

Vortrag 3 Setup, Schlüssel und erste Anwendung

Datenbanken

Lukas Wais

Codersbay

Version: 19. April 2023

Inhaltsverzeichnis

Schlüssel

Setup

Installation MariaDB

Befehle

Projekt

Schlüssel in relationalen Datenbanksystemen

Sie werden dazu verwendet Tupel (=Zeilen) einer Relation (Tabelle) **eindeutig** zu identifizieren. Ein Schlüssel ist eine Spalte, oder eine Gruppe von Spalten, die so ausgewählt sind, dass jedes Tupel über die Werte dieser Spaltengruppe eine eindeutige Wertekombination hat.

Arten von Schlüsseln I

- ▶ **Superschlüssel:** Menge von Attributen in einer Relation, welche die Tupel eindeutig identifizieren.
Triviales Beispiel: Die Menge aller Attribute einer Relation gemeinsam. Die Elemente von Mengen sind eindeutig. Das bedeutet auch, dass ein Superschlüssel Spalten beinhaltet, die für die Schlüsseleigenschaft nicht notwendig sind.

- ▶ **Schlüsselkandidat:** Minimale Teilmenge der Spalten eines Superschlüssels, welche die Identifizierung eines Tupels ermöglicht. Minimal bedeutet, dass keine Spalte weggelassen werden kann (er würde sonst nicht mehr identifizieren).

Arten von Schlüsseln II

- ▶ **Primärschlüssel:** Der ausgewählte Schlüsselkandidat, der die für die Abbildung der Relation tatsächlich verwendet wird.
- ▶ **Fremdschlüssel:** Attribut, oder auch Attributkombination, einer Tabelle die auf einen Primärschlüssel verweist. Er zeigt an welche Tupel der Relation in Verbindung stehen.

Eigenschaften eines Primärschlüssels

- ▶ Möglichst klein (=möglichst wenig Attribute/möglichst simpler Datentyp)
- ▶ Während des DB-Lebenszyklus soll er sich nicht ändern; zeitlich stabil → sonst müssten sich auch Fremdschlüssel entsprechend ändern.
- ▶ Eindeutige Identifizierbarkeit der Objekte. Beispiel: Wählt man als Primärschlüssel Name und Geburtsdatum darf es keine Person mit gleichem Namen und Geburtstag geben. Ein künstlicher Schlüssel, auch genannt Surrogatschlüssel, wie etwa eine ID wäre hier besser.

Beispiel

Wir wählen folgende Tabelle:

$\{ISBN, Buchtitel, Autor\}$

Schlüsselkandidaten:

¹Durch die Wahl der Primärschlüssels, werden alle anderen Schlüsselkandidaten zu Alternativschlüsseln

Beispiel

Wir wählen folgende Tabelle:

$\{ISBN, Buchtitel, Autor\}$

Schlüsselkandidaten:

$\{ISBN\}, \{Buchtitel, Autor\}$

Primärschlüssel:

¹Durch die Wahl der Primärschlüssels, werden alle anderen Schlüsselkandidaten zu Alternativschlüsseln

Beispiel

Wir wählen folgende Tabelle:

$\{ISBN, Buchtitel, Autor\}$

Schlüsselkandidaten:

$\{ISBN\}, \{Buchtitel, Autor\}$

Primärschlüssel:

$\{ISBN\}$

Alternativschlüssel: ¹

$\{Buchtitel, Autor\}$

¹Durch die Wahl der Primärschlüssels, werden alle anderen Schlüsselkandidaten zu Alternativschlüsseln

Beispiel Fremdschlüssel

Ist Chef von

Tabelle Mitarbeiter ist wie folgt definiert:

{Personalnummer, Vorname, Nachname, ...}

Vorgesetzter	Untergebener
123	456
123	789

Primärschlüssel: Personalnummer von Vorgesetzter + Untergebener.
Fremdschlüssel sind die einzelnen Personalnummern der Mitarbeiter.

Installation DBMS

1. Installation Betriebssystem für DBMS (meist Linux)
2. Installation DBMS
3. Konfiguration DBMS; Benutzer anlegen, Backup, ...
4. Erstellung der Tabellen

Installation auf Ubuntu

Eine Installationsanleitung und weitere Informationen zu MariaDB gibt es im Ubuntu Wiki:
<https://wiki.ubuntuusers.de/MariaDB/>.

Einige Befehle zur Erstellung einer Datenbank I

- ▶ Erstellen: `CREATE DATABASE databasename;`
- ▶ Löschen: `DROP DATABASE databasename;`

Vorsicht beim Löschen, es werden die gesamten Daten in der Tabelle gelöscht und ein Rollback ist **nicht** möglich. Wiederherstellung aus einem Backup ist möglich.

- ▶ Tabelle erstellen:

```
CREATE TABLE tablename (  
    column1 datatype,  
    column2 datatype,  
    column3 datatype,  
    ....  
);
```

Einige Befehle zur Erstellung einer Datenbank II

- ▶ Inhalt, aber nicht die Tabelle löschen: **TRUNCATE TABLE** tablename;
- ▶ Änder/löschen/updaten von Spalten in existierenden Tabellen:
ALTER TABLE tablename
ADD columnname datatype;
- ▶ Erstellung einer Tabelle mit Fremd- und Primärschlüssel und Constraint:
CREATE TABLE Orders (
 OrderID **int NOT NULL**,
 OrderNumber **int NOT NULL**,
 PersonID **int**,
 PRIMARY KEY (OrderID),
 FOREIGN KEY (PersonID) **REFERENCES** Persons(PersonID)
);

Constraints

Regeln für Daten in der Tabelle. Sie können Spalten oder die ganze Tabelle betreffen.

- ▶ NOT NULL - stellt sicher, dass eine Spalte keinen NULL-Wert haben kann.
- ▶ UNIQUE - Stellt sicher, dass alle Werte in einer Spalte unterschiedlich sind.
- ▶ PRIMARY KEY - Eine Kombination aus NOT NULL und UNIQUE - identifiziert jede Zeile in einer Tabelle eindeutig.
- ▶ FOREIGN KEY - Verhindert Aktionen, die Verknüpfungen zwischen Tabellen zerstören würden.
- ▶ CHECK - Stellt sicher, dass die Werte in einer Spalte eine bestimmte Bedingung erfüllen.
- ▶ DEFAULT - Setzt einen Standardwert für eine Spalte, wenn kein Wert angegeben ist.
- ▶ CREATE INDEX - Wird verwendet, um sehr schnell Daten aus der Datenbank zu erstellen und abzurufen (Indexierung).

Erstes Datenbankprojekt Part 1

- ▶ Installiere eine MariaDB auf Ubuntu.
- ▶ Verbinde dich von deinem Hostrechner mit einem DB Client deiner Wahl.
- ▶ Setze die Zootabelle von der letzten Aufgabe um. Erstelle die entsprechende Tabelle, inklusive constraints.
- ▶ Schreibe ein Programm in Python oder JavaScript welche zumindest die CRUD Operationen erlauben. Ein Konsolendialog ist ausreichend.

Erstes Datenbankprojekt Part 2

- ▶ Der Konsolendialog soll nun in einen RESTful Webservice verwandelt werden.
- ▶ Verwende hierfür entweder [Flusk](#) oder [Express.js](#)
- ▶ Überlege dir entsprechende URLs und Teste die Funktionsweise mit [Postman](#).
- ▶ Programmiere ein Frontend mit HTML und JavaScript.

Abstrakte RESTful Architektur

