

# Einführung in Datenbanken

## — Übungsblatt 2 (Lexikalische Syntax von SQL) —

Ihre Lösungen laden Sie bitte in die Übungsplattform in StudIP hoch ([StudIP-Eintrag der Vorlesung], Reiter „LTI-Tool“, dann auf „Anwendung starten“).

**Einsendeschluss ist Montag, der 30.10.2023, 18<sup>00</sup>.**

Hausaufgaben müssen einzeln bearbeitet werden. „Zu ähnliche Lösungen“ führen automatisch zu 0 Punkten für alle Beteiligten. Das gilt auch dann, wenn Sie nicht direkt abgeschrieben haben, sondern nur zufällig die gleiche Quelle benutzt haben (ohne diese zu nennen).

Wir haben für SQL-Anfragen vereinbart, dass bei korrekter Nennung der Quelle (als SQL-Kommentar im vorgeschriebenen Format) selbst identische Abgaben akzeptiert werden (ohne Punktabzug).

### Aufgabe 1 (6 Punkte)

Beantworten Sie folgende Fragen zur lexikalischen Syntax von SQL (alle zusammen in einer `.txt`-Datei):

- a) Ist `12.` eine syntaktisch korrekte Zahlkonstante?
  - Falls ja: Beschreiben Sie (in wenigen Worten) welchen Weg man durch die Syntaxgraphen auf die Folien 3-19 bis 3-21 nehmen muss.
  - Falls nein: Erklären Sie, was der Fehler ist, und wie man ihn korrigieren kann.
- b) Ist `1,25` eine syntaktisch korrekte Zahlkonstante? Wie bei a) beschreiben Sie entweder den Weg durch die Syntaxgraphen oder den Fehler und eine mögliche Korrektur.
- c) Ist `09` eine syntaktisch korrekte Zahlkonstante? (Antwort wie bei a) und b).)
- d) Wie viele Zeichen hat die Zeichenkette `''''`? Gesucht ist also die Länge der Zeichenkette. Geben Sie den Zahlwert an und eine kurze Begründung.
- e) Wie viele Zeichen hat die Zeichenkette `'\n'`? (Wie bei d) bitte mit Begründung.)
- f) Ist `"abc"` eine syntaktisch korrekte Zeichenkette? Falls nicht: Was ist es? Wie könnte man den Fehler korrigieren?

## Aufgabe 2 (5 Punkte)

Diese Aufgabe bezieht sich auf die Tabelle `film` im Datenbank-Schema `sakila_public` im Adminer:

```
https://dbs.informatik.uni-halle.de/edb?pgsql=db&username=student_gast&
db=postgres&ns=sakila_public
```

Das Passwort für den Adminer (bzw. eigentlich das zugrundeliegende DBMS PostgreSQL) finden Sie im Reiter „Informationen“ im StudIP-Eintrag der Vorlesung.

Die „Sakila“-Datenbank ist eine bekannte Beispieldatenbank von MySQL für einen Videoverleih. Wie bei einer Bibliothek kann man sich DVDs ausleihen (gegen eine Gebühr). Während man heute bei Streaming-Diensten Filme über das Internet anschaut, musste man früher im Laden ein Speichermedium ausleihen, und dann zu Hause in seinen DVD-Spieler stecken.

Die Sakila-Datenbank enthält relativ viele Tabellen, die auch teils recht groß sind. Leider sind die Daten nicht real, selbst die Film-Titel sind zusammengewürfelt aus Worten, die allerdings teilweise in bekannten Film-Titeln vorkommen.

Die Tabelle `film` hat folgende Spalten:

- `film_id`: Eindeutige Nummer des Films.
- `title`: Titel des Films.
- `description`: Kurze Beschreibung des Films.
- `release_year`: Jahr, in dem der Film erschienen ist (leider bei allen Filmen 2006).
- `language_id`: Bei allen Filmen 1: Englisch.
- `original_language_id`: Originalsprache des Films, leider bei allen Filmen leer.
- `rental_duration`: Anzahl Tage, nach denen man den Film zurückbringen muss (zwischen 3 und 7 — bei sehr nachgefragten Neuerscheinungen sind es weniger Tage).
- `rental_rate`: Preis, den man für das Ausleihen des Films bezahlen muss.
- `length`: Länge des Films in Minuten (von 46 bis 185).
- `replacement_cost`: Kaufpreis der DVD (der zu zahlen wäre, wenn die DVD verloren geht).
- `rating`: Angabe, ob der Film für Kinder und Jugendliche geeignet wäre, entsprechend FSK, aber hier nach dem amerikanischen MPA/MPAA-System („Motion Picture Association of America“): G, PG, PG-13, R, NC-17 (siehe Wikipedia).
- `last_update`: Zeitstempel der letzten Änderung des Datensatzes, bei allen Zeilen 2006-02-15 05:03:42.
- `special_features`: Zusatzmaterial auf der DVD (Array von Zeichenketten).

