

Dokumentation der Datenmodellierung und SQL-Anweisungen

Hendrik Siemens

14. März 2024

Inhaltsverzeichnis

1	Einleitung	2
2	ER-Modell	2
2.1	Entitäten und Attribute	2
2.2	Beziehungen	2
3	SQL-Anweisungen	3
3.1	Tabellenstruktur	3
3.2	Datenimport und Verteilung	5
3.3	Datenabfrage	7
4	Schlussfolgerung	8

1 Einleitung

Die Dokumentation beschreibt die Entwicklung eines ER-Modells und die Implementierung eines Datenmodells für die Verwaltung von Schulbuchausleihen. Ziel ist es, die bestehenden manuellen Prozesse zu digitalisieren und zu automatisieren, um Effizienz und Genauigkeit zu verbessern.

2 ER-Modell

Basierend auf der vorliegenden Listenform wurde ein ER-Modell entworfen, welches die Datenstruktur für die Schulbuchverwaltung abbildet.

2.1 Entitäten und Attribute

- **Schüler:** SchülerID, Name, Vorname, Straße, PLZ, Ort
- **Klasse:** KlassenID, Klassenname, LehrerID
- **Lehrer:** LehrerID, Name
- **Buch:** BuchID, ISBN, Titel
- **Buchausleihe:** SchülerID, BuchID, Ausleihdatum, Rückgabedatum

2.2 Beziehungen

- Schüler sind Klassen zugeordnet.
- Klassen werden von Lehrern unterrichtet.
- Bücher werden von Schülern ausgeliehen.

3 SQL-Anweisungen

Nachfolgend sind die SQL-Anweisungen für die Umsetzung des ER-Modells und den Datenimport aufgeführt.

3.1 Tabellenstruktur

```
1 CREATE TABLE Schueler (  
2     SchuelerID INT AUTO_INCREMENT PRIMARY KEY,  
3     Name VARCHAR(255),  
4     Vorname VARCHAR(255),  
5     Strasse VARCHAR(255),  
6     PLZ VARCHAR(5),  
7     Ort VARCHAR(255)  
8 );  
9  
10 CREATE TABLE Klasse (  
11     KlassenID INT AUTO_INCREMENT PRIMARY KEY,  
12     Klassenname VARCHAR(255),  
13     LehrerID INT,  
14     FOREIGN KEY (LehrerID) REFERENCES Lehrer(LehrerID)  
15 );  
16  
17 CREATE TABLE Lehrer (  
18     LehrerID INT AUTO_INCREMENT PRIMARY KEY,  
19     Name VARCHAR(255)  
20 );  
21  
22 CREATE TABLE Buch (  
23     BuchID INT AUTO_INCREMENT PRIMARY KEY,  
24     ISBN VARCHAR(13),  
25     Titel VARCHAR(255)  
26 );  
27  
28 CREATE TABLE Buchausleihe (  
29     SchuelerID INT,  
30     BuchID INT,  
31     Ausleihdatum DATE,
```

```
32     FOREIGN KEY (SchuelerID) REFERENCES Schueler(SchuelerID),
33     FOREIGN KEY (BuchID) REFERENCES Buch(BuchID)
34 );
35
36 CREATE TABLE TempImport (
37     Name VARCHAR(255),
38     Vorname VARCHAR(255),
39     Strasse VARCHAR(255),
40     PLZ VARCHAR(5),
41     Ort VARCHAR(255),
42     Klassenname VARCHAR(255),
43     LehrerName VARCHAR(255),
44     ISBN VARCHAR(13),
45     Buchtitel VARCHAR(255)
46 );
```

3.2 Datenimport und Verteilung

```
1 LOAD DATA INFILE '/home/hendrik/Documents/test.csv'
2 INTO TABLE TempImport
3 FIELDS TERMINATED BY ','
4 OPTIONALLY ENCLOSED BY '"'
5 LINES TERMINATED BY '\n'
6 IGNORE 1 LINES
7 (Name, Vorname, Strasse, PLZ, Ort, Klassenname, LehrerName, ISBN,
   ↪ Buchtitel);
8
9
10 INSERT INTO Schueler (Name, Vorname, Strasse, PLZ, Ort)
11 SELECT DISTINCT Name, Vorname, Strasse, PLZ, Ort
12 FROM TempImport
13 ON DUPLICATE KEY UPDATE SchuelerID = SchuelerID;
14
15 INSERT INTO Lehrer (Name)
16 SELECT DISTINCT LehrerName
17 FROM TempImport
18 WHERE LehrerName IS NOT NULL AND LehrerName != '';
19
20 INSERT INTO Klasse (Klassenname, LehrerID)
21 SELECT DISTINCT
22     t.Klassenname,
23     l.LehrerID
24 FROM TempImport t
25 JOIN Lehrer l ON t.LehrerName = l.Name
26 WHERE t.Klassenname IS NOT NULL AND t.Klassenname != '';
27
28 INSERT INTO Buch (ISBN, Titel)
29 SELECT DISTINCT
30     ISBN,
31     Buchtitel
32 FROM TempImport
33 WHERE ISBN IS NOT NULL AND ISBN != '';
34
```

```
35 INSERT INTO Buchausleihe (SchuelerID, BuchID, Ausleihdatum,  
    ↪ Rueckgabedatum)  
36 SELECT  
37     s.SchuelerID,  
38     b.BuchID,  
39     ti.Ausleihdatum,  
40     ti.Rueckgabedatum  
41 FROM TempImport ti  
42 JOIN Schueler s ON ti.Name = s.Name AND ti.Vorname = s.Vorname  
43 JOIN Buch b ON ti.ISBN = b.ISBN  
44 WHERE ti.Ausleihdatum IS NOT NULL;  
45  
46 DROP TABLE IF EXISTS TempImport;
```

3.3 Datenabfrage

```
1 SELECT
2   s.SchuelerID,
3   s.Name AS SchuelerName,
4   s.Vorname,
5   s.Strasse,
6   s.PLZ,
7   s.Ort,
8   k.Klassenname,
9   l.Name AS LehrerName,
10  b.ISBN,
11  b.Titel AS Buchtitel
12 FROM
13   Schueler s
14   JOIN Buchausleihe ba ON s.SchuelerID = ba.SchuelerID
15   JOIN Buch b ON ba.BuchID = b.BuchID
16   JOIN Klasse k ON s.SchuelerID = k.KlassenID
17   JOIN Lehrer l ON k.LehrerID = l.LehrerID
18 ORDER BY
19   s.SchuelerID, b.BuchID;
```

4 Schlussfolgerung

Diese Dokumentation stellt einen ersten Schritt in Richtung einer effizienteren und fehlerfreien Verwaltung von Schulbüchern dar. Weitere Optimierungen und Anpassungen können implementiert werden, um die Benutzerfreundlichkeit und Funktionalität des Systems zu verbessern.