Was ist der Unterschied zwischen relationalen und nicht-relationalen Datenbanken?

AWS-Konto erstellen

Kostenlose Datenbank-Angebote entdecken
Kostenlose Angebote für Datenbank-Services in der Cloud anzeigen

Datenbank-Services überprüfen
Mit einem umfassenden Angebot an Datenbank-Services schneller innovieren

Datenbank-Schulungen durchsuchen
Beginnen Sie mit der Schulung für Datenbanken mit Inhalten, die von AWS-Experten erstellt wurden

Über die neuesten AWS-Datenbank-Produktneuheiten und Best Practices lesen

Was ist der Unterschied zwischen relationalen und nicht-relationalen Datenbanken?

Relationale und nicht-relationale Datenbanken sind zwei Methoden der Datenspeicherung für Anwendungen. Eine relationale Datenbank (oder SQL-Datenbank) speichert Daten im tabellarischen Format mit Zeilen und Spalten. Die Spalten enthalten Datenattribute und die Zeilen haben Datenwerte. Sie können die Tabellen in einer relationalen Datenbank verknüpfen, um tiefere Einblicke in die Zusammenhänge zwischen verschiedenen Datenpunkten zu erhalten. Andererseits verwenden nicht-relationale Datenbanken (oder NoSQL-Datenbanken) eine Vielzahl von Datenmodellen für den Zugriff auf und die Verwaltung von Daten. Sie sind speziell für Anwendungen optimiert, die große Datenmengen, geringe Latenz und flexible Datenmodelle erfordern, was durch die Lockerung einiger Datenkonsistenzbeschränkungen anderer Datenbanken erreicht wird.

Lesen Sie über relationale Datenbanken »

Lesen Sie mehr über NoSQL-Datenbanken »

Wie speichern relationale Datenbanken Daten?

Relationale Datenbanken speichern Daten in Tabellen mit Spalten und Zeilen. Jede Spalte steht für ein bestimmtes Datenattribut und jede Zeile steht für eine Instance dieser Daten.

Sie geben jeder Tabelle einen Primärschlüssel — eine Kennungsspalte, die die Tabelle eindeutig identifiziert. Sie verwenden den Primärschlüssel, um Beziehungen zwischen Tabellen herzustellen. Sie verwenden es, um Zeilen zwischen Tabellen miteinander zu verknüpfen, als Fremdschlüssel in einer anderen Tabelle.

Sobald zwei Tabellen miteinander verbunden sind, erhalten Sie mit einer einzigen Abfrage Daten von beiden. Sie schreiben SQL-Abfragen, um mit der relationalen Datenbank zu interagieren.

Beispiel für gespeicherte Daten

Die folgenden Tabellen veranschaulichen diesen Ansatz:

Produkt-ID (Primärschlüssel)	Produktname	Produktkosten
P1	Produkt_A	100 USD
P2	Produkt_B	50 USD
P3	Produkt_C	80 USD

Kunden_ID	Kundenname	Artikel_gekauft (Fremdschlüssel)
C1	Kunde_A	P2
C2	Kunde_B	P1
C3	Kunde_C	P3

Wie speichern nicht-relationale Datenbanken Daten?

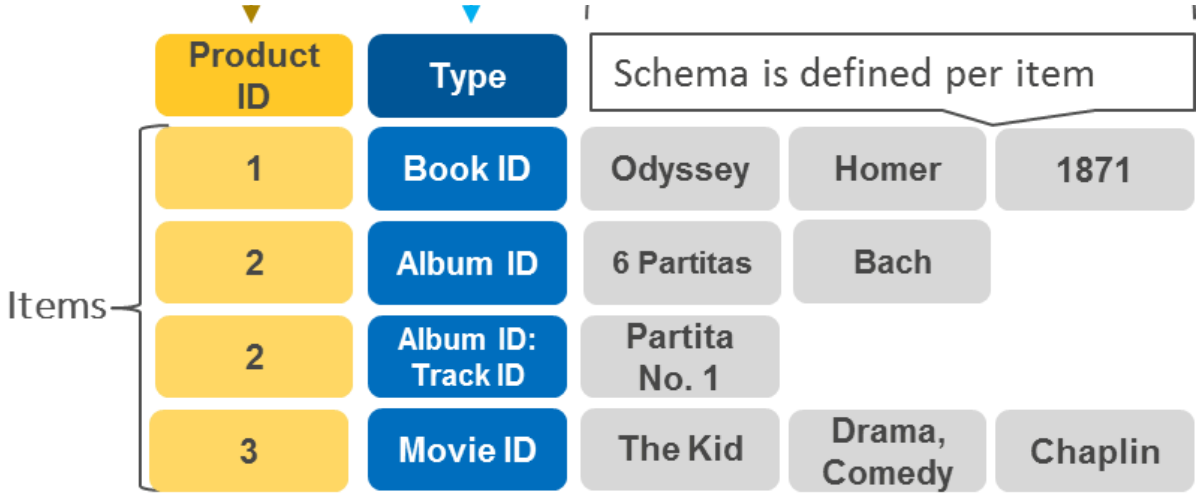
Aufgrund der unterschiedlichen Art und Weise, wie sie Daten ohne Schema verwalten und speichern, gibt es mehrere verschiedene nicht-relationale Datenbanksysteme. Daten ohne Schema sind Daten, die ohne die Einschränkungen gespeichert werden, die relationale Datenbanken erfordern.

