

## Aufgabenblatt 4

### JDBC

- Abgabetermin: **Dienstag, 17.06.14**
- 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.
- Abgabesystem unter  
<https://www.dcl.hpi.uni-potsdam.de/submit/>
- **Die Abgabe besteht diesmal aus folgenden drei Elementen:**
  - PDF-Dateien *im A4-Format*
  - eine PDF-Datei **Aufgabe-1.pdf** (Lösung zu Aufgabe 1)
  - eine PDF-Datei **Aufgabe-2.pdf** (Ausgabe der zwei gefragten Keywords)
  - eine Java-Datei mit folgendem Dateinamen und entsprechendem Klassennamen:  
**ImdbSuche.java**

### Aufgabe 1: Sichten

a) Erstelle auf Basis der IMDb-Relationen eine View mit den folgenden Informationen:

- die Namen aller Schauspieler und Schauspielerinnen,
- deren Geschlecht ('m'/'f'),
- deren Filme (movie\_id) und
- deren Rolle

Das Schema soll also wie folgt aussehen: `ActorMW(name, sex, movie_id, role)`. Gib das Statement in deiner Abgabe an. **2 P**

b) Für welche beiden Anfragen aus Aufgaben 1 und 2 des letzten Aufgabenblatts (SQL) kann diese View *sinnvoll* verwendet werden? Gib die entsprechenden Anfragen an (Umformulieren ist nicht nötig). **2 P**

## Aufgabe 2: Keyword-Suche für IMDb

In dieser Aufgabe bauen wir eine IMDb-Keyword-Suche. Es soll ein Java-Programm geschrieben werden, das ein Keyword entgegennimmt und alle Filme und Schauspieler(innen) ausgibt, in deren Namen dieses Keyword enthalten ist. Der Zugriff auf die Datenbank (aus dem letzten Aufgabenblatt) erfolgt über JDBC. Hier zwei Beispiele, die die gewünschte Ausgabe darstellen:

Beispiel 1: Keyword = Krug

```
java ImdbSuche -d db2db1 -s 127.0.0.1 -p 50001 -u db2inst1  
-pw ws2011 -k Krug
```

Erwartete Ausgabe:

MOVIES

ACTORS

```
Krug, Manfred  
    PLAYED IN  
        Tatort - Leiche im Keller  
        Tatort - Lockvögel  
        Tatort - Rattenlinie  
    CO-STARS  
        Brauer, Charles (3)  
        Hart, Kurt (2)  
        Hoppe, Edgar (2)  
        Martens, Dirk (I) (2)  
        Becker, Rolf (1)
```

Beispiel 2: Keyword = Blacks

```
java ImdbSuche -d db2db1 -s 127.0.0.1 -p 50001 -u db2inst1  
-pw ws2011 -k Blacks
```

Erwartete Ausgabe:

MOVIES

```
Blacksmith, The, 1922, Comedy, Short  
    Fox, Virginia  
    Keaton, Buster  
    Roberts, Joe (I)
```

ACTORS

```
Blackshaw, Anthony  
    PLAYED IN  
        Fiction Makers, The  
    CO-STARS  
        Armitage, Graham (1)  
        Ashmore, Peter (1)  
        Ball, Ralph (1)  
        Boyd, Roy (1)  
        Clegg, Tom (1)
```

20 P

### Programmablauf

Der Benutzer soll das Programm mit folgenden Parametern starten können:

- -d <Datenbankname>
- -s <IP-Adresse des Datenbank-Servers> [in der VM mittels *ip addr* ermitteln (Eintrag *eth0*), falls im Hostsystem entwickelt wird]
- -p <Port der Datenbank-Instanz> [in der VM *50001*]
- -u <Username> [in der VM *db2inst1*]
- -pw <Passwort> [in der VM *ws2011*]
- -k <Keyword>

Das Programm stellt mittels JDBC eine Verbindung zur Datenbank her (3 P) und ermittelt alle notwendigen Informationen, um folgende Ausgabe zu ermöglichen:

- a) Die Ausgabe beginnt mit dem Teil *MOVIES*. Hier sollen alle Filme gelistet werden, in denen das Keyword im Titel vorkommt. Die Ausgabe ist aufsteigend sortiert nach dem Titel. **1 P**
- b) Für jeden Film wird ausgegeben:
  - 1) Der Titel, das Jahr, und alle Genres des Films (jeweils durch Komma getrennt). **2 P**
  - 2) Außerdem umfasst die Ausgabe für jeden Film die beteiligten Schauspieler(innen), sortiert nach dem Namen. Es werden jedoch nur die ersten fünf ausgegeben. **3 P**
- c) Im zweiten Teil *ACTORS* werden alle Schauspieler(innen) ausgegeben, in deren Namen das gegebene Keyword vorkommt. Das Ergebnis soll dabei aufsteigend nach den gefundenen Namen sortiert werden. **2 P**
- d) Für jede(n) Schauspieler(in) wird ausgegeben:
  - 1) Alle Filme, in denen der Schauspieler mitgewirkt hat (*PLAYED IN*). **3 P**
  - 2) Die Schauspieler(innen), die in denselben Filmen mitgewirkt haben (*CO-STARS*). Hinter jedem/-r Schauspieler(in) steht in Klammern die Anzahl der Filme, in denen beide mitgespielt haben. Es sollen die Top 5 ausgegeben werden, sortiert nach Anzahl gemeinsamer Filme, dann nach Name. **6 P**

Weitere Hinweise:

- Alle Ausgaben sollen grundsätzlich nur einmal erfolgen (z.B. sollen keine Filme doppelt ausgegeben werden).
- Der Name eines Schauspielers kennzeichnet ihn eindeutig.
- Die Nutzung von *PreparedStatement*s wird empfohlen. Vergleiche testweise die Performance mit normalen *Statements* für häufig aufgerufene Anfragen.

### Abgabeform

- Es soll *eine* Java-Klasse abgegeben werden. Die Klasse soll vom DB2-JDBC-Treiber abhängen und von keinen weiteren externen Bibliotheken.
- Es soll darüberhinaus als PDF die Ausgabe des Programms für die folgenden beiden Anfragen abgegeben werden:
  - Keyword = *Prior*
  - Keyword = *Schwarz*

### Allgemeine Hinweise zur Entwicklung

Grundsätzlich kann auch außerhalb der VM im Host-Betriebssystem mit Eclipse entwickelt und von dort aus auf die laufende VM zugegriffen werden. Die individuelle IP-Adresse der VM lässt sich auf der Konsole mit dem Befehl *ip addr* ermitteln, dessen Ausgabe folgendermaßen aussieht:

```
db2inst1@db2expc:~> ip addr
1: lo: <LOOPBACK,UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 brd 127.255.255.255 scope host lo
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,NOTRAILERS,UP> mtu 1500 qdisc pfifo_fast ...
    inet 192.168.163.135/24 brd 192.168.163.255 scope global ...
    inet6 fe80::20c:29ff:fe80:9c99/64 scope link
        valid_lft forever preferred_lft forever
3: sit0: <NOARP> mtu 1480 qdisc noop
    link/sit 0.0.0.0 brd 0.0.0.0
```

Die fett gedruckte IP-Adresse ist diejenige, über die die VM erreichbar ist.

Der Standard-Port der DB2-Instanz lautet *50001*.

Für den Zugriff auf die DB2-Datenbank mittels Java wird der *DB2-JDBC*-Treiber benötigt. Dieser steht auf der Übungsseite zum Download bereit.