

SQL Code

```
1 -- Gruppenaufgabe 1
2 -- SELECT: определяет столбцы для вывода (проекция)
3 -- FROM: указывает таблицы или источники данных
4 -- WHERE: фильтрует строки до группировки
5 -- GROUP BY: объединяет строки с одинаковыми значениями (агрегация)
6 -- HAVING: фильтрует группы после группировки (работает с агрегатными функциями)
7 -- ORDER BY: сортирует итоговый результат
8
9 -- Gruppenaufgabe 2
10 -- a) Welche Namen haben die Studierenden?
11     SELECT NAME
12     FROM STUDENTEN;
13
14 -- b) Welche Vorlesungen haben einen Umfang von 4 SWS?
15     SELECT *
16     FROM Vorlesungen
17     WHERE SWS = 4;
18
19 -- c) Welche AssistentInnen (Name) arbeiten an den Fachgebieten ,Ideenlehre' oder 'Sprachtheorie'?
20     SELECT NAME
21     FROM ASSISTENTEN
22     WHERE FACHGEBIET = 'Ideenlehre' OR FACHGEBIET = 'Sprachtheorie';
23
24 -- d) Welche Namen gibt es unter Studierenden oder ProfessorInnen?
25     SELECT Name
26     FROM STUDENTEN
27     UNION
28     SELECT Name
29     FROM PROFESSOREN;
30
31
32 -- e) Welche Vorlesung (Titel) wird von welchem/r ProfessorIn (Name) gelesen?
33
34 -- explizit JOIN
35     SELECT v.Titel, p.Name
36     FROM VORLESUNGEN v
37     JOIN PROFESSOREN p ON v.GELESENVON = p.PERSNR;
38
39 -- explizit JOIN
40     SELECT v.Titel, p.Name
41     FROM Vorlesungen v
42     JOIN Professoren p ON v.gelesenVon = p.PersNr;
43
44 -- implizit JOIN
45     SELECT v.Titel, p.Name
46     FROM VORLESUNGEN v, PROFESSOREN p
47     WHERE v.GELESENVON = p.PERSNR AND p.PERSNR < 50;
```

```

48
49
50
51 -- f) Welche Vorlesungen (Titel) werden von Professor Augustinus gelesen?
52 SELECT v.Titel
53 FROM VORLESUNGEN v
54 JOIN PROFESSOREN p ON v.GELESENVON = p.PERSNR
55 WHERE p.NAME = 'Augustinus';
56
57 -- g) Welche AssistentInnen (Name) arbeiten für ProfessorInnen, die Vorlesung
en mit 4 SWS halten?
58
59 -- explizit JOIN
60 SELECT a.Name
61 FROM Assistenten a
62 JOIN Professoren p ON a.Boss = p.PersNr
63 JOIN Vorlesungen v ON p.PersNr = v.gelesenVon
64 WHERE v.SWS = 4;
65
66 -- implizit JOIN
67 SELECT Name
68 FROM ASSISTENTEN, VORLESUNGEN
69 WHERE GELESENVON = BOSS and SWS = 4;
70
71
72 -- Gruppenaufgabe 3
73 -- а) Имена профессоров, которые читают лекции с 2 SWS, для которых есть усло
вия (предшествующие лекции)
74 -- πName (σVorlNr = Nachfolger ( σSWS=2 (Professoren × PersNr = gelesenVon
Vorlesungen) × voraussetzen ) )
75 -- деревоπ (Name)
76 -- |
77 -- σ (VorlNr = Nachfolger)
78 -- |
79 -- x (Kreuzprodukt)
80 -- / \ 
81 -- σ (SWS=2)      voraussetzen
82 -- |
83 -- × (PersNr=gelesenVon)
84 -- / \ 
85 -- Professoren Vorlesungen
86 -- Optimierung: Selektion nach unten schieben, Kreuzprodukt eliminieren
87 --
88 -- π (Name)
89 -- |
90 -- × (PersNr = gelesenVon)
91 -- / \ 
92 -- Professoren      × (VorlNr = Nachfolger)
93 -- / \ 
94 -- σ (SWS=2)      voraussetzen
95 -- |
96 -- Vorlesungen

