DHBW
Duale Hochschule
Baden-Württemberg
Karlsruhe

Fakultät Wirtschaft

Studiengang Wirtschaftsinformatik

Überwindung der Grenzen relationaler Datenbanken in Big-Data-Umgebungen

Assignment

im Modul

Advanced Database Technology

Im Rahmen der Prüfung zum Bachelor of Science (B. Sc.)

Verfasser:	Manuel Rettig
Kurs:	WWI23B3
Dozentin:	Maria Dertinger
Abgabedatum:	18. März 2025

Selbstständigkeitserklärung

Ich versichere hiermit, dass ich die vorliegende Arbeit mit dem Thema:

Überwindung der Grenzen relationaler Datenbanken in Big-Data-Umgebungen

selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Ich versichere zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

Karlsruhe, _____

Manuel Rettig

Inhaltsverzeichnis

1 Einleitung

1.1 Relationale Datenbanken

Das relationale Modell bildet die Grundlage heutiger relationaler Datenbankmanagementsysteme (RDBMS) und wurde im Jahr 1970 erstmals durch den Mathematiker Edgar F. Codd konzeptioniert und vorgeschlagen. [Cod70] Auf Basis dieser Forschungsarbeite entstanden in den folgenden Dekaden zahllose Datenbanksysteme wie beispielsweise IBM's Db2, die Oracle Database und der Microsoft SQL Server. Die über 50 Jahre alte Technologie durchlief im Laufe der Jahrzehnte mehrere Evolutionszyklen, um mit den Wandelnden Anforderungen mitzuhalten. Zusammen mit mit der Structured Query Language (kurz: **SQL**) bilden relationale Datenbanken bis heute den de facto Standard für das Speichern digitaler Daten und Informationen aller Art und Güte. In der montalich aktualisierten Rangliste der Popularität verschiedener DBMS der Webseite *db-engines.com* sind im März 2025 sieben der 10 meistgenutzten Datenbanken von primär relationaler Natur. [Red25]

Relationale Datenbanken basieren auf dem relationalen Modell welches aus dem mathematischen Fachgebiet der Mengenlehre abgeleitet wurde. Daten werden in Tabellen, den sogenannten Relationen gespeichert. Diese Tabelle sowie deren Spalten und unterstützten Datenformate folgen einem fest definiertem Schema. Bevor Datenzeilen eingefügt und gespeichert werden können muss dieses Schema erstellt werden. Für die Absprache wird die universelle Sprache SQL verwendet, welche mengenorientierte Operatoren wie Vereinigung oder das Kartesische-Produkt mit relationenorientierten Operatoren wie Selektion und Projektion verbindet. Die Grundstruktur folgt dabei dem SELECT-FROM-WHERE Ausdruck, mit welchem Daten von einer oder mehreren Tabellen unter Selektions- und Filterbedingungen (WHERE) abgefragt und tabellarisch angezeigt werden.

Des Weiteren bietet die große Auswahl an Implementierungen verschiedener Hersteller ein breites Spektrum an Auswahl und Expertenwissen. Für die Mehrheit der

1.2 Big Data - Große Daten?

Der Begriff Big Data (dt. Große Daten) beschreibt den allgegenwärtigen Trend wachsender Datenmengen. Hierbei liegt der Fokus nicht lediglich auf der Wachsenden Größe einzelner Datensätze wie eine einfache Übersetzung vermuten ließe, sondern vielmehr der horizontalen Skalierung und somit einer überwältigenden Anzahl von Datensätzen welche in Echtzeit verarbeitet und gespeichert werden müssen. Passende Beispiele hierfür sind beispielsweise der Datenfluss großer Online-Shops, Soziale-Netzwerke und digitale Streaming-Platformen. (Meier, 2018, p. 5)

