Institut für Informatik
Datenbanken und Informationssysteme
Prof. Dr. Stefan Conrad
Kirill Bogomasov

Datenbanksysteme
2. Praktisches Projekt

Einführung

Im Laufe des praktischen Projekts sollen Sie ein Werkstattportal 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 eines dazugehörigen RESTful Web Services

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

Wichtige allgemeine Hinweise

Für die ersten zwei Schritte haben Sie jeweils einen Bearbeitungszeitraum von einer Woche, für die Implementierung der Datenbank in SQLite und die abschließende Umsetzung eines dazugehörigen RESTful Web Services haben Sie jeweils zwei 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. Näheres zu der Veranstaltung finden Sie auf unserer Homepage.

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

Die Abgabe dieses vierten Teils ist bis Montag, den 20.07.2020, um 10:00 Uhr über das Abgabesystem möglich. Achten Sie auf die **aktuelle Aufgabenstellung** am Ende des Übungsblattes.

Anwendungsszenario

Das Ziel der gesamten praktischen Übung besteht in der Implementierung eines Systems zur Verwaltung eines Werkstattportals. Die explizite Aufgabe für dieses Blatt finden Sie weiter unten.

Es sollen folgende Sachverhalte dargestellt werden:

In unserer Datenbank sind **Werkstätten** hinterlegt. Zu jeder Werkstatt liegt ein Name, sowie Information zum Preis je AW[1] vor. Jede Werkstatt kann mehrere **Adressen** besitzen. Zu jeder Adresse werden die Stadt, PLZ, Straße und eine Hausnummer gespeichert. Jede Werkstatt hat mindestens einen **Ansprechpartner**.

Ein Ansprechpartner vertritt genau eine Werkstatt. Dabei besitzt er eine Mitarbeiternummer (MA-Nr.) und ist **User**. Zu jedem User werden Vorname, Nachname, Passwort, sowie eine eindeutige E-Mail-Adresse gespeichert. Ansprechpartner erstellen beliebig viele **Aufträge**.

Kunden sind ebenfalls User. Optional kann einem Kunden eine Rückrufnummer hinterlegt werden. Jeder Kunde hat genau eine Adresse. Alle Kunden können Werkstätten nach dem Schulnotensystem bewerten. Manche Kunden hinterlassen eine Werbeeinwilligung und dürfen deshalb kontaktiert werden. Kunden können unendlich viele **Fahrzeuge** besitzen. Jedes Fahrzeug gehört dabei genau einem Kunden.

Ein Fahrzeug besitzt ein eindeutiges Kennzeichen, einen Eintrag zur Erstzulassung sowie HU-Datum (MM/YYYY). Jedes Fahrzeug ist genau einem **Modell** zugeordnet. Jedes Modell hat einen Typ (z.B. Limousine, Cabrio, ...), eine Fahrzeugklasse (z.B. PKW, LKW, ...), Leistungsangabe (in kW), eine Bezeichnung (z.B. Golf, Tiguan, ...) und einen Hersteller. Zu jedem Fahrzeug können beliebig viele **Aufträge** existieren. Ein Auftrag ist hingegen genau einem Fahrzeug zugeordnet. Jeder Auftrag wird von genau einem Ansprechpartner erstellt. Ein Auftrag enthält das Auftragsdatum, die AW-Anzahl, sowie einen beschreibenden Text.

Die Durchführung eines Auftrags benötigt eine beliebige Anzahl an **Ersatzteilen**. Jedes Ersatzteil besitzt eine eindeutige Ersatzteilnummer, eine Bezeichnung, eine optionale Abbildung sowie Herstellerinformation. Manche Ersatzteile können durch ein anderes Ersatzteil ersetzt werden und können widerrum ebenfalls Ersatz für ein anderes Ersatzteil sein.

