Die Mongo DB

Vorstellungsrunde

- Name
- Rolle im Unternehmen
- Themenbezogene Vorkenntnisse
- Konkrete Problemstellung
- Individuelle Zielsetzung

Umgebung

- Eigener Rechner
  - Download der Mongo-Community Edition und Installation jeweils für benutztes Betriebssystem
    - https://www.mongodb.com/try/download/community
  - Alternativ
    - Docker-Image
      - https://hub.docker.com/_/mongo

Seminarzeiten

JAVACREAM
Training
Consulting
Projectmanagement

8:00 - 16:00

Mittagspause: 11:45 - 13:00

Pausen: 9:45 - 10:15, 14:15:14:30

Ausgangssituation

Kleiner Hinweis

- Name der Datenbank
  - MongoDB -> “Humengous” = gigantisch/enorm
    - “Menges-DB”

Historie: “NoSql-Bewegung”

- 2003
  - “Big Data”
  - Besser: “Big & Fast Data”
- Relationale Datenbanksysteme hatten im Endeffekt Skalierungsprobleme
  - Hochskalieren erfolgt vertikal, mehr CPU, mehr RAM, mehr Storage
- NoSql erfindet neue Datenbank-Typen
  - Key-Value
  - Dokumenten-orientiert -> MongoDB
  - Graphen-orientiert
  - Spalten-orientiert

NoSql revisited

- Der Name “NoSql”
  - Besser “NoRelational”
    - Präziser: “No” = “Not Only”
- Die Umsetzung eines abstrakten EntityModells erfolgt nicht zwangsläufig durch ein relationales Modell
- NoSQL-Kategorien
  - Key-Value-Modell
    - select value from store where key=’key1’
  - Dokumenten-orientierte Modellierung -> Details später
  - Graphen-Modell
    - Tausende von Joins sind ohne relevanten Performance-Verlust zu realisieren
  - Spalten-orientierte Modellierung
    - basiert auf einem Objekt-Modell
  - Relationales Modell
    - Fremdschlüssel und Verknüpfungstabellen

Was ist Dokumenten-orientiert?

- Jedes Dokument hat eine eindeutige Dokumenten-ID
  - Weltweite Eindeutigkeit ist zu garantieren
    - URI: Zugriffsprotokoll://host/collection/id
- Beziehungen zwischen Dokumenten erfolgen über eine Verlinkung
  - = Angabe der URI des “anderen” Dokuments
- Dokumente haben ein Schema, eine Struktur
  - “Schema on Read”
    - Eine Abfrage legt das benötigte Schema fest und bekommt dann auch nur die Dokumente, die dem Schema genügen
  - “Schema on Write”
    - Validierung beim Schreib-Vorgang, eine Datensenke akzeptiert nur Daten, die einer vorgegebenen Struktur entsprechen

Dokumenten-orientierte Datenbanken

- Breite Produktpalette
  - Open Source-Produkte, z.B. Couchbase
- Kommerzielles Produkt: MongoDB
  - Community-Edition
    - Frei einsetzbar, alle kein offizieller Support
    - Bei uns im Training
  - Lizenzpflichtige Version
    - Support
    - Tooling
    - …
  - MongoDB Atlas
    - Cloud-basierte Lösung

Training

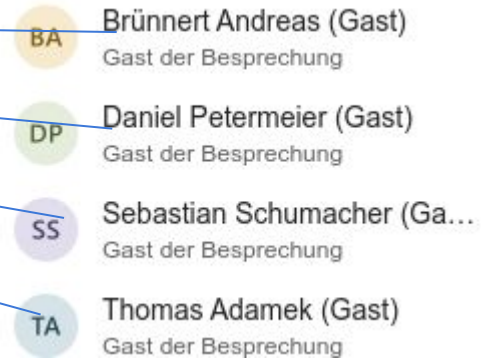
VKB

Aufbau unserer Trainings-Umgebung

- Rechner mit Docker-Runtime
  - Integrata-Cegos stellt hierfür Ubuntu-basierte Deskmate-Maschinen bereit

