HSR
HOCHSCHULE FÜR TECHNIK
RAPPERSWIL

FHO Fachhochschule Ostschweiz

Übung 2: OR Mapper (Fortsetzung)

Datenbanksysteme 2, FS 2018

PROF. STEFAN KELLER, MARCEL HUBER

Rapperswil, 2018-02-27

IFS INSTITUTE FOR SOFTWARE

Prof. Stefan Keller ■ Marcel Huber
HSR Hochschule für Technik Rapperswil ■ IFS ■ Oberseestrasse 10 ■ CH-8640 Rapperswil
T +41 55 222 46 30 ■ F +41 55 222 46 20 ■ ifs@hsr.ch ■ https://ifs.hsr.ch

Übung 2: OR Mapper (Fortsetzung)

Zielsetzung

Vertiefung in der Anwendung von JPA als OR Mapper.

- Änderungen an Relationen
- Abbildung von Vererbung
- `JPQL`
- explizites Locking

Vorbereitung

Diese Übung baut auf dem Resultat der letzten Übung[1] und der Datenbank `bank`[2] auf.

Vorlagen

Die Projektvorlage für diese Aufgabe ist in dieser Zip-Datei[3]. In dieser finden Sie das Eclipse-Projekt `dbs2.jpa_exercise_continued`[4] mit Java Klassen inklusive JPA-Annotationen. Die JPA Libraries und PostgreSQL-JDBC Drivers für Java-8 sind im Ordner `lib/`[5] **bereits enthalten**.

In der Vorlage zur Übung finden Sie die Datei `persistence.xml`[6], welche Ihnen als Vorlage zur Konfiguration Ihres JPA-Programms dienen kann. Sie müssen dieses eventuell noch an Ihre Umgebung anpassen wenn Sie beispielsweise einen anderen Port zum PostgreSQL[7] Server verwenden.

Datenbank `bank`

- Für diese Übung benötigen Sie PostgreSQL[8] und die Datenbank `bank`[9]

  Import von `bank` geschieht durch Eingabe des folgenden Kommandos innerhalb des entsprechenden Unterverzeichnis

  ```
  psql -U postgres -f 0_runAllScripts.sql
  ```

[1] https://gitlab.dev.ifs.hsr.ch/m1huber/Dbs2Uebungen/blob/v1.2.2/OR-Mapper_JPA/README.md
[2] https://gitlab.dev.ifs.hsr.ch/m1huber/Dbs2Uebungen/blob/v1.2.2/Databases/bank
[3] https://gitlab.dev.ifs.hsr.ch/m1huber/Dbs2Uebungen/-/jobs/artifacts/master/raw/OR-Mapper_JPA_ff.zip?job=OR-Mapper_JPA_ff
[4] https://gitlab.dev.ifs.hsr.ch/m1huber/Dbs2Uebungen/blob/v1.2.2/OR-Mapper_JPA_ff/.project
[5] https://gitlab.dev.ifs.hsr.ch/m1huber/Dbs2Uebungen/blob/v1.2.2/OR-Mapper_JPA_ff/lib
[6] https://gitlab.dev.ifs.hsr.ch/m1huber/Dbs2Uebungen/blob/v1.2.2/OR-Mapper_JPA_ff/src/META-INF/persistence.xml
[7] https://www.postgresql.org
[8] https://www.postgresql.org
[9] https://gitlab.dev.ifs.hsr.ch/m1huber/Dbs2Uebungen/blob/v1.2.2/Databases/bank

Prof. Stefan Keller ■ Marcel Huber
HSR Hochschule für Technik Rapperswil ■ IFS ■ Oberseestrasse 10 ■ CH-8640 Rapperswil
T +41 55 222 46 30 ■ F +41 55 222 46 20 ■ ifs@hsr.ch ■ https://ifs.hsr.ch

oder durch Ausführen von `run.bat`[10] oder `run.sh`[11] im entsprechenden Verzeichnis.

Die Daten befinden sich in diesem Ordner[12] und in dieser zip Datei[13].

