

## 1.1 Apache Couch Datenbank

Datenbanken (DB) sollen Flexibilität und einen sicheren Gebrauch über einen großen Zeitraum gewährleisten. Sie dienen dazu Unmengen von Datenansammlungen strukturiert abzuspeichern und verwalten diese zeitgleich. Diese Datenorganisation ist in zwei Bereiche gegliedert, zum einen in die Datenbasis (die aus den zu verwaltenden Daten besteht) zum andern aus einem Datenbankmanagementsystem (DBMS), welches die Daten verwaltet, die Kontrolle insofern darüber hat und dem Benutzer diese schlussendlich zur Verfügung stellt. Das DBMS besteht wiederum erstens aus einem Programmsystem das mithilfe von Werkzeugen und Methoden die Datenbank einrichtet wie auch diese dementsprechend pflegt. Zweitens aus einer Anwendungssoftware welche Zugriffe (Lesen, Ändern, Hinzufügen, Löschen von Daten) auf der Datenbank tätigt und so die Datensicherheit als auch den Datenschutz garantiert. [23, 24] Das Datenbanksystem (DBS) setzte sich letztendlich aus dem DBMS und der Datenbank (Datenbestand) zusammen. An dieser Stelle sein noch einmal ein paar Terminologien hervorgehoben:

- Redundanz: Überfluss, identische Daten sind mehrfach (in unterschiedlichen Tabellen) gespeichert
- Konsistenz: keine widersprüchlichen Daten (z.B. Adresse ist zweimal, unterschiedlich gespeichert)
- Integrität (Integritätsregel): Korrektheit der Daten (physikalisch, logisch und semantisch) (z.B. referentielle Integrität: Es können nur Verweise auf Datensätze eingetragen werden, die in der DB tatsächlich existieren, DBMS prüft)
- Transaktion: Mehrere, zusammengehörige Anweisungen (z.B. Geld überweisen) werden entweder alle ausgeführt oder wieder rückgängig gemacht.

Die letzten drei Punkte bilden zusammen mit der Dauerhaftigkeit das ACID<sup>1</sup> – Prinzip in der Informatik.

Ein paar Vorteile einer Datenbank sind schon zu Beginn in der **Fehler! Verweisquelle konnte nicht gefunden werden.** genannt worden. Neben der hohen Wiederverwendbarkeit oder einer guten Suchvariabilität um Korrelationen zu finden zum Beispiel. Ansonsten ist eine leistungsfähige DB fortdauernd Aktuell zudem die räumliche Unabhängigkeit von entscheidenden nutzen was eine Mehrbenutzerfähigkeit zulässt. Nachteile wie die Abhängigkeit von der Technik, das Wissen eine Datenbank fachgerecht anzuwenden oder auch die immer ansteigenden nicht enden Datenmengen (zum Teil wegen redundanten Daten) welche für einen erhöhten Speicherbedarf sorgen, sind dabei nicht vernachlässigbar.

---

<sup>1</sup> atomicity, consistency, isolation und durability

Datenmodelle sind dabei maßgebend wie Daten prinzipiell gespeichert und später manipuliert werden. Es besteht aus drei Bestandteilen:

1. Datenstruktur: Objekte und deren Beziehung
2. Operationen und den Beziehungen innerhalb Operationen
3. Integritätsbedingungen auf Objekte und Beziehungen: Hiermit sind Regeln festgelegt, die die Menge der zulässigen Zustände bzw. Zustandsübergänge definieren.

Im Allgemeinen gibt es die Unterscheidung in Folgende Datenmodelle:

- Hierarchisches Datenmodell
- Netzwerkartiges Datenmodell
- Relationale<sup>2</sup> Datenmodell
- Objektorientiertes Datenmodell
- Objektrelationales Datenmodell
- Nicht – relationale Datenbankmodelle

Im weiteren Verlauf soll das nicht – relationale Modell zur flexiblen Speicherung von hierarchischer oder unstrukturierten Daten näher erläutert werden. Da dieses die Grundlage für Not only SQL (NoSQL) – Datenbanksysteme (DBS) sind. Kennzeichen eines NoSQL – DBS sind:

- Das vorhandene Datenbankmodell ist nicht – relational
- Das System ist schemafrei bzw. nur schwache Schemarestriktionen
- Die Architektur ist von Anfang an auf eine horizontale Skalierbarkeit<sup>3</sup> dimensioniert
- Die Struktur des Konsistenzmodells weicht von dem ACID – Prinzip ab

Eines dieser DBS ist das dokumentenorientierte Datenbanksystem. Die Daten sind als Schlüsselpaare (Key – Value – Pair) bestehend aus einem eindeutigen Schlüssel (**Key**) zusammen mit einem dazugehörigen Wert (**Value**) abgespeichert. Im Value selbst werden Dokumente eines bestimmten Dateiformates abgelegt, dessen Daten strukturiert sind. Typische Datenformate hierfür: JSON, die binäre Variante BSON oder auch XML. Der Aufbau der Daten muss nicht spezifiziert werden und kann von Dokument zu Dokument voneinander abweichen. Es ist jedoch sinnvoll unterschiedlich strukturierte Dokumente separat zu speichern.

