Aufgabenblatt 4
SQL

- Abgabetermin: **Dienstag, 07.06.11**
- 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 dbs1-2011@hpi.uni-potsdam.de mit Subject
    Abgabe DBS I: Aufgabenblatt <n> <Namen>
  - ausschließlich pdf-Dateien
  - eine Datei pro Aufgabe mit folgendem Dateinamen:
    blatt<aufgabenblattNr>aufgabe<aufgabenNr><Nachnamen>.pdf
    Bitte **keine Leerzeichen, Unterstriche, Umlaute, Sonderzeichen**, ... im Dateinamen!
  - ***jedes* Blatt beschriftet mit Namen**
  - Wir korrigieren die Abgaben aufgabenweise. Das beschriebene Verfahren vereinfacht uns die Arbeit erheblich!

Vorbereitungen für das Aufgabenblatt: IMDB in DB2 laden

Für die praktischen Aufgaben stellen wir eine Virtual Machine zur Verfügung mit:

- SUSE Linux Enterprise Server
- DB2 Express-C 9.7
- eine db2-Instanz db2inst1 (= Nutzer unter dem ihr euch anmeldet)
- eine (leere) Datenbank db2db1

Hinweise zum Umgang mit der virtuellen Maschine (VM)

- auf den Poolrechnern: Start der VM mittels Klick auf

  `Windows > All Programs > VMware > DB2 Express-C 9.7 32-bit`

- auf dem eigenen Rechner:
  - Voraussetzung: Installierter VMware Player (Freeware)
  - VM kopieren vom Lehrveranstaltungs-Share unter
    `.../FG_Informationssysteme/DBS I/Uebung2011/uebung4-sql/vm`
    (alle Dateien werden benötigt, die ausführbare Datei ist DB2 Express-C 9.7 32-bit.vmx)
  - Starten der VM per
    - Starte VMware Player > Play DB2 Express-C 9.7 32-bit.vmx oder
    - Doppelklick auf DB2 Express-C 9.7 32-bit.vmx

- Die virtuellen Festplatten der VM sind nicht schreibbar, dh. alle gemachten Änderungen gehen nach einem Neutstart der VM verloren, sodass sie sich wieder im „Auslieferungszustand“ befindet. Falls du die VM auf deinem eigenen Rechner verwendest, besteht jedoch die Möglichkeit sie im VMware Player zu „suspenden“ (Virtual Machine > Power > Suspend).
- **Auf den Poolrechnern darf die VM nicht suspended werden**, da du sie sonst für deine Kommilitonen auf dem Rechner blockierst. D. h. VM herunterfahren per `Computer > Shutdown`

- Login-Daten für die VM: User *db2inst1*, Passwort *pefahe22*
- Beim Start fragt die VM ob Software-Updates installiert werden sollen. Das ist nicht notwendig (Remind me later).
- Einstellung für deutsches Tastaturlayout

  ```
  Computer > Control Center > Hardware > Keyboard >
  Layouts > Add > Germany > set as Default
  ```

- Zugriff auf Internet ist freigegeben, d. h. Dateien (Skripte) können auf diesem Wege gespeichert werden.

Möglichkeiten zum Ausführen von Anfragen

- Queries direkt in der Shell ausführen (Beachte die Anführungsstriche!):

  ```
  db2 "select * from actor"
  ```

- Auf der Kommandozeile ein selbstgeschriebenes SQL-Skript starten:

  ```
  db2 -tvf <skriptname>
  ```

- Anfragen direkt im *DB2-Control-Center* im *Query-Executor* eingeben (Zum Ausführen markieren und Klick auf „Play"-Button oder *Strg+Enter*; Falls nichts markiert ist, werden alle Anfragen ausgeführt.)
- Hinweis: Vor dem Ausführen von Anfragen / Anweisungen musst du dich mit der Datenbank verbinden:

  ```
  db2 connect to DB2DB1
  ```

  Trennen der Verbindung nach Beendigung aller Anfragen:

  ```
  db2 connect reset
  ```

