Unterschiede zwischen SQL und NoSQL Datenbanken

Sie lernen…

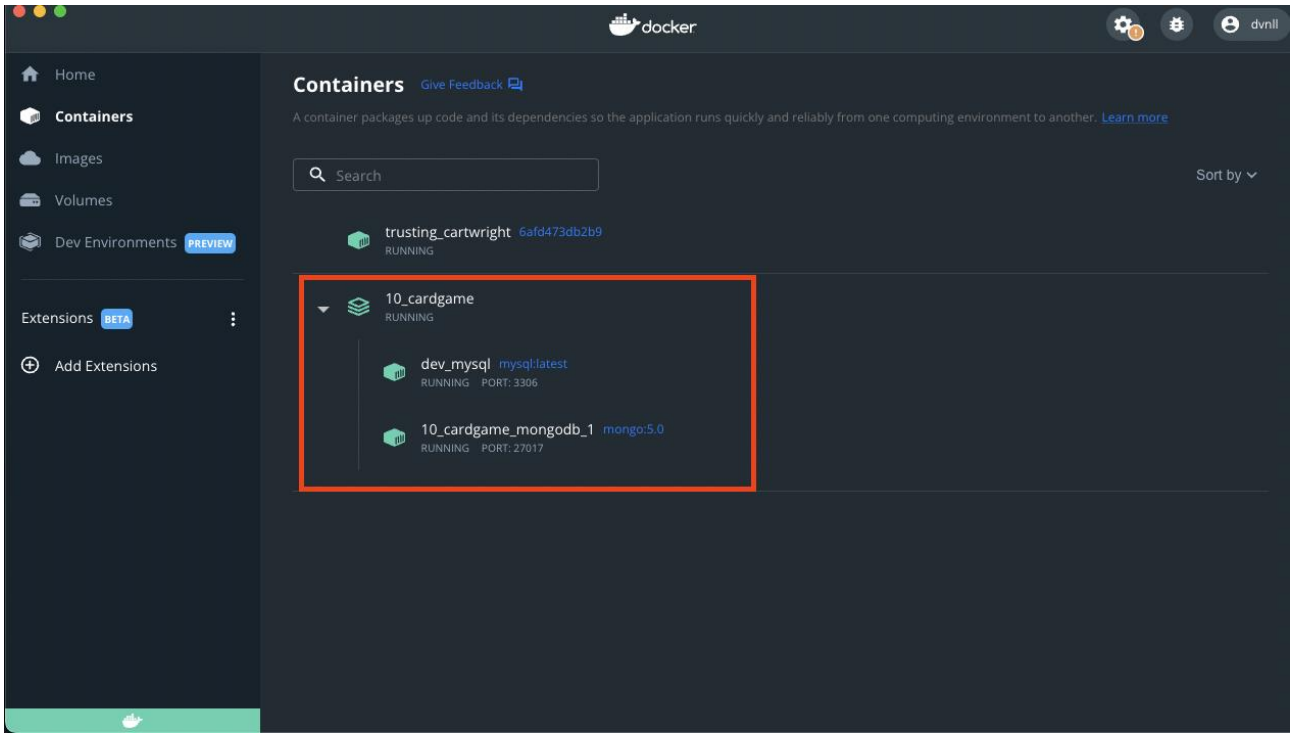
- ***Was die grundlegenden Unterschiede zwischen SQL Datenbanken und NoSQL Datenbanken sind***
- ***Wie man Datenbanken in eine Java-Applikation einbindet***

Vorbereitung

Die Basis bildet das Docker-Compose mit Mysql und mongoDB als Datenbanken und das Autoquartett-Programm, welches Sie aus dem Modul 319 kennen.

Starten Sie Ihre Umgebung mit Docker-Compose (im Ordner mit dem docker-compose.yml):

docker-compose up

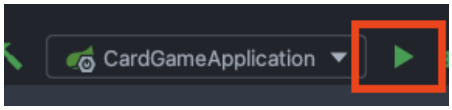


Hinweis: Port, User und Passwort sind abhängig von Ihrer Installation.
Standard dieser Vorlage sind diese:

Der User heisst: root.
Das Passwort heisst: root

Nachdem die Datenbanken gestartet sind, können Sie das Projekt im IntelliJ öffnen und warten, bis Maven alle Module geladen hat.

Danach können Sie die Applikation mit Hilfe von



Starten. Das Resultat sehen Sie auf der nächsten Seite:

Card Game

Spieler 1 und Spieler 2 haben je ein verdecktes Deck von Karten.

Spieler 1 deckt eine Karte auf und entscheidet sich für ein Attribut, das er setzt.

Spieler 2 deckt eine Karte auf und die Karte mit dem besseren Wert im Attribut gewinnt.