```

```

97  SELECT DISTINCT Name
98  FROM VORLESUNGEN v
99  JOIN PROFESSOREN p ON v.GELESENVON = p.PERSNR
100 JOIN VORAUSSETZEN vor ON v.VORLNR = vor.NACHFOLGER
101 WHERE v.SWS = 2;
102
103
104 ---
105
106
107 -- Hausaufgabe 1
108 -- a) Finden Sie die Studierenden (Name, MatrNr), die den Professor Sokrates
aus Vorlesung(en) kennen
109 SELECT s.Name, s.MatrNr
110 FROM Studenten s
111 JOIN HOEREN h ON s.MatrNR = h.MATRNR
112 JOIN Vorlesungen v ON h.VorlNr = v.VORLNR
113 JOIN Professoren p ON v.GELESENVON = p.PersNr
114 WHERE p.Name = 'Socrates'
115
116 -- b) Finden Sie die Studierenden (Name, MatrNr), die Vorlesungen hören, die
auch der Student Fichte hört.
117 SELECT distinct s.Name, s.MatrNr
118 FROM Studenten s
119 JOIN hoeren h ON s.MatrNr = h.MATRNR
120 WHERE h.VORLNR IN (
121     SELECT h2.VorlNr
122     FROM HOEREN h2
123     JOIN STUDENTEN s2 ON h2.MATRNR = s2.MATRNR
124     WHERE s2.NAME = 'Fichte'
125     ) AND s.NAME != 'Fichte';
126
127 -- c) Finden Sie die AssistentInnen von ProfessorInnen (PersNr, Name), die die
Studenten Fichte unterrichtet haben
128 SELECT a.PersNr, a.Name
129 FROM Assistenten a
130 JOIN Professoren p ON a.Boss = p.PersNr
131 JOIN Vorlesungen v ON v.gelesenVon = p.PersNr
132 JOIN hoeren h ON v.VorlNr = h.VorlNr
133 JOIN Studenten s ON h.MatrNr = s.MatrNr
134 WHERE s.Name = 'Fichte';
135
136 -- d) Geben Sie die Namen der ProfessorInnen an (PersNr, Name), die der Stu-
dent Xenokrates aus Vorlesungen kennt.
137 SELECT DISTINCT p.PersNr, p.Name
138 FROM Professoren p
139 JOIN Vorlesungen v ON p.PersNr = v.gelesenVon
140 JOIN hoeren h ON v.VorlNr = h.VorlNr
141 JOIN Studenten s ON h.MatrNr = s.MatrNr
142 WHERE s.Name = 'Xenokrates';
143
144 -- e) Welche Vorlesungen (Titel) werden von Studierenden im Grundstudium (1.-
```

4- Semester) gehört?

```
145  SELECT v.Titel
146  FROM VORLESUNGEN v
147  JOIN hoeren h on v.VORLNR = h.VORLNR
148  JOIN Studenten s ON h.MATRNR = s.MATRNR
149  WHERE s.SEMESTER BETWEEN 1 AND 4;
150
151
152  -- Hausaufgabe 2
153  -- a) Geben Sie den SQL-Befehl zum Erstellen einer neuen Tabelle Abschlussarbeiten an.
154  -- Diese soll eine eindeutige Nummer zur Identifizierung der Abschlussarbeiten,
155  -- den Titel der Arbeit und das Abgabedatum enthalten
156
157  CREATE TABLE Abschlussarbeiten (
158      ArbeitsNr INT PRIMARY KEY,
159      Titel VARCHAR(255),
160      Abgabedatum DATE
161 );
162
163  -- b) Geben Sie den SQL-Befehl zum Löschen der in a) erstellten Tabelle an.
164  DROP TABLE Abschlussarbeiten;
165
166  -- c) Fügen Sie mit Hilfe eines SQL-Befehls die Daten eines neuen Studenten mit dem
167  -- Namen Platon und der Matrikelnummer 1234 in die entsprechende Tabelle ein.
168
169  INSERT INTO STUDENTEN (MATRNR, NAME, SEMESTER)
170  VALUES (1234, 'Platon', 1);
171
172  -- d) Nutzen Sie einen SQL-Befehl, um den in c) hinzugefügten Studenten wieder aus
173  -- der Studenten Tabelle zu löschen.
174
175  DELETE FROM STUDENTEN
176  WHERE MATRNR = 1234;
177
178  -- e) Ändern Sie die Besoldung der Professoren, sodass Professoren mit C3 nun mit C4
179  -- besoldet werden
180  UPDATE PROFESSOREN
181  SET Rang = 'C4'
182  WHERE Rang = 'C3';
```