Remote Rechner der Integrata-Cegos

Deskmate-User	Deskmate Passwort	Ubuntu User-ID	Passwort
tn28.raum01@integrata-cegos.de	4023_tn28	sl01	sl01
tn29.raum01@integrata-cegos.de	4023_tn29	sl01	sl01
tn30.raum01@integrata-cegos.de	4023_tn30	sl01	sl01
tn31.raum01@integrata-cegos.de	4023_tn31	sl01	sl01
tn32.raum01@integrata-cegos.de	4023_tn32	sl01	sl01
tn33.raum01@integrata-cegos.de	4023_tn33	sl01	sl01
Deskmate-Link:	https://integrata-cegos.deskmate.me/		



Support:
Zentrale IT
Telefon:
+49 711 62010 355
ZentraleIT@integrata-cegos.de

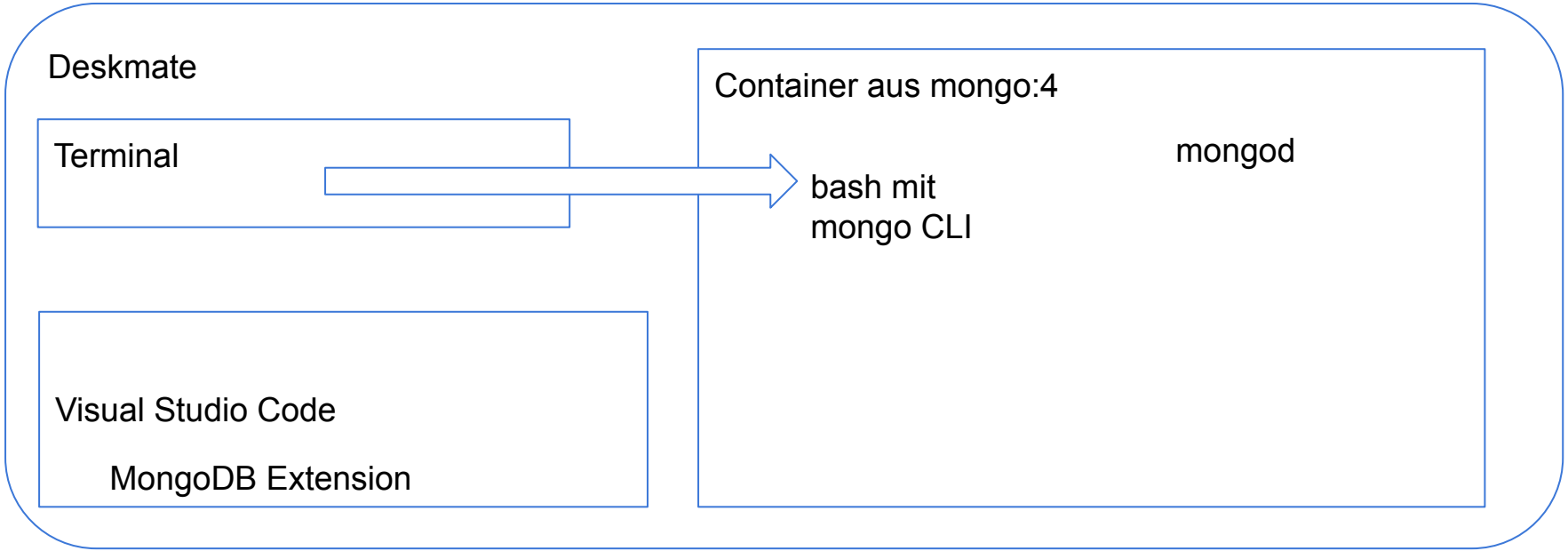
VORSICHT

- BITTE KEINE AKTUALISERUNGEN AKZEPTIEREN!

