Aufgabenblatt 3
SQL

- Abgabetermin: **Sonntag, 09.06.13**
- 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.
- Abgabesytem unter
  `https://www.hpi.uni-potsdam.de/naumann/sites/abgabe/dbs1_2013`
  - ausschließlich PDF-Dateien
  - *eine Datei pro Aufgabe* namens Aufgabe-<aufgabenNr>.pdf
  - *jedes* Blatt beschriftet mit Namen

Vorbereitungen für Aufgabenblätter 3 und 4: IMDb in DB2 laden

Für die praktischen Aufgaben stellen wir die virtuelle Maschine "DB2 Express-C 9.7 32-bit" zur Verfügung. Die virtuelle Maschine (VM) ist folgendermaßen eingerichtet:

- Betriebssystem: *SUSE Linux Enterprise Server*
- Datenbank: *DB2 Express-C 9.7*
- Datenbank-Instanz: *db2inst1*
- Datenbank-Name: *db2db1*

In der virtuellen Maschine wurde ein Nutzer für das Betriebssystem und die Datenbank angelegt mit den folgenden Zugangsdaten:

- Nutzername: *db2inst1*
- Passwort: *ws2011*

Hinweise zum Umgang mit der VM

- Arbeit an einem Poolrechner:
  - Start der VM über:
    `Windows > All Programs > VMware > DB2 Express-C 9.7 32-bit`
- Arbeit mit dem eigenen Rechner:
  - Voraussetzung: VMware Player (Freeware)
  - VM kopieren von Laufwerk S: `\\fs3\bbs\DBS1-VM`
    - Alle Dateien werden benötigt
    - Die ausführbare Datei ist *DB2 Express-C 9.7 32-bit.vmx*
  - Starten der VM über:
    - Starte VMware Player > Open a Virtual Machine > *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, d.h. alle gemachten Änderungen gehen nach einem Neutstart der VM verloren, sodass sich die VM 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. Daher muss die VM immer heruntergefahren werden (Computer > Shutdown)
- Beim Start fragt die VM evtl. 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
  ```

- Der Zugriff auf das Internet ist nicht freigegeben. Daten können aber per Drag-and-Drop in das VM-Fenster mit dem Host-Rechner ausgetauscht werden.

Möglichkeiten zum Ausführen von Anfragen

- Kommandozeile
  - Shell öffnen: Computer > Gnome Terminal
  - Verbindung mit der Datenbank herstellen:
    `db2 connect to DB2DB1`
  - Queries ausführen (Beachte die Anführungsstriche!):
    `db2 "select * from actor"`
  - Ein (selbstgeschriebenes) SQL-Skript ausführen:
    `db2 -tvf <skriptname>`
  - Verbindung mit der Datenbank trennen:
    `db2 connect reset`
- IBM Data Studio
  - *Data Studio* starten:
    `Desktop > DB2 Data Studio`
  - Projekt anlegen:
    ```
    File > New > Data Development Project > Next > Finish >
    Edit now > Edit ... > Driver Properties >
    "User name" und "Password" eintragen >
    "Save password" Haken setzen > OK > Finish > Yes
    ```
  - Script anlegen:
    `Rechtsklick auf Projekt > New > SQL or SQuery Script > Finish`
  - SQL-Anfragen ausführen:
    - SQL in Script eintippen
    - SQL zum Ausführen markieren (falls nichts markiert ist, werden alle Anfragen ausgeführt)
    - Klick auf „Play“-Button

Laden der Daten

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

  Die Datentypen sind wie folgt zu wählen:

  - Die Attribute `id`, `year` und `order` sind vom Typ INTEGER
  - Alle anderen Attribute sind vom Typ VARCHAR(127)

  Definiere auch die Primärschlüssel für die einzelnen Tabellen.

- Daten importieren:
  Die Daten der Tabellen stellen wir als CSV-Dateien zur Verfügung.
  - Lade die Daten von der Übungsseite herunterladen und entpacke sie.
  - Kopiere die Dateien mittels Drag-and-Drop in die VM, z.B. auf den Desktop.
  - Lade die Daten (per IMPORT- oder LOAD-Statement) aus den CSV-Dateien in die entsprechenden Tabellen.
- **Wichtiger Hinweis**: 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 nächste Übung benötigt!**

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

Hinweise zum IMDb-Datensatz

- Die Tabellen `actor`, `actress`, `producer` und `genre` enthalten eine Spalte mit dem Namen "id". Hierbei handelt es sich um Zeilennummern und nicht um Werte, die etwa einen Film oder eine Person eindeutig identifizieren. Die Tabellen `actor` und `actress` speichern daher eher Rollen von Schauspielern in verschiedenen Filmen als Schauspieler selbst.
- Nimm an, dass Schauspieler, Schauspielerinnen und Produzenten eindeutig über ihre Namen identifiziert werden können.
- Nimm an, dass "movie_id" jeweils ein Fremdschlüssel auf "mid" ist.

Hinweise zur Bearbeitung der Aufgaben

- Benenne aggregierte Spalten so um, dass sinnvolle Spaltennamen ausgegeben werden.
- 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.
- Wenn nach dem Ergebnis einer SQL-Anfrage gefragt ist, dann gibt maximal 15 Tupel und die Anzahl aller Tupel an.

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.

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

b) 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**

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 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**

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

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

Aufgabe 2: Deutsch → SQL

Die Aufgabenstellung aus Aufgabe 1 gilt 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) 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 3: 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 4: SQL → Deutsch

Gib natürlichsprachlich wieder, wonach folgende SQL-Anfragen 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) **Zusatzaufgabe**

Hinweis: Der Operator `TABLESAMPLE BERNOULLI (`*n*`)` ist IBM UDB-spezifisch und wählt zufällig *n* % Tupel der Ergebnismenge aus.

```
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!

4 Zusatzpunkte