

Praktikum zur Vorlesung Datenbanken

Dr.-Ing. Katharina Emmerich
Dr. Andrey Krekhov
Mohamed Abokahf
Fabian Kvas

Wintersemester 2022/2023

Aufgaben Block 2

**Bitte beachten Sie, dass die Bearbeitung und Abgabe der Aufgaben
in Dreier- oder Viererteams erfolgen muss. Einzelabgaben und
Zweierteams sind nicht erlaubt.**

Datum	
Team (Account)	
Passwort	

Aktuelle Informationen, Ansprechpartner, Materialien und Uploads unter:
<https://moodle.uni-due.de/course/view.php?id=36483>

2 Block 2: SQL mit DB2 (15 Punkte)

Abgabe bis 23.12.2022, 23:59 Uhr

Für die folgenden Aufgaben steht eine Beschreibung der verwendeten Datenbank im Anhang A auf Seite 6 zur Verfügung.

Hinweis: Block 2 ist unabhängig von Block 1 und 3 (während Block 3 auf Block 1 aufbaut).

Aufgabe 1: Katalogisieren (1 Punkt)

In diesem Praktikum soll die DB2 aus der Kommandozeile heraus angesprochen werden. Es gibt auch grafische, Java-basierte Werkzeuge, auf die hier jedoch verzichtet werden soll.

Die Datenbank, mit der in diesem Block gearbeitet wird, befindet sich auf einem entfernten Rechner und muss daher in Ihren Rechner importiert werden. Machen Sie sich mit den unterschiedlichen Modi vertraut, die in der Dokumentation beschrieben werden. Katalogisieren Sie dann die Instanz **dbmaster** auf dem Rechner **bijou.is.inf.uni-due.de** mit Server-Port 50005.

Katalogisieren Sie die Datenbank **IMDB** lokal unter einem Alias. Lassen Sie sich dann sowohl das Node Directory als auch das Database Directory anzeigen.

Abgabe:

- Reichen Sie die angezeigten Einträge für Instanz (node) und Datenbank (database) als Textdatei (.txt) ein (kopieren Sie den Text aus der Kommandozeile). Denken Sie daran, zusätzlich in der Textdatei alle Gruppenmitglieder (Name, Vorname, Matrikelnummer) anzugeben.

Aufgabe 2: Sichten (2 Punkte)

- (a) Erstellen Sie eine Sicht **dbpXXX¹.lowRatedMovies**, welche alle Filme beinhalten soll, die schlechter als durchschnittlich bewertet wurden.

★ Die Sicht soll 4 Spalten umfassen: **MOVIEID**, **TITLE**, **YEAR** und **RANK**.

- (b) Erstellen Sie eine Sicht **dbpXXX².spielbergMovies**, die alle Filme enthält, die der Produzent „Spielberg, Steven“ produziert hat. Darüber hinaus sollen die Genres der Filme angezeigt werden. Hat ein Film mehr als ein Genre, so soll pro Genre eine Zeile angezeigt werden.

★ Die Sicht soll 3 Spalten umfassen: **MOVIEID**, **TITLE**, und **GENRE**.

Abgabe:

- Reichen Sie die verwendeten SQL-Befehle als Textdatei (.txt) ein. Geben Sie auch hier alle Gruppenmitglieder (Name, Vorname, Matrikelnummer) an.

¹Dabei steht „XXX“ für Ihre Gruppennummer, z.B. dbp001.

²Dabei steht „XXX“ für Ihre Gruppennummer, z.B. dbp001.

Aufgabe 3: Verwendung von Sichten (2 Punkte)

- (a) Verwenden Sie die Sicht `dbpXXX.lowRatedMovies` aus Aufgabe 2, um pro Jahr die Filme mit der jeweils niedrigsten Bewertung auszugeben. Es kann dabei mehrere Filme pro Jahr mit der niedrigsten Bewertung geben (oder keinen, falls alle Filme eines Jahres eine überdurchschnittliche Bewertung bekommen haben). Die Ausgabe soll absteigend nach Jahr sortiert sein (von neu zu alt).
- ★ Die Ausgabe soll genau 3 Spalten umfassen: `YEAR`, `TITLE`, und `RANK`.
- (b) Verwenden Sie die Sicht `dbpXXX.spielbergMovies` aus Aufgabe 2, um alle Filme von Steven Spielberg auszugeben, die 3 oder mehr Genres besitzen. Dabei soll zu jedem Film die Anzahl (Amount) an zugeordneten Genres angezeigt werden.
- ★ Die Ausgabe soll genau 2 Spalten umfassen: `TITLE` und `AMOUNT`.

Abgabe:

- Reichen Sie die SQL-Befehle (SELECT-Statements) als Textdatei (.txt) ein. Geben Sie auch hier alle Gruppenmitglieder (Name, Vorname, Matrikelnummer) an.

