

Arbeitsblatt zu SQL-Befehlen


Erste Übungen - Lösungen

Aufgabe 1

Gegeben seien die Relationen
Student(MatrnNr, Name)
Vorlesung(VorlNr, Titel, PersNr)
Professor(PersNr, Name)
und die Beziehungsrelation
hört(MatrnNr, VorlNr)

Erörtere jeweils, was die folgenden SQL-Anweisungen bewirken:

Anweisungen mit Lösungen

- a) `SELECT * FROM Student`
Listet die Werte aller Spalten aus der Tabelle *Student* auf. 
- b) `SELECT *`
`FROM Student`
`WHERE (MatrnNr LIKE "156-*") OR (Name LIKE "Ab*")`
Listet die Werte aller Spalten aus der Tabelle *Student* auf, bei denen die *Matrikelnummer* mit „156“ oder der *Name* mit „Ab“ beginnt.
- c) `SELECT DISTINCT PersNr FROM Vorlesung`
(spezielle) Projektion: Listet die vorhandenen, verschiedenen Ausprägungen der Spalte *PersNr* aus der Tabelle *Vorlesung* auf.
- d) `SELECT Name, MatrNr AS Matrikelnummer FROM Student`
Die Spalte *MatrNr* heißt in der Ergebnisrelation jetzt *Matrikelnummer*.
- e) `SELECT Titel, VorlNr FROM Vorlesung WHERE PersNr = 12`
Selektion: Listet alle *Vorlesungen* des *Professors* mit der *PersNr* 12 auf.
- f) `SELECT Titel, VorlNr FROM Vorlesung ORDER BY PersNr`
Projektion mit Gruppierung: Listet alle *Vorlesungs-Titel*, sortiert nach haltenden *Professoren*, auf.
- g) `SELECT Vorlesung.VorlNr, Vorlesung.Titel, Professor.Name, Professor.PersNr FROM Professor INNER JOIN Vorlesung ON Vorlesung.PersNr = Professor.PersNr`
Innerer natürlicher Verbund: Listet die Werte der Spalten *VorlNr* und *Titel* aus der Tabelle *Vorlesung* sowie der Spalten *Name* und *Persnr* aus der Tabelle *Professor* für alle *Vorlesungen* auf.
- h) `SELECT Vorlesung.VorlNr, Vorlesung.Titel, Professor.Name, Professor.PersNr FROM Professor LEFT OUTER JOIN Vorlesung ON Vorlesung.PersNr = Professor.PersNr`
Äußerer linker natürlicher Verbund: Listet die Werte der Spalten *VorlNr* und *Titel* aus der Tabelle *Vorlesung* sowie der Spalten *Name* und *PersNr* aus der Tabelle *Professor* für alle *Vorlesungen* auf. Professoren, die keine *Vorlesungen* halten werden auch mit aufgelistet.
- i) `SELECT Professor.Name, Professor.PersNr`
`FROM Professor LEFT OUTER JOIN Vorlesung ON Professor.PersNr = Vorlesung.PersNr`
`WHERE Vorlesung.PersNr IS NULL`
Listet alle *Professoren* auf, die keine *Vorlesungen* halten.
- j) `SELECT Professor.Name, Professor.PersNr`
`FROM Professor`
`WHERE NOT EXISTS (SELECT * FROM Vorlesung WHERE Vorlesung.PersNr = Professor.PersNr)`
Unterabfrage mit Existenz-Quantor: Das gleiche mit einer Unterabfrage.
- k) `SELECT COUNT(Vorlesung.PersNr) AS Anzahl, Professor.Name, Professor.PersNr FROM Professor LEFT OUTER JOIN Vorlesung on Professor.PersNr = Vorlesung.PersNr GROUP BY Professor.Name, Professor.PersNr`
Gruppierung, Aggregation und äußerer linker natürlicher Verbund: Zählt die Anzahl der *Vorlesungen* pro *Professor* (ohne Nullwerte!).

z.B.

