| D | ES3UE | | | WS 2021 | Übung 2 |
|-------------|-------------------|--------|-----------------|------------------------|--------------------|
| | | | | Abgabetermin per e-Lea | arning: 20.10.2021 |
| | DES3UEG1: Niklas | Name _ | Angelos Angelis | Aufw | 4 and in h |
| \boxtimes | DES3UEG2: Niklas | | | | |
| | DES3UEG3: Traxler | Punkte | | Kurzzeichen Tutor | |

Hinweise und Richtlinien:

Die Übungsausarbeitungen müssen den im eLearning angegebenen Formatierungsrichtlinien entsprechen. Nichtbeachtung dieser Formatierungsrichtlinien führt zu Punkteabzug.

Ziel dieser Übung ist es, die Basiselemente der Anfragesprache SQL mit einem neuen Datenbank-Schema zu wiederholen. Dabei wird das Schema der "Sakila-Datenbank" für die folgenden Übungen eingeführt.

1. Grundlagen (6 Punkte)

- 1. Erstellen Sie eine Liste aller Schauspieler. Verketten Sie Vor- und Nachname (getrennt durch ein Leerzeichen) und nennen Sie die neue Spalte "Name". Sortieren Sie das Ergebnis nach dem Nachnamen. (0,5 Punkte)
- 2. Geben Sie eine Liste mit Titel und Länge all jener Filme aus, die länger als 180 Minuten dauern. (0,5 *Punkte*)
- 3. Geben Sie eine Liste mit Filmtitel aus, deren Namen an vierter Stelle ein 'A' enthält, geben Sie die Titel so aus, dass jeweils der erste Buchstabe eines Wortes mit einem Großbuchstaben beginnt (zB Atlantis Cause). (0,5 Punkte)
- 4. Geben Sie den Titel jener Filme aus, bei denen eine Originalsprache eingetragen ist. (0,5 Punkte)
- 5. Geben Sie die Anzahl der verliehenen Filme zwischen 1.1.2015 und 31.12.2015 aus (Start des Verleihvorgangs, rental_date). (1 Punkt)
- 6. Geben Sie alle Inventar-Ids aus, die noch nie verliehen wurden. (1 Punkt)
- 7. Erstellen Sie eine Liste mit Kunden (Vor- und Nachname), die in Newcastle, Linz und London wohnen. (1 Punkt)
- 8. Geben Sie für den Kunden mit der ID 240 alle Verleihvorgänge mit dem Start des Verleihvorgangs und dem bezahlten Betrag aus. Geben Sie das Datum im Format 'Mittwoch, 14. Okt. 2020' aus, verwenden Sie dazu die Funktion to_char und recherchieren Sie bei Bedarf in der Oracle-Dokumentation. (1 Punkt)

2. Gruppierungen und Unterabfragen

(12 Punkte)

- 1. Geben Sie die Titel aller Filme aus, die länger dauern als der Film mit der ID 50 und deren Ersetzungskosten größer sind als der Film mit der ID 101. (1,5 Punkte)
- 2. Geben Sie die Titel aller Filme aus, die kürzer als 60 Minuten dauern und in den gleichen Kategorien spielen als die Filme mit den IDs 10, 20 oder 30. (1,5 Punkte)
- 3. Erstellen Sie eine Liste aus Schauspieler (Vor- und Nachname) und Anzahl der Filme, in denen sie mitspielen. Die Liste soll nur jene Schauspieler enthalten, die in mehr als 35 Filmen mitspielen. Zählen Sie nur jene Filme, die mindestens 60 Minuten dauern. (1 Punkt)

- 4. Erstellen Sie eine Liste mit den Titeln und der Länge jener Filme, die länger als der Durchschnitt sind. (1 Punkt)
- 5. Ermitteln Sie die Namen jener Filmkategorien zu denen weniger als 60 Filme gehören. (*1 Punkt*)
- 6. Ermitteln Sie alle Filme, die die längsten in ihrem Erscheinungsjahr sind und geben Sie Titel, Dauer (*length*) sowie Erscheinungsjahr aus. (1 Punkt)
- 7. Geben Sie den durchschnittlich bezahlten Preis (Tabelle *payment*) pro Filmkategorie aus (sollte ein Film zu mehreren Kategorien gehören, zählt er zu allen). Runden Sie auf zwei Nachkommastellen. (1 Punkt)
- 8. Geben Sie die neun zuletzt verliehenen Filme und das Verleihdatum im Format 'YYYY-MM-DD' aus. (1,5 Punkte)
- 9. Geben Sie alle Schauspieler (mind. Vor-/Nachname) aus, die in mehr als 15 **verschiedenen** Film-Kategorien spielen. Achten Sie dabei darauf, dass manche Schauspieler unter Umständen den gleichen Namen tragen. (*1 Punkt*)
- 10. Welcher Film wurde pro Film-Kategorie als erstes ausgeliehen (*rental_date*)? Geben Sie Film-Titel, Kategorie-Name und Verleihdatum aus. Sortieren Sie nach dem Kategorie-Namen. (1,5 *Punkte*)

