

Datenbanken mit SQL

Arbeitsweise von und Grundlagen des Umgangs
mit Relationalen Datenbanken

Was ist ein RDBMS?

- Ein Serverdienst, der das Speichern von Daten in Tabellen anbietet.
- Zwischen den Datensätzen können Verbindungen (Relationen) implementiert werden.
- Zugriffe können über eine eigene Nutzerverwaltung gesteuert werden.
- Zugriffe können über per Netzwerk verbundene Anwendungen oder eine Shell erfolgen.

Relationales
DatenBank
Management
System

Besonderheiten

- Aus Nutzersicht erfolgt kein Zugriff auf Dateien.
- Für die einzelnen Tabellenspalten können Datentypen festgelegt werden.
- Eingabe und Ausgabe erfolgen nach den Industrie Standards für SQL, unabhängig von Betriebssystem oder Implementierung

Was ist Normalisierung?

- In Ausgaben gewünschte, wiederkehrende gleiche Daten sollen nicht mehrmals gespeichert werden.
- Ein solcher wiederholt gebrauchter Datensatz wird nur ein mal gespeichert.
- Wo er benötigt wird, wird statt der Daten eine Verbindung(Relation) zu diesem Datensatz gespeichert.

Beispiel: Autorendaten für Dokumente

- Ein Dokument hat einen Titel, Inhalt, Datum etc und einen Autor.
- Ein Autor hat einen Namen, eine Adresse, einen Geburtstag etc
- Im Datensatz des Dokuments wird statt der Daten des Autors die ID des Datensatzes des Autors eingefügt.

Technische Umsetzung von Relationen

- Relationen sind rein semantische Konstruktionen, ein RDBMS muss diese nicht eigenständig umsetzen.
- Anwendungen sind so programmiert, dass die Relationen nutzbar werden
- Relationen können in gängigen Programmiersprachen oder direkt in SQL genutzt werden.

Beispiel für die Umsetzung

- 1.) In PHP:

- `$dokument_sql="SELECT * FROM documents;";`

- `$document_data=read_sql($dokument_sql);`

- `$author_sql="SELECT * FROM authors WHERE aid=$document_data['aid'];";`

- `$author_data=read_sql($author_sql);`

- `$document_data` und `$author_data` können zusammen angezeigt werden.

Beispiel für die Umsetzung

- 2.) In SQL:

```
SELECT documents.content, authors.name  
FROM documents  
INNER JOIN authors
```

```
ON documents.aid=authors.aid;
```

- Gibt einen Datensatz aus, der zu jedem Dokument Inhalt den Namen des Autors enthält.
- Die Daten des Autors müssen bei beiden Methoden nur genau ein mal gespeichert werden, egal, wieviele Dokumente zu ihm/ihr gehören.

Vor/Nachteile

- Pro: komplex strukturierte Datensätze aus Text und Zahlen können effizient gespeichert und leicht programmiert werden. Besonders nachträgliche Änderungen der Inhalte sind einfach.
- Contra: Binäre Daten(Bilder, Musik etc) sind als einfache Dateien erheblich besser zu speichern und zu laden. Einmal festgelegte Tabellenstrukturen lassen sich nach einer produktiven Laufzeit nur noch schwer ändern.

Häufig verwendete SQL RDBMS

- Mysql / MariaDB
- Oracle
- Microsoft SQL Server
- PostgreSQL
- SQLite

Alternativen: NoSQL

- NoSQL Datenbanken werden nicht mit SQL abgefragt und haben meist keine Tabellenstruktur(Schema) im herkömmlichen Sinne.
- Ihre Struktur kann Dokumentbasiert(jeder Datensatz hat eine eigene Struktur) sein oder Listenartig(Key/Value Store, Wide Column)

Beispiele NoSQL

- MongoDB (Dokumentbasiert, für extrem große Datensätze, Abfrage via Java Script Dialekt, u.a.: LHC, New York Times etc)
- Redis (schneller, leicht programmierbarer Key/Value Store)
- Cassandra(Wide Column, für extrem große Datenbanken, u.a. Facebook)