**Aufgaben db\_forz**

**Implementieren Sie die Datenbank db\_forz\_alternativ mittels des SQL-Skripts.**

**Lesen Sie die zugehörigen Daten ein.**

**Bei genauerem Hinsehen bemerken Sie, dass eine beliebige Zeichenfolge als Wochentag eingegeben werden kann. Dies gefährdet jedoch die Integrität Ihrer Datenbank. Analysieren Sie Ihre eingelesenen Daten und ergänzen Sie darauf basierend die Tabelle t\_stunde um einen geeigneten Constraint. (Hinweis: Man kann Tabellen nachträglich mittels des Befehls alter table ergänzen)**

alter table t\_stunde add constraint pruefe\_wochentag CHECK (wochentag in ('Mo', 'Di', 'Mi', 'Do', 'Fr'));

**Es besteht auch die Möglichkeit, nachträglich bereits bestehende Constraints zu entfernen. Entfernen Sie Ihren gerade definierten Constraint und vergewissern Sie sich durch eine Eingabe, dass die Regelung nicht mehr greift. Ergänzen Sie anschließend die Tabelle wieder um den gerade entfernten Constraint.**

alter table t\_stunde drop constraint pruefe\_wochentag;

alter table t\_stunde add constraint pruefe\_wochentag CHECK (wochentag in ('Mo', 'Di', 'Mi', 'Do', 'Fr'));

**Analog können völlig beliebige Beginn- und Endzeiten für die einzelnen Stundenblöcke hinterlegt werden. Schränken Sie dies ebenfalls auf die aktuelle gültigen Beginn- sowie Endezeiten ein!**

alter table t\_stunde add constraint pruefe\_beginn CHECK (beginn in ('07:25', '08:15', '09:25', '09:30', '10:15', '10:20', '11:15', '12:05', '13:00', '13:50', '14:45', '15:35'));

alter table t\_stunde add constraint pruefe\_ende CHECK (ende in ('08:10', '09:00', '10:10', '10:15', '11:00', '11:05', '12:00', '12:50', '13:45', '14:35', '15:30', '16:20'));

**Prüfen Sie zudem, ob sich die Datenbank um einen Constraint ergänzen lässt, der in Widerspruch zu bereits hinterlegten Daten steht.**

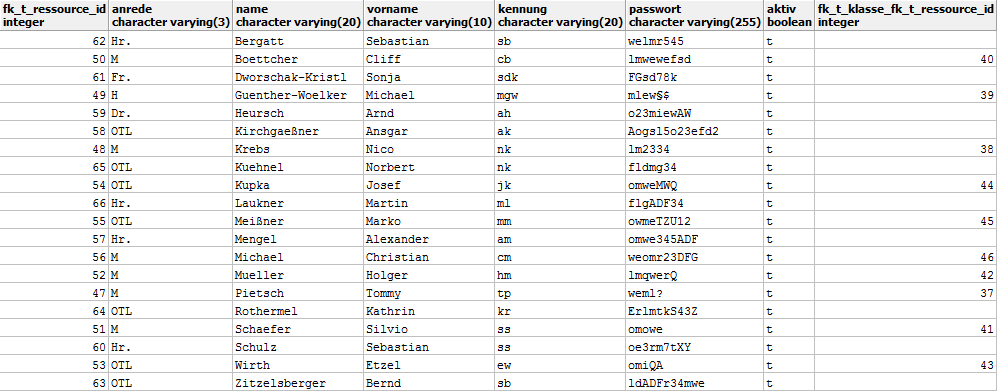
Dies ist nicht möglich.

**Freitag ist eine Stunde eingetragen, die nach Routinestundenplan gar nicht existiert. Löschen Sie sie!**

delete from t\_stunde where wochentag = 'Fr' and beginn = '11:15';

**Lassen Sie sich sämtliche Daten zu Lehrern anzeigen. Sortieren Sie dabei aufsteigend nach Nachnamen, sowie, bei identischen Nachname, absteigend nach zugehöriger RessourcenID.**

**Ergebnis:**

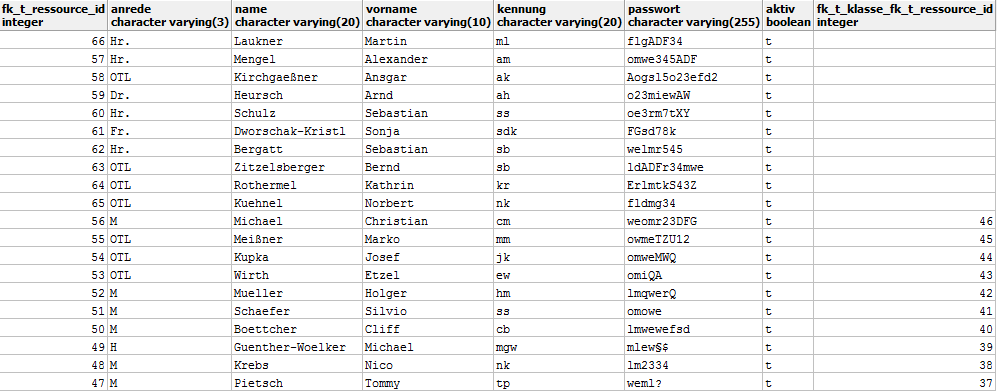


select \*

from t\_Lehrer

order by name ASC, fk\_t\_ressource\_id DESC;