3. Insert, Update und Delete

(6 Punkte)

- 1. Erstellen Sie eine neue Tabelle "new_film", diese soll den gleichen Aufbau wie die Tabelle "film" haben, jedoch nur die neuesten Filme (jene Filme, die das höchste Erscheinungsjahr in der Datenbank aufweisen) enthalten. (1 Punkt)
- 2. Fügen Sie den Film "Jason Bourne" mit ID = 1001 in Englisch (*language_id* = 1) mit 5 Tagen Verleihdauer (zum Preis von 1,79) mit Ersetzungskosten von 16,99 in die Tabelle ein. (*I Punkt*)
- 3. Erhöhen Sie den Leihpreis der Filme in der Tabelle "new_film" um 15%, wenn der Leihpreis kleiner als 2 ist. (0,5 Punkte)
- 4. Erstellen Sie eine View, die alle Filme der Tabelle "new_film" enthält, deren Leihpreis maximal 2 beträgt, vergeben Sie die Check-Option. Die View soll nur den Filmtitel, die Beschreibung, den Leihpreis und die Länge enthalten. (1 Punkt)
- 5. Welche Auswirkungen haben COMMIT und ROLLBACK an dieser Stelle (nachdem Sie die View erstellt haben). (0,5 Punkte)
- 6. Können Sie die Datensätze Ihrer neuen View verändern? Wenn ja, führen Sie die Erhöhung der Filme (10%) ein weiteres Mal durch, diesmal auf Ihre View. Wenn nein, warum nicht? (0,5 Punkte)
- 7. Löschen Sie alle Einträge aus der Tabelle new_film, deren Leihpreis über 1.79 liegt. (0,5 *Punkte*)
- 8. Können Sie den Leihpreis der Filme in der View nun erhöhen? Erklären Sie dieses Verhalten. (0,5 Punkte)
- 9. Löschen Sie die Tabelle "new_film" und Ihre erstellte View wieder. (0,5 Punkte)

Aufgabe 1)

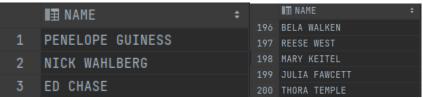
--1. Erstellen Sie eine Liste aller Schauspieler. Verketten Sie Vor- und Nachname (getrennt durch

--ein Leerzeichen) und nennen Sie die neue Spalte "Name". Sortieren Sie das Ergebnis nach dem

-- Nachnamen. (0,5 Punkte)

 ${\tt SELECT \ (FIRST_NAME \ | \ | \ ' \ | \ | LAST_NAME) \ AS \ Name}$

FROM ACTOR;



--2. Geben Sie eine Liste mit Titel und Länge all jener Filme aus, die länger als 180 Minuten dauern. (0,5 Punkte)

SELECT TITLE, LENGTH

FROM FILM

WHERE LENGTH > 180;

| | II TITLE | ‡ | ■■ LENGTH ÷ | 34 | 調 TITLE SMOOCHY CONTROL | ‡ | II LENGTH ÷ 184 |
|---|------------------|----------|-------------|----|-------------------------------|----------|--------------------|
| 1 | BAKED CLEOPATRA | | 182 | | SOLDIERS EVOLUTION | | 185 |
| 2 | ANALYZE HOOSIERS | | 181 | 36 | SONS INTERVIEW | | 184 |
| 3 | CATCH AMISTAD | | 183 | | SORORITY QUEEN STAR OPERATION | | 184 181 |
| 4 | CHICAGO NORTH | | 185 | 39 | SWEET BROTHERHOOD | | 185 |

