

Einleitung

Datenbanken sind die Basis zur Speicherung und Verwaltung von Daten. Sie unterstützen Programme in der Einhaltung der Datenintegrität und zur Vermeidung von Redundanzen. In der vorliegenden Arbeit ist die erstellte Datenbank die Grundlage für eine Buchtausch-App, die es Nutzern ermöglicht, Bücher lokal einzustellen und auszuleihen.

In der implementierten Datenbank werden alle Informationen gespeichert, welche für die Nutzung der App notwendig sind, von den Nutzerdaten bis zu den Tauschdaten. Die Daten werden dabei in einer geeigneten Form gespeichert, sodass eine fehler- und konfliktfreie Nutzung der App möglich ist. Basierend auf der Datenbank ist es in der App möglich, Nutzer zu registrieren, Bücher einzustellen und auszuleihen, Favoritenlisten anzulegen sowie Nutzer und Bücher nach einem Tausch zu bewerten. Die vorliegende Arbeit umfasst neben der technischen Umsetzung und Metadaten auch die technische Reflexion.

Technische Umsetzung

Das auf der Datenbank basierende ER-Modell wurde als UML Klassendiagramm erstellt. Dadurch können sowohl Konditional Beziehungen als auch die Kardinalitäten der Beziehungen lesbarer dargestellt werden. Die Entitäten des Modells können in 4 Kategorien unterteilt werden: User, Book, Borrowingperiod und Rating. Die zentralen Beziehungen der Kategorien: Userdata <-> Booksummary, Booksummary <-> Book, Booksummary <-> Borrowingperiod und Borrowingperiod <-> "Rating" spielen eine entscheidende Rolle zur Verwaltung und Nutzung der Daten. Durch diese sind alle weiteren Entitäten miteinander verbunden und es können die Hauptfunktionalitäten der App vollumfänglich genutzt werden. Die Attributnamen sowie dessen Datentypen wurden basierend auf einer eindeutigen und konsistenten Datenspeicherung gewählt. Auch eine gleichlautende Logik der Bezeichnungen wurde angestrebt.

Die Erstellung aller Tabellen wurde durch CREATE Statements durchgeführt. Anschließend wurden die Trigger bei den Entitäten Bookrating, UserratingLender und UserratingReceiver ergänzt. Die Trigger dienen dazu, dass die abgegebenen Bewertungen als Durchschnittswert berechnet und anschließend anderen Nutzern zur Verfügung gestellt wird. Zur Verbesserung der Datenintegrität wurden ebenfalls Trigger erstellt, falls Bewertungen geändert oder gelöscht werden.

In jeder Entität wurden Primärschlüssel verwendet, wobei die meisten davon einfache Primärschlüssel sind {ID}. In den Entitäten BookAuthor und Favouritelist wurde auf zusammengesetzte Primärschlüssel {PK} zurückgegriffen, da die Eindeutigkeit je Datenzeile gegeben ist und dadurch ein künstlicher Primärschlüssel nicht zwingend erforderlich ist. Ebenfalls wurden mehrere Fremdschlüssel verwendet, um Beziehungen zwischen den Entitäten herzustellen. Besonders dabei sind die zwei n:m Beziehungen bei BookAuthor (Book <-> Author) und Favouritelist (Userdata <-> Booksummary). Ebenfalls gibt es drei Dreifachbeziehungen: Bookrating, UserratingLender und UserratingReceiver. Bei allen Fremdschlüsselbeziehung wurde ein Verhalten definiert, welches die Datenintegrität erfüllt z.B. die übernehmende Änderung des referenzierten Datensatzes (ON UPDATE CASCADE) und ein Verhindern des Löschens (ON DELETE RESTRICT). Insgesamt erreicht die Datenbank die 3 Normalform, wodurch eine hohe Normalisierung erreicht wird.

