Studie: Persistenzstrategie für die Datenverarbeitung

Im Rahmen der Anforderung der Plan&Los GmbH, wurde eine Studie zur Wahl Persistenzstrategie für die Datenverarbeitung erstellt. Im folgenden Dokument werden die Technologien relationale DB, dokumentenorientierte DB und Graphen-DB anhand verschiedener Kriterien bewertet.
Als Kriterien wurden Integritätssicherung, Redundanzarmut, Datensicherheit, Datenschutz, Mehrbenutzerbetrieb, Datenunabhängigkeit, zentrale Kontrolle und Skalierbarkeit herangezogen.
Ziel der Studie ist es eine Handlungsempfehlung der Plan&Los GmbH vorzulegen.

Relationale Datenbank:

Eine Relationale Datenbank (RD) besteht aus einer Menge von Relationen in denen logisch zusammengehörige Daten gespeichert werden. Eine Relation ist eine Menge von Datensätzen (Tupel). Die RD hat die Form einer Tabelle, wobei die Spalten als Attribute und die Zeilen als Datensätze bezeichnet werden. Entitätstypen, sowie auch Beziehungen eines Relationship-Modells, werden über diese Relationen abgebildet. Eine Relation ist gekennzeichnet durch:

- einen eindeutigen Bezeichner (Name)
- ein oder mehrere Attribute (Spalten)
- beliebig viele Tupel (Zeilen)
- einen Primärschlüssel, bestehend aus einem oder mehreren Attributen

Vorteile:
In relationalen Datenbanken ist die Datenintegrität durch Schlüsselintegrität gegeben. Dies hat den Vorteil, dass jedes Attribut eindeutig durch die Schlüsselkandidaten bzw. Primärschlüssel unterscheidbar ist. Parallel wird hierdurch die Datensicherheit der Datenbank aufrechterhalten, da Zugriffe nur mit festgelegten Identifikatoren ermöglicht werden können.
Ebenfalls lassen sich Änderungen von Datensätzen nur an einer Stelle vornehmen, was den Datenschutz der Datenbank garantiert.
Da der Inhalt der Datensätze, aufgrund der Mengendefinition der Relationen, immer unterschiedlich sein muss, gibt es keine zwei Tupel die identisch sind. Folglich stellt die Redundanzarmut einer relationalen DB einen weiteren Vorteil dar.
Durch die einheitlichen Abfragemethoden einer relationalen Datenbank, wird die zentrale Kontrolle deutlich erleichtert. Hierbei kann administrativ von einem Rechner aus der Zugriff auf die gesamte Datenmenge gewährleistet werden.

Nachteile:
Aufgrund der fest definierten Schematik von relationalen Datenbanken, ist die Extraktion von Einzeldaten bzw. die Suche nach komplexeren Zusammenhängen stark leistungsmindernd und stellt einen deutlichen Nachteil dar.
Folglich ist der Mehrbenutzerbetrieb von relationalen Datenbanken durch die verminderte Performanz etwas eingeschränkt.
Des Weiteren können benutzerdefinierte Abfragemethoden nicht gewährleistet werden, da die Datenbank mit einer festgelegten Zugriffssprache agiert. Dadurch können keine personalisierten Referenzen erzeugt werden.

Verwendung:
Durch die garantierte Datenkonsistenz und der flexiblen Abfragemöglichkeit haben sich relationale Datenbanksysteme zum de-facto-Industriestandard entwickelt.

Studie: Persistenzstrategie für die Datenverarbeitung

Dokumentenorientierte Datenbank:

Die dokumentenorientierte Datenbank speichert ihre Daten beispielsweise mittels JSON- oder XML-Formaten in einzelnen Dokumenten ab. Ein Dokument ist in diesem Zusammenhang als eine strukturierte Zusammenstellung bestimmter Daten zu verstehen. Innerhalb eines Dokuments sind die Daten in Form von Key/Value-Paaren gespeichert. Im Gegensatz zu Key-Value-Stores ist jedoch die Dokumentenstruktur für die Datenbanken transparenter gestaltet, wodurch diese Datenbanken schemafrei aufgebaut sind.

Vorteile:
Als großen Vorteil durch die nichtvorhandenen Schemarestriktionen, ist die Flexibilität der dokumentenorientierten Datenbanken sehr hoch. Dadurch kann der Mehrbenutzerbetrieb problemlos durchgeführt werden, da jeder Nutzer eine andere Dokumentenstruktur verwaltet und somit problemlos auf seine Daten zugreifen kann.
Zusätzlich bietet die aggregierte Speicherform Vorteile bei der horizontalen Skalierung des Systems. Somit kann die Datenbank horizontal erweitert werden ohne jegliche Verbindungen innerhalb neuangeschlossener Speichermöglichkeiten herstellen zu müssen.
Um die Datensicherheit der dokumentenorientierten Datenbank zu gewährleisten, wird im Fall einer MongoDB o. ä., kontinuierlich ein mitlaufendes Backup erzeugt (ReplicaSet). Dies geschieht indem ein „Hauptserver“ von anderen „Mitservern“ gespiegelt wird. Ebenfalls ist die Integritätssicherung dadurch geleistet.
Der Datenschutz von dokumentenorientierten Datenbanken variiert je nach Anbieter. Dieser kann von einer HTTPS-Endpunktverschlüsselung bis hin zu einer simplen administrativen Zugangsrechtkontrolle reichen.