Laden der Daten

- Erstelle entsprechend der gegebenen Relationen die nötigen Tabellen (per CREATE TABLE) in der Datenbank db2db1:
  - movie(<u>mid</u>, title, year)
  - actor(<u>id</u>, name, movie_id, role, order)
  - actress(<u>id</u>, name, movie_id, role, order)
  - producer(<u>id</u>, name, movie_id, role)
  - genre(<u>id</u>, movie_id, genre)

  Die Datentypen sind wie folgt zu wählen: Attribute mit dem Namen `id`, `year` oder `order` sind vom Typ INTEGER, alle anderen Attribute vom Typ VARCHAR(127). Definiere die Primärschlüssel, aber keine Fremdschlüssel.
- Lade die Daten (per IMPORT oder LOAD) aus den **.csv*-Dateien in die entsprechenden Tabellen. Die *.csv-Dateien liegen in der VM unter /home/db2inst1/Documents
- Hinweis für die Arbeit auf den Poolrechnern: Kopiere die notwendigen Statements zum Anlegen der Relationen und zum Laden der Daten in eine Datei (Skript), so dass du die Datenbank schnell neu erstellen kannst. Die Datenbank wird auch für die JDBC-Übung benötigt!

Die DB2-Dokumentation findest du unter:
http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp

Aufgabe 1: Deutsch → SQL

Nenne für jede der folgenden natürlichsprachlichen Fragen eine geeignete SQL-Anfrage und führe sie auf den Daten der IMDB aus. Gib auf deiner Abgabe die Anfrage und deren Ergebnis an.

Hinweise:

- Benenne aggregierte Spalten so um, dass sinnvolle Spaltennamen ausgegeben werden.
- Einige der Tabellen enthalten eine Spalte mit dem Namen ID. Hierbei handelt es sich jedoch um Zeilennummern und nicht um Werte, die etwa einen Film oder eine Person eindeutig identifizieren müssen.
- Treffe *soweit nötig* geeignete Annahmen bezüglich des DB-Schemas (Fremdschlüssel).
- Anfragen auf Schauspielerinnen *und* Schauspielern sind explizit in der Frage formuliert. D. h. falls eine Frage nur *Schauspieler* erwähnt, soll auch nur die Tabelle `actor` angefragt werden.

a) Gib alle Producer aus, die keine zugehörigen Einträge in der Filmtabelle haben (nach Producernamen sortiert)! Jeder Producer soll dabei nur einmal ausgegeben werden. **3 P**

b) Wieviele Schauspielerinnen gibt es? **2 P**

c) Gib die Titel aller Filmpaare aus, in denen mindestens ein gemeinsamer Schauspieler mitspielt! Sortiere das Ergebnis nach dem Titel des zweiten Films. **3 P**

d) Gib alle Filme (Titel, Jahr) aus, deren Titel mit "Tatort" beginnen, und zusätzlich wie viele Schauspielerinnen an diesen Folgen jeweils beteiligt sind. **3 P**

e) Gib die Namen der Personen (Schauspieler und Produzenten) an, die an der Serie "Edge of Night, The" beteiligt waren, und zwar einmal nach Mengen- und ein weiteres Mal nach Multimengensemantik. (Hinweis: UNION benutzen) **4 P**

f) Formuliere *eine* Anfrage, die die Jahreszahl und die Anzahl der in diesem Jahr veröffentlichten Filme abfragt für

   - das höchste vorkommende Jahr und
   - das Jahr mit den meisten veröffentlichten Filmen

   5 P

Aufgabe 2: Deutsch → SQL (cont.)

Die Aufgabenstellung und die Hinweise aus Aufgabe 1 gelten weiter.

a) Gib alle Schauspieler an, die *auch* in Filmen des Genres „Action" spielen und deren Name mit „T" beginnt. **3 P**

b) Gib alle Schauspieler an, die *nur* in Filmen des Genres „Action" spielen und deren Name mit „T" beginnt. **4 P**

c) Gib die Namen aller Producer an, die 2001 Filme in beliebten Genres gedreht haben. Ein „beliebtes Genre" sei ein Genre in dem mindestens 200 Filme gedreht wurden. **5 P**

d) Erstelle eine Top-3 Liste der Filme mit den meisten Schauspielern und Schauspielerinnen (Name)! Sortiere entsprechend. Hinweis: Recherchiere hierzu die `FETCH FIRST` Klausel. **4 P**

e) Erstelle eine Top-3 Liste der Schauspieler und Schauspielerinnen (Name) mit den meisten Filmen! Sortiere entsprechend. **4 P**

