

# Erste Übungen SQL-Befehle

## Aufgabe 1:

a) *SELECT \* FROM Student*

Der Befehl gibt alle Daten der Tabelle Studenten aus – also Matrikelnummer und Schülernamen.

b) *SELECT \* FROM Student WHERE (MatrNr LIKE "156-\*)" OR (Name LIKE "Ab\*")*

Der Befehl gibt alle Daten der Tabelle Studenten (Matrikelnummer und Name) aus für Studenten, deren Matrikelnummer entweder mindestens 156 beträgt oder deren Namen mit „Ab“ anfängt.

c) *SELECT DISTINCT PersNr FROM Vorlesung*

Gibt die PersNr (also die ID der Professoren) aus der Liste Vorlesungen aus. Dabei handelt es sich um alle Professoren, die Vorlesungen halten – jedoch bewirkt das DISTINCT, dass jeder PersNr nur einmal aufgelistet wird.

d) *SELECT Name, MatrNr AS Matrikelnummer FROM Student*

Der Befehl gibt die Spalte Name und die Spalte MatrNr aus der Liste Student aus. Dabei steht der Name (entgegengesetzt der ursprünglichen Definierung) an erster Stelle, während die MatrNr in der Überschrift zur Matrikelnummer umbenannt wird.

e) *SELECT Titel, VorlNr FROM Vorlesung WHERE PersNr = 12*

Der Befehl gibt den Titel und die Vorlesungsnummer aller Vorlesungen aus, die durch den Professor mit der PersNr 12 gehalten werden.

f) *SELECT Titel, VorlNr FROM Vorlesung ORDER BY PersNr*

Der Befehl gibt die Titel und Vorlesungsnummer aller Vorlesungen an, geordnet nach der PersNr (also der ID des Dozenten).

g) *SELECT Vorlesung.VorlNr, Vorlesung.Titel, Professor.Name, Professor.PersNr FROM Professor INNER JOIN Vorlesung ON Professor.PersNr = Vorlesung.PersNr*

Der Befehl erzeugt den Join aus den Vorlesungen und den Professoren. Dabei werden nur jene Einträge gejoint (verbunden), bei denen die PersNr in der Klasse Professor und Vorlesung identisch sind. Im Anschluss werden Vorlesungsnummer, Titel, Name des Professors und seine PersNr in eben beschriebener Reihenfolge ausgegeben.

h) *SELECT Vorlesung.VorlNr, Vorlesung.Titel, Professor.Name, Professor.PersNr FROM Professor LEFT OUTER JOIN Vorlesung ON Professor.PersNr = Vorlesung.PersNr*

Diese SQL-Anweisung gibt eine Liste aller Vorlesungen in Kombination mit dem haltenden Professor aus. Dabei kann es geschehen, dass der Wert des haltenden Professors NULL ist.

i) *SELECT Professor.Name, Professor.PersNr*

*FROM Professor LEFT OUTER JOIN Vorlesung ON Professor.PersNr = Vorlesung.PersNr  
WHERE Vorlesung.PersNr IS NULL*

Der SQL-Befehl gibt eine Liste aller Professoren mit ihrem Namen und PersNr aus, die keine Vorlesung halten (also deren Vorlesung.PersNr im Join leer bleibt).

j) *SELECT Professor.Name, Professor.PersNr*

*FROM Professor*

*WHERE NOT EXISTS (SELECT \* FROM Vorlesung WHERE Vorlesung.PersNr = Professor.PersNr)*

Diese Anweisung gibt über den Befehl WHERE NOT EXISTS die Namen jener Professoren aus, die keine Vorlesungen halten (beziehungsweise deren PersNr keiner PersNr im Vorlesungskatalog zugeordnet ist).