Nachteile:
Ein deutlicher Nachteil der nichtvorhandenen Schemarestriktionen, ist ein Mangel der Redundanzarmut. Da keine Notwendigkeit besteht Primärschlüssel zu nutzen, ist der Benutzer nicht gezwungen Relationen festzulegen. Dadurch ist eine Mehrfachspeicherung von Daten möglich.
Des Weiteren stellen dokumentenorientierte Datenbanken keine Abfragemöglichkeiten bezüglich der gespeicherten Dokumente bereit da sie relationsfrei sind. Um diese Möglichkeiten dennoch nutzen zu können, müssen eigens programmierte Befehle erzeugt werden.
Die zentrale Kontrolle von dokumentenorientierten Datenbanken ist nicht vorhanden. Aufgrund keiner festgelegten Strukturen sind alle Datenzugriffe individuell gestaltet. Dadurch kann keine administrative Verwaltung ermöglicht werden.

Verwendung:
Dokumentenorientierte Datenbanken sind, durch ihr allgemein gehaltenes Datenmodell, vielseitig einsetzbar. Aufgrund genannter Kriterien eignen sich diese besonders gut für Web-Applikationen, da dort die verbreitetsten Datenaustauschformate XML und JSON, ähnlich wie in dokumentenorientierten Datenbanken, sind.

Studie: Persistenzstrategie für die Datenverarbeitung

Graphendatenbank:

Graphendatenbanken speichern Daten in einem Graphen, in welchem die einzelnen Datenelemente durch Knoten abgebildet werden, die bestimmte Attribute besitzen. Diese Knoten sind über Beziehungen miteinander verbunden, die Beziehungen wiederum können gerichtet und benannt sein und ebenfalls Attribute besitzen. Beziehungen zwischen einzelnen Knoten existieren sowohl auf einer logischen Ebene und auch als eine direkte physische Verbindung zwischen einzelnen Knoten in der Datenbank.
Bei der Datenabfrage gibt es prinzipiell zwei Arten von Abfragen. Bei der ersten Möglichkeit kann man eine gezielte Suche nach einem Knoten oder einer Kante, wenn diese ein bestimmtes Attribut besitzen. Die zweite Möglichkeit Daten abzufragen sind Graph-Traversierungen, bei Traversierungen werden die einzelnen Elemente des Graphen schrittweise durchlaufen auf diese Weise können sowohl einfache Nachbarschaftsabfragen als auch umfangreiche Wegfindungsprobleme gelöst werden.

Vorteile:
Bei großen Datenmengen können Datenabfragen in der Regel schnell erfolgen, da Graphendatenbanken den Beziehungen bei der Abfrage zur Laufzeit folgen und nicht erst Berechnungen durchführen müssen um die gesuchten Daten bereitzustellen.
Da Graphendatenbanken keinem starren Schema unterliegen, sind sie von Natur aus erweiterungsfähig und bestens für die kontinuierliche Datenzunahme geeignet, wodurch es unproblematisch ist Aktualisierungen und Strukturänderungen vorzunehmen.
Die Modellierung der Graphendatenbanken ist einfach, der Grund dafür ist die Whiteboard-Freundlichkeit, denn man kann einen Graphen auf einem Whiteboard zeichnen und diesen dann genauso in die Datenbank übernehmen. Durch diese einfache und übersichtliche Modellierung können Redundanzen von vornherein vermieden werden.
Die meisten Graphendatenbanken stellen sicher, dass die Datensicherheit gewährleistet ist z.B., wenn Netzwerkstörungen oder Serverausfälle auftreten gehen keine Daten verloren.

Nachteile:
Es gibt bisher keine einheitliche Abfragesprache und daher gibt es viele Abfragesprachen sowie Graphenmodelle.
Graphendatenbanken erlauben einem schnelle Datenabfragen, wenn es sich diese nur auf individuelle Objekte beziehen aber, wenn man eine große Anzahl an Daten gleichzeitig analysieren möchte, gibt es erhebliche Performanzprobleme.

Verwendung:
Graphendatenbanken werden meist dort eingesetzt, wo große Mengen an Informationen hochgradig miteinander vernetzt sind, dies ist beispielsweise in den Bereichen der Logistik oder auch bei sozialen Netzwerken der Fall (Twitter, Google), denn besonders hier ist es wichtig einen effizienten Umgang mit solchen Datenstrukturen zu haben.

Studie: Persistenzstrategie für die Datenverarbeitung

Empfehlung:

Aufgrund der mangelhaften Performanz einer relationalen Datenbank bei größeren Datenmengen, raten wir zu einer **dokumentenorientierten Datenbank**.
Die nötige Struktur zur Erfassung von Daten einer typisch relationalen Datenbank kann auch über eine dokumentenorientierte Datenbank generiert werden.
Da Relationen nicht zwingend notwendig sind kann der Datenspeicher durch einfaches Hinzufügen von weitern Speicherelementen erfolgen und nicht durch teure Optimierung.
Ebenfalls sind dokumentenorientierte Datenbanksystem der zu verwaltenden Datenmengen angemessen und sind somit der Graphendatenbanken vorzuziehen.

Quellen:

- https://entwickler.de/online/datenbanken/graphendatenbanken-fuenf-gute-gruende-fuer-den-umstieg-115004.html
- https://dbs.uni-leipzig.de/file/seminar_1112_stuber_ausarbeitung.pdf
- https://dbs.uni-leipzig.de/file/seminar_1112_tran_ausarbeitung.pdf
- https://tdwi.org/articles/2017/03/14/good-bad-and-hype-about-graph-databases-for-mdm.aspx
- https://www.searchenterprisesoftware.de/meinung/Pro-und-Contra-Native-versus-nicht-native-Graphdatenbanken
- http://wi-wiki.de/doku.php?id=bigdata:dokumentdb
- https://www.gesis.org/fileadmin/upload/forschung/publikationen/gesis_reihen/iz_arbeitsberichte/ab5.pdf
- http://gisbsc.gis-ma.org/GISBScL4/de/html/GISBSc_VL4_V_lo6.html