Mongos und Docker

- docker create --name mongodb -p 27017:27017 mongo:4
- docker exec -it mongodb /bin/bash
  - mongo
  - Interaktive Shell
- Dazu Visual Studio Code mit Mongos Extension

Umgebung


Deskmate
Terminal
Visual Studio Code
MongoDB Extension
Container aus mongo:4
bash mit
mongo CLI
mongod

MongoDB First Contact

Das API der MongosDB

- **Datenbank-Server mit Server-Socket auf 27017**
- **Befehlssatz ist JavaScript**
  - Das Dokumenten-Format in MongosDB ist JSON
- Andere APIs
  - REST-Schnittstelle ist vorhanden, allerdings eigentlich nur in der Lizenz-Version
  - Besser: GraphQL
- MongoDB-Treiber für andere Programmiersprachen sind vorhanden
  - Java
  - C#
  - Python

Exkurs: Datenformate für Dokumente

- XML ist ein **akademisch** sehr geeignetes Datenformat für Dokumente
  - Schema
  - Link-Element
- JSON-Format
  - de facto Standard im Internet
    - Für Web-Anwendungen ist damit keine Transformation nötig
  - Es fehlt
    - Schema-Beschreibung
    - Standardisierte Angabe für Links
  - MongoDB nutzt JSON in einer Erweiterung
    - BSON-Spezifikation
    - Ermöglicht Schemata und Verlinkungen

Operationen auf Dokumenten

- CRUD-Operationen
  - Create
    - insertOne(object)
  - Read
    - find
      - Mit Abfrage-Lriterien
  - Update
    - saveOrUpdate, Details hierzu später
  - Delete
    - deleteMany
    - deleteOne