Das Datenbanksystem eignet sich besonders für die Verarbeitung umfangreicher Ansammlungen strukturierter Daten. Die derzeit (zum Zeitpunkt wo die Arbeit erstellt wurde) Systemanbieter einer dokumentenorientierten Datenbank sind unter anderem MongoDB, Couchbase und Apache CouchDB. [25] 21–22;368-369; 370 - 374

---

<sup>2</sup> In Beziehung stehen zu hier: Schlüssel (Primärschlüssel, Fremdschlüssel)

<sup>3</sup> die Verarbeitung von Daten und Anwendung ist auf viele Rechnerknoten verteilt

Apache CouchDB steht für **Cluster of unreliable commodity hardware Data Base**. Das Grundkonzept dieser Datenbank ist sehr anschaulich und für jeden leicht verständlich, weiter Besonderheiten sind:

- Ein JSON – Dokument dient als Datenspeicher
- Für Anfragen wird HTTP<sup>4</sup> verwendet
- Zuverlässigkeit
- Die Datenspeicherung ist konsistent

Die primäre Verwendung der CouchDB sah der Erfinder Damien Katz bei Entwicklungen von Webanwendungen vor. Außerdem als Konsequenz der Apache Lizenz 2.0 gruppiert sie sich in die Open Source Software ein.

Das lesen, ändern, hinzufügen, löschen von Dokumenten ist über eine HTTP API<sup>5</sup> welche das typische Datenformat JSON verwendet möglich. Die CouchDB beinhaltet verschiedene Dokumentenarten, die sich in:

- Datendokumente: Sie enthalten die Informationen, bei dem erstellen wird automatisch eine eindeutige Dokumenten – ID zugeordnet. Diese ID dient später als Schlüssel.
- Virtuelle Dokumente: Da das Verwalten von allen Informationen in einem Dokument gegen das Prinzip einer dokumentenorientierten Datenbank ist, jedoch in vereinzelter Begebenheiten die Verknüpfung mehrerer Dokumententypen erwünscht ist. Stellen die virtuellen Dokumente Views bereit, an die Anfragen gestellt werden können und die entsprechende Informationen zurückgeben.
- Design Dokument: Diese Dokumente enthalten Source Code, dessen Ausführung der CouchDB JavaScript – Query – Server übernimmt. Aus diesem Grund können individuelle Oberflächen implementiert werden, die aus folgenden Komponenten bestehen:
  - View – Funktion
  - Show – Funktion
  - List – Funktion
  - Update – Funktion
  - Validierungs – Funktionen

Abschließend nun ein Fazit. Für die spezifische Anwendungsumgebung von Webanwendungen ist CouchDB gut verwendbar. Ein weiterer Vorteil ist der zuverlässige Umgang von Eingabe wie auch Speicherung der Daten was im Allgemeinen alle dokumentenorientierte Datenbanken ausmacht. Was jedoch das Bearbeiten von Daten

---

<sup>4</sup> Hypertext Transfer Protocol

<sup>5</sup> application programming interface

angeht sind oft komplexe Anfragen notwendig, was eher für ein relationales System spricht. Davon abgesehen durch den webanwendungsnahen Aufbau ist das Prinzip sehr komfortabel für den Anwender und kann somit durchaus für ihn interessant sein. [26]

- [23] SIEPERMANN, Markus [Dr.] ; LACKES, Richard [Prof. Dr.]: *Datenbankmanagementsystem (DBMS)*. URL <http://wirtschaftslexikon.gabler.de/Archiv/74907/datenbankmanagementsystem-dbms-v10.html> – Überprüfungsdatum 2018-02-28
- [24] SIEPERMANN, Markus [Dr.] ; MECKEL, Astrid [Dr.] ; LACKES, Richard [Prof. Dr.]: *Datenbank*. URL <http://wirtschaftslexikon.gabler.de/Archiv/55473/datenbank-v13.html> – Überprüfungsdatum 2018-02-28
- [25] KUDRAß, Thomas: *Taschenbuch Datenbanken*. 2., neu bearbeitete Auflage. München : Carl Hanser Verlag GmbH & Co. KG, 2015
- [26] BERTELSMEIER, Birgit ; FÄSKORN, Heide: *Apache CouchDB*. URL [http://wikis.gm.fh-koeln.de/wiki\\_db/Datenbanken/CouchDB](http://wikis.gm.fh-koeln.de/wiki_db/Datenbanken/CouchDB). – Aktualisierungsdatum: 2013-07-05 – Überprüfungsdatum 2018-02-28