Als Nächstes erläutern wir einige der gängigen Typen von nicht-relationalen Datenbanken.

Schlüssel-Werte-Datenbanken

Eine Schlüssel-Werte-Datenbank speichert Daten als Sammlung von Schlüssel-Werte-Paaren. In einem Paar dient der Schlüssel als eindeutige Kennung. Sowohl Schlüssel als auch Werte können alles sein, von einfachen Objekten bis hin zu komplexen zusammengesetzten Objekten.

Lesen Sie mehr über Schlüssel-Werte-Datenbanken »



Dokumentdatenbanken

Dokumentdatenbanken haben dasselbe Dokumentmodellformat, das Entwickler in ihrem Anwendungscode verwenden. Sie speichern Daten als JSON-Objekte, die flexibler, halbstrukturierter und hierarchischer Natur sind.

Das folgende Beispiel zeigt, wie gespeicherte Daten in einer Dokumentdatenbank aussehen können.

{

Name des Unternehmens: „Beliebiges Unternehmen“,

Adresse: {Straße: „1212 Main Street“, Ort: „Anytown"},

Telefonnummer: „1-800-555-0101",

Branche: [„Lebensmittelverarbeitung“, „Geräte"]

Typ: „privat“,

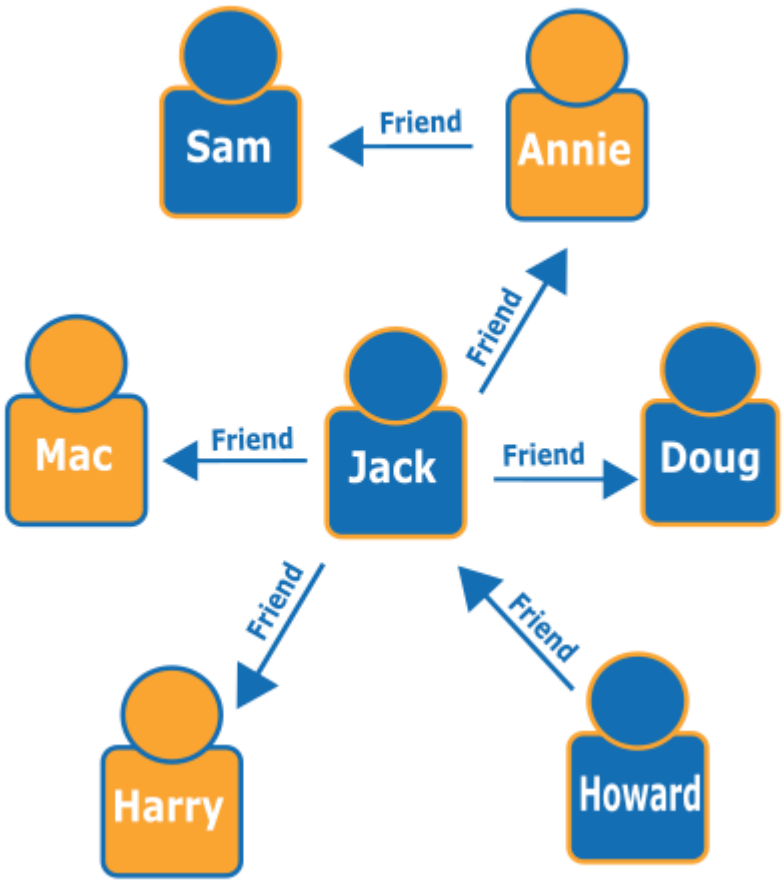
Anzahl der Mitarbeiter: 987

}

Lesen Sie mehr über Dokumentendatenbanken »

Ein Edge hat immer einen Startknoten, einen Endknoten, einen Typ und eine Richtung. Er kann beispielsweise über- und untergeordnete Beziehungen, Aktionen und Besitzverhältnisse beschreiben.