Aufgabe 3: SQL → Deutsch

Gib natürlichsprachlich wieder, wonach die folgenden SQL-Queries suchen.

a)
```
WITH ProdAct AS
     (
     SELECT Prod.Name AS PName,
         Act.Name AS AName,
         COUNT(Act.MOVIE_ID) AS CountM
     FROM (
         SELECT NAME, MOVIE_ID FROM ACTOR
         UNION ALL
         SELECT NAME, MOVIE_ID FROM ACTRESS
     ) AS Act
     INNER JOIN MOVIE AS Mov ON Act.MOVIE_ID = Mov.MID
     INNER JOIN PRODUCER AS Prod ON Mov.MID = Prod.MOVIE_ID
     GROUP BY Prod.Name, Act.Name
     ORDER BY Prod.Name, Act.Name
)

SELECT ProdAct.AName, ProdAct.PName, CountM
FROM ProdAct,
     (
     SELECT AName, MAX(CountM) AS maxValue
     FROM ProdAct GROUP BY AName
     ) AS maxCount
WHERE ProdAct.AName = maxCount.AName AND
     ProdAct.CountM = maxCount.maxValue
```

5 P

b) Hinweis: Der Operator `TABLESAMPLE BERNOULLI (prozent)` ist IBM UDB-spezifisch und gibt zufällig ausgewählte Tupel der Ergebnismenge zurück.

```
SELECT *
FROM ACTOR AS a WHERE a.MOVIE_ID IN
     (       SELECT DISTINCT m.MID
             FROM MOVIE AS m
             TABLESAMPLE BERNOULLI (0.10)
             INNER JOIN GENRE AS g ON g.MOVIE_ID = m.MID
             WHERE g.GENRE NOT LIKE '%Adult%'
         UNION
             SELECT a.MOVIE_ID AS mid
             FROM ACTOR AS a
             TABLESAMPLE BERNOULLI (0.01)
             INNER JOIN PRODUCER AS p ON p.NAME = a.NAME
             WHERE a.MOVIE_ID = p.MOVIE_ID
     )
UNION
     SELECT *
     FROM ACTOR AS a
     TABLESAMPLE BERNOULLI(2)
```

Gib zusätzlich an ob Tupel doppelt ausgegeben werden und begründe deine Antwort! **6 P**

Aufgabe 4: Relationale Algebra → SQL

Formuliere die folgenden drei Anfragen der relationalen Algebra als SQL-Anfragen!

Verwendetes Schema:

- Stadt (StadtName, LandID, p1950, p2000, p2015)
  wobei `p1950`, `p2000` und `p2015` die Bevölkerungszahlen in diesen Jahren darstellen
- Land (LandID, Name, Kontinent, Hauptstadt, Bevoelkerung)
- Geographie (LandID, Landfläche, Wasserfläche, Küstenlänge, urbar)
  wobei `urbar` die urbare Fläche des Landes beschreibt

a) $\pi_{Name,Kontinent}(\sigma_{Bevoelkerung>200.000.000}(Land))$ **2 P**

b) $\pi_{Name}(\sigma_{(Bevoelkerung<2*p1950)\vee(Bevoelkerung<4*p2000)}(\sigma_{StadtName=Hauptstadt}(Stadt \bowtie Land)))$ **3 P**

c) $\pi_{Name}(Land \bowtie Geographie) -$
$\pi_{Name}(Land \bowtie (\sigma_{G1.urbar<G2.urbar}(\rho_{G1}(Geographie) \times \pi_{urbar}(\rho_{G2}(Geographie)))))$ **4 P**

Aufgabe 5: 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 Anfragen aus Aufgabe 1 und 2 kann diese View genutzt werden? Gib die entsprechenden Anfragen an. **4 P**