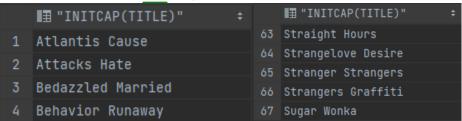
--3. Geben Sie eine Liste mit Filmtitel aus, deren Namen an vierter Stelle ein 'A' enthält, geben Sie

--die Titel so aus, dass jeweils der erste Buchstabe eines Wortes mit einem Großbuchstaben beginnt (zB Atlantis Cause). (0,5 Punkte)

SELECT INITCAP(TITLE)

FROM FILM

WHERE TITLE LIKE ' A%';



--4. Geben Sie den Titel jener Filme aus, bei denen eine Originalsprache eingetragen ist. (0,5

--Punkte)

SELECT TITLE

FROM FILM

WHERE ORIGINAL LANGUAGE ID IS NOT NULL;

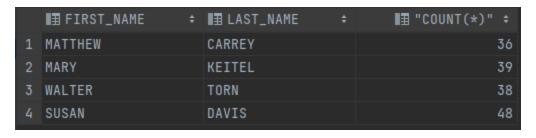
| | | _ | | |
|---|----------------------|-----|-------------------|----------|
| | II TITLE | | ₽ TITLE | ‡ |
| 1 | ARMY FLINTSTONES | 496 | MYSTIC TRUMAN | |
| 2 | ARSENIC INDEPENDENCE | 497 | NATIONAL STORY | |
| 3 | ARTIST COLDBLOODED | 498 | NECKLACE OUTBREAK | |
| 4 | ATLANTIS CAUSE | 499 | NEIGHBORS CHARADE | |
| 5 | ATTACKS HATE | 500 | NEMO CAMPUS | |

```
--5. Geben Sie die Anzahl der verliehenen Filme zwischen 1.1.2015 und
31.12.2015 aus (Start des
--Verleihvorgangs, rental date). (1 Punkt)
alter session set nls date format = 'dd.mm.yyyy';
SELECT COUNT(*)
FROM RENTAL
WHERE RENTAL DATE BETWEEN '01.01.2015' AND '31.12.2015';
         III "COUNT(*)" $
                      7050
--6. Geben Sie alle Inventar-Ids aus, die noch nie verliehen wurden. (1
SELECT I.INVENTORY ID
FROM INVENTORY I
    LEFT JOIN RENTAL R on I.INVENTORY ID = R.INVENTORY ID
WHERE RENTAL ID IS NULL;
          📭 INVENTORY_ID 🕏
--7. Erstellen Sie eine Liste mit Kunden (Vor- und Nachname), die in
Newcastle, Linz und London
--wohnen. (1 Punkt)
SELECT FIRST NAME, LAST NAME
FROM CUSTOMER C
    JOIN ADDRESS A on C.ADDRESS ID = A.ADDRESS ID
    JOIN CITY CI on A.CITY ID = CI.CITY ID
WHERE REGEXP LIKE(CI.CITY, 'Newcatsle|Linz|London');
    J∏ FIRST_NAME

⇒ JEE LAST_NAME
 1 JILL
                            HAWKINS
 2 MATTIE
                            HOFFMAN
 3 CECIL
                            VINES
--8. Geben Sie für den Kunden mit der ID 240 alle Verleihvorgänge mit dem
Start des Verleihvorgangs und dem bezahlten Betrag aus. Geben Sie das Datum
im Format 'Mittwoch, 14. Okt.
--2020' aus, verwenden Sie dazu die Funktion to char und recherchieren Sie
bei Bedarf in der
--Oracle-Dokumentation. (1 Punkt)
SELECT R.RENTAL ID, TO CHAR( R.RENTAL DATE, 'DAY, dd. Mon. YYYY' ), P.AMOUNT
FROM CUSTOMER C
    JOIN RENTAL R on C.CUSTOMER ID = R.CUSTOMER ID
    JOIN PAYMENT P on C.CUSTOMER ID = P.CUSTOMER ID
WHERE C.CUSTOMER ID = 240
     III RENTAL_ID : III TO_CHAR(R.RENTAL_DATE,'DAY,DD.MON.YYYY'
           643 MONTAG , 17. Feb. 2014
246 DIENSTAG , 31. Dez. 2013
          460 DIENSTAG , 27. Mai. 2014
2196 SAMSTAG , 21. Mrz. 2015
2264 FREITAG , 18. Apr. 2014
     NET RENTAL_ID : NET TO_CHAR(R.RENTAL_DATE, 'DAY, DD.MON.YYYY')
          2196 SAMSTAG , 21. Mrz. 2015
2264 FREITAG , 18. Apr. 2014
2872 SONNTAG , 18. Mai. 2014
```

