

Tutorium 8 - Datenbanksysteme

21.12.2022 – Finn Kapitza



Aufgabe 8.1 – Anfragen in SQL

Professoren (persNr, name, raum)

Studenten (matrNr, name, semester)

Vorlesungen (vorlNr, titel, ects, gelesenVon [Professoren])

Voraussetzungen (vorausgesetzt [Vorlesungen], vorlesung [Vorlesungen])

hoeren (matrNr [Studenten], vorlNr [Vorlesungen])

Pruefungen (matrNr [Studenten], vorlNr [Vorlesungen], note)

Studenten		
matrNr	name	semester

Professoren		
persNr	name	raum

hoeren	
matrNr	vorlNr

Pruefungen		
matrNr	vorlNr	note

Vorlesungen			
vorlNr	titel	ects	gelesenVon

Voraussetzungen	
vorausgesetzt	vorlesung

Aufgabe 8.1.a

Bestimmen Sie für **Jede Vorlesung** die Durchschnittsnote **über alle** in dem Fach stattgefundenen Prüfungen

Anzuzeigen Sind:

- Vorlesungsnummer/-titel
- Durchschnittsnote gerundet auf 3 Nachkommastellen

Informationen stehen in:

- Vorlesungen
 - Prüfungen
- } vorlNr

Aufgabe 8.1.a

SELECT vorlNr, titel, **ROUND**(**AVG**(note), 3) **AS** Durchschnittsnote

FROM Vorlesungen **NATURAL JOIN** Pruefungen

GROUP BY vorlNr, titel

Aufgabe 8.2.b

Bestimmen sie die Professoren, zusammen mit der **Anzahl der unterschiedlichen** Vorlesungen, die er hält.

Anzuzeigen sind:

- Personalnummer
- Professornamen
- Anzahl **unterschiedlicher** Vorlesungen (0 wenn Prof. keine hält)
-> in absteigender Reihenfolge

Informationen sind in:

- | | | |
|---------------|---|------------|
| - Professoren | } | persNr |
| - Vorlesungen | | gelesenVon |

Aufgabe 8.1.b

```
SELECT persNr, name, COUNT(vorlNr) AS Anzahl_Vorlesungen  
  
FROM Professoren LEFT JOIN Vorlesungen ON persNr = gelesenVon  
  
GROUP BY persNr, name  
  
ORDER BY Anzahl DESC
```

Aufgabe 8.1.c

Bestimme für **jeden Studenten**, wie **viele andere** Studenten aus einem **höheren Semester** dieser kennt (mind. Eine Vorlesung zusammen besuchen)

Anzuzeigen sind:

- Matrikelnummer / Name
- Anzahl Studenten aus höherem Semester mit denen eine Vorlesung besucht wird

Informationen sind in:

- Studenten x2
 - hoeren x2
- } **matrNr**

Aufgabe 8.1.c

```
SELECT s1.matrNr, s1.name, COUNT(DISTINCT s2.matrNr) AS Freunde  
  
FROM Studenten s1, hoeren h1, Studenten s2, hoeren h2  
  
WHERE s1.matrNr = h1.matrNr AND s2.matrNr = h2.matrNr AND  
  
        h1.vorlNr = h2.vorlNr AND s1.semester < s2.semester  
  
GROUP BY s1.matrNr, s1.name  
HAVING Freunde > 2
```

Aufgabe 8.1.d

Bestimmen Sie die Professoren, für die **mindestens zwei Studenten**
mindestens drei Vorlesungen des jeweiligen Professors hören

Anzuzeigen sind:

- Personalnummer und Name vom Professor

Informationen in:

- Professoren
 - Vorlesungen
 - hoeren
- } matrNr
- } persNr = gelesenVon

Aufgabe 8.1.d – Liste der Studenten mit der jeweiligen Anzahl an Vorlesungen die Sie von den Professoren hören (mind. 3)

CREATE VIEW A AS

(

SELECT matrNr, gelesenVon, **COUNT**(*) as anz

FROM Vorlesungen **NATURAL JOIN** hoeren

GROUP BY matrNr, gelesenVon

HAVING COUNT(*) >= 3

)

Aufgabe 8.1.d – Finales Ergebnis

SELECT persNr, name

FROM Professoren **JOIN** A **ON** persNr = gelesenVon

GROUP BY persNr, name

HAVING COUNT(*) >= 2

Aufgabe 8.1.e

Bestimme für **jede** Vorlesung, welche **mind. Eine** andere Vorlesung voraussetzt, die **prozentualen Anteile** der teilnehmenden Studenten **pro Semester**

Anzuzeigen sind:

- Vorlesungsnummer und Titel
- Semester
- Prozentualer Anteil des Semesters an allen Zuhörenden

Aufgabe 8.1.e

Informationen in:

- Vorlesungen
- Studenten
- Hören

Sortieren nach:

- Alphabetisch nach Titel
- Absteigend nach prozentualem Anteil
- Aufsteigend nach Semester

