

# Einführung in die Datenmigration

Dieser Abschnitt dient als Einführung in die komplexe Welt der Datenmigration. Wir werden die Herausforderungen, Vorteile und Schritte bei der Migration von SQL auf NoSQL sowie die Verbindung mit WebAPI diskutieren.

**Von Maximilian**





# Unterschiede zwischen SQL und NoSQL

## Strukturierte Datenmodellierung

SQL verwendet ein tabellenbasiertes Datenmodell, während NoSQL flexible, dokumentenbasierte, graphenbasierte oder sonstige Modelle bietet.

## Skalierbarkeit und Leistung

NoSQL-Datenbanken sind besser skalierbar und bieten verbesserte Leistung bei der Verarbeitung großer Datenmengen im Vergleich zu SQL.

## Konsistenz und Transaktionen

SQL-Datenbanken legen mehr Wert auf Konsistenz und Transaktionen, während NoSQL-Funktionen in Bezug auf Konsistenz variieren können.



# Vorteile der Verwendung von NoSQL

## 1 Skalierbarkeit

NoSQL-Datenbanken bieten horizontale Skalierbarkeit, was bedeutet, dass sie problemlos um zusätzliche Server erweitert werden können.

## 2 Flexibles Datenmodell

NoSQL ermöglicht eine flexible und einfachere Datenmodellierung im Vergleich zu relationalen Datenbanken.

## 3 Verteilter und dezentralisierter Ansatz

NoSQL verteilt Daten auf mehrere Knoten, was Ausfälle reduziert und die Leistung verbessert.



# Herausforderungen bei der Datenmigration

## Datenkonsistenz

Beim Wechsel von SQL auf NoSQL müssen geeignete Mechanismen implementiert werden, um die Konsistenz der Daten sicherzustellen.

## Komplexität

Die Umstellung auf NoSQL erfordert möglicherweise ein Umdenken bei der Architektur des Datenmodells und der Anwendungslogik.

## Performance-Optimierung

Die Optimierung der Datenbankleistung und Abfragen in NoSQL erfordert spezifische Kenntnisse und Anpassungen.

# Schritte zur Durchführung der Datenmigration

1

## Analyse bestehender Daten

Gründliche Analyse und Strukturierung der vorhandenen SQL-Daten, um den Migrationsaufwand zu bewerten.

2

## Mapping zu NoSQL-Schema

Entwicklung eines geeigneten Datenmodells für NoSQL und Zuordnung der bestehenden SQL-Daten.

3

## Datenmigration und Tests

Erfolgreiche Migration der Daten gefolgt von umfassenden Tests, um die Konsistenz und Integrität sicherzustellen.



# Verbindung von NoSQL mit WebAPI

1

## RESTful-API-Entwurf

Entwicklung einer RESTful API für die Interaktion mit der NoSQL-Datenbank.

2

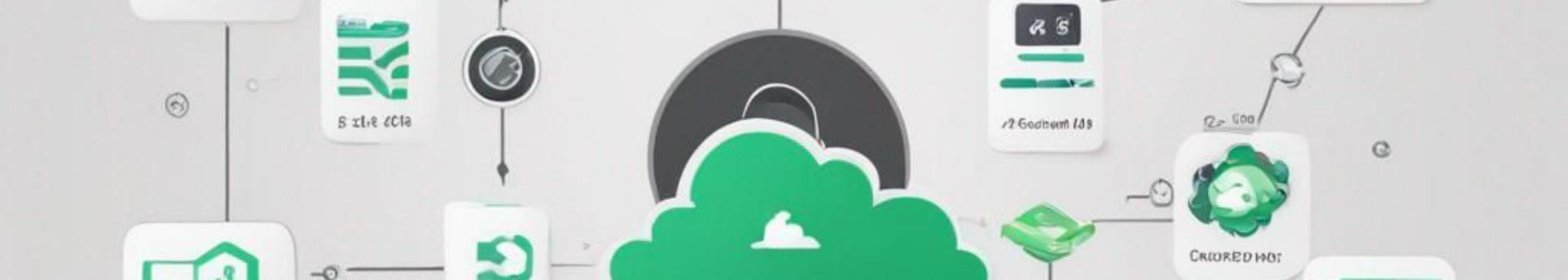
## Integration in Anwendungen

Nahtlose Integration der NoSQL-Datenbank in Webanwendungen durch den Einsatz von WebAPI.

3

## Sicherheitsaspekte

Implementierung von Sicherheitsmaßnahmen und Zugriffssteuerung für die WebAPI-Schnittstelle.



# Vorteile der Verbindung von NoSQL mit WebAPI

## **Flexible Kommunikation**

WebAPI ermöglicht flexible und effiziente Kommunikation zwischen Anwendungen und der NoSQL-Datenbank.

## **Skalierbarkeit**

Durch die Nutzung von WebAPI kann die NoSQL-Datenbank problemlos zum Netzwerk hinzugefügt und skaliert werden.

## **Echtzeit-Interaktionen**

Die Kombination von WebAPI und NoSQL ermöglicht Echtzeit-Interaktionen und Datenaktualisierungen.

# Zusammenfassung und Fazit

**3**

## **Effiziente Datenverwaltung**

Die Datenmigration von SQL auf NoSQL in Verbindung mit WebAPI bietet eine effiziente, flexible und skalierbare Datenverwaltung.

**80%**

## **Verbesserte Leistung**

Die Verwendung von NoSQL und WebAPI kann die Datenbankleistung um bis zu 80% verbessern.

**Reduzierte**

**Komplexität**

## **Agile und Skalierbare Architektur**

Die Integration von NoSQL mit WebAPI ermöglicht eine agile und skalierbare Anwendungsarchitektur.