Lesen Sie mehr über Graphdatenbanken »



Hauptunterschiede: relationale und nicht-relationale Datenbanken

Relationale und nicht-relationale Datenbanken speichern und verwalten Daten sehr unterschiedlich. In den folgenden Abschnitten werden spezifische Unterschiede erörtert.

Aufbau

Nicht-relationale Datenbanken sind flexibler und nützlicher für Daten mit wechselnden Anforderungen. Sie können sie verwenden, um Bilder, Videos, Dokumente und andere halbstrukturierte und unstrukturierte Inhalte zu speichern.

Datenintegritätsmechanismus

Atomizität, Konsistenz, Isolation und Haltbarkeit (ACID) beziehen sich auf die Fähigkeit der Datenbank, die Datenintegrität trotz Fehlern oder Unterbrechungen bei der Datenverarbeitung aufrechtzuerhalten.

Ein relationales Datenbankmodell folgt strengen ACID-Eigenschaften. Dies bedeutet, dass eine Reihe aufeinanderfolgender Vorgänge immer zusammen abgeschlossen werden. Wenn ein einziger Vorgang fehlschlägt, schlägt der gesamte Satz von Vorgängen fehl. Dies garantiert die Datengenauigkeit zu jeder Zeit.

Im Gegensatz dazu bieten nicht-relationale Datenbanken ein flexibleres Modell, da sie grundsätzlich verfügbar, weich und schließlich konsistent sind (BASE).

Nicht-relationale Datenbanken garantieren Verfügbarkeit, aber keine sofortige Konsistenz. Der Datenbankstatus kann sich im Laufe der Zeit ändern und wird schließlich konsistent. Einige nicht-relationale Datenbanken bieten möglicherweise ACID-Konformität mit der Leistung oder andere Kompromisse.

Leistung

Die Leistung relationaler Datenbanken hängt von ihrem Festplattensubsystem ab. Um die Datenbankleistung zu verbessern, können Sie SSDs verwenden und die Festplatte optimieren, indem Sie sie mit einem redundanten Array unabhängiger Festplatten (RAID) konfigurieren. Um eine Spitzenleistung zu erzielen, müssen Sie auch Indizes, Tabellenstrukturen und Abfragen optimieren.

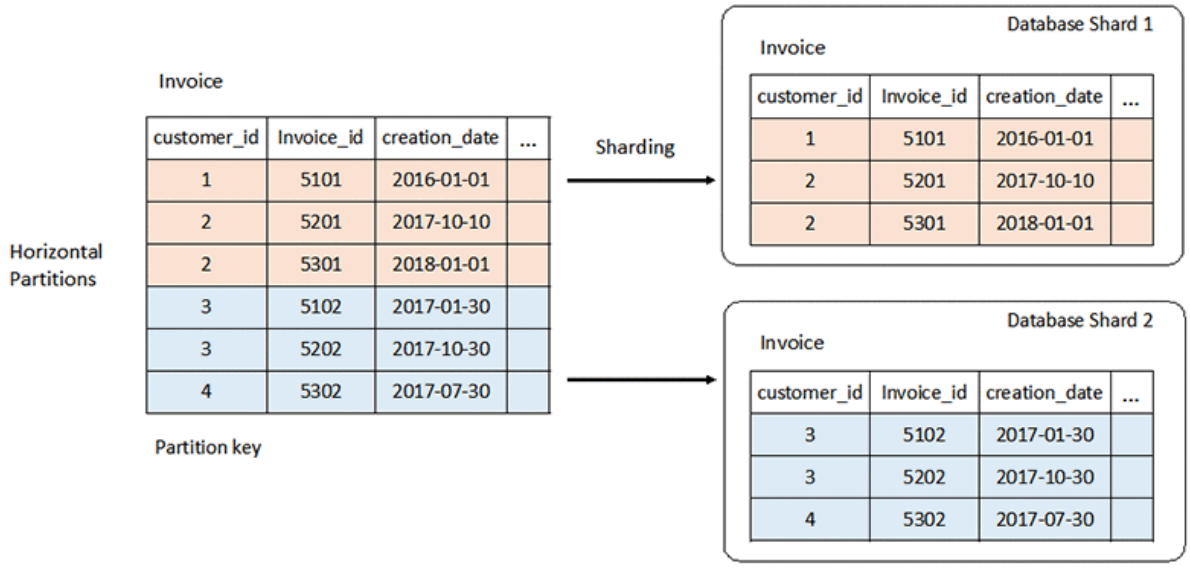
Im Gegensatz dazu hängt die Leistung von NoSQL-Datenbanken von der Netzwerklatenz, der Größe des Hardware-Clusters und der aufrufenden Anwendung ab. Es gibt verschiedene Möglichkeiten, die Leistung einer nicht-relationalen Datenbank zu verbessern:

- Erhöhen Sie die Clustergröße
- Minimieren Sie die Netzwerklatenz
- Index und Cache

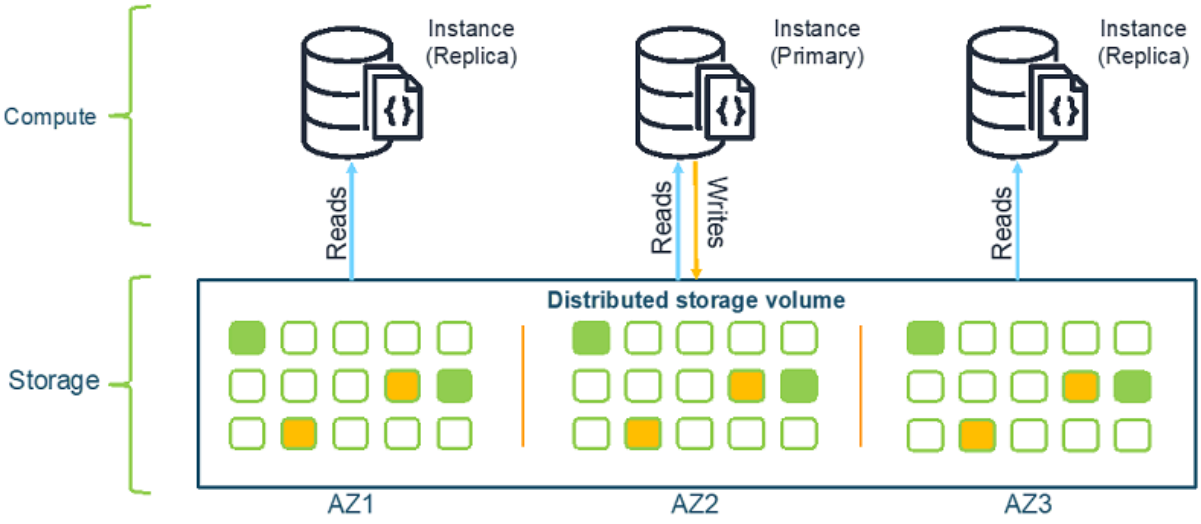
NoSQL-Datenbanken bieten im Vergleich zu relationalen Datenbanken eine höhere Leistung und Skalierbarkeit für bestimmte Anwendungsfälle.

schreibgeschützte Workloads serverübergreifend duplizieren. Die horizontale Skalierung für Lese-Schreib-Workloads erfordert jedoch spezielle Strategien wie Partitionierung und Sharding.

Lesen Sie mehr über Datenbank-Sharding »



Im Gegensatz dazu sind NoSQL-Datenbanken hochgradig skalierbar. Sie können ihre Workload einfacher auf viele Knoten verteilen. Diese Datenbanken können große Datenmengen verarbeiten, indem sie in kleinere Gruppen partitionieren und die Datensätze auf mehrere Knoten verteilen.



und Zugriffshäufigkeit vorhersehbar sind. Möglicherweise bevorzugen Sie auch ein relationales Datenbankverwaltungssystem, wenn Beziehungen zwischen Entitäten wichtig sind. Wenn Sie beispielsweise einen großen Datensatz mit einer komplexen Struktur und komplexen Beziehungen haben, möchten Sie, dass sich die Beziehungen durch Analytik und Benutzerfreundlichkeit auszeichnen.

Im Gegensatz dazu eignet sich ein nicht-relationales Modell besser für die Speicherung von Daten, die in Form oder Größe flexibel sind oder sich in Zukunft ändern können.

In einigen Fällen passen Datenbeziehungen auch einfach nicht gut in das tabellarische Primär- und Fremdschlüsselformat. Um beispielsweise die Freunde und Beziehungen in einem sozialen Netzwerk zu modellieren, benötigen Sie eine Tabelle mit Hunderten von Zeilen in einer relationalen Datenbank.

Im Gegensatz dazu kann dies in einer nicht-relationalen Datenbank in einer einzigen Zeile dargestellt werden. Das folgende Beispiel zeigt Dateneinträge für ein Mitglied mit vier Freunden in einer nicht-relationalen Datenbank.