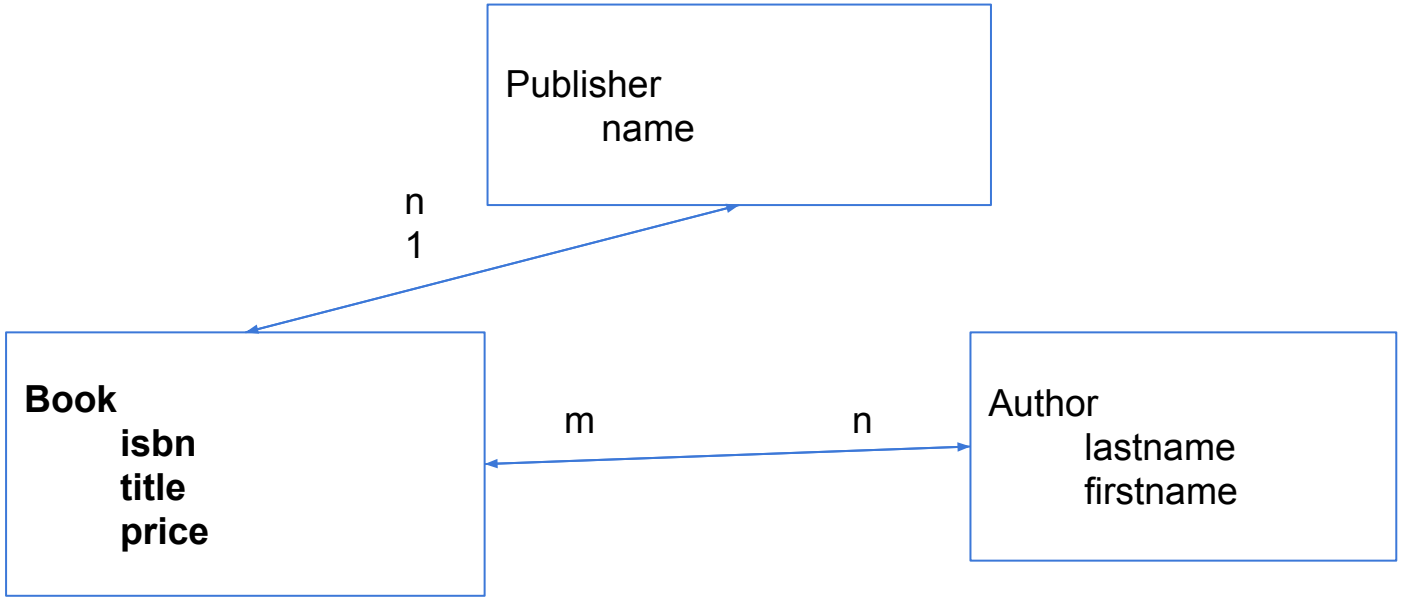
insertOne

- Parameter ist das zu erzeugende Object
- MongoDB erzeugt intern eine relativ eindeutige Id
  - Diese ObjectID ist nur im Zusammenspiel mit host/collection weltweit eindeutig

Exkurs: Notwendige Anzahl von Collections

- Ansatz 1: Collections werden wie Tabellen verwendet
  - In den allermeisten Fällen total falsch
    - Dies führt zu einem schlechten Dokumenten-Modell mit viel zu viel Joins zwischen verschiedenen Dokumenten
- Ansatz 2:
  - Es genügt pro Datenbank-Instanz eine einzige Collection
    - Das ist prinzipiell völlig in Ordnung
      - Schema on Read ist hocheffizient implementiert
- In der Realität werden Collections aber noch zusätzlich benutzt
  - Halten von Konfigurationseinstellungen
  - Berechtigungskonzept
  - Übersichtlichkeit

Das Entity-Modell für unser Training



Step 1

ToDo: Buch-Daten erzeugen

- Schreiben Sie eine Funktion, die ein paar Test-Bücher in die Datenbank legt
- Ablauf
  - drop der Collection publishing
  - create publishing
  - Erzeugen der Test-Daten
  - CHECK: DB, Collection und Daten sind vorhanden