Ersatzteile sind genau einer **Kategorie** zugeordnet. Eine Kategorie besitzt dabei eine eindeutige Bezeichnung (z.B. Reifen). Ersatzteile werden zu einem Stückpreis durch einen **Lieferanten** bereitgestellt. Der Lieferant ist ein User und hat einen Händlernamen.

[1] https://www.pitlane4.de/was-ist-eine-aw-eigentlich/

4. Aufgabenteil

Allgemein

Ihre Aufgabe besteht nun darin, ein Java-Programm zu schreiben, durch das Anwender die Möglichkeit haben, mit Ihrer Datenbank zu kommunizieren.

Dieses Programm muss eine REST-API in Form eines RESTful Web Services sein und soll den gelisteten Anforderungen genügen.

Anforderungen

Das Programm soll die Grundfunktionalität eines Systems zur Buchung von Festivals bieten. **Die genaue Auflistung der Anforderungen finden Sie auf dem Zusatzblatt[2].**

Hilfsmittel

Sie dürfen Tools wie beispielsweise Gradle, Frameworks wie beispielsweise Jersey oder Spring und Libraries wie beispielsweise Lombok benutzen, die Ihnen die Entwicklung vereinfachen, solange Sie dadurch noch immer **alle Anforderungen einhalten**.

Es steht Ihnen ein Template für die REST-API zur Verfügung, das Sie benutzen können, aber nicht müssen.

Abgabe

Die Abgabe soll in Form eines ZIP-Archivs erfolgen, welches das Programm als Quellcode bzw. IDE-Projekt und die korrigierte und mit Beispieleinträgen befüllte Datenbank enthält, mit der das Programm kommuniziert. Halten Sie sich bei der Einreichung Ihrer Abgabe an das bereits vom ersten Übungsblatt bekannte Format.

Die Abgabe soll ein als ZIP-Archiv komprimierter Ordner mit Namen dbs-propra-ss2020-<vorname>-<nachname>, wobei <vorname> Ihr Vorname und <nachname> Ihr Nachname ist, und folgender Ordnerstruktur sein:

- phase1 (Ordner mit Inhalt aus Phase 1)
- phase2 (Ordner mit Inhalt aus Phase 2)
- phase3 (Ordner mit Inhalt aus Phase 3)
- phase4 (Ordner für Phase 4)
- phase4/data (Ordner für Datenbank spezifische Daten)
- phase4/data/schema.sql (nur SQLite-DDL-Anweisungen für SQL-Schema)
- phase4/data/data.sql (nur SQLite-DML-Anweisungen für Beispieleinträge)
- phase4/data/database.db (SQLite-Datenbank, welche aus schema.sql und data.sql erstellt werden kann)

[2] https://pad.hhu.de/s/S1tt1j9hB

- phase4/gradle (Ordner für Gradle spezifische Dateien)
- phase4/gradle/wrapper (Ordner für Gradle Wrapper spezifische Dateien)
- phase4/gradle/wrapper/gradle-wrapper.jar (Gradle Wrapper)
- phase4/gradle/wrapper/gradle-wrapper.properties (Einstellungen für Gradle Wrapper)
- phase4/src (Ordner für Quellcode)
- phase4/build.gradle (Buildskript für Gradle)
- phase4/gradlew (ausführbares Skript für Gradle Wrapper)
- phase4/gradlew.bat (ausführbares Skript für Gradle Wrapper)
- phase4/settings.gradle (Einstellungen für Gradle)
- phase4/README.adoc (kritische Entscheidungen und die verwendete SQLite-Version)

Hinweise

Teilen Sie sich die Bearbeitungszeit von 2 Wochen gut ein: Versuchen Sie, so gut es geht, in der ersten Woche fertig zu werden, um für eventuell auftretende Bugs oder Probleme noch genug Zeit zu haben.

Die Korrektur wird zum Teil automatisiert ablaufen. Es ist folglich **zwingend erforderlich, sich genauestens an die Anforderungen zu halten**! Bei Fragen dazu, wenden Sie sich bitte an die Tutoren.