

## Datenbanksysteme 2. Praktisches Projekt

# Einführung

Im Laufe des praktischen Projekts sollen Sie ein Online-Tagebuch entwickeln. Die Entwicklung dieses Projekts soll dabei in vier Schritten ablaufen:

1. ER-Modellierung der Datenbank
2. Überführung der ER-Modellierung in ein relationales Modell
3. Implementierung der Datenbank in SQLite
4. Umsetzung des dazugehörigen Anwendungsprogramms in Java

Die einzelnen Arbeitsschritte bauen hierbei aufeinander auf. Die aktuelle Aufgabenstellung wird im letzten Abschnitt des Kapitels „Aufgabenstellung“ beschrieben.

## Wichtige allgemeine Hinweise

Für die ersten drei Schritte haben Sie jeweils einen Bearbeitungszeitraum von einer Woche, für die abschließende Implementierung haben Sie drei Wochen Zeit. Um die praktischen Übungen zu bestehen, müssen Sie *jeden* der vier Abschnitte bestehen. Dies bedeutet, dass Sie den zweiten Schritt nur bearbeiten können, wenn Sie den ersten *bestanden* haben. Klären Sie daher kritische Punkte und Fragen *vor* der Abgabe in den Sprechstunden, da es aufgrund des Zeitplans keine Nachbearbeitungszeit geben kann.

Sie müssen alle Arbeitsschritte *alleine* und selbstständig bearbeiten. Gruppenarbeiten, auch bei nicht ganz identischen Abgaben, führen zum Ausschluss aller Beteiligten.

Die Abgabe dieses dritten Teils ist bis Montag, den 12.11.2018, um 10:00 Uhr über das Abgabesystem möglich. Achten Sie auf die **aktuelle Aufgabenstellung** am Ende des Übungsblattes.

newpage

## Anwendungsszenario

Das Ziel der gesamten praktischen Übung besteht in der Implementierung eines Online-Tagebuchs. Die explizite Aufgabe für dieses Blatt finden Sie weiter unten.

Es sollen folgende Eigenschaften dargestellt werden:

Ein **Benutzer** besitzt einen Vor-, sowie Nachnamen, eine eindeutige E-Mail-Adresse und ein Passwort.

Ein **Autor** ist ein Benutzer. Ein Benutzer kann beliebig viele Autoren abonnieren. Ein Autor kann von beliebig vielen Benutzern abonniert werden. Der Autor kann beliebig viele andere Autoren empfehlen, sowie von beliebig vielen anderen Autoren empfohlen werden. Der Autor besitzt ein Pseudonym und ein optionales Avatar. Er legt einen Preis für sein Tagebuch fest, sofern es mindestens eine private Seite enthält (sonst gilt Preis = 0). Der Benutzer kann zu jedem Autor genau eine Bewertung abgeben.

Eine Bewertung besteht aus einem Text und einer Benotung (1-5 Sterne).

Ein Autor kann jeden Tag (Datum) genau eine **Seite** in seinem Tagebuch anlegen. Jede erstellte Seite hat eine Nummer und einen Typ (privat/öffentlich). Jede Seite wird von genau einem Autor angelegt. Sie enthält mindestens einen Eintrag.

Ein Autor kann beliebig viele **Einträge** schreiben. Jeder Eintrag wird von genau einem Autor geschrieben und enthält einen Titel, eine Uhrzeit, einen Text und gehört zu genau einer Seite (nur einer eigenen Seite des Autors). Außerdem kann ein Eintrag beliebig viele **Bilder** enthalten.

Jedes **Bild** gehört zu genau einem Eintrag. Das Bild wird in der Datenbank als Blob abgelegt. Zu manchen Bildern werden die GPS-Koordinaten (Längen- und Breitengrad) gespeichert.

Zu jedem Bild können beliebig viele **Tags** gesetzt werden. Jedes Tag ist eindeutig und kann beliebig oft verwendet werden. Sowohl die Tags als auch die Bilder werden von Autoren erstellt/zugewiesen/hochgeladen.

Ein Benutzer führt **Transaktionen** unter Angabe von Zweck aus. Optional kann zu einer Transaktion ein Gutscheincode angegeben werden. Mit dem Gutscheincode können die Tagebücher ebenfalls erworben werden, sofern der Wert ausreicht. Ist der Gutscheinwert höher als der Preis eines Tagebuches, freut sich der Autor ganz besonders über den Verkauf.

Jede **Transaktion** enthält einen Betrag, ein Datum und eine Zahlungsmittelwahl (z.B. Paypal). Jede Transaktion wird von genau einem Autor empfangen.