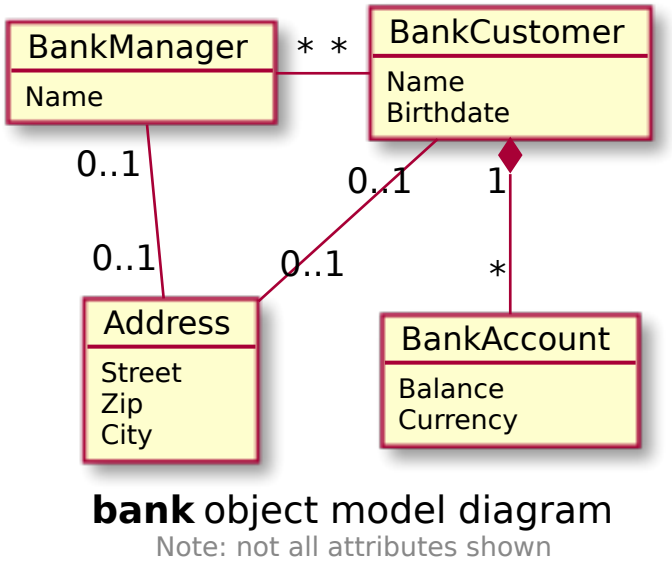
Aufgaben

Aufgabe 1: Bidirektionale Beziehung

Für Ihre JPA Anwendung soll die `N:M`-Beziehung zwischen den Entities `BankManager` und `BankCustomer` bei Änderungen konsistent bleiben. Wenn also beispielsweise `BankCustomer K` nicht mehr von `BankManager M` betreut wird, so darf `BankManager M` auch nicht mehr in der Manager-Liste des `BankCustomer K` erscheinen. Zur Veranschaulichung ist nachfolgend nochmals das Objektmodell abgebildet.

1. Implementieren Sie die konsistente Beziehung für Änderungen und testen Sie es mit einem Beispiel
   - indem Sie eine neue Beziehung einfügen
   - und eine existierende Beziehung entfernen.

Persistentes Objektmodell:

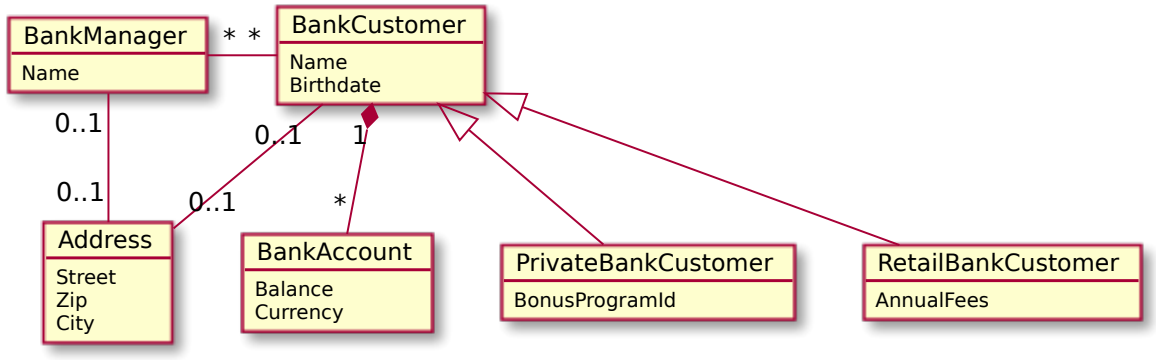


Aufgabe 2: Vererbung

Neu sollen zwei Sub-Klassen von `BankCustomer`, wie im folgenden Klassendiagramm abgebildet, eingeführt werden:

[10] https://gitlab.dev.ifs.hsr.ch/m1huber/Dbs2Uebungen/blob/v1.2.2/Databases/bank/run.bat
[11] https://gitlab.dev.ifs.hsr.ch/m1huber/Dbs2Uebungen/blob/v1.2.2/Databases/bank/run.sh
[12] https://gitlab.dev.ifs.hsr.ch/m1huber/Dbs2Uebungen/blob/v1.2.2/Databases/bank
[13] https://gitlab.dev.ifs.hsr.ch/m1huber/Dbs2Uebungen/-/jobs/artifacts/master/raw/Databases.zip?job=Databases