Mitglieds_ID

Freundes_ID

M1

M2

M1 {Name des Mitglieds: „Mitglied 1“

M3 Freunde des Mitglieds: „Mitglied 2, Mitglied 3, Mitglied 4, Mitglied 5“}

M1

M4

M1

M5

Kategorie	Relationale Datenbank	Nicht-relationale Datenbank
Datenmodell	Tabellarisch.	Schlüsselwert, Dokument oder Diagramm.
Datentyp	Strukturiert.	Strukturierte, halbstrukturierte und unstrukturierte Daten
Datenintegrität	Hoch bei voller ACID-Konformität.	Eventuelles Konsistenzmodell.
Performance	Verbessert, indem dem Server mehr Ressourcen hinzugefügt wurden.	Verbessert durch das Hinzufügen weiterer Serverknoten.
Skalierung	Horizontale Skalierung erfordert zusätzliche Datenverwaltungsstrategien.	Die horizontale Skalierung ist einfach.

Wie kann AWS Ihre relationalen und nicht-relationalen Datenbankanforderungen unterstützen?

Amazon Web Services (AWS) bietet viele Services für relationale und nicht-relationale Datenbankanforderungen.

AWS-Services für relationale Datenbanken

Amazon Relational Database Service (Amazon RDS) umfasst eine Reihe verwalteter Services, die das Einrichten, Betreiben und Skalieren von relationalen Datenbanken in der Cloud vereinfachen. Cloud-Datenbanken bieten viele Vorteile wie Leistung, Skalierbarkeit und Kosteneffizienz. Sie können relationale Datenbank-Engines wie diese verwenden:

- Amazon RDS für SQL Server zur Bereitstellung mehrerer Editionen von SQL Server (2014, 2016, 2017 und 2019)
- Amazon RDS für MySQL zur Unterstützung der MySQL-Community-Edition-Versionen 5.7 und 8.0

AWS-Services für nicht-relationale Datenbanken

AWS bietet auch mehrere NoSQL-Datenbankservices, um all Ihre NoSQL-Anforderungen zu erfüllen. Hier sind einige Beispiele:

- Amazon DynamoDB ist ein Schlüssel-Werte-Datenbankservice, der für Workloads jeder Größenordnung eine konsistente Latenz im einstelligen Millisekundenbereich bietet.
- Amazon DocumentDB (mit MondoDB-Kompatibilität) ist eine beliebte dokumentenorientierte Datenbank mit leistungsstarken und intuitiven APIs für eine flexible und iterative Entwicklung.
- Amazon MemoryDB für Redis ist ein langlebiger In-Memory-Datenbankservice. Er bietet eine Lese- und Schreiblatenz von Mikrosekunden für eine ultraschnelle Leistung.
- Amazon Neptune ist ein vollständig verwalteter Graphdatenbankservice zum Erstellen und Ausführen von leistungsstarken Graphanwendungen.
- Amazon OpenSearch Service wurde speziell für die Bereitstellung von Visualisierungen und Analysen maschinengenerierter Daten nahezu in Echtzeit entwickelt.

Beginnen Sie mit relationalen und nicht-relationalen Datenbanken auf AWS, indem Sie noch heute ein Konto erstellen.

Nächste Schritte mit AWS

Erfahren Sie, wie Sie mit relationalen Datenbanken auf AWS beginnen können

Mehr über AWS erfahren

- Was ist AWS?
- Was ist Cloud Computing?
- Barrierefreiheit bei AWS
- Was ist DevOps?
- Was ist ein Container?
- Was ist ein Data Lake?
- Was ist künstliche Intelligenz (KI)?
- Was ist generative KI?
- Was ist Machine Learning (ML)?
- AWS-Cloud-Sicherheit
- Neuerungen
- Blogs
- Pressemitteilungen
- Erste Schritte
- Schulung und Zertifizierung
- AWS-Lösungsbibliothek
- Architekturzentrum
- Häufig gestellte Fragen zu Produkt und Technik
- Berichte von Analysten
- AWS-Partner

Entwickler in AWS

- Entwicklerzentrum
- SDKs und Tools
- .NET auf AWS
- Python in AWS
- Java in AWS
- PHP in AWS
- JavaScript in AWS

Hilfe

- Kontakt
- Erhalten Sie Hilfe von Experten
- Support-Ticket aufgeben
- AWS re:Post
- Wissenscenter
- AWS Support – Überblick
- Rechtliche Dokumente
- Stellenangebote bei AWS