Aufgabe 8.1.e – Vorlesungen mit Voraussetzung

```
CREATE VIEW Vorlesung_mit_Voraussetzung AS  
(  
  SELECT vorlNr, titel  
  FROM Vorlesungen  
  
  WHERE vorlNr IN (  
    SELECT vorlesung  
    FROM Voraussetzungen)  
)
```

Aufgabe 8.1.e – Anzahl der Gesamten Hörer Pro Vorlesung

```
CREATE VIEW Anzahl_Gesamt AS  
(  
SELECT vorlNr, COUNT(*) AS anz_ges  
  
FROM hoeren  
  
GROUP BY vorlNr  
)
```


Aufgabe 8.1.e – Anzahl der Hörer Pro Vorlesung pro Semester

```
CREATE VIEW Anzahl_Semester AS  
(  
SELECT vorlNr, semester, COUNT(*) AS anz_sem  
  
FROM hoeren NATURAL JOIN Studenten  
  
GROUP BY vorlNr, semester  
)
```

Aufgabe 8.1.e – Finales Ergebnis

```
SELECT vorlNr, titel, semester, ROUND(anz_sem/anz_ges *100, 2) AS Prozent  
  
FROM Anzahl_Semester NATURAL JOIN Anzahl_Gesamt  
      NATURAL JOIN Vorlesung_mit_Voraussetzung  
  
WHERE anz_sem < anz_ges  
  
GROUP BY titel ASC, Prozent DESC, semester ASC
```

Aufgabe 8.2.a – Finden Sie die Artikelnummer, die Artikelbezeichnung, den Lagerort und den Lagerbestand aller in Hamburg oder München gelagerten Artikel

SELECT art_nr, art_bez, lagerort, lagerbest

FROM Inventar

WHERE lagerort **IN** ("Hamburg", "Muenchen")

Aufgabe 8.2.b – Finden Sie für alle Bestellungen des Artikels „203333“ die Auftragsnummer, die Artikelnummer, die bestellte Menge und alle Lagerbestände und Lagerorte, an denen ausreichend Stückzahl vorhanden ist

SELECT auftr_nr, art_nr, menge, lagerbest, lagerort

FROM Auftragsposten **NATURAL JOIN** Inventar

WHERE art_nr = “203333” **AND** lagerbest >= menge

Aufgabe 8.2.c – Finden Sie die Nummer derjenigen Kunden, die keinen Auftrag erteilt haben

SELECT kund_nr

FROM Kunde

WHERE kund_nr **NOT IN** (

SELECT kund_nr

FROM Verkauf

)

Aufgabe 8.2.d – Finden Sie die verschiedenen Lagerorte von Artikeln, die einen Lagerbestand von mindestens 8 Exemplaren haben

SELECT DISTINCT lagerort

FROM Inventar

WHERE lagerbest >= 8

Aufgabe 8.2.e – Finden sie den Vornamen und Nachnamen aller Angestellter, die einen oder mehrere Verkäufe an Kunden aus Stuttgart bearbeitet haben

SELECT DISTINCT vorname, nachname

FROM Personal **NATURAL JOIN** Verkauf **NATURAL JOIN** Kunde

WHERE ort = "Stuttgart"

Aufgabe 8.2.f – Finden Sie den Nachnamen, Vornamen, Einsatzort und Gehalt aller Angestellten. Sortiere das Ergebnis lexikographisch nach dem Einsatzort, bei gleichem Einsatzort absteigend nach dem Gehalt

SELECT nachname, vorname

FROM Personal

ORDER BY einsatz, gehalt **DESC**

Aufgabe 8.2.g – Finden Sie die Personalnummern und das Gehalt der Angestellten mit minimalen oder maximalem Gehalt

```
SELECT pers_nr, gehalt
FROM Personal
WHERE gehalt = (
    SELECT MIN(gehalt)
    FROM Personal)
OR gehalt = (
    SELECT MAX(gehalt)
    FROM Personal)
```

Aufgabe 8.2.h – Finden Sie für jeden Einsatzort die Anzahl der dort eingesetzten Angestellten

SELECT einsatz, **COUNT**(*)

FROM Personal

GROUP BY einsatz

Aufgabe 8.2.i – Finden sie die durchschnittliche Anzahl an Angestellten über alle Einsatzorte

```
SELECT AVG(anzahl)
FROM (
    SELECT einsatz, COUNT(*) AS anzahl
    FROM Personal
    GROUP BY einsatz) AS temp
```

Aufgabe 8.2.j – Finden Sie für jeden Artikel die Artikelnummer und die Summe der Lagerbestände in den Lagerorten, die den Artikel führen. Nur Artikel mit Gesamtbestand von mehr als 10 sollen angezeigt werden

SELECT art_nr, **SUM**(lagerbest)

FROM Inventar

GROUP BY art_nr

HAVING SUM(lagerbest) > 10



LUDWIG-
MAXIMILIANS-
UNIVERSITÄT
MÜNCHEN



Finn Kapitza – Finn.Kapitza@campus.lmu.de