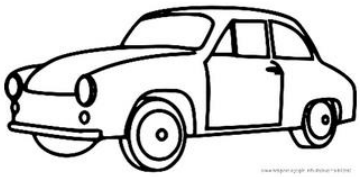
Die Verlierer Karte wird in das Deck des Siegers verschoben.

Und der Sieger zeigt seine nächste Karte...

Spieler 1

Noch 2 Karten im Deck

Leere Karte



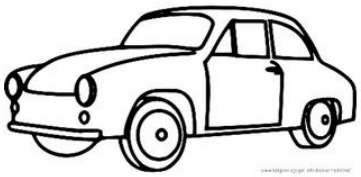
show left

move to right

Spieler 2

Noch 2 Karten im Deck

Leere Karte

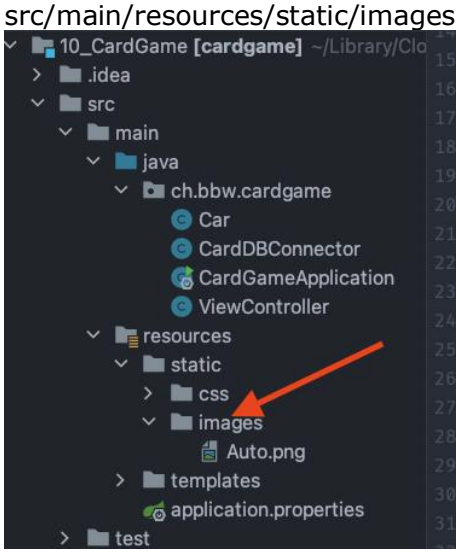


show right

move to left

1 Datenbank & Applikation kennenlernen

1. Öffnen Sie das file init.sql – der Syntax sollte Ihnen aus den Modulen 162 und 164 bekannt sein. Beantworten Sie die folgenden Fragen:
    a. Was macht dieses File?
    b. Wie sieht die Datenbank aus?
    c. Wie viele Autos sind in der Datenbank?
2. Ändern Sie die Bilder für die bestehenden Autos.
    a. Dafür müssen Sie zuerst welche finden, die zum Model passen (Google reicht für unsere Zwecke aus, brauchen keine speziellen Lizenzen).
    b. In einem zweiten Schritt fügen Sie die Bilder unter src/main/resources/static/images hinzu:



    c. Zuletzt müssen Sie noch die Datenbankeinträge anpassen, damit diese auf die richtigen Bilder zeigen. Ausgangslage:

WHERE ORDER BY

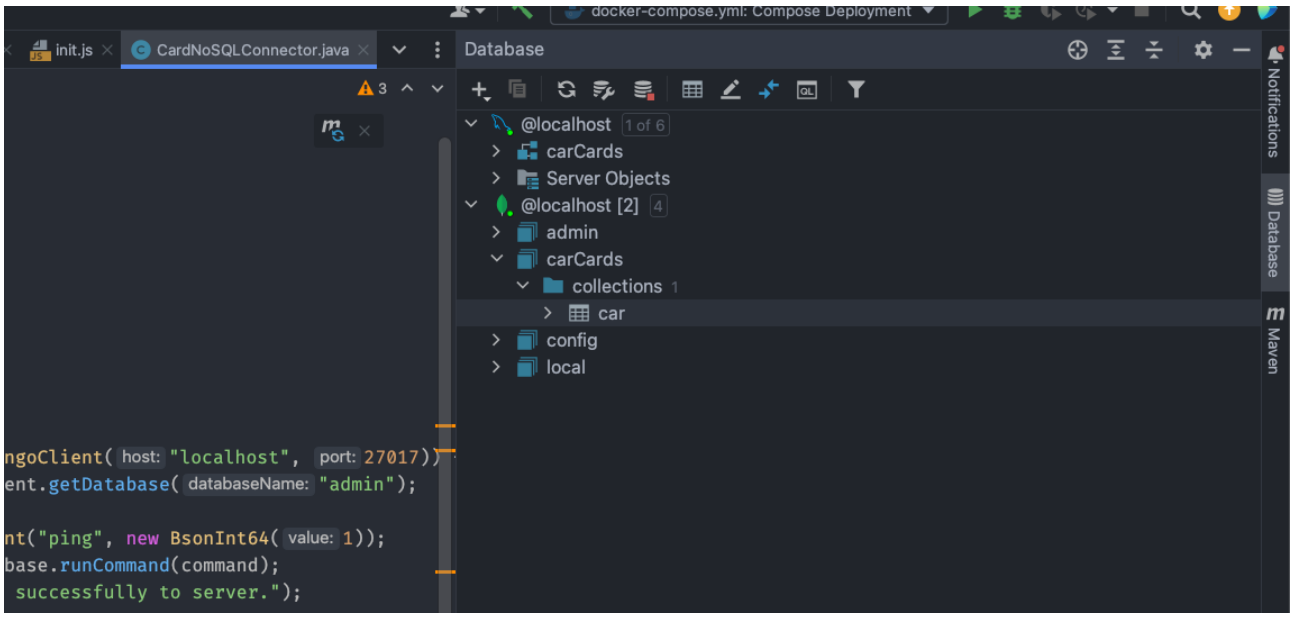
	id	imageUrl	tradeName	model	prize
1	1	images/Auto.png	Audi	Flaschback 300	50000
2	2	images/Auto.png	Opel	Manta SE	20000
3	3	images/Auto.png	VW	Golf GL	12000
4	4	images/Auto.png	Fiat	500	15000