Dummy Daten und Testabfragen

Die Daten für die Tabellen wurden mit Hilfe der INSERT Statements eingefügt. Für jede Tabelle wurden, sofern möglich, mindestens 10 Einträge als Dummy Daten eingetragen. Bei den Tabellen Country, Language, BookCover, Status, BookQuality, wurde auf diesen Umfang verzichtet, da weitere Daten die realitätsnahe und anwendbare Nutzung, vor allem zu Beginn, behindert hätten. Im Laufe der App Nutzung kann es sein, dass weitere Daten hinzugefügt werden, dies wird selbstverständlich unterstützt. Für jede Tabelle wurde durch ein allgemeines SELECT Statement die gespeicherten Werte ausgegeben, um die korrekte Speicherung zu überprüfen. Zusätzlich wurde ein Testfall von der Nutzer Registrierung bis zur Bewertung nach einem Tausch durchgeführt. Dabei wurden spezielle Abfragen mit JOIN Befehlen durchgeführt, um die korrekte Referenzierung zu überprüfen. Bei allen Abfragen wurden keine Fehler oder Komplikationen festgestellt.

Metadaten und Systemübersicht

In der Datenbank sind 24 Tabellen, in denen insgesamt 252 Datensätze gespeichert sind. Die genaue Aufteilung der Datensätze ist in der folgenden Tabelle aufgeschlüsselt. Da es sich hierbei noch um einen Test handelt, sind die Datensätze begrenzt. Bei der Nutzung von realen Daten ist ein Umfang von mehreren Tausen Einträgen zu erwarten.

Tabelle 1: Dateneinträge je Tabelle

Tabelle	Anzahl Datensätze
Country	1
City	10
Address	10
Logindata	10

Tabelle	Anzahl Datensätze
Weight	16
Dimensions	16
Book	16
Author	14

Userdata	10
Favouritelist	10
Genre	10
Publisher	14
Language	2
YearOfPublication	10
BookCover	2
NumberOfPages	10

BookAuthor	14
Status	5
Bookquality	5
Booksummary	16
Borrowingperiod	15
Bookrating	16
UserratingLender	10
UserratingReceiver	10

Quelle: Eigene Darstellung

Reflexion und Fazit

Die entwickelte Datenbank erfüllt vollständig alle funktionalen Anforderungen. Besonders positiv hervorzuheben ist die klare Trennung zwischen Nutzerdaten, Buchdaten, Ausleihvorgängen und Bewertungen. Durch die vorgenommene Normalisierung auf die 3. Normalform wird auch eine hohe Datenintegrität und geringe Redundanzen sichergestellt. Herausfordernd war hingegen die Konzeption der Bewertungstabellen im Hinblick auf die Verknüpfung mehrerer Tabellen. Die Modellierung komplexer Beziehungen hat sich in der Praxis jedoch als stabil erwiesen.

Im Verlauf der Projektarbeit wurde das Datenmodell mehrfach anhand des erhaltenen Feedbacks überarbeitet. So wurden z.B. einige Rollen und Aktionen näher spezifiziert, um das Verständnis zu erhöhen, die Suchfunktionen für Bücher weiter ausgebaut (z.B. BookCover, NumberOfPages) und SQL Statements detaillierter dokumentiert, um das Verständnis für andere Entwickler zu erhöhen. Außerdem wurden Entitäten neu angeordnet, neue hinzugefügt und Attributbezeichnungen und Datentypen aus Logikgründen angepasst.

Für eine Weiterentwicklung der Datenbank ist es möglich sich Gedanken zu Benachrichtigungen, einer Frontend Anbindung für eine Web- und App Nutzung sowie einer Mehrsprachigkeit zu machen. Die Weiterentwicklung der Datenbank und letztendlichem Programm sollte auch in Abstimmung mit den Wünschen der Nutzer geschehen, um eine hohe Zufriedenheit und weitreichende Nutzung zu erzielen.

Das entwickelte Datenbanksystem entspricht in vollem Umfang den vorgegebenen Zielen. Alle erforderlichen Prozesse und Eigenschaften einer Buchtauchplattform wurden modelliert und technisch umgesetzt. Ebenfalls hat die Datenbank eine ausreichende Normalform, ist stabil und wurde getestet. Somit erfüllt das System sowohl den funktionale als auch qualitative Anforderungen an ein realitätsnahes Data Mart-Projekt.