Prof. Stefan Keller ■ Marcel Huber
HSR Hochschule für Technik Rapperswil ■ IFS ■ Oberseestrasse 10 ■ CH-8640 Rapperswil
T +41 55 222 46 30 ■ F +41 55 222 46 20 ■ ifs@hsr.ch ■ https://ifs.hsr.ch



bank inheritance diagram
Note: not all attributes shown

- `PrivateBankCustomer` (mit spezifischem Attribut `BonusProgramId (int)`)
- `RetailBankCustomer` (mit spezifischem Attribut `AnnualFees (double)`)

1. Wählen Sie eine der vorgestellten Vererbungsmodellierungen auf der DB (`Single Class, Joined Table, Table per Class`) aus.
2. Erweitern Sie die Datenbank `bank` zur Unterstützung dieser Sub-Typen.
   - Erzeugen neuer Tabellen mittels `CREATE TABLE ...`, abhängig vom gewählten Vererbungsmodell.
   - Spaltenerweiterung der Tabelle `bankcustomer` mittels `ALTER TABLE ... ADD COLUMN ...`. Denken Sie dabei auch an einen sinnvollen `DEFAULT` Wert.
   - Setzen Sie den entsprechenden Klassentyp für bestehenden Einträge.
   - Erstellen von ForeignKey Constraints mittels `ALTER TABLE ... ADD FOREIGN KEY ...`, abhängig vom gewählten Vererbungsmodell.
   - Füllen Sie geeignete Werte für `BonusProgramId` und `AnnualFees` ein.

   Tipp Verwenden Sie dazu eine separate sql-Skript Datei, z.B. `5_inheritance.sql`
3. Implementieren Sie das Mapping der Klassen in JPA anhand des von Ihnen gewählten Vererbungsmodells.

Aufgabe 3: JPQL

Setzen Sie folgende Abfragen als JPQL in Ihrem Java-Programm für die Datenbank `bank` um. Erstellen Sie dazu jeweils ein `DynamicQuery` **und** ein `NamedQuery`.

1. Menge aller Kunden mit der Summe der Kontostände pro Kunde.
2. Alle `PrivateBankCustomer` mit Alter >= 30 Jahre, sortiert nach Namen.

Aufgabe 4: Lost Updates

In der letzten Übung wurde die Methode `transfer()` zur Kontoüberweisung auf Basis von JPA implementiert.

Prof. Stefan Keller ■ Marcel Huber
HSR Hochschule für Technik Rapperswil ■ IFS ■ Oberseestrasse 10 ■ CH-8640 Rapperswil
T +41 55 222 46 30 ■ F +41 55 222 46 20 ■ ifs@hsr.ch ■ https://ifs.hsr.ch

1. Definieren Sie ein Szenario, welches einen Lost Update Fehler verursacht.
2. Korrigieren Sie anschliessend den Code mit Hilfe von expliziten Locks `EntityManager.lock()`, so dass diese Lost Updates vermieden werden.
3. Testen Sie das gleiche Szenario noch einmal.

Hinweis:

- Um einen Lost Update zu verursachen, können Sie beispielsweise das Programm zu einem bestimmten Zeitpunkt anhalten und derweilen eine externe Transaktion (in der SQL-Konsole) ausführen.

Musterlösung

Die Musterlösung zu den Übungen finden Sie im Branch OR-Mapper_JPA_ff-Solutions[14] oder als zip-Datei[15].

[14] https://gitlab.dev.ifs.hsr.ch/m1huber/Dbs2Uebungen/tree/OR-Mapper_JPA_ff-Solutions/OR-Mapper_JPA_ff
[15] https://gitlab.dev.ifs.hsr.ch/m1huber/Dbs2Uebungen/-/jobs/artifacts/OR-Mapper_JPA_ff-Solutions/raw/OR-Mapper_JPA_ff-Solutions.zip?job=OR-Mapper_JPA_ff-Solutions

Prof. Stefan Keller ■ Marcel Huber
HSR Hochschule für Technik Rapperswil ■ IFS ■ Oberseestrasse 10 ■ CH-8640 Rapperswil
T +41 55 222 46 30 ■ F +41 55 222 46 20 ■ ifs@hsr.ch ■ https://ifs.hsr.ch