Aufgabe 4: SQL-Anfragen (7 Punkte)

Formulieren Sie die folgende Anfragen in SQL:

- (a) Welcher Film hat die meisten Schauspieler? Es könnten mehrere Filme sein. Geben Sie neben dem Titel des Films auch die Anzahl der Schauspieler aus.
- ★ Die Ausgabe soll 2 Spalten umfassen: `TITLE`, `AMOUNT`
- (b) Wie viele Schauspieler (Actors) hat ein Film mit dem Genre Mystery durchschnittlich?
- ★ Die Ausgabe soll 1 Spalte umfassen: `AMOUNT`
- (c) Ermitteln Sie für jeden Regisseur (Director) den durchschnittlichen Profit (gross) pro Film. Sortieren Sie die Ausgabe aufsteigend nach dem Profit.
- ★ Die Ausgabe soll 2 Spalten umfassen: `NAME` und `MEAN_GROSS`
- (d) Welche Regisseure haben mindestens einmal selbst in einem ihrer eigenen Filme mitgespielt? Geben Sie die Namen nach dem Alphabet sortiert aus und achten Sie dabei darauf, dass dieselben Namen nicht mehrfach vorkommen. Sie können bei dieser Aufgabe davon ausgehen, dass ein Regisseur (Director) und ein Schauspieler (Actor) dieselbe Person sind, wenn sie den gleichen Namen besitzen.
- ★ Die Ausgabe soll 1 Spalte umfassen: `NAME` (Regisseur)
- (e) Welche Autoren (Writer) haben **ausschliesslich** an Filmen mitgewirkt, die zwischen 1920 und 1929 veröffentlicht wurden? Geben Sie alle Namen aus.
- ★ Die Ausgabe soll 1 Spalte umfassen: `NAME` (Writer)

- (f) Ermitteln Sie für jeden Film die mitwirkenden Schauspieler und Regisseure (Rolle). Pro Film und Mitwirkendem soll eine Zeile ausgegeben werden. Die zugehörige Rolle (Actor/Director) der Mitwirkenden soll ebenfalls ausgegeben werden. Die Ausgabe soll absteigend nach dem Filmtitel sortiert werden.

★ Die Ausgabe soll 4 Spalten umfassen: **TITLE** (Film), **YEAR** (Film), **NAME** (Mitwirkender) und **ROLE** (Actor/Director)

- (g) Berechnen Sie für jedes Genre den relativen Marktanteil in Prozent gemessen an der Anzahl an Filmen, die das jeweilige Genre besitzen. Sind beispielsweise von 80 Filmen 10 davon Horrorfilme, so beträgt der Marktanteil für Horrorfilme 12.5%. Filme, die kein Genre haben, sollen grundsätzlich nicht berücksichtigt werden.

★ Die Ausgabe soll 3 Spalten: **GENRE**, **MOVIES_PER_GENRE**, **MARKET_SHARE**.

Abgabe:

- Reichen Sie die SQL-Befehle als Textdatei (.txt) ein. Geben Sie auch hier alle Gruppenmitglieder (Name, Vorname, Matrikelnummer) an. Achten Sie darauf, dass klar zu erkennen ist, welcher Befehl zu welcher Teilaufgabe gehört.

Aufgabe 5: Rekursive Anfrage (1 Punkt)

Das Spiel „Six Degrees of Kevin Bacon“ geht so (Beschreibung von <http://www.cs.virginia.edu/oracle>):

The object of the game is to start with any actor or actress who has been in a movie and connect them to Kevin Bacon in the smallest number of links possible. Two people are linked if they've been in a movie together. We do not consider links through television shows, made-for-tv movies, writers, producers, directors, etc. For example, you might wonder how Alfred Hitchcock can be connected to Kevin Bacon. One answer is that:

Alfred Hitchcock was in Show Business at War (1943) with Orson Welles, and Orson Welles was in A Safe Place (1971) with Jack Nicholson, and Jack Nicholson was in A Few Good Men (1992) with Kevin Bacon!

*Then we can count how many links were necessary and assign the actor or actress a **Bacon number**.*

Finden Sie alle Schauspieler und Schauspielerinnen mit einer maximalen Bacon-Nummer von zwei. Als Gedankenübung schreiben Sie ein rekursives SQL-Statement, dass die Schauspieler und Schauspielerinnen für beliebige Bacon-Nummern findet.

Tipp: In SELECT-Statements kann man sich das Ergebnis von Berechnungen zurückgeben lassen. Beispiel: `SELECT 1+3 FROM title;`

Abgabe:

- Reichen Sie die SQL-Befehle als Textdatei (.txt) ein. Geben Sie auch hier alle Gruppenmitglieder (Name, Vorname, Matrikelnummer) an.

