

# Advanced Database Techniques - Aufgabe 1 - Entwicklung eines Datenbankschemas

## Jochen Schmidt

Student, Informatik  
FH Würzburg-Schweinfurt  
Fakultät Informatik  
Matrikelnummer: 511xxx

Email: jochen.schmidt1@fhws.student.de

## Lennard Rose

Student, Informatik  
FH Würzburg-Schweinfurt  
Fakultät Informatik  
Matrikelnummer: 511xxx

Email: lennard.rose@fhws.student.de

## Moritz Zeitler

Student, Informatik  
FH Würzburg-Schweinfurt  
Fakultät Informatik  
Matrikelnummer: 5118094

Email: moritz.zeitler@fhws.student.de

*Im Kurs 'Advanced Database Techniques' an der FHWS soll im Laufe des Kurses eine Datenbank, von den Studenten gewählt auf Herz und Nieren geprüft werden. Die Gruppe um Lennard Rose, Jochen Schmidt und Moritz Zeitler entschied sich daraufhin Elasticsearch als Ebene diese zu verwenden, unter dem Bewusstsein das Elasticsearch eigentlich keine klassische Datenbank ist sondern eher eine Suchmaschine. Bei der ersten Aufgabe soll nun für ein freigeschätztes Thema ein Datenbankschema erstellt werden. Dies wird im folgenden genauer beleuchtet.*

## 1 Datenbank Thema: Fitness und Ernährungs Tracker

### 1.1 Verwaltung

### 1.2 Trainingspläne

### 1.3 Ernährung

## 2 Datenbankschema-Anpassungen für Elasticsearch

Um das bereits vorgestellte Datenbankschema besser verwendbar zu machen, in Bezug auf Elasticsearch und Kibana, müssen einige Anpassungen vorgenommen werden. Die aus Relationalen Datenbanken bekannten 'joins', und somit das Erstellen von vielschichtigen Queries über mehrere Tabellen, vergleichbar mit Indizes in Elasticsearch, müssen größtenteils entfernt werden. SQL-Style joins sind in Elasticsearch so kostenintensiv das sie eigentlich grundsätzlich verboten sind [1]. Elasticsearch betrachtet die Welt daher auf einer eher horizontalen Ebene [2], Objekte die in Elasticsearch gespeichert werden sollten daher geflatten werden (aus dem englischen 'flatten').

Hier gibt es verschiedene Optionen um diese flache Welt zu realisieren. Unter anderem könnte zum Beispiel auf Applikationsseite eine Art 'join' über mehrer Indizes implementiert werden.

Da sich der Kurs aber nur im die Datenbank drehen soll, stellt dies keine ernsthafte Option dar. Hingegen sinnvoller sind hier entweder sogenannte 'nested objects' [3] oder eine Art 'Parent-Child-Beziehung' zwischen den Objekten [4]. Grundsätzlich gilt aber für beide Optionen das die Daten im gleichen Index abgelegt werden. Dies bedeutet das zwar die Abfragen an sich komplexer werden, aber nicht über mehrere Indizes hinweg gesucht wird.

## 2.1 Kibana spezifische Änderungen

### 3 Fazit

### 4 Lessons Learned

### References

- [1] Reference, E., 2021. Joining queries.
- [2] Gormley, C., and Tong, Z., 2015. *Elasticsearch: The Definitive Guide - A Distributed Real-Time Search and Analytics Engine*. "O'Reilly Media, Inc.", Sebastopol.
- [3] Gormley, C., and Tong, Z., 2015. *Elasticsearch: The Definitive Guide - A Distributed Real-Time Search and Analytics Engine*. "O'Reilly Media, Inc.", Sebastopol.
- [4] Gormley, C., and Tong, Z., 2015. *Elasticsearch: The Definitive Guide - A Distributed Real-Time Search and Analytics Engine*. "O'Reilly Media, Inc.", Sebastopol.