

noSQL mit MongoDB

Nach den relationalen Datenbanksystemen, die wir am Beispiel MariaDB erkundet haben, wollen wir uns nun mit noSQL-Datenbanksystemen auseinandersetzen und widmen uns in dieser Übung speziell dem Dokumenten-Datenbanksystem **MongoDB** [1]. MongoDB speichert Daten als JSON-Objekte [2].

MongoDB in Betrieb nehmen

Um MongoDB verwenden zu können, müssen wir es verfügbar machen.

Folgende Möglichkeiten bieten sich an:

1. Im Codespace verwenden

Wenn Sie wie in Übung 1 einen Codespace des Repositories verwenden wollen, dann aktualisieren Sie bitte Ihren Fork des DBTI25-Repositories [3] (Sync Fork auf der Eingangsseite Ihres Forks aktivieren).

Bitte stellen Sie sicher, dass Sie alle Änderungen, die Sie im bisherigen Codespace gemacht haben, in Ihren Fork gepushed haben (`git commit` und `git push`), damit alle Ihre Änderungen im Repository stehen, der bisherige Codespace gelöscht und durch einen Codespace mit MongoDB ersetzt werden kann.

Starten Sie einen neuen Codespace und inspizieren Sie das `Dockerfile` unter `.devcontainer`, um herauszufinden, wie MongoDB unter Linux (Ubuntu) installiert werden kann.

Starten Sie bitte den MongoDB-Server: `service mongod start` und dann den MongoDB-Client: `mongosh`

2. MongoDB selbst installieren

Laden Sie bitte die aktuelle Version von MongoDB unter <http://www.mongodb.org/downloads>). Entpacken Sie bitte die Distribution lokal und legen Sie bitte ein Datenverzeichnis `$HOME/data` bzw. `H:\data` an.

Dann starten Sie bitte:

(a) den MongoDB-Server:

`mongod --dbpath=$HOME/data` (bzw. `mongod --dbpath H:\data`)

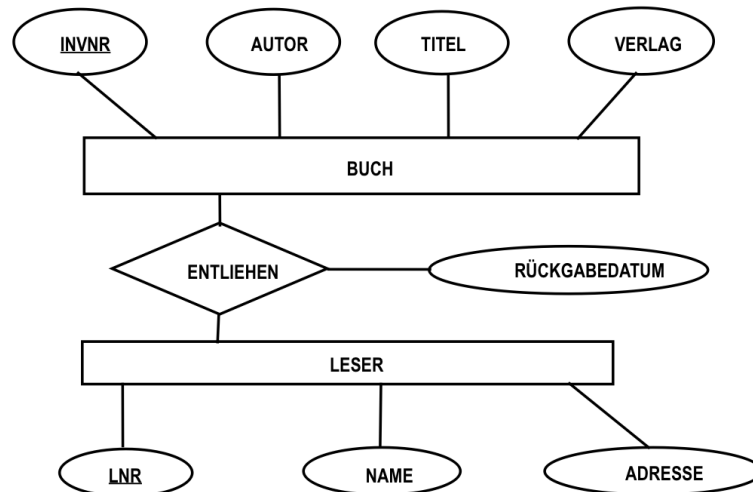
und

(b) den MongoDB-Client: `mongosh`

MongoDB-CRUD

Informationen zur interaktiven Benutzung von MongoDB findet sich im Buch *The Little MongoDB Book* [4] von Karl Seguin und eine Zusammenfassung der MongoDB-Befehle auf einer Seite im Cheat-Sheet [5] von Ovidiu Anical.

Wir stellen uns ein einfaches Datenmodell einer Bibliothek vor, in dem *Bücher* durch Inventarnummer `INVNR`, Autor, Titel und Verlag dargestellt werden. Außerdem sollen Daten über *Leser* (Lesernummern `LNR`, Name und Adresse) festgehalten werden. Zwischen Büchern und Lesern besteht die 1:N-Beziehung *Entliehen*, für die zusätzlich auch noch das Rückgabedatum festgehalten wird.



1. Bitte lösen Sie die folgenden Aufgaben interaktiv im Mongo-Client.

- (a) Legen Sie bitte ein JSON-Dokument für ein Exemplar des Buchs „Die Känguru-Chroniken: Ansichten eines vorlauten Beuteltiers“ von Marc-Uwe Kling (Ullstein-Verlag) und ein Dokument für den Leser Friedrich Funke, Bahnhofstraße 17, 23758 Oldenburg an. (Stichwort `insertOne`)

Legen Sie bitte mindestens fünf weitere Bücher und Leser-Dokumente Ihrer Wahl an. (Stichwort `insertMany`)

Wie lässt sich der Sachverhalt, dass ein Leser ein bestimmtes Buch ausgeliehen hat, ausdrücken?

