

Portfolio: Datenbankmanagementsystem Buchtausch-App

Kurs: Datenbankmanagement-Systeme

Phase: 1.3 Finalisierungsphase

Autor: D.T

Datum: 10. Februar 2026

FUNKTIONALITÄT UND PROJEKTÜBERSICHT

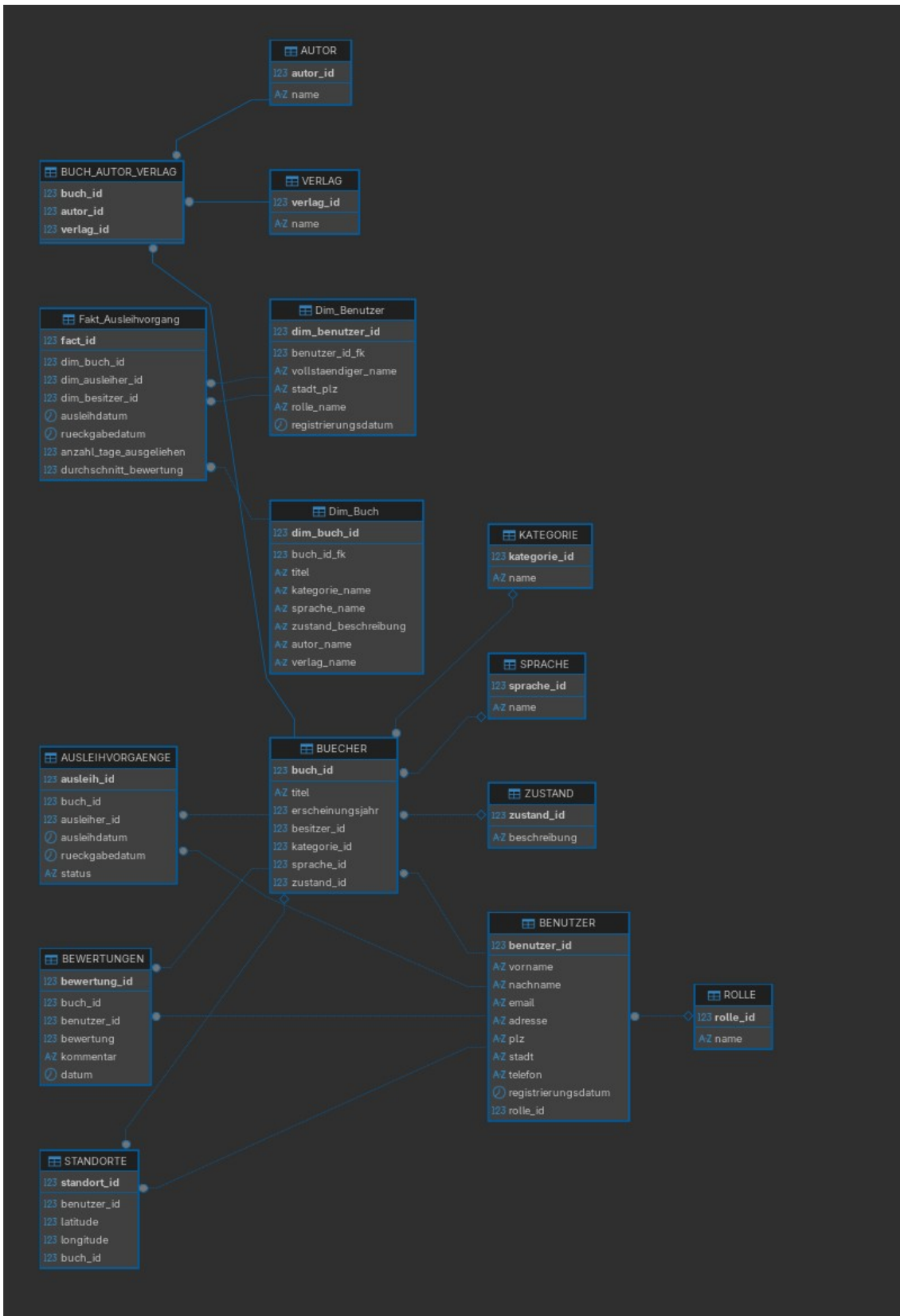
1.1 Projektbeschreibung

Die "Buchtausch-App" ist ein relationales Datenbanksystem, das entwickelt wurde, um den Austausch von Büchern zwischen Nutzern effizient zu verwalten. Das System ermöglicht es Benutzern, eigene Bestände zu registrieren, Bücher anderer Nutzer auszuleihen, Standorte zu verwalten und Bewertungen abzugeben.

1.2 Kernfunktionalitäten

Das System zeichnet sich durch folgende technische Merkmale aus:

- **Zentrale Bestandsverwaltung:** Umfassende Abbildung von Büchern inklusive Metadaten wie Kategorien, Sprachen und physischem Zustand.
- **Komplexe Beziehungslogik (n:m:m):** Eine Besonderheit des Systems ist die Auflösung der dreifachen Beziehung zwischen Büchern, Autoren und Verlagen über die Tabelle BUCH_AUTOR_VERLAG. Dies ermöglicht eine redundanzfreie Zuordnung, selbst wenn ein Buch mehrere Autoren oder verschiedene Verlagsausgaben hat.
- **Leihprozess & Integrität:** Die Tabelle AUSLEIHVORGAENGE steuert den Status der Bücher. Durch Foreign-Key-Constraints (InnoDB) wird sichergestellt, dass keine inkonsistenten Daten (z.B. Ausleihe eines gelöschten Buches) entstehen.
- **Geodaten-Anbindung:** Über die Tabelle STANDORTE können Bücher und Nutzer geografisch (Latitude/Longitude) verortet werden, was die Basis für eine spätere Umkreissuche bildet.
- **Analytischer Data Mart:** Parallel zum operativen Schema wurde ein Star-Schema (Data Mart) implementiert. Dieses erlaubt schnelle analytische Abfragen über Ausleihgewohnheiten, ohne die Performance des operativen Systems zu beeinträchtigen.



METADATEN UND TECHNISCHE OPTIMIERUNG

2.1 Datenbank-Metadaten

Nach der erfolgreichen Implementierung und Befüllung mit Testdaten ergeben sich folgende statistische Kennzahlen für das System:

| Metrik | Wert |
|------------------------------------|-----------------------|
| Anzahl der Tabellen | 15 (inkl. Data Mart) |
| Gesamtzahl der Datensätze | 144 Einträge |
| Speicher-Engine | InnoDB (ACID-konform) |
| Datenbankvolumen (Daten + Indizes) | 0,56 MB |
| Unterstützte Relationen | 1:1, 1:n, n:m, n:m:m |

2.2 Umsetzung des Tutor-Feedbacks

Basierend auf dem Feedback aus der zweiten Phase wurden gezielte Verbesserungen am System vorgenommen:

- Performance-Optimierung durch Indizierung:** Um die Abfragegeschwindigkeit bei steigenden Datensätzen zu gewährleisten, wurden zusätzliche Indizes auf häufig genutzte Fremdschlüsselspalten gesetzt.
 - `idx_buch_besitzer` auf `BUECHER(besitzer_id)`
 - `idx_ausleihe_buch` auf `AUSLEIHVORGAENGE(buch_id)`
 - `idx_standort_koordinaten` auf `STANDORTE(latitude, longitude)`
- Validierung der referenziellen Integrität:** Es wurden zusätzliche Testläufe durchgeführt, um sicherzustellen, dass Löschweiterleitungen (Cascades) oder Einschränkungen (Restricts) korrekt greifen, insbesondere bei der komplexen Verknüpfung von Autoren und Verlagen.
- Erweiterte Testfälle:** Die Testdaten wurden auf insgesamt 144 Zeilen erweitert, um Grenzfälle (z.B. ein Benutzer ohne Ausleihen oder ein Buch mit mehreren Autoren) besser abzubilden.

REPOSITORY UND INSTALLATION

3.1 GitHub Repository

Der vollständige Quellcode, inklusive aller SQL-Skripte, Dokumentationen und Screenshots, ist unter folgendem Link verfügbar:

GitHub Link: https://github.com/Unadvised7311/Buchtausch_App

3.2 Installationsanleitung (Kurzform)

Um das System in einer lokalen MariaDB-Umgebung zu replizieren, folgen Sie diesen Schritten:

1. Klonen Sie das Repository oder laden Sie die Datei `buchtausch_app.sql` herunter.
2. Öffnen Sie Ihr DBMS (z. B. DBeaver).
3. Erstellen Sie eine neue Verbindung zu Ihrer MariaDB-Instanz.
4. Führen Sie das gesamte Skript aus (Alt + X in DBeaver). Alle Tabellen werden automatisch angelegt, befüllt und optimiert.

3.3 Fazit und Reflexion

Die Entwicklung des Buchtausch-App-DBMS demonstriert den Übergang von einem konzeptionellen ER-Modell zu einer hochperformanten, normalisierten Datenbank. Besonders lehrreich war die Implementierung des Data Marts, da hier deutlich wurde, wie unterschiedliche Datenstrukturen (3NF für Transaktionen vs. Star-Schema für Analysen) innerhalb eines Systems koexistieren können. Durch das Feedback der Tutoren wurde zudem das Bewusstsein für Datenbank-Performance (Indizierung) geschärft.