

**Praktikum Datenbanken / DB2**  
**Woche 5: SQL als Datenmanipulationssprache**

Musterlösung

- (b) Legt eine neue Tabelle für Schauspieler an, die Attribute Name und Geschlecht besitzt, oder erweitert eure existierende Tabelle entsprechend. Das Geschlecht kann entweder 'männlich' oder 'weiblich' sein. Die Dateien *actors-1950+* und *actresses-1950+* enthalten Daten zu Schauspielern, respektive Schauspielerinnen. Fügt diese Datensätze in die neue (oder geänderte Tabelle) ein, setzt dabei das Attribut Geschlecht auf den korrekten Wert.

```
CONNECT TO imdb23;

CREATE TABLE schauspieler_5b (
    name          VARCHAR(75) NOT NULL PRIMARY KEY,
    geschlecht    CHARACTER(10)
);

LOAD FROM "/usr/projects/db2doc/data/actors-1950+" OF DEL
MODIFIED BY COLDEL|
METHOD P(1)
MESSAGES "/tmp/group23_5b.log"
INSERT INTO schauspieler_5b (name);

UPDATE schauspieler_5b
    SET geschlecht='männlich';

LOAD FROM "/usr/projects/db2doc/data/actresses-1950+" OF DEL
MODIFIED BY COLDEL|
METHOD P(1)
MESSAGES "/tmp/group23_5b.log"
INSERT INTO schauspieler_5b (name);

UPDATE schauspieler_5b
    SET geschlecht='weiblich'
    WHERE geschlecht IS NULL;
```

- (c) Legt eine neue Tabelle für die Daten der Datei *costume-designers* an.

Führt ein ALTER TABLE-Statement aus, um ein neues Attribut **Aufgabe** einzuführen. Benutzt UPDATE, um bei jedem Kostümdesigner die Aufgabe 'Designer' einzutragen.

Wenn zwei Personen den gleichen Namen haben, wurde dieser in der Datenbasis durch Nachstellung einer römischen Zahl in Klammern eindeutig gemacht. Schreibt ein UPDATE-Statement, das die römischen Zahlen durch arabische Zahlen ersetzt:

I = 1, II = 2, III = 3, IV = 4, V = 5, VI = 6, VII = 7, VIII = 8, IX = 9,  
X = 10, XI = 11, XII = 12, XIII = 13, XIV = 14, XV = 15

Löscht alle Tupel aus der Tabelle, die sich auf Filme beziehen, die nach dem aktuellen Datum fertiggestellt werden. Hinweis: Benutzt für den Vergleich einen Registerwert.

```
CREATE TABLE designer (  
    name          VARCHAR(50)    NOT NULL,  
    produktion    VARCHAR(75)    NOT NULL,  
    jahr          INTEGER        NOT NULL,  
  
    PRIMARY KEY (name,produktion,jahr)  
);  
  
LOAD FROM "/usr/projects/db2doc/data/costume-designers" OF DEL  
MODIFIED BY COLDEL|  
METHOD P(1,2,3)  
MESSAGES "/tmp/group23_5c.log"  
INSERT INTO designer (name,produktion,jahr);  
  
ALTER TABLE designer  
ADD aufgabe CHARACTER(10);  
  
UPDATE designer  
SET aufgabe='Designer';  
  
UPDATE designer  
SET name=CASE  
    WHEN name LIKE '%(I)' THEN replace(name,'(I)','(1)')  
    WHEN name LIKE '%(II)' THEN replace(name,'(II)','(2)')  
    WHEN name LIKE '%(III)' THEN replace(name,'(III)','(3)')  
    WHEN name LIKE '%(IV)' THEN replace(name,'(IV)','(4)')  
    WHEN name LIKE '%(V)' THEN replace(name,'(V)','(5)')  
    WHEN name LIKE '%(VI)' THEN replace(name,'(VI)','(6)')  
    WHEN name LIKE '%(VII)' THEN replace(name,'(VII)','(7)')  
    WHEN name LIKE '%(VIII)' THEN replace(name,'(VIII)','(8)')  
    WHEN name LIKE '%(IX)' THEN replace(name,'(IX)','(9)')  
    WHEN name LIKE '%(X)' THEN replace(name,'(X)','(10)')  
    WHEN name LIKE '%(XI)' THEN replace(name,'(XI)','(11)')  
    WHEN name LIKE '%(XII)' THEN replace(name,'(XII)','(12)')  
    WHEN name LIKE '%(XIII)' THEN replace(name,'(XIII)','(13)')  
    WHEN name LIKE '%(XIV)' THEN replace(name,'(XIV)','(14)')  
    WHEN name LIKE '%(XV)' THEN replace(name,'(XV)','(15)')  
ELSE  
    name  
END;  
  
DELETE FROM designer  
WHERE jahr > year(CURRENT DATE);  
  
CONNECT RESET;  
TERMINATE;
```