2 Grenzen relationaler Datenbanken

2.1 Eingeschränkte Skalierbarkeit

Das rasante Datenwachstum der letzten zwei Dekaden bedingte die Notwendigkeit kontinuierlich expandierender digitaler Infrastruktur, wie etwa Server und Datenbanken. Die parallel zunehmende Rechenleistung und Speicherkapazitäten einzelner Computerchips und Festplatten konnten diesem Trend nicht kompensieren, sodass die vertikale Skalierung durch Erhöhung der Rechenleistung einzelner Computer an ihre technischen und wirtschaftlichen Grenzen stieß. Um das Problem zu lösen, wurde vermehrt auf die horizontale Skalierung gesetzt, bei der mehrere physikalisch getrennte Rechenknoten zusammenarbeiten. Diese Netzwerke bilden die Grundlage heutiger Rechenzentren und Supercomputer. Die genannten Limitierungen konnten durch Parallelisierung gelöst werden und ermöglichen theoretisch unbegrenzte Kapazität. Darüber hinaus führt die Verteilung zu einer höheren Verfügbarkeit und Ausfallsicherheit. Ist ein Knoten, beispielsweise auf Grund von Wartungsarbeiten nicht erreichbar, wird die verlorene Rechenleistung von anderen Knoten ausgeglichen, ohne merkliche Auswirkungen auf die Anwendung und deren Nutzer. Relationale Datenbanken sind traditionell für die vertikale Skalierung ausgelegt, und speziell optimiert für den Betrieb auf einem einzelnen Server. Die effektive und effiziente (optimale) horizontale Skalierung relationaler Datenbanken ist aufgrund der strengen ACID-Eigenschaften und der Einschränkung des CAP-Theorems nicht, oder nur mit erheblichem Kostenaufwand, möglich. Ein Ansatz war der sogenannte Memcache, wobei die häufigeren Lesezugriffe auf mehrere Replikationsserver verteilt wurden. Beim Sharding werden große Tabellen über mehrere Datenbankserver partitioniert. Es stellte sich jedoch heraus, dass diese komplizierten Ansätze nicht nur viele Nachteile und hohe Kosten verursachen, sondern die notwendige Skalierbarkeit, wie etwa von Sozialen Netzwerken oder Online-Shops benötigt, nicht erreichbar ist. [Har15, S. 41-43]

2.2 Fehlende Flexibilität & Impedance Mismatch

Für die Erstellung der Tabellen einer relationalen Datenbank müssen die Spalten und deren Datentypen streng definiert werden. Jede Spalte benötigt dabei einen eindeutig identifizierbaren Primärschlüssel, um die Zeilen voneinander unterscheiden zu können. Komplexe Datenmodelle werden zunächst normalisiert, um Redundanzen zu eliminieren, dabei wird das Datenmodell auf mehrere Tabellen verteilt, und miteinander via Primär- und Fremdschlüsseln verknüpft. Die vollständige Modellierung des Datenmodells kann unterstützten, um die Abhängigkeiten und Zusammenhänge zu visualisieren. Beim Einfügen eines Datensatzes, müssen die Länge genau mit der Anzahl der Spalten und den festgelegen Datentypen übereinstimmen. Ist dies nicht der Fall, lehnt das Datenbankmanagmentsystem die Daten ab, und gibt eine entsprechende Fehlermeldung zurück.

Dieser Ansatz ermöglicht neben weiteren Vorteilen eine hohe Datenkonsistenz, jedoch auf Kosten der Schema-Flexibilität. Der Ansatz der agilen Softwareentwicklung fordert einen schmalen iterativ-dynamischen Entwicklungsprozess, sodass häufige Änderungen oder Erweiterungen des Datenmodell notwendig werden. Moderne RDBMS unterstützten zwar die Modifikation des Schemas bestehender Tabellen, diese erfordern jedoch besondere Vorsicht und erfüllen nicht die geforderte einfache Flexibilität und verlässliche Stabilität, dynamischer Datenmodelle. [Har15, S. 197]

[**<empty citation>**]

3 Lösungsansätze in Big-Data Umgebungen

Im Big-Data Umfeld wurden die Grenzen relationaler Datenbanken früh erkannt und an entsprechenden Lösungsansätzen gearbeitet. Als größtes Hindernis distanzierte man sich zunehmend von dem relationalen Datenmodell, hin zu weniger komplizierten und statischen Datenmodellen.

3.1 NoSQL-Datenbanken als Alternative

3.2 NewSQL für bessere Skalierbarkeit

3.3 Verteilte Datenverarbeitung und Cloud-Technologien

4 Zusammenfassung

4.1 Fazit

4.2 Ausblick

Quellenverzeichnis

[Cod70] E. F. Codd. „A Relational Model of Data for Large Shared Data Banks“. In: *Commun. ACM* 13.6 (1970), S. 377–387. ISSN: 0001-0782. DOI: 10.1145/362384.362685. URL: https://dl.acm.org/doi/10.1145/362384.362685 (besucht am 27. 02. 2025).

[Har15] Guy Harrison. *Next Generation Databases*. Berkeley, CA: Apress, 2015. ISBN: 978-1-4842-1330-8 978-1-4842-1329-2. DOI: 10.1007/978-1-4842-1329-2. URL: http://link.springer.com/10.1007/978-1-4842-1329-2 (besucht am 02. 03. 2025).

[Red25] Red Gate Software Ltd. *DB-Engines Ranking*. DB-Engines. 2025. URL: https://db-engines.com/de/ranking (besucht am 02. 03. 2025).