# Recherchen zu Sharding

Bearbeitet: **Jerome Pracht**

Projekt: **Chatsystem für Large Scale It & Clourd Computing**

Inhalt

[Recherchen zu Sharding 1](#_Toc110203361)

[Was ist Sharding? 1](#_Toc110203362)

[Definition 1](#_Toc110203363)

[Techniken 2](#_Toc110203364)

[Docker 2](#_Toc110203365)

[Citus 2](#_Toc110203366)

[Fazit 3](#_Toc110203367)

# Was ist Sharding?

Zunächst habe ich mich gefragt, was Sharding ist.

## Definition

Die genaue Definition von Sharding ist auch unter dem deutschen Begriff [Denormalisierung](https://de.wikipedia.org/wiki/Denormalisierung#Fragmentierung) zu finden.

In eigenen Worten kann ich es so beschreiben, dass die Datenbank Tabellen in kleinere Tabellen mit dem gleichen Aufbau aufgespalten werden, sodass bei CRUD Operationen die Geschwindigkeit stark beschleunigt werden kann, da nur auf Teiltabellen gearbeitet wird und die anderen Teiltabellen unabhängig davon genutzt werden können. Hierbei werden die Normalformen, welche sonst von Datenbanken gefordert werden, zurückgenommen. Hierbei gibt es horizontales und vertikales Sharding. Beim horizontalen Sharding werden die Attribute von Einträgen mit gleicher ID, auf viele kleinere Tabellen aufgeteilt. Beim vertikalen Sharding wird das Schema der Tabelle beibehalten, jedoch sind über alle Teiltabellen, auf welche die Tabelle aufgeteilt wird, nur einzigartige ID´s erlaubt.

# Techniken

In diesem Abschnitt möchte ich auf die Techniken eingehen, mit welchen man das Sharding umsetzen kann. Da das Projekt auf Docker aufsetzt und im Projekt eine PostgreSQL Datenbank verwendet wird, habe ich meine Recherchen auf Techniken beschränkt, welche sich für unser Projekt eignen.

## Docker

Für die Umsetzung in Docker gibt es einige Möglichkeiten. Unter anderem die Option das Sharding in einem Cluster umzusetzen. Dieses kann mittels z. B. mittels Kubernetes geschaffen werden. Hierzu muss man unter den Einstellungen unter dem Punkt ***Kubernetes***Kubernetes aktivieren. Docker erstellt dann die nötigen Container.

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

### Citus

[Citus](https://www.citusdata.com/blog/2015/09/17/docker-run-pgshard-cluster/) ist eine Softwareerweiterung, mit derer man Sharding betreiben kann. Diese kann ganz alleine, in verbindung mit Docker oder Azure genutzt werden.  
  
Ich habe mich an der Umsetzung mit Docker versucht. Hierzu habe ich die das citus-docker Repo aus GitHub geklont und dieses Ausgeführt. Leider habe ich trotz vieler Recherchen und Nachfragen im Team meine Fehler beim Ausführen dieser Dateien nicht finden können. Ich habe die Erweiterung mit und ohne Kubernetes Aktivierung in Docker getestet. Immer bekam ich einen Fehler. Infolgedessen konnte ich leider auch kein Sharding für unser Projekt bereitstellen.

# Fazit

Das Projekt hat zu Anfang sehr viel Spaß gemacht. Sich zu überlegen, wie die Komponenten miteinander interagieren können, um das gewünschte Chatsystem zur Verfügung zu stellen, welches skalierbar und cloudfähig sein sollte. Aufgrund der späten und eher schwachen Erklärungen, wie wir die Anwendungen direkt in der Cloud ausführen können, haben wir uns gegen eine Cloud Umsetzung entschieden.

Back to topic: Sharding hat auch Nachteile. Meist wird (wesentlich) mehr Speicher benötigt und ein gutes Controlling ist ein Muss, damit die schnellen Zugriffe auch fehlerfrei zustande kommen können. Daher muss man immer abwägen, ob sich Sharding wirklich lohnt einzusetzen. In unserem Projekt hätte sich vertikales Sharding ziemlich gut anwenden lassen. Durch die Möglichkeit die Chats in den Tabellen mit einzigartigen ID´s zu versehen, dann aber auf vielen Untertabellen zu verteilen, hätte eine schnelle Chatverbindung und auch einen schnellen Nachrichten Austausch gewährleistet.

Die Skalierbarkeit wäre an dieser Stelle auch sehr entscheidend gewesen und das ist mit Sharding (und ausreichend freiem Speicher) gut umsetzbar. Sobald eine Tabelle zu groß wird, kann diese aufgeteilt werden und die Geschwindigkeit bleibt bestehen. Während der Recherchen habe ich des Öfteren gelesen, dass dies auch mittels einer festzulegenden maximal Belegung der Tabellen automatisiert erfolgen kann.