    d. Laden Sie die Webseite neu, die Autos sollten nun Ihre Bilder anzeigen
3. Fügen Sie weitere Autos hinzu. Dabei muss immer eine gerade Anzahl Autos in der Datenbank vorhanden sein.

Sie kennen nun die Applikation und das Datenmodell. In einem nächsten Schritt machen wir uns mit dem neuen Konzept von NoSQL anhand einer anderen Datenbank vertraut: MongoDB

2 MongoDB Anbindung

Unter dem Port 27017 befindet sich die MongoDB. Die Datenbank kann mit DataGrip oder direkt im IntellIJ geöffnet werden (user: root, pw: root):



Auch hier gibt es eine «Tabelle» car. Allerdings heissen Tabellen in MongoDB «Collections».

1. Verbinden Sie sich mit der MongoDB und vergleichen Sie die Inhalte der Collection Car mit der Tabelle Car aus der vorherigen Aufgabe.
    a. Was stellen Sie fest?
    b. Wo liegen die Unterschiede?
2. Schauen Sie sich nun das File init.js an. Es macht das gleiche, wie das init.sql
    a. Aber was ist anders?
    b. Kennen Sie diese Syntax?

3 MongoDB Hintergrund

In MongoDB wird nicht von Tabellen und Datensätze gesprochen, sondern von Collections mit Dokumenten.

Jeder «Datensatz» ist hier ein Dokument, weshalb man die MongoDB auch als Dokumentenbasierte Datenbank bezeichnet.

In der NoSQL Welt gibt es auch ganz andere Modelle wie:

- ***Key-Value pair***
- ***Column based***
- ***Document Oriented***
- ***Graph databases***

Wir werden uns später im Modul mit den anderen Modellen befassen.

MongoDB speichert die Dokumente im JSON Format. JSON steht dabei für JavaScript Object Notation. In den Web-Modulen werden Sie noch öfters auf dieses Format treffen, deshalb hier eine kleine Einführung:

Ein Objekt (oder «Datensatz») wird mit {} begonnen und beendet:

Die Attribute sind dabei «key-value» Paare mit folgendem Syntax:

''attributName'':''attributWert''

Bei Zahlen können beim Attributwert die Hochkommas weggelassen werden.

Arrays werden mit [] angegeben:

[1,2,3,4,5] ist ein Array/eine Liste mit den Zahlen 1-5.

Arrays können ähnlich wie in Java nicht nur Zahlen, sondern auch Texte und wiederum andere Objekte beinhalten.

4 MongoDB in Applikation einbinden

In diesem Schritt werden wir nun die MySQL Datenbank durch die MongoDB ersetzen. Die Applikation sollte wie gewöhnlich weiter funktionieren.

1. Informieren Sie über das JSON Format
    a. Wo findet es überall Anwendung? (Top Usecases)
    b. Wie sieht der Syntax aus? (Repetition Aufgabe 3)
    c. Was sind die üblichsten Datentypen und deren Schreibweise?
    d. Erstellen Sie ein Dokument me.json mit einem JSON Objekt, welches Attribute zu Ihrer Person festhält (Name, Nachname, Geburtsdatum, etc…)
2. Wechseln Sie in der Applikation die DB Verbindung zu NoSQL. (Unter ViewController.java Zeile 34)
    a. Wie heisst die Klasse, die Sie nun nutzen müssen?
    b. Worin unterscheidet sich diese Klasse von der vorherigen?
    c. Was bleibt gleich oder ähnlich?
3. Nun müssen Sie auch hier neue Autos hinzufügen. Erstellen Sie insgesamt 4 neue Autos in Ihrer MongoDB. 2 davon über ein GUI und 2 davon direkt mit Code in einer Console. Unter 3. Finden Sie Hilfe zum JSON Syntax, im init.js finden Sie Beispiele für das hinzufügen.
    a. Welche Unterschiede und Gemeinsamkeiten können Sie feststellen?
    b. Wo sehen Sie Vorteile/Nachteile der verschiedenen Syntax?