MatrNr	Name
123456	Müller
123457	Meier
123458	Brandt

Abteilung	ID	Name	Leiter
	0	Raumfahrt	1
	1	Fuhrpark	4
	2	Verwaltung	2

arbeitet_an	ID	Mitarbeiter	Projekt
	0	1	0
	1	1	1
	2	2	2
	3	5	0
	4	0	1
	5	0	2

Mitarbeiter	ID	Name	Vorname	Abteilung
	0	Müller	Anton	NULL
	1	Geiger	Sven	0
	2	Schwab	Anita	2
	4	Görgens	Margit	1
	5	Hurz	Willy	NULL

Projekt	ID	Bezeichner	Abteilung	Verantwortlicher
	0	Apollo 13	0	5
	1	Challenger	0	4
	2	Webseiten	2	0

Aufgabe 2

- Finde den Leiter der Raumfahrtabteilung.
- Gib alle Abteilungsleiter aus.
- Notiere alle Projekte, die zur Verwaltungsabteilung gehören.
- Finde heraus, wer für das Apollo 13 Projekt verantwortlich ist.
- Schreibe alle Namen von Leuten nieder, die am Challenger Projekt arbeiten.
- Gib alle Projektverantwortlichen aus.
- Erörtere, ob es eine Person gibt, die am Challenger Projekt arbeitet und gleichzeitig Leiter einer Abteilung ist.
- Liste alle Mitarbeiter auf, die am Apollo 13 Projekt oder am Webseiten-Projekt arbeiten.

Lösungen

1.

```
SELECT Mitarbeiter.Name, Mitarbeiter.Vorname
FROM Mitarbeiter, Abteilung
WHERE Abteilung.Leiter = Mitarbeiter.ID
AND Abteilung.Name = "Raumfahrt"
```

Ergebnis:	Name	Vorname
	Geiger	Sven

```
AND Projekt.ID = arbeitet_an.Projekt
AND arbeitet_an.Mitarbeiter = Mitarbeiter.ID
```

Ergebnis:	Name	Vorname
	Geiger	Sven
	Müller	Anton

6.

```
SELECT Mitarbeiter.Name, Mitarbeiter.Vorname
FROM Mitarbeiter INNER JOIN Projekt
ON Mitarbeiter.ID = Projekt.Verantwortlicher
```

Ergebnis:	Name	Vorname
	Hurz	Willy
	Görgens	Margit
	Müller	Anton

7.

```
SELECT Mitarbeiter.Name, Mitarbeiter.Vorname
FROM Mitarbeiter, Projekt, arbeitet_an, Abteilung
WHERE Projekt.Bezeichner = "Challenger"
AND Projekt.ID = arbeitet_an.Projekt
AND arbeitet_an.Mitarbeiter = Mitarbeiter.ID
AND arbeitet_an.Mitarbeiter = Abteilung.Leiter
```

Ergebnis:	Name	Vorname
	Geiger	Sven

8.

```
SELECT Mitarbeiter.Name, Mitarbeiter.Vorname
FROM Mitarbeiter, Projekt, arbeitet_an
WHERE (Projekt.Bezeichner = "Apollo 13" OR
Projekt.Bezeichner = "Webseiten")
AND Projekt.ID = arbeitet_an.Projekt
AND arbeitet_an.Mitarbeiter = Mitarbeiter.ID
```

Ergebnis:	Name	Vorname
	Geiger	Sven
	Hurz	Willy
	Schwab	Anita
	Müller	Anton

2.

```
SELECT Mitarbeiter.Name, Mitarbeiter.Vorname
FROM Mitarbeiter INNER JOIN Abteilung
ON Mitarbeiter.ID = Abteilung.Leiter
```

Ergebnis:	Name	Vorname
	Geiger	Sven
	Görgens	Margit
	Schwab	Anita

3.

```
SELECT Projekt.Bezeichner
FROM Projekt INNER JOIN Abteilung
ON Abteilung.ID = Projekt.Abteilung
WHERE Abteilung.Name = "Verwaltung"
```

Ergebnis:	Bezeichner
	Webseiten

4.

```
SELECT Mitarbeiter.Name, Mitarbeiter.Vorname
FROM Mitarbeiter, Projekt
WHERE Projekt.Bezeichner = "Apollo 13"
AND Projekt.Verantwortlicher = Mitarbeiter.ID
```

Ergebnis:	Name	Vorname
	Hurz	Willy

5.

```
SELECT Mitarbeiter.Name, Mitarbeiter.Vorname
FROM Mitarbeiter, Projekt, arbeitet_an
WHERE Projekt.Bezeichner = "Challenger"
```