**Ermitteln Sie sämtliche Informationen zu Lehrern, sortieren Sie dabei absteigend nach der Identifikationsnummer der zugehörigen Klasse. Lehrer, die keine Klassenlehrer sind sollen zuerst angezeigt werden.**



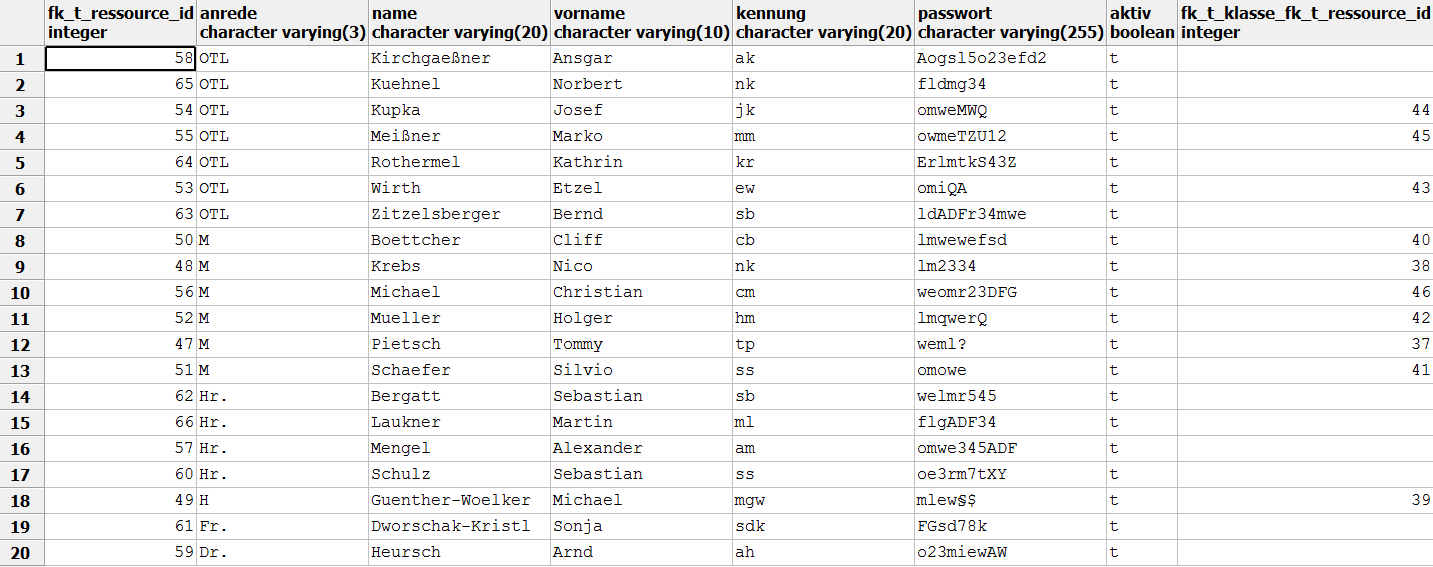
select \*

from t\_Lehrer

order by fk\_t\_klasse\_fk\_t\_ressource\_id DESC NULLS FIRST ;

**Ermitteln Sie sämtliche Informationen zu Lehrern, sortieren Sie dabei absteigend nach der Anrede, aufsteigend nach dem Nachnamen sowie aufsteigend nach dem Vornamen.**

Ergebnis:



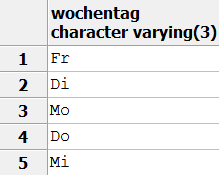
select \*

from t\_Lehrer

order by anrede DESC, name ASC, vorname ASC;

**Lassen Sie sich sämtliche Wochentage einmal anzeigen.**

Ergebnis:

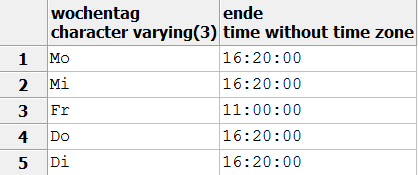


select distinct wochentag

from t\_Stunde;

**Lassen Sie sich zu jedem Wochentag die Uhrzeit anzeigen, zu der die letzte Stunde des jeweiligen Tages endet. Sortieren Sie die Wochentage absteigend. Arbeiten Sie ohne Gruppierung!**

Ergebnis:



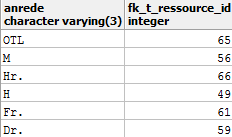
select distinct on(wochentag) wochentag, ende

from t\_Stunde

order by wochentag DESC;

**Ermitteln Sie zu jeder Personengruppe, basierend auf der Anrede, die jeweils höchste zugeordnete Ressourcen\_ID, geben Sie diese beiden Werte aus! Arbeiten Sie ohne Gruppierung!**

Ergebnis:



