

1. Funktionalität des DBMS

Die entwickelte Datenbank für die Buchtausch-App bildet den vollständigen Lebenszyklus einer Buchausleihe ab und unterstützt dabei sämtliche Kernprozesse sowie ergänzende Funktionen für Kommunikation und Support. Grundlage ist ein relationales Schema in MariaDB/MySQL mit 21 normalisierten Tabellen (3NF/BCNF), das Datenintegrität und Redundanzfreiheit gewährleistet.

Benutzer- und Rollenverwaltung

Zentrale Entität ist die Tabelle *benutzer*, in der alle Stammdaten wie Benutzername, Name, E-Mail und Adresse gespeichert werden. Für Sicherheit und Konsistenz sorgen eindeutige Schlüssel (UNIQUE auf Benutzername und E-Mail) sowie ein Passwort-Hash. Über Spezialisierungstabellen werden Rollen abgebildet:

- *admin* für Administratorrechte (Pflege, Sperrungen, Datenkorrekturen),
- *verleiher* für Benutzer, die eigene Exemplare bereitstellen,
- *ausleiher* für Benutzer, die Bücher ausleihen.

CHECK-Constraints stellen sicher, dass der Status eines Benutzers ausschließlich „aktiv“ oder „gesperrt“ annehmen kann. Damit wird verhindert, dass inkonsistente Zustände entstehen.

Verwaltung von Werken und Exemplaren

Bibliographische Daten werden in der Entität *buch_werk* geführt. Sie enthält Informationen wie Titel, ISBN, Erscheinungsjahr, Sprache und Verlag. Über m:n-Beziehungen zu den Entitäten *autor* und *genre* wird eine flexible Modellierung realisiert, sodass ein Werk mehrere Autoren und Genres haben kann. Aus einem Werk können beliebig viele *buch_exemplar*-Einträge hervorgehen, die einzelnen Verleihern gehören. Jedes Exemplar hat Attribute wie Zustand, Status, maximale Ausleihdauer und Abholort. Zustände (z. B. „neu“, „sehr gut“, „antik“) sind in einer eigenen Entität normiert und durch ein UNIQUE-Constraint abgesichert. Für Abholorte und Zeitfenster existieren eigene Tabellen, die es ermöglichen, detailliert festzulegen, wo und wann ein Buch abgeholt werden kann.

Prozesse: Ausleihe, Reservierung, Bewertung

Die eigentlichen Transaktionen werden über die Prozesse *ausleihe* und *reservierung* abgebildet.

- Bei der Ausleihe wird ein Exemplar mit einem Ausleiher verknüpft, inklusive Start- und Enddatum sowie einem Status („aktiv“, „abgeschlossen“). Redundanzen wurden vermieden, indem die Spalte *verleiher_id* entfernt wurde – diese Information lässt sich bereits aus der Beziehung zum Exemplar ableiten. Zusätzlich sichern CHECK-Constraints die Konsistenz der Datumsbereiche.

- Die Reservierung stellt eine Dreifachbeziehung zwischen Ausleiher, Exemplar und Zeitraum dar. Auch hier werden Statuswerte geprüft und die Gültigkeit der Datumsbereiche erzwungen.
- Bewertungen ermöglichen es, dass jedes Werk von einem Benutzer einmalig mit Sternen (1–5) und Kommentar bewertet werden kann. Dies wird durch ein UNIQUE-Constraint auf (benutzer_id, werk_id) und einen CHECK auf den Sternebereich garantiert. Ein Default-Timestamp stellt sicher, dass das Erstellungsdatum automatisch erfasst wird.

Kommunikation und Support

Zur Unterstützung des Austauschs zwischen den Benutzern wurde ein Chatsystem integriert. Über die Tabellen *chat_thread* und *chat_nachricht* können Nachrichten thematisch gebündelt werden. Default-Werte für Timestamps und Lesestatus („gelesen“ = FALSE) gewährleisten, dass Nachrichten chronologisch korrekt erfasst und verarbeitet werden.

Für technische oder organisatorische Probleme existiert die Entität *hilfe_ticket*. Benutzer können Tickets mit Betreff und Beschreibung anlegen, deren Status („offen“, „in Bearbeitung“, „geschlossen“) ebenfalls durch Constraints kontrolliert wird. Dadurch wird ein konsistenter Workflow im Supportprozess ermöglicht.

Geografische Funktionalität

Eine wesentliche Erweiterung gegenüber den ursprünglichen Anforderungen ist die Integration von Geodaten. Die Adresstabelle wurde um Latitude- und Longitude-Koordinaten ergänzt, die über CHECK-Constraints auf gültige Wertebereiche begrenzt sind. Indizes auf PLZ, Ort und Koordinaten beschleunigen Suchanfragen.

Darauf aufbauend wurden spezielle Views (*v_abholort_geo*, *v_exemplar_geo*) erstellt, die die Verknüpfung von Exemplaren mit ihren Abholorten und geografischen Daten ermöglichen. Über die implementierte Haversine-Funktion lassen sich Umkreissuchen mit realistischen Distanzberechnungen durchführen. Zudem wurde eine Stored Procedure (*search_exemplare_nearby*) entwickelt, die kombinierte Suchanfragen erlaubt: Benutzer können sowohl über PLZ, Ort oder Straße als auch über Radiusparameter Bücher in ihrer Umgebung finden. Damit wird die Praxisnähe der App erheblich gesteigert, da Nutzer gezielt nach lokal verfügbaren Büchern suchen können.

Qualitätssicherung und Testabfragen

Die Funktionalität des Systems wurde umfassend getestet. Dazu wurden alle Tabellen mit mindestens zehn Dummy-Datensätzen gefüllt, die realistische Szenarien abbilden (z. B. unterschiedliche Zustände, reservierte oder ausgeliehene Exemplare, verschiedene Genres und Autoren). Für jede Entität existiert mindestens eine Testabfrage, die in phpMyAdmin ausgeführt und dokumentiert wurde. Neben einfachen SELECT-Abfragen wurden auch komplexere SQL-Techniken eingesetzt, darunter INNER, LEFT, RIGHT und NATURAL JOINS, Aggregationen (GROUP BY, HAVING), Sortierungen (ORDER BY) sowie Views für wiederkehrende Abfragen. Damit konnte die Praxisnähe und Stabilität der Datenbank nachgewiesen werden.

2. Metadaten & Kennzahlen

Die finale Datenbank umfasst folgende Metadaten. Screenshots der Abfragen sind im ZIP-Ordner enthalten.

Gesamtübersicht:

Kennzahl	Wert
Anzahl Tabellen	21
Datenbankgröße gesamt	0,85 MB

Detailübersicht je Tabelle:

Tabelle	Zeilen	Speicher (KB)
adresse	20	80 KB
benutzer	20	48 KB
buch_exemplar	20	80 KB
buch_werk	12	48 KB
ausleihe	12	48 KB
reservierung	12	48 KB
bewertung	12	48 KB
autor	15	32 KB
genre	12	32 KB
werk_autor	12	32 KB
werk_genre	17	32 KB
zustand	10	32 KB
abholort	12	48 KB
zeitfenster	12	32 KB
verlag	10	48 KB
chat_thread	10	32 KB
chat_nachricht	12	48 KB
hilfe_ticket	10	32 KB
admin	3	16 KB
verleiher	10	16 KB
ausleiher	16	16 KB

3. GitHub

Das Projekt wurde in einem GitHub-Repository veröffentlicht. Der Link zum Repository lautet:

<https://github.com/HlrA-RGB-0/buchtausch-app-db>