

Aufgabenblatt 5 JDBC und BCNF

- Abgabetermin: **Mittwoch, 8.7.09, 11:00 Uhr**
- Zur Prüfungszulassung muss ein Aufgabenblatt mit mind. 25% der Punkte bewertet werden und alle weiteren Aufgabenblätter mit mindestens 50% der Punkte.
- Die Aufgaben sollen in Zweiergruppen bearbeitet werden.
- Abgabe:
 - per E-Mail an `dbs1-2009@hpi.uni-potsdam.de` mit Subject
Abgabe DBS I: Aufgabenblatt <n> Namen
 - **Aufgabe 1: eine java-Datei namens <datenbankEurerGruppe>.java,**
Aufgabe 2: eine pdf-Datei namens blatt5aufgabe2<Nachnamen>.pdf
 - jedes Blatt beschriftet mit Namen und Matrikelnummern
 - Wir korrigieren die Abgaben aufgabenweise. Das beschriebene Verfahren vereinfacht uns die Arbeit erheblich!
- Der DB2-JDBC Treiber (`db2jcc.jar` für JDBC Version ≤ 3.0 bzw. `db2jcc4.jar` für JDBC Version 4.0) liegt im DB2-Installations-Verzeichnis unter `SQLLIB\java` (auf den Poolrechnern unter `C:\Program Files\IBM\SQLLIB\java`).
- Die Instanz (mit allen Datenbanken) ist über Port 50000 zu erreichen.
- Eine gute Hilfe bietet u. a. das Manual „Developing Java Applications“, zu finden unter <http://www-1.ibm.com/support/docview.wss?rs=71&uid=swg27009552>.

Aufgabe 1: Konsolidierung der IMDB-Datenbank

In den vergangenen Übungen wurden bereits einige Unzulänglichkeiten unseres Ausschnittes der IMDB-Datenbank deutlich (z. B. SchauspielerInnen, die in Filmen mitspielen, die nicht in der Tabelle `MOVIE` enthalten sind). Das Ziel dieser Aufgabe besteht unter anderem darin, solche Probleme zu beheben.

Schreibe unter Verwendung von JDBC ein Java-Programm, das sämtliche nachfolgend aufgeführten Aktionen nacheinander ausführt und die folgenden Parameter akzeptiert:

- `-db <Datenbankname>`
- `-s <Schemaname>` // zu konsolidierendes Schema
- `-u <Benutzername>`
- `-p <Passwort>`

Das Programm soll so geschrieben sein, dass es mit den Argumenten aufgerufen werden kann, sich unter Nutzung dieser Informationen automatisch zur angegebenen Datenbank verbindet und die Verbindung am Ende selbständig wieder schließt. Es soll keinerlei Benutzerinteraktion geben.

- a) Erstelle eine Tabelle `ACT`, die wie folgt aufgebaut ist und zur Speicherung von Schauspielern und Schauspielerinnen dienen soll.

Tabellename: ACT

Spaltenname	Datentyp	Nullwert zugelassen?	Primärschlüsselattribute
Name	<code>varchar(255)</code>	nein	×
<code>isFemale</code>	<code>smallint</code>	nein	–

Gib zusätzlich die Bedingung an, dass `isFemale` nur die Werte 0 (für den Wahrheitswert falsch) und 1 (für den Wahrheitswert wahr) annehmen kann.¹ 3 P

¹Beachte, dass DB2 den Datentyp `BOOLEAN` nicht unterstützt.

- b) Speichere jeden Schauspieler bzw. jede SchauspielerIn, der bzw. die in der Tabelle `ACTOR` bzw. `ACTRESS` gespeichert ist *und* in mindestens einem in der Tabelle `MOVIE` gespeicherten Film mitspielt, in der neu angelegten Tabelle `ACT`. **5 P**
- c) Zwischen dem Entitätstyp `Film` und dem Entitätstyp `SchauspielerIn` besteht eine $n:m$ -Beziehung: ein/e Schauspieler/-in kann in mehreren Filmen mitspielen; in einem Film können mehrere SchauspielerInnen mitspielen. Ein solcher Relationship-Typ wird bei der Übersetzung ins relationale Modell auf eine Relation abgebildet, die sämtliche Primärschlüsselattribute der Tabellen `MOVIE` und `ACT` als zusammengesetzten Primärschlüssel besitzt. Darüber hinaus bilden eventuell vorhandene Attribute des Relationship-Typs die Nichtschlüsselattribute der entsprechenden Relation.
- Lege eine entsprechende Tabelle `ACTMOVIE` an, die zur Speicherung dieser Relation dient. Definiere entsprechende Fremdschlüssel, die sichern, dass Tupel in dieser Tabelle ausschließlich Filme in `MOVIE` und SchauspielerInnen in `ACT` referenzieren. Gib bei der Definition der Fremdschlüssel `ON DELETE CASCADE` an.
- Beachte, dass die Attribute (`Role`, `Order`) des Relationship-Typs in der vorliegenden Übersetzung als Attribute in der Tabelle `ACTOR` (bzw. `ACTRESS`) vorliegen. Beachte bei der Definition ferner, dass beide Attribute optional sind. **5 P**
- d) Befülle die angelegte Tabelle `ACTMOVIE` ausgehend von den Daten, die in den Tabellen `ACTOR` bzw. `ACTRESS` gespeichert sind. Offensichtlich nicht vorhandene Werte für die Attribute `Role` und `Order` sollen jeweils durch den `NULL`-Wert repräsentiert werden. **7 P**
- e) Lösche nun diejenigen Tupel aus den Tabellen `ACTOR` und `ACTRESS`, die sich aus den in `ACTMOVIE` und `ACT` gespeicherten Daten ergeben. Damit verbleiben in `ACTOR` bzw. `ACTRESS` nur noch solche Tupel, die auf einen Film verweisen, der nicht in `MOVIE` existiert. Setze für diese Tupel den Wert des Attributs `Movie_ID` auf den `NULL`-Wert. Beachte, dass es dafür evtl. zuvor erforderlich ist, die Definition der Tabelle `ACTOR` bzw. `ACTRESS` entsprechend zu ändern. **5 P**

Aufgabe 2: BCNF

Gegeben sei das folgende Relationenschema:

`Verschiffung(Schiffsname, Schiffstyp, FahrtID, Ladung, Hafen, Ankunftsdatum)`

Hinweise:

- Ankunftsdatum betrifft die Ankunft eines Schiffes in dem jeweiligen Hafen.
- Zur Angabe der Lösung ist auch die folgende „Kurzschreibung“ der Relation (speziell ihrer Attribute) erlaubt: `Verschiffung(SN, ST, F, L, H, A)`

Gegeben seien weiter die folgenden FDs:

- `Schiffsname` → `Schiffstyp`
- `FahrtID` → `Schiffsname`, `Ladung`
- `Schiffsname`, `Ankunftsdatum` → `FahrtID`, `Hafen`

- a) Identifiziere alle Schlüssel unter Verwendung der Hüllenbildung. **2 P**
- b) Normalisiere die Relation zur BCNF. Stelle die notwendigen Schritte und Entscheidungen nachvollziehbar dar. **4 P**