

Übung Informationssysteme

Blatt 3

Aufgabe 1 (SQL, QBE). Gegeben sei die Film-Datenbank aus Aufgabe 1 von Blatt 2 mit folgendem Datenbankschema:

Film : { [ID, Titel, Jahr, FSK] }
Person : { [ID, Name, Vorname] }
Kino : { [ID, Name, Telefon, Adresse] }
Mitwirkung : { [Film, Person, Funktion] }
Vorstellung : { [Film, Datum, Kino] }

Formulieren Sie die folgenden Anfragen in SQL und QBE:

1. Welche Kinos gibt es in Bonn?
2. Gibt es verschiedene Personen mit demselben Namen?
3. Welche nicht jugendfreien Filme laufen im „Metropol“?
4. Mit welchen Regisseuren hat Michelle Pfeiffer zusammengearbeitet?
5. Welche Schauspieler führten auch Regie?
6. Hat jemand an allen Filmen von Woody Allen mitgewirkt?
7. Welche Filme wurden produziert, aber nie gezeigt?

Aufgabe 2 (Aggregatoperatoren in SQL). Erstellen Sie SQL-Anweisungen für die folgenden Anfragen:

1. Bei wievielen Filmen hat jede Person mitgewirkt? (Angabe der Personen-ID genügt)
2. Bei wievielen Filmen wirkt eine Person im Durchschnitt mit?
3. Wie heißt die Person, die bei den meisten Filmen mitgewirkt hat?
4. Erstellen Sie eine Übersicht: jahresweise die Anzahl an Filmen und die durchschnittliche Anzahl an Vorstellungen.

Aufgabe 3 (Nullwerte in SQL). Gegeben sei eine (unter den für die Übungen eingerichteten ORACLE-Accounts verfügbare) Vertreter-Relation mit Nullwerten:

Vertreter	Name	Vorname	Grundgehalt	Bonus
	Meier	Georg	5.000	300
	Schmitz	Josef	5.000	-
	Berger	Luise	-	500
	Baumann	Fritz	-	500

Untersuchen Sie die folgenden SQL-Anfragen:

1. `SELECT Name FROM Vertreter WHERE Bonus=500`
2. `SELECT Name FROM Vertreter WHERE Bonus<>500`
3. `SELECT COUNT(Name) FROM Vertreter`
4. `SELECT COUNT(Grundgehalt) FROM Vertreter`
5. `SELECT COUNT(*) FROM Vertreter`
6. `SELECT SUM(Grundgehalt+Bonus) FROM Vertreter`
7. `SELECT SUM(Grundgehalt)+SUM(Bonus) FROM Vertreter`
8. `SELECT SUM(Bonus) FROM Vertreter GROUP BY Grundgehalt`

Diskutieren Sie die Ergebnisse.

Aufgabe 4 (Rekursion, Embedded SQL). Es sei eine Relation VS gegeben, die eine Vater-Sohn-Beziehung zwischen zwei Personen ausdrücken soll:

$$VS : \{ [\underline{\text{Vater}} : \text{string}, \text{Sohn} : \text{string}] \}$$

1. Geben Sie eine SQL-Anweisung an, die den Ur-Großvater (väterlicherseits) von „Jim Knopf“ ausgibt.
2. Wie würde man in SQL die gesamte Abstammungshierarchie ausgeben?
3. Formulieren Sie mit Hilfe eines C-Programms und Embedded-SQL eine Anfrage, die sämtliche männlichen Vorfahren von „Jim Knopf“ ausgibt.

Aufgabe 5 (Datenintegrität). Formulieren Sie die Anweisungen, die zur Erstellung der Film-Datenbank (siehe Aufgabe 1) notwendig sind in SQL. Überlegen Sie insbesondere, welche Maßnahmen zur Einhaltung der referentiellen Integrität getroffen werden müssen und welche weiteren Integritätsbedingungen sinnvoll wären.