedia WHERE tytuł = 'Shrek';  
  
obsada  
-----  
{'Diaz', 'Murphy', 'Myers', 'Zamachowski'}  
  
(1 rows)  
cqlsh:filmy>
```

Na końcu dodamy kolumnę 'recenzja' typu map:

```
ALTER TABLE komedia ADD recenzja map<text, text>;  
UPDATE komedia SET recenzja = { 'New York Times': 'Bardzo pochlebna recenzja', 'Wyborcza':  
'Arcydzieło' } WHERE tytuł = 'Kac Vegas';  
SELECT * FROM komedia WHERE tytuł = 'Kac Vegas';
```

```
maria_dev@sandbox-hdp:/home/maria_dev  
cqlsh:filmy> ALTER TABLE komedia ADD recenzja map<text, text>;  
cqlsh:filmy> UPDATE komedia SET recenzja = { 'New York Times': 'Bardzo pochlebna recenzja', 'Wyborcza': 'Arcydzieło' } WHERE tytuł = 'Kac Vegas';  
cqlsh:filmy> SELECT * FROM komedia WHERE tytuł = 'Kac Vegas';  
  
tytuł | czas_trwania | nagrody | obsada | recenzja | rok | tytuł_oryginalny  
-----  
Kac Vegas | 100 | null | null | {'New York Times': 'Bardzo pochlebna recenzja', 'Wyborcza': 'Arcydzieło'} | 2009 | The Hangover  
  
(1 rows)  
cqlsh:filmy>
```

Pozostaje nam zaktualizować recenzje gazety wyborczej:

```
UPDATE komedia SET recenzja = recenzja + {'Wyborcza': 'Bardzo zabawny film'} WHERE tytuł = 'Kac Vegas';  
SELECT recenzja FROM komedia WHERE tytuł = 'Kac Vegas';
```

```
maria_dev@sandbox-hdp:/home/maria_dev  
cqlsh:filmy> UPDATE komedia SET recenzja = recenzja + {'Wyborcza': 'Bardzo zabawny film'} WHERE tytuł = 'Kac Vegas';  
cqlsh:filmy> SELECT recenzja FROM komedia WHERE tytuł = 'Kac Vegas';  
  
recenzja  
-----  
{'New York Times': 'Bardzo pochlebna recenzja', 'Wyborcza': 'Bardzo zabawny film'}  
  
(1 rows)  
cqlsh:filmy>
```