Aufgabe 6: Trigger (2 Punkte)

Erstellen Sie zunächst die Tabelle `dbpXXX.frequentRatedMovies` (ersetzen Sie dabei XXX wieder durch Ihre Team-Account-Nummer):

```
CREATE TABLE dbpXXX.frequentRatedMovies (  
  movieID INTEGER NOT NULL,  
  title VARCHAR(400),  
  year VARCHAR(100),  
  PRIMARY KEY (movieID)  
);
```

Schreiben Sie nun zwei Trigger mit dem Namen `dbpXXX.trigger1` und `dbpXXX.trigger2`, die Folgendes leisten:

- `trigger1`: Sobald ein Film seine 100. Bewertung erhält, soll er in die Tabelle `dbpXXX.frequentRatedMovies` mit den angegebenen Attributen aufgenommen werden.
- `trigger2`: Sobald ein Rating zu einem Film gelöscht wird und dieser Film dadurch weniger als 100 Bewertungen hat, soll der Film aus der Tabelle `dbpXXX.frequentRatedMovies` entfernt werden.

Abgabe:

- Reichen Sie die SQL-Befehle als Textdatei (.txt) ein. Geben Sie auch hier alle Gruppenmitglieder (Name, Vorname, Matrikelnummer) an.

Anhang

A Beschreibung der IMDB-Datenbank

Für die SQL-Aufgaben ist eine Beispieldatenbank mit Filmdaten vorgegeben. Die Datenbank IMDB solltet Ihr in der ersten Woche unter einem lokalen Alias bereits katalogisiert haben.

Der relevante Ausschnitt des Schemas der Datenbank ist im Anschluss beschrieben. Ihr könnt den Aufbau der Tabellen auch direkt von der Datenbank erfragen. Die Daten der Tabellen geben teilweise kein vollständiges Bild der Welt wieder, sollen aber für unsere Zwecke genügen.

A.1 UML-Diagramm der IMDB-Datenbank

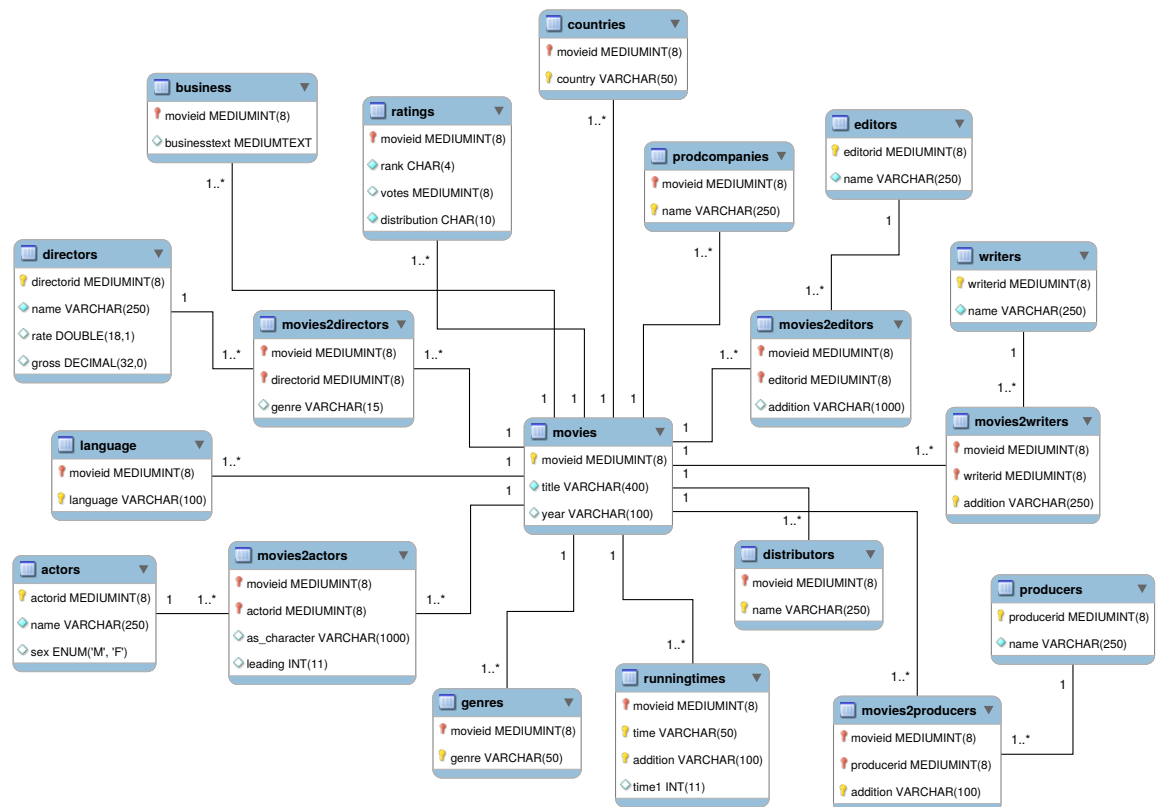


Abbildung 1: UML-Diagramm der IMDB-Datenbank