- (b) Suchen Sie in Ihrer Mongo-Datenbank nach einem Buch mit dem Autor Marc-Uwe Kling. (Stichwort `find`)
- (c) Ermitteln Sie, wie viele Bücher Ihre Datenbank verwaltet. (Stichwort `countDocuments` oder `count`)
- (d) Ermitteln Sie bitte alle Leser, die mehr als ein Buch ausgeliehen haben, absteigend sortiert nach Anzahl der entliehenen Bücher.
- (e) Welche Operationen auf der Datenbank muss man ausführen, um widerzuspiegeln, dass ein Leser ein Buch ausleiht bzw. zurückgibt? Lassen Sie bitte Friedrich Funke das Känguru-Buch ausleihen und wieder zurückgeben.
- (f) Statt getrennte JSON-Collections für Leser, Bücher und Entliehen zu haben, können entliehene Bücher auch in einer Liste innerhalb von Leser-Dokumenten verwaltet werden.

Legen Sie bitte (zusätzlich) ein Leser-Dokument für Heinz Müller, Klopstockweg 17, 38124 Braunschweig an, der das obige Känguru-Chroniken-Buch und auch noch „Der König von Berlin“ von Horst Evers (Rowohlt-Verlag) ausgeliehen hat.

Welche Vor- und Nachteile hat diese Art die Daten abzulegen?

- (g) Welche Operationen auf der nun anders strukturierten Datenbank muss man ausführen, um widerzuspiegeln, dass ein Leser ein Buch ausleiht bzw. zurückgibt? Lassen Sie bitte Heinz Müller das Känguru-Buch zurückgeben. Friedrich Funke soll es wieder ausleihen.

2. Auch MongoDB lässt sich aus Java ansprechen. Bitte machen Sie sich damit vertraut, wie man die MongoDB-CRUD-Operationen aus Java durchführt. Details dazu finden sich in der MongoDB-Dokumentation unter [6]. Die Entsprechung eines JDBC-Treibers für MongoDB nennt sich *Java Sync Driver*. Er ist im DBTI25-Repository für Maven im Verzeichnis Mongo schon im `pom.xml` als Abhängigkeit vermerkt.

Bitte wiederholen Sie die Teilaufgaben (a) bis (g) aus Aufgabe 1. aber verwenden Sie nun nicht mehr `mongosh` sondern schreiben Sie dafür ein Java-Programm. Am besten strukturieren Sie Ihr Programm als JUnit-Tests, die jeweils auf einer passend initialisierten Datenbank arbeiten.

Die Firma-Datenbank in MongoDB

Wir wollen nun sehen, wie eine relationale Datenbank mit Tabellen und Datensätzen in eine MongoDB-Datenbank mit Collections und Dokumenten übergehen kann.

1. Überlegen Sie, wie die bisher als relationale Datenbank vorliegende Firma-Datenbank geeignet als Sammlung von MongoDB-Collections zu repräsentieren wäre. Muss die Datenbank in Normalform stehen und müssen alle Zusammenhänge zwischen Entitäten über Fremdschlüsselbeziehungen repräsentiert werden?
2. Bitte dokumentieren Sie Ihre Überlegungen in einem Dokument *Konzept zur Darstellung der Datenbank Firma in MongoDB*.
3. Schreiben Sie bitte ein Java-Programm, dass die Firmen-Datenbank aus MariaDB ausliest und in der von Ihnen entworfenen Struktur in MongoDB ablegt.
4. Bitte erzeugen Sie für Ihr Konzept und für Ihr Java-Programm einen Pull-Request, um diese Aufgabe einzureichen. Die Abgabe dieser Aufgabe wird beurteilt und führt bei erfolgreicher Bearbeitung zu einem Noten-Bonus in der anstehenden Klausur von bis zu 0,3.
5. Wiederholen Sie bitte die CRUD-Operationen aus Übung 1 nun für Ihre in MongoDB abgelegte Firma-Datenbank:
 - (a) CREATE: Erzeugen Sie einen neues Dokument in der Collection `personal` für den neuen Mitarbeiter **Henrik Krause** in der `Produktion`, mit Personalnummer `417`, Gehaltsstufe `it1` und in der `tkk` krankenversichert ist.
 - (b) READ: zeigen Sie bitte alle Dokumente der Collection `personal` an.
 - (c) UPDATE: Ändern Sie die Collection `gehalt` so, dass das Gehalt in der Gehaltsstufe `it1` um 10% erhöht wird.
 - (d) DELETE: Der Mitarbeiter **Lutz Tietze** scheidet aus der Firma aus. Bitte löschen Sie sein Dokument aus der `Personal`-Collection.
 - (e) Beantworten Sie die Frage: *Welche MitarbeiterInnen arbeiten in der Abteilung Verkauf?*

[1] <https://www.mongodb.org>

[2] <https://www.json.org/json-en.html>

[3] <https://github.com/fh-wedel/DBTI25>

[4] <https://lms.fh-wedel.de/mod/resource/view.php?id=56922>

[5] <https://lms.fh-wedel.de/mod/resource/view.php?id=56923>

[6] <https://www.mongodb.com/docs/drivers/java/sync/current/crud/>