

Cheatset Datenbankdesign

Rüdiger Voigt

01.09.2017

1 Die wichtigsten Befehle

SELECT
INSERT
UPDATE
DELETE

2 WHERE-Bedingung

- Schränkt ein welche Zeilen ausgewählt werden.
- Funktioniert mit SELECT, UPDATE, DELETE.

3 Die wichtigsten Datentypen

CHAR Folge beliebiger Zeichen mit fixer Länge.
VARCHAR Folge beliebiger Zeichen mit variabler Länge und Maximallänge
TEXT Ähnlich zu VARCHAR, aber mit deutlich höherer Maximallänge.
INT Integer / Ganzzahl
FLOAT Gleitkommazahl. ACHTUNG: Rundungsfehler bei bestimmten Zahlen. Nicht für Geldbeträge geeignet!
DECIMAL Speichert Kommazahlen korrekt.

4 Struktur

Dieses "Cheatset" ist als *grobe Orientierungshilfe* gedacht. Bei einem komplexen Projekt kämen weitere Schritte (Stakeholder-Meeting, Entwicklung einer Oberfläche, Usability-Tests mit Endanwender, ...) hinzu. Erfahrungsgemäß wird der Entwurf auch mehrfach überarbeitet oder im Projektverlauf abgeändert.

- Sie wählen die passende Storage Engine und eine geeignet Collation.
Auf einem MariaDB-System sind die Storage-Engine XtraDB (gibt sich im Frontend als InnoDB aus / beherrscht Foreign Key Constraints und Transaktionen) und die Collation `utf8mb4_unicode_general_ci` meistens eine sehr gute Wahl.
- Sie überlegen, welchen Zweck Ihre Datenbank hat. Welche Daten brauchen Sie? Welche davon können Sie mit welchem Aufwand erheben? Welche können Sie importieren?

- Welcher Datentyp liegt jeweils vor?
Das ist nicht immer einfach zu beantworten, oder eindeutig. Eine deutsche Postleitzahl lässt sich als Integer abbilden, aber im Ausland enthalten Sie oft auch Buchstaben.
- Ausgehend davon planen Sie Ihre Tabellenstruktur. Beachten Sie:
 - Die Normalformen! (Zumindest die ersten Drei.)
 - Welche Daten gehören logisch zusammen?
 - Jede Zeile braucht einen Primary Key. Das ist Unique-Identifizierer und so können Sie die Zeile im Zweifel eindeutig identifizieren / abfragen / löschen. Wenn sich keine Spalte aufdrängt, verwenden Sie eine id Spalte (Integer mit Auto Increment). Durch Definition als PK wird ein Index für diese Spalte erzeugt.
- Welche Beziehungen möchten Sie abbilden?
Sind diese einfach abzubilden (Eine Spalte verweist immer auf einen anderen Wert in einer Tabelle), oder sind es komplexere Beziehungen (zum Beispiel das Buch mit mehreren Autoren: Zuordnungstabelle)
- Legen Sie Indizes da an, wo es sinnvoll ist.
Prinzipiell alle Spalten nach welchen Sie sortieren / filtern. Ein Index zum Beispiel auf ein LONGTEXT Feld macht dafür keinen Sinn.
Sie können im nächsten Schritt Foreign Key Constraints nur definieren, wenn auf den entsprechenden Spalten der parent und der child Tabelle ein Index liegt!
- Legen Sie Foreign Key Constraints an. Ihre Ziele dabei:
 - Sie gewährleisten Konsistenz indem Sie verhindern, dass Daten bei Abhängigkeiten in unzulässiger Weise gelöscht oder geändert werden!
 - Beziehungen zwischen verschiedenen Tabellen werden deutlich.
- Füllen Sie Ihr Projekt mit Testdaten.
- Prüfen Sie die Funktionalität durch Abfragen und Ausprobieren von typischen Fällen.
Zum Beispiel gehen Mitarbeiter irgendwann in Rente und werden dann aus der Datenbank gelöscht. Geht das oder besteht hier ein Problem etwa in Form von FK Constraints?
- Speichern Sie komplexe und oder wichtige Abfragen als View ab.
- Wenn bestimmte Aktionen in bestimmter Reihenfolge oder in bestimmter Art und Weise auf eine oder mehrere Tabellen zugreifen müssen, schreiben Sie Funktionen / Stored Procedures.

Eine aktuelle Fassung der Unterlagen finden Sie unter:

<https://www.ruediger-voigt.eu/kurs-datenbanken-und-sql.html>