Projekt NoSQL

Ein Beispiel anhand einer fiktiven Social-Media Plattform mit MongoDB

Gruppenmitglieder:

Mina Schmachtel, Julian Reith, Jakob Molfenter
und Daniel Kern

GitHub Link:

https://github.com/3x3cut0r/ProjektDatenbankNoSQL

NoSQL Datenbanken

Die Gruppe hat sich für eine MongoDB entschieden. MongoDB ist eine Dokumentenbasierte Datenbank, die in C++ geschrieben ist. Daten werden unstrukturiert oder halbstrukturiert in Dokumenten gespeichert. Als strukturierte Datenbanken zählen relationale Datenbanken, da sie ihre Einträge klar strukturiert in Tabellen speichern. Auch die Spalten und welche Werte diese beinhalten, werden im Voraus definiert. Die Beziehungen zwischen den Tabellen entsprechen dabei einem vordefinierten Schema.

Ein klassisches Beispiel für halbstrukturierte Daten sind JSON ähnliche Dateien in den die Werte als Key-Value-Pair (Schlüssel-Werte-Paar) gespeichert werten. Eine solche Art von Datenspeicherung verwendet MongoDB.

Daten werden als Key-Value-Pair in Dokumente geschrieben und die Dokumente in sogenannten Collections (Sammlungen) zusammengefasst.

Durch diese Art der Datenspeicherung ist die MongoDB flexibel in dem was gespeichert werden kann. Es kann ohne Anpassungen ein weiterer Key-Value-Pair zu einem Dokument hinzugefügt werden, welches ein bis dahin unbekanntes Attribut beinhaltet. Auch spielen Datentypen keine Rolle da sie in der Datenbank nicht definiert werden müssen. So kann ein Dokument das bisher nur Texte und Zahlen enthalten hat, ohne Anpassung des Schemas auch Bilder oder Videos speichern. Deshalb ist bei MongoDB von einer “schema-less” Datenbank die Rede. So können auch Objekte eines Dokumentes weitere Objekte beinhalten.

Umsetzung

Die Gruppe hat sich zur Veranschaulichung der Vorteile für einen Beispielcode entschieden, der eine Social-Media Plattform rudimentär abbildet. Betrachtet man die Speicherung von Posts, so können diese sehr unterschiedlich ausfallen. Ein Post (wird im Code als “Tweet” bezeichnet) kann nur aus Text, einem Datum und einer Überschrift bestehen. Er kann aber auch ein Bild und Likes beinhalten. Möchte man zu Beginn nur Texte in den Tweets zulassen und Bilder erst zu einem späteren Zeitpunkt implementieren, so müsste das Schema der relationalen Datenbank angepasst werden.

Bei einer MongoDB kann das Bild entweder als weiterer Key-Value Eintrag, in einem Dokument, das den Tweet beinhaltet, gespeichert werden. Soll zum Bild noch ein Alternativtext (alt) und eine ID gespeichert werden so kann es als weiteres Objekt innerhalb des “Tweet -Objektes” gespeichert werden

Vorteil Flexibilität

Bei der Einführung neuer Funktionen in die Anwendung muss sich bei einer relationalen (SQL-)Datenbank ein neues Schema überlegt werden. Die Datenbank benötigt eine neue Struktur, eventuell neue Relationen bei Einführung neuer Tabellen und muss abschließend erneut auf Konsistenz hin überprüft werden. Dies fällt bei einer NoSQL-Datenbank ohne feste Datenstruktur weg, da neue Attribute, wie im Beispiel bei Version 1.1 “likes” oder “dislikes”, einfach in vorhandene Dokumente ergänzt werden können. Ein neues Schema, wenn man dies bei einer No-SQL-Datenbank so nennen darf, kann somit “on-the-fly" erstellt und verwendet werden.

Um die Flexibilität der Datenbank deutlich zu machen, wächst der Funktionsumfang der Social-Media Plattform mit der Zeit. Zum Zeitpunkt des Launches (Version 1.0) der Plattform sind lediglich Tweets, die einen Titel und einen Text beinhalten und Retweets, die sich auf einen vorhandenen Tweet beziehen, möglich. Beispiele hierfür sind im Schema der Datenbank zu finden (docker/mongodb/conf/00-schema.js). Es können weitere Tweets der Version 1.0 über eine Postman-Collection im Ordner “tweets v1.0” erstellt werden. Alle in der Datenbank gespeicherten Tweets können mit dem Postman-Request “tweets/fetchAll” abgerufen werden.

Schon nach kurzer Zeit wird von den Benutzern die Funktion des Likens und Dislikens gefordert. Diese Funktion soll auf unserer Plattform mit der Version 1.1 implementiert werden. In der Postman-Collection im Ordner “tweets v1.1” kann nun ein Tweet erstellt werden der Likes und Dislikes beinhaltet. Über den Postman-Request “like a tweet” kann der im Schema enthaltene Tweet geliked werden. Gibt man sich diesen nun mit dem Postman-Request “get single tweet” aus, so sieht man, dass sich die Likes um eines erhöht haben. Das gleiche ist auch mit Dislikes möglich. Hierzu muss der Postman-Request “dislike a tweet” aufgerufen werden.

Andere Social-Media Plattformen beginnen Bilder mit in ihre Tweets aufzunehmen. Auch diese Funktion soll auf unserer Plattform mit der Version 1.2 implementiert werden. Der Benutzer kann nun ein Bild zu seinem Tweet hinzufügen und dabei einen Alternativtext zum Bild mit angeben. In der Postman-Collection im Ordner “tweets v1.2” kann nun ein Tweet erstellt werden, der ein Bild beinhaltet. Mit dem Postman-Request “update tweet with image” wird der erste Tweet mit einem Bild versehen.