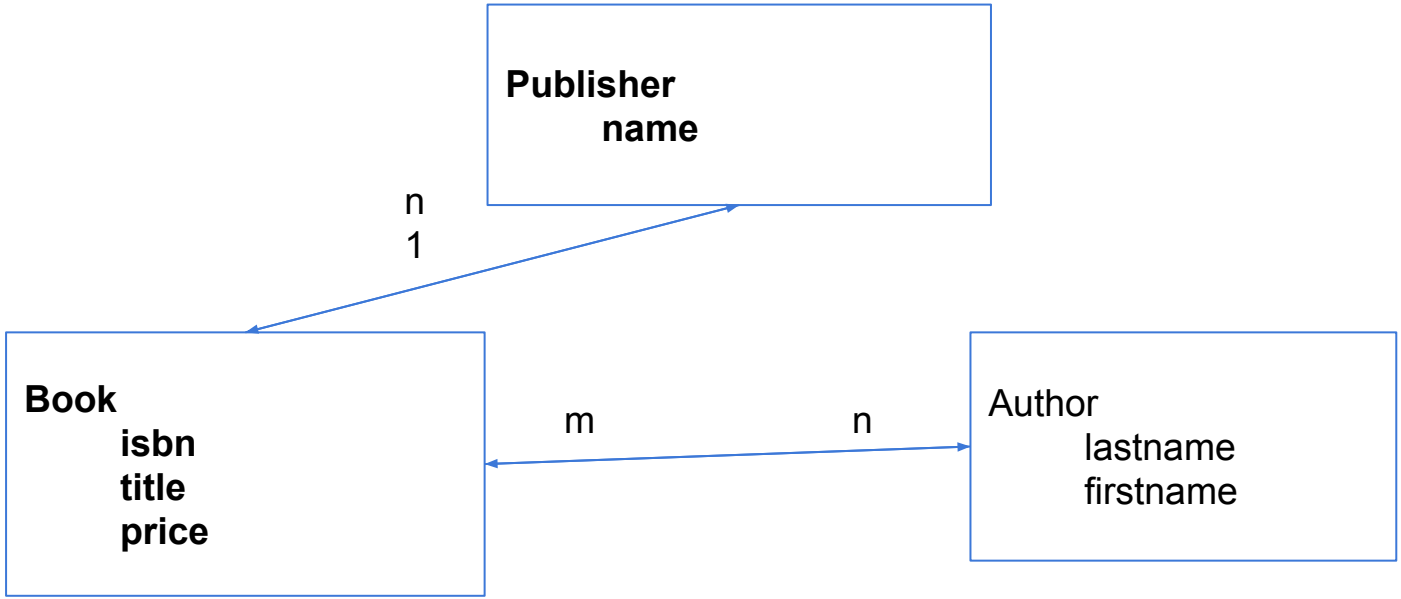
Criteria-Objekte

- Such-Operationen
  - find()
  - find(criteriaObject)
    - criteriaObject: Ein JSON-Objekt bzw. in unserer Umgebung ein JavaScript-Objekt
- Hinweis
  - _id ist auch ein Kriterium, allerdings: ObjectId(“hash”)
- Exkurs: zu den ObjectIds
  - Bestandteile
    - Timestamp
    - 5 Zeichen sind Prozess-abhängig
    - Interner Counter
-

Exkurs: Abfragesprachen

- SQL?
  - Passt nicht auf Dokumenten-basierte Abfragen
- N1QL
  - gesprochen: “Nickel”
  - SQL-Erweiterung als Standard für eine Dokumenten-basierte Abfragesprache
  - MongoDB unterstützt N1QL nicht
- Statt dessen
  - etwas proprietäres

Das Entity-Modell für unser Training



Step 2

ToDo

- Publisher zusätzlich mit einem Address-Dokument {city: “”, street: “”)
  - {“address.city”: “Berlin”}
- Book zusätzlich mit einer Kurzbeschreibung (description)
  - {$text: {$search: “MongoDb”}
- zusätzlich: auch die Projektion kann ein $elemMatch enthalten
  - Im Endeffekt ein Subquery auf die Ergebnis-Liste

Abbildung in Dokumente

- Embedded Documents
  - Besteht also aus mehreren Dokumenten, die sich allesamt in einem Haupt-Dokument befinden
    - Als Aggregat sind alle Sub-Dokumente im Lebenszyklus an das Hauptdokument gebunden
    - Nur das Haupt-Dokument hat eine Object-ID
  - “Sawitzki”: Atomar Document
- Verlinkungen auf andere Dokumente
  - Die klassische Dokumenten-orientierte Modellierung

Dokumenten-orientierte Datenbanken

- Primärfokus
  - Embedded Documents
    - Verwaltung solcher Dokumente ist äußerst effizient möglich
  - Transaktionssicherheit auf Embedded Documents ist trivial
- Bei verlinkten Dokumenten erfolgt das Joinen
  - auf Client-Seite (der ursprüngliche Ansatz)
  - auf Server-Seite (mittlerweile unterstützt, aber aus Performance-Sicht nicht unbedenklich)
  - Transaktionssicherheit ist entweder gar nicht gewährleistet (Client-Joins) oder sehr aufwändig

MongoDb

- Ursprüngliche Abfrage-Sprache bezieht sich auf Embedded Documents
- Später
  - Map-Reduce
  - Aggregate-Pipeline

Im Training nächster Schritt

- Embedded Publisher-Books-Dokumente
- Klassische Formulierung von Kriterien
- Programmatisch ist ein Embedded Document nichts anderes als ein Objekt-Geflecht
  - Als Literal {name: “Springer”, books: [{isbn: “ISBN1”, …}]
  - let publisher = {name: “Springer”, books: []}
  - publisher.books[0] = book1
- ToDo
  - 2 Publisher (“Springer”, “Addison”)
  - Springer bekommt die 5 Bücher, Addison ein Demo-Buch