k) *SELECT COUNT (Vorlesung.PersNr) AS Anzahl, Professor.Name, Professor.PersNr*

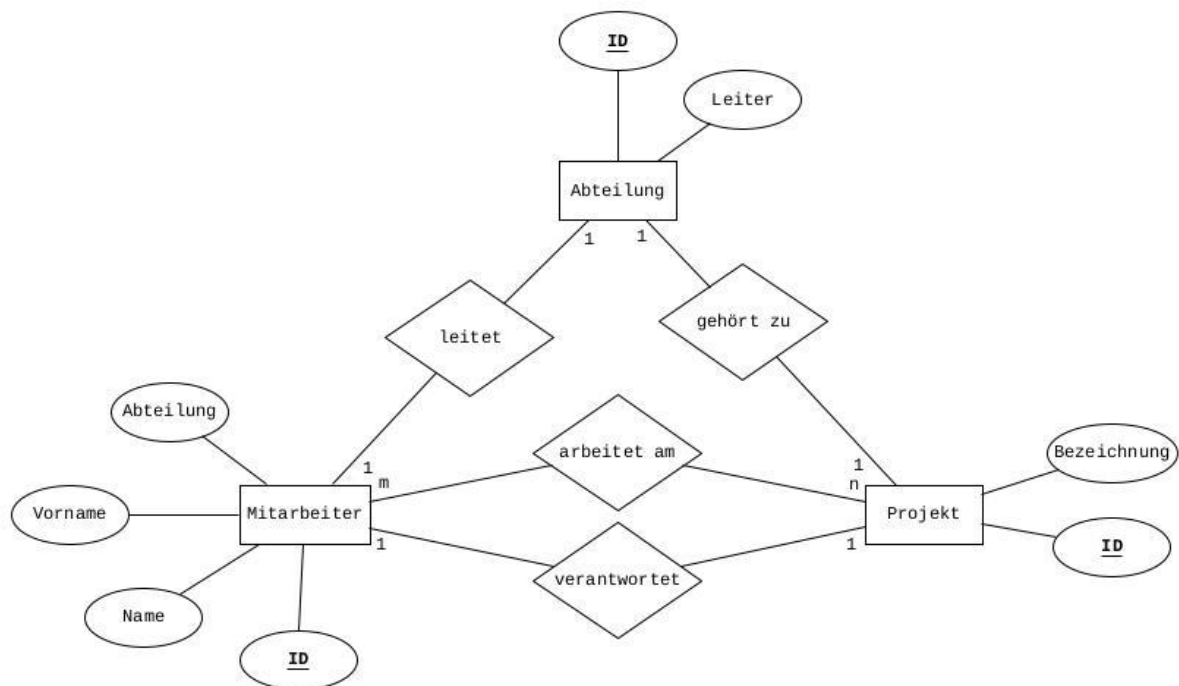
*FROM Professor LEFT OUTER JOIN Vorlesung ON Professor.PersNr = Vorlesung.PersNr*

*GROUP BY Professor.Name, Professor.PersNr*

Über den Count-Befehl soll diese SQL-Anweisung die Anzahl der Vorlesungen, die ein Professor hält, zählen. Im Anschluss werden die Ergebnisse gruppiert nach Namen und PersNr des Professors.

## Aufgabe 2:

a)



b)

(1) *Leiter der Raumfahrtbehörde*

*SELECT Mitarbeiter.Name, Mitarbeiter.Vorname FROM Mitarbeiter JOIN Abteilung WHERE Abteilung.Name = ,Raumfahrt‘*

(2) *Alle Abteilungsleiter*

*SELECT Mitarbeiter.Name, Mitarbeiter.Vorname FROM Mitarbeiter JOIN Abteilung*

(3) *Projekte der Verwaltungsabteilung*

*SELECT Projekt.Bezeichnung FROM Projekt WHERE Projekt.Abteilung = 2*

*//2 ist die ID der Verwaltungsabteilung, was den Join überflüssig macht*

*(4) Verantwortlicher Apollo 13 Projekt*

```
SELECT Mitarbeiter.Name, Mitarbeiter.Vorname FROM Projekt JOIN Mitarbeiter WHERE  
Projekt.Bezeichnung = ,Apollo 13‘
```

*(5) Namen der Mitarbeiter Challenger-Projekt*

```
SELECT Mitarbeiter.Name, Mitarbeiter.Vorname FROM Projekt JOIN arbeitet_am JOIN Mitarbeiter  
WHERE Projekt.Bezeichnung = ,Challenger‘
```

//Hier wäre auch der Verzicht auf den Join mit Projekt möglich, in dem man auf die Projekt-ID zurückgreift

*(6) Liste der Projektverantwortlichen*

```
SELECT Mitarbeiter.Name, Mitarbeiter.Vorname FROM Projekt JOIN Mitarbeiter  
//Der Join von Mitarbeitern und Projekte wird nur über die Mitarbeiter-IDs, welche als  
Verantwortliche eingetragen sind, durchgeführt
```

*(7) Abteilungsleiter als Mitarbeiter des Challenger-Projekts*

Sven Geiger, welcher Abteilungsleiter der Raumfahrt ist, arbeitet auch am Challenger-Projekt.

```
SELECT Mitarbeiter.Name, Mitarbeiter.Vorname FROM Projekt JOIN Mitarbeiter JOIN Abteilung  
JOIN arbeitet_am WHERE Projekt.Bezeichnung = ,Challenger‘
```

//Zusätzlich möglich über spezielle Projekt-IDs

*(8) Mitarbeiter Apollo 13 und Website*

```
SELECT Mitarbeiter.Name, Mitarbeiter.Vorname FROM Projekt JOIN arbeitet_am JOIN Mitarbeiter  
WHERE Projekt.Bezeichnung = ,Apollo 13‘ OR Projekt.Bezeichnung = ,Website‘
```

//Zusätzlich möglich über spezielle Projekt-IDs