Aufgabe 2)

--1. Geben Sie die Titel aller Filme aus, die länger dauern als der Film mit der ID 50 und deren Ersetzungskosten größer -- sind als der Film mit der ID 101. (1,5 Punkte) SELECT TITLE FROM FILM WHERE LENGTH > (SELECT LENGTH FROM FILM WHERE FILM ID = 50) AND REPLACEMENT COST > (SELECT REPLACEMENT COST FROM FILM WHERE FILM ID = 101); **III TITLE** 1 CONSPIRACY SPIRIT 2 GANGS PRIDE 3 WIFE TURN 4 SOLDIERS EVOLUTION 5 SWEET BROTHERHOOD --2. Geben Sie die Titel aller Filme aus, die kürzer als 60 Minuten dauern und in den gleichen Kategorien spielen als -- die Filme mit den IDs 10, 20 oder 30. (1,5 Punkte) SELECT F.TITLE FROM FILM F JOIN FILM CATEGORY FC on F.FILM ID = FC.FILM ID WHERE F.LENGTH < 60 AND CATEGORY ID IN (SELECT CATEGORY ID FROM FILM CATEGORY WHERE FILM ID IN (10,20,30)); 1 SENSE GREEK 2 CRANES RESERVOIR 3 DIVORCE SHINING 7 HANOVER GALAXY 8 HEAVENLY GUN 9 LEGEND JEDI 10 MINORITY KISS 11 SIMON NORTH 12 ACE GOLDFINGER 13 AIRPORT POLLOCK 14 COMMANDMENTS EXPRESS --3. Erstellen Sie eine Liste aus Schauspieler (Vor- und Nachname) und Anzahl der Filme, in denen --sie mitspielen. Die Liste soll nur jene Schauspieler enthalten, die in mehr als 35 Filmen mitspielen. --Zählen Sie nur jene Filme, die mindestens 60 Minuten dauern. (1 Punkt) SELECT A.FIRST NAME, A.LAST NAME, COUNT(*) FROM ACTOR A JOIN FILM ACTOR FA on A.ACTOR ID = FA.ACTOR ID JOIN FILM F on FA.FILM ID = F.FILM ID WHERE f.LENGTH > 60 GROUP BY A.FIRST NAME, A.LAST NAME **HAVING** COUNT (FA.ACTOR ID) >35;



--4. Erstellen Sie eine Liste mit den Titeln und der Länge jener Filme, die länger als der Durchschnitt sind. (1 Punkt)

SELECT TITLE, LENGTH

FROM FILM

WHERE LENGTH > (SELECT AVG(LENGTH)

FROM FILM);

| | JE TITLE | II≣ LENGTH ≎ | ## TITLE | ‡ | ■■ LENGTH ÷ |
|--|----------------------|--------------|------------------------|----------|-------------|
| | ARMY FLINTSTONES | 148 | SWEET BROTHERHOOD | | 185 |
| | ARSENIC INDEPENDENCE | 137 | TADPOLE PARK | | 155 |
| | ARTIST COLDBLOODED | 170 | TALENTED HOMICIDE | | 173 |
| | ATLANTIS CAUSE | 170 | TELEGRAPH VOYAGE | | 148 |
| | BABY HALL | | TELEMARK HEARTBREAKERS | | 152 |
| | | | | | |

 $--5.\ Ermitteln$ Sie die Namen jener Filmkategorien zu denen weniger als 60 Filme gehören. (1

--Punkt)

SELECT NAME

FROM CATEGORY

WHERE CATEGORY_ID IN (SELECT CATEGORY_ID

FROM FILM_CATEGORY
GROUP BY CATEGORY_ID
HAVING COUNT(*) < 60);</pre>



--6. Ermitteln Sie alle Filme, die die längsten in ihrem Erscheinungsjahr sind und geben Sie Titel,

--Dauer (length) sowie Erscheinungsjahr aus. (1 Punkt)

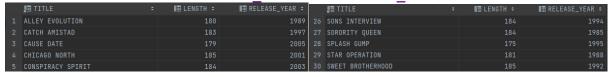
SELECT F.TITLE, F.LENGTH, F.RELEASE YEAR

FROM FILM F

WHERE F.LENGTH = (SELECT MAX(F2.LENGTH)

FROM FILM F2

WHERE F.RELEASE YEAR = F2.RELEASE YEAR);



--8. Geben Sie die neun zuletzt verliehenen Filme und das Verleihdatum im Format 'YYYY-MMDD' aus. (1,5 Punkte)

```
SELECT I.FILM_ID, TO_CHAR(R.RENTAL_DATE, 'YYYY-MMDD')
FROM INVENTORY I
JOIN RENTAL R on I.INVENTORY_ID = R.INVENTORY_ID
WHERE ROWNUM <= 9</pre>
```

ORDER BY RENTAL DATE DESC;

| | | 2200/ |
|---|-------------|---|
| | ■ FILM_ID ÷ | ■ TO_CHAR(R.RENTAL_DATE, 'YYYY-MMDD') ÷ |
| 1 | 333 | 2015-0628 |
| 2 | 535 | 2015-0516 |
| 3 | 870 | 2015-0111 |
| 4 | 613 | 2014-1229 |
| 5 | 80 | 2014-1011 |
| 6 | 450 | 2014-0826 |
| 7 | 373 | 2014-0526 |
| 8 | 565 | 2014-0125 |
| 9 | 510 | 2014-0124 |

```
--9. Geben Sie alle Schauspieler (mind. Vor-/Nachname) aus, die in mehr als
15 verschiedenen
--Film-Kategorien spielen. Achten Sie dabei darauf, dass manche Schauspieler
unter Umständen
--den gleichen Namen tragen. (1 Punkt)
SELECT A.ACTOR ID, A.FIRST NAME, A.LAST NAME
FROM ACTOR A
         JOIN FILM ACTOR FA on A.ACTOR ID = FA.ACTOR ID
         JOIN FILM CATEGORY FC on FA.FILM ID = FC.FILM ID
GROUP BY A.ACTOR ID, A.FIRST NAME, A.LAST NAME
HAVING COUNT (FC.CATEGORY ID) > 15;
    IR ACTOR_ID : IR FIRST_NAME : IR LAST_NAME :
          30 SANDRA
--10. Welcher Film wurde pro Film-Kategorie als erstes ausgeliehen
(rental_date)? Geben Sie FilmTitel, Kategorie-Name
-- und Verleihdatum aus. Sortieren Sie nach dem Kategorie-Namen. (1,5Punkte)
SELECT F.TITLE, C.NAME, R.RENTAL DATE
FROM FILM F
    JOIN FILM CATEGORY FC on F.FILM ID = FC.FILM ID
    JOIN CATEGORY C on C.CATEGORY ID = FC.CATEGORY ID
    JOIN INVENTORY I on FC.FILM ID = I.FILM ID
    JOIN RENTAL R on I.INVENTORY ID = R.INVENTORY ID
WHERE R.RENTAL DATE = (SELECT MIN (RENTAL DATE)
                       FROM RENTAL R2
                            JOIN INVENTORY I2 on R2. INVENTORY ID =
12.INVENTORY ID
                            JOIN FILM CATEGORY FC2 on I2.FILM ID =
FC2.FILM ID
                       WHERE FC.CATEGORY ID = FC2.CATEGORY ID)
ORDER BY C.NAME
```

| | III TITLE ÷ | III NAME ≎ | ■ RENTAL_DATE ÷ | III TITLE | ÷ IIII NAME | III RENTAL_DATE ÷ |
|---|--------------------|------------|-----------------|---------------|-------------|-------------------|
| 1 | MINDS TRUMAN | Action | 2013-12-05 | MASKED BUBBLE | | 2013-12-05 |
| _ | HINDS INCHAN | ACCION | 2013-12-03 | | | 2013-12-05 |
| 2 | PRIMARY GLASS | Action | 2013-12-05 | | | 2013-12-05 |
| _ | | | | | | 2013-12-05 |
| 3 | DARN FORRESTER | Action | 2013-12-05 | | | 2013-12-05 |
| 4 | POTTER CONNECTICUT | Animation | 2013-12-05 | | | 2013-12-10 |
| - | | | | | | 2013-12-10 |
| 5 | FARGO GANDHI | Children | 2013-12-08 | | | 2013-12-10 |

Aufgabe 3)

```
--1. Erstellen Sie eine neue Tabelle "new film", diese soll den gleichen
Aufbau wie die Tabelle
--"film" haben, jedoch nur die neuesten Filme (jene Filme, die das höchste
Erscheinungsjahr in
--der Datenbank aufweisen) enthalten. (1 Punkt)
CREATE TABLE new film (
  new film id INTEGER NOT NULL,
  title VARCHAR2 (255) NOT NULL,
  description VARCHAR2 (255),
  release year INTEGER,
  language id INTEGER NOT NULL,
  original language id INTEGER,
  rental duration INTEGER NOT NULL,
  rental rate DECIMAL(4,2) NOT NULL,
  length INTEGER DEFAULT NULL,
  replacement cost DECIMAL(5,2) NOT NULL,
  rating VARCHAR2(20),
  special features VARCHAR2 (255),
  last update TIMESTAMP NOT NULL,
  CONSTRAINT new film pk PRIMARY KEY (new film id),
  CONSTRAINT fk new film language FOREIGN KEY (language id) REFERENCES
language (language id) DEFERRABLE,
 CONSTRAINT fk new film language original FOREIGN KEY (original language id)
REFERENCES language (language id) DEFERRABLE
);
INSERT INTO new film
SELECT *
FROM FILM
WHERE RELEASE YEAR = (SELECT MAX(RELEASE YEAR)
                      FROM FILM);
--2. Fügen Sie den Film "Jason Bourne" mit ID = 1001 in Englisch (language id
= 1) mit 5 Tagen
--Verleihdauer (zum Preis von 1,79) mit Ersetzungskosten von 16,99 in die
Tabelle ein. (1 Punkt)
INSERT INTO FILM (FILM ID, TITLE, LANGUAGE ID, ORIGINAL LANGUAGE ID,
RENTAL DURATION, RENTAL RATE, REPLACEMENT COST, LAST UPDATE)
VALUES (1001, 'Jason Bourne', 1, 1, 5, 1.79, 16.99, CURRENT TIMESTAMP);
--3. Erhöhen Sie den Leihpreis der Filme in der Tabelle "new film" um 15%,
wenn der Leihpreis
--kleiner als 2 ist. (0,5 Punkte)
UPDATE new film
SET rental rate = rental rate*1.15
WHERE rental_rate < 2;</pre>
--4. Erstellen Sie eine View, die alle Filme der Tabelle "new film" enthält,
deren Leihpreis maximal 2 beträgt,
-- vergeben Sie die Check-Option. Die View soll nur den Filmtitel, die
Beschreibung, den Leihpreis und die Länge enthalten. (1 Punkt)
CREATE VIEW FILM LESSTHAN 2 AS
SELECT
    TITLE, DESCRIPTION, RENTAL RATE, LENGTH
```

FROM

NEW FILM

WHERE

RENTAL_RATE < 2 WITH CHECK OPTION;</pre>

--5. Welche Auswirkungen haben COMMIT und ROLLBACK an dieser Stelle (nachdem Sie die View erstellt haben). (0,5 Punkte)

Ein COMMIT speichert die Datenänderungen in die Datenbank, sodass der Zustand der vorherigen Daten überschrieben wird. Ab diesem Zeitpunkt können alle Benutzer die Ergebnisse sehen.

Ein ROLLBACK hingegen verwirft alle noch nicht gespeicherten Änderungen, wodurch der vorherige Zustand mit Hilfe vom UNDO-Tablespace hergestellt wird. Zudem werden die betroffenen gesperrten Zeilen wieder freigegeben.

--6. Können Sie die Datensätze Ihrer neuen View verändern? Wenn ja, führen Sie die Erhöhung der

--Filme (10%) ein weiteres Mal durch, diesmal auf Ihre View. Wenn nein, warum nicht? (0,5 $\,$

--Punkte)

Es geht nicht weil die Check Option dies verhindert. Nämlich würde dann die erhöhung die rental_rates mancher filme erhöhen, diese sollen dann aber durch der with check Option in der View nicht angezeigt

--7. Löschen Sie alle Einträge aus der Tabelle new_film, deren Leihpreis über 1.79 liegt. (0,5 --Punkte)

DELETE FROM NEW FILM WHERE RENTAL RATE > 1.79;

- --8. Können Sie den Leihpreis der Filme in der View nun erhöhen? Erklären Sie dieses Verhalten.
- -- (0,5 Punkte)

--9. Löschen Sie die Tabelle "new_film" und Ihre erstellte View wieder. (0,5 Punkte)

drop view FILM_LESSTHAN_2;
drop table NEW FILM