



Einsendeaufgabe 1

zum Lernheft

Datenbankentwicklung für Webanwendungen mit MySQL

Weiterentwicklung einer Game-Community-Datenbank

Einführung

Ihr Kunde ist Geschäftsführer einer kleinen, aber im deutschsprachigen Raum sehr etablierten Game-Community. Bisher besteht die Community im wesentlichen aus einem Forum und einem Bereich mit Spieleberichten. Es gibt eine Handvoll von Autoren, die diese Berichte schreiben. Die Datenbanken für normale Autoren und Benutzer waren bisher getrennt. Der Kunde hat nun einige neue Anforderungen, die ein Überarbeiten der Datenbankstruktur erfordern.

Bestehende Datenbank

Hier sehen Sie das DDL der bestehenden Datenbank:

```
CREATE TABLE articles (id INTEGER PRIMARY KEY AUTO_INCREMENT, name VARCHAR(80),  
body TEXT, created_at DATETIME, updated_at DATETIME, author_id INTEGER);  
CREATE TABLE authors (id INTEGER PRIMARY KEY AUTO_INCREMENT, name VARCHAR(50),  
first_name VARCHAR(30), salutation VARCHAR(10), email VARCHAR(60),  
password_hash varchar(20), area_of_expertise VARCHAR(80), created_at DATETIME,  
updated_at DATETIME);  
CREATE TABLE forums (id INTEGER PRIMARY KEY AUTO_INCREMENT, name VARCHAR(40),  
description TEXT);  
CREATE TABLE posts (id INTEGER PRIMARY KEY AUTO_INCREMENT, user_id INTEGER,  
topic_id INTEGER, body TEXT);  
CREATE TABLE topics (id INTEGER PRIMARY KEY AUTO_INCREMENT, forum_id INTEGER,  
title VARCHAR(30));  
CREATE TABLE users (id INTEGER PRIMARY KEY AUTO_INCREMENT, name VARCHAR(40),  
first_name VARCHAR(40), salutation VARCHAR(10), created_at DATETIME, email  
VARCHAR(80), password_hash varchar(20));  
CREATE TABLE games (id INTEGER PRIMARY KEY AUTO_INCREMENT, title VARCHAR(100),  
year_published INTEGER, description TEXT, created_at DATETIME);  
CREATE TABLE article_is_about_games (article_id INTEGER, game_id INTEGER,  
PRIMARY KEY(article_id, game_id));
```

Aufgabe

Entwickeln Sie ein Migrationsskript, das die alte Struktur in eine neue überführt – entsprechend den neuen Kundenanforderungen (siehe unten). Achten Sie darauf, dass die neue Struktur alle Daten der alten speichern kann – solange das nicht im Widerspruch mit den neuen Anforderungen steht. Außerdem sollten Sie die Migration so verlustfrei wie möglich gestalten. Versuchen Sie also, bestehende Daten beizubehalten. Sie müssen jedoch **keine Datenmigration** durchführen.

Achten Sie beim Migrationsskript vor allem darauf, dass die SQL-Anweisungen keine überflüssigen Informationen enthalten, wie z.B. Zeichenkodierung oder Storage Engine.

Ergebnisse

Reichen Sie bitte folgende Dokumente ein:

- ein Domänenmodell, das die neuen Anforderungen abdeckt, als UML-Klassendiagramm (im Datei-Format PNG).
- ein daraus abgeleitetes physisches Datenbankmodell als UML-Klassendiagramm (im Datei-Format PNG).
- ein SQL-Migrationsskript, das die alte DB-Struktur in die neue überführt.



Anforderungen

Der Kunde formuliert folgende Anforderungen:

- Autoren sollen keinen getrennten Login mehr haben. Sie loggen sich zukünftig auf der gleichen Seite ein wie normale Benutzer. Normale Benutzer haben natürlich dennoch kein Recht, Artikel zu verfassen. [A1]
Hinweis: Integrieren Sie dazu die Autoren in die Tabelle `users`. Mit Hilfe eines Attributes `is_author` schaffen Sie die Grundlage zur Überprüfung des Autorenrechts.
- Ein Benutzer oder Autor meldet sich nicht mehr mit seiner E-Mail-Adresse, sondern einem dafür neu eingeführten Benutzernamen (`user_login_name`) an. [A2]
- Die Idee, dass ein Artikel sich auf mehrere Spiele beziehen kann, hat sich nicht bewährt. In Zukunft bezieht sich jeder Artikel immer auf ein bestimmtes Spiel. Zu einem Spiel kann es aber, nach wie vor, mehrere Artikel geben. [A3]
- Benutzer dürfen künftig ein Bild von sich hochladen (das Bild muss nicht in der Datenbank selbst gespeichert werden). [A4]
- Bestehende Forenposts dürfen noch 5 Minuten lang vom Poster geändert werden - danach nicht mehr. [A5]
- In Zukunft soll jeder Benutzer gleich sehen können, welche Forenbeiträge (Posts) er sich bereits angeschaut hat. [A6]
- Die alte Datenbank hat praktisch keine Validierungen integriert (leere Namen, doppelte E-Mail-Adressen etc.). Stellen Sie fest, welche Prüfungen erforderlich sind, und implementieren Sie diese. [A7]

Häufige Fragen

1. Soll das Migrationsskript den neuen Stand des Datenbankschemas erzeugen oder das bestehende Schema in das neue überführen?
 - Letzteres. Sie können das Skript testen, in dem Sie das SQL der bestehenden Datenbank einfach aus diesem Dokument übernehmen und auf einer leeren Datenbank ausführen. Auf dem resultierenden Schema muss das Migrationsskript fehlerfrei laufen.
2. Sollen die Daten selbst auch migriert werden oder nur das Schema?
 - Das Schema reicht aus. In der Praxis würden Sie natürlich auch die Daten migrieren, aber das ist nicht Teil dieser Aufgabe.
3. Werden ALTER- oder CREATE-Statements benötigt?
 - Bei einer Migration von einem alten Stand auf einen neuen können grundsätzlich beide Arten von Statements (und weitere) benötigt werden - überwiegend allerdings eher ALTER-Statements.