Eine detaillierte Auflistung der geforderten Funktionalität des entgeltigen Programms folgt auf dem letzten Aufgabenblatt. Die Benutzer des Systems sollen unter anderem folgende Aktionen durchführen können:

- Ein neuer Benutzer soll die Möglichkeit haben sich zu registrieren.
- Ein registrierter Benutzer soll die Möglichkeit haben sich anzumelden.
- Ein registrierter Benutzer soll die Möglichkeit haben öffentliche Seiten von beliebigen Tagebüchern zu lesen.
- Ein registrierter Benutzer soll Zugriff zu allen Seiten eines Tagebuches erhalten, sofern er diesen erworben hat.
- Ein Autor soll einen Avatar hochladen können.
- Ein Autor soll sein Tagebuch führen können. Auch eine spätere Korrektur der Seiteninhalte soll möglich sein.
- Es soll möglich sein, nach Autoren zu suchen.
- Es soll möglich sein, nach einem oder mehreren Tags zu suchen und sich Bilder zu Einträgen auf öffentlichen Seiten anzeigen zu lassen. Bei privaten Inhalten soll nur das Vorhandensein dieser Information angezeigt werden und die Möglichkeit gegeben werden, diese zu erwerben.

## Aufgabenstellung (3.Teil)

Geben Sie gegebenenfalls eine **korrigierte** Version Ihres Relationenmodells an. Setzen Sie dieses dann in SQLite (empfohlene Version 3.16.2) um. Legen Sie dabei für *jede* Tabelle mindestens 2 Beispieleinträge an und geben Sie abschließend die entstandene SQLite Datenbank gezippt und die Statements zur Erzeugung und Füllung der Datenbank als eine reine Textdatei (\*.txt) ab. Beachten Sie bitte hierbei, dass die Daten bereits beim Einfügen in die Datenbank kontrolliert werden müssen (z.B. Wertebereich) und nicht erst später in der GUI. Falls Sie vor der Abgabe Ihre Datenbank auf einem unserer Rechner testen möchten, benötigen Sie Zugangsdaten. Diese bekommen Sie in Ihrer Sprechstunde bei Ihrem Tutor.

Geben Sie weiterhin Anfragen an, die Folgendes auf Ihrer Datenbank beantworten:

- Geben Sie genau zwei Tags aus, die am häufigsten verwendet werden.
- Geben Sie genau die Autoren an, die mit ihren Online-Tagebüchern am meisten verdient hat (inkl. Verdienst).
- Geben Sie genau die Einträge aus, die den Titel "Silvester in London" besitzen und von Autoren mit einer Durchschnittsbewertung zwischen vier und fünf Sternen (einschließlich) geschrieben sind.
- Geben Sie genau die Tagebücher aus, die Bilder enthalten, welche am Äquator (GPS) entstanden sind.
- Geben Sie genau die Autoren aus, die mindestens drei öffentliche und keine privaten Seiten angelegt haben.

Achten Sie bitte darauf, dass Ihre Datenbank für die Anfragen passende Einträge enthält und eine nicht leere Ergebnismenge liefert.

# Hinweise

## 1 PRAGMA

Es empfiehlt sich folgende PRAGMA-Statements zu verwenden:

```
PRAGMA auto_vacuum = 1;  
PRAGMA automatic_index = 1;  
PRAGMA case_sensitive_like = 0;  
PRAGMA defer_foreign_keys = 0;  
PRAGMA encoding = "UTF-8";  
PRAGMA foreign_keys = 1;  
PRAGMA ignore_check_constraints = 0;  
PRAGMA journal_mode = WAL;  
PRAGMA query_only = 0;  
PRAGMA recursive_triggers = 1;  
PRAGMA reverse_unordered_selects = 0;  
PRAGMA secure_delete = 0;  
PRAGMA synchronous = NORMAL;
```

## 2 Datentypen

Es dürfen die folgenden Datentypen benutzt werden:

- INTEGER
- DOUBLE
- VARCHAR
- TEXT
- BOOLEAN
- BLOB
- TIME
- DATE
- DATETIME

Nähere Informationen zu den Datentypen sind der SQLite-Dokumentation zu entnehmen.

### 3 Wertebereich

Folgende Bedingungen müssen eingehalten werden:

Wertebereiche müssen entsprechend des Anwendungsszenarios sinnvoll gewählt und geprüft werden. Somit sollen zum Beispiel Namen nur Zeichen aus dem deutschen Alphabet enthalten und gültige E-Mail-Adressen dem Format "local-part@domain-part" entsprechen. Passwörter müssen mindestens sechs Zeichen lang sein und dabei mindestens einen Großbuchstaben und mindestens eine Ziffer enthalten. Gültige Preise müssen als Betrag in Ziffern mit englischem Dezimaltrennzeichen gespeichert sein. Alle Datumsangaben müssen dem Standard-SQLite-Format entsprechen. Für die spätere Verwendung besteht die Notwendigkeit, Bilder im PNG-Format zu speichern. Alle nicht optionalen Attribute, die in mindestens einem Element des (zum entsprechenden Attribut gehörigen) Wertebereichs mindestens ein Nicht-Ziffer-Zeichen enthalten, müssen auf eine Länge größer null geprüft werden. Alle vorgegebenen Wertebereiche sind einzuhalten. Weiterhin dürfen Wertebereiche, die keine Zahlenmengen sind, nicht zwischen Groß- und Kleinschreibung unterscheiden.