Schreiben Sie eine SQL-Anfrage, die für alle Filme mit `film_id` bis 20 Folgendes ausgibt:

- die Film-Id (nennen Sie die Ausgabespalte bitte `Film-ID`),
- den Titel des Films (Spalte `title`),
- den Entleihpreis (Spalte `rental_rate` umbenannt in `Miete`)
- den Netto-Entleihpreis, wenn `rental_rate` 19% MwSt enthält (netto ist ohne MwSt, also das Geld, das der Unternehmer behalten darf, die Mehrwertsteuer muss er ja an das Finanzamt abführen), nennen Sie diese Spalte bitte `Netto` und runden Sie das Ergebnis auf zwei Nachkommastellen durch Aufruf von `ROUND(..., 2)`,
- den Kaufpreis (Spalte `replacement_cost` umbenannt in `Kauf`),
- wie viele Prozent des Kaufpreises (`replacement_cost`) die `rental_rate` ist (MwSt spielt hier keine Rolle, da beide Preise sie enthalten). Runden Sie das Ergebnis bitte mittels `ROUND(...)` auf eine ganze Zahl und nennen Sie die Spalte `Prozent`.

Bei den Namen der Ergebnisspalten achten Sie bitte auch auf die vorgegebene Groß-/Kleinschreibung.

Das erwartete Ergebnis ist:

Film-ID	title	Miete	Netto	Kauf	Prozent
1	ACADEMY DINOSAUR	0.99	0.83	20.99	5
2	ACE GOLDFINGER	4.99	4.19	12.99	38
3	ADAPTATION HOLES	2.99	2.51	18.99	16
4	AFFAIR PREJUDICE	2.99	2.51	26.99	11
5	AFRICAN EGG	2.99	2.51	22.99	13
6	AGENT TRUMAN	2.99	2.51	17.99	17
7	AIRPLANE SIERRA	4.99	4.19	28.99	17
8	AIRPORT POLLOCK	4.99	4.19	15.99	31
9	ALABAMA DEVIL	2.99	2.51	21.99	14
10	ALADDIN CALENDAR	4.99	4.19	24.99	20
11	ALAMO VIDEOTAPE	0.99	0.83	16.99	6
12	ALASKA PHANTOM	0.99	0.83	22.99	4
13	ALI FOREVER	4.99	4.19	21.99	23
14	ALICE FANTASIA	0.99	0.83	23.99	4
15	ALIEN CENTER	2.99	2.51	10.99	27
16	ALLEY EVOLUTION	2.99	2.51	23.99	12
17	ALONE TRIP	0.99	0.83	14.99	7
18	ALTER VICTORY	0.99	0.83	27.99	4
19	AMADEUS HOLY	0.99	0.83	20.99	5
20	AMELIE HELLFIGHTERS	4.99	4.19	23.99	21

**Beachten Sie bitte:**

- Anfragen, die in PostgreSQL nicht ausführbar sind (wegen Syntaxfehlern) werden automatisch mit 0 Punkten bewertet! Eventuelle Anmerkungen müssen als SQL-Kommentar geschrieben werden.
- Abweichungen vom erwarteten Ergebnis im Beispielszustand führen dagegen nicht automatisch zu 0 Punkten (aber schon zu Punktabzug). Sie müssen keine besonderen Tricks machen, um die obige Ausgabe zu erreichen. Teile der Anfrage, die mit der Aufgabenstellung gar nichts zu tun haben, führen eher zu stärkeren Punktabzügen.
- Die Aufgabenstellung gibt keine bestimmte Sortierung vor und der PostgreSQL-Optimierer hat die Freiheit, die Ergebniszeilen in beliebiger Reihenfolge zu liefern. Die Musterlösung lieferte die obige Reihenfolge. Bei Bedarf können Sie Ihrer Anfrage am Ende `ORDER BY 1` (d.h. sortiere nach der ersten Spalte) hinzufügen, um die Tabellen besser vergleichen zu können. Das ist aber keine Bedingung.
- Es kann auch maximal ein Punkt für Stilfehler abgezogen werden.
- Bei Unklarheiten und Problemen können Sie Fragen im Forum in StudIP stellen. Bitte posten Sie aber keine Anfragen, die der Lösung nahe kommen. Bei komischen Syntaxfehlern, die Sie sich nicht erklären können, können Sie probieren, dem Dozenten oder dem Übungsleiter eine persönliche EMail mit der Anfrage zu schicken.
- Schreiben Sie Ihren Namen (sowie Matrikelnummer etc.) bitte nicht in die Abgabe, da wir die Anfragen auch für ein Forschungsprojekt verwenden wollen. Das soll natürlich in anonymisierter Form geschehen, aber wir können nicht garantieren, dass wir identifizierbare Informationen auch im Text der Anfrage in allen Fällen finden und eliminieren.
- Auch in der Klausur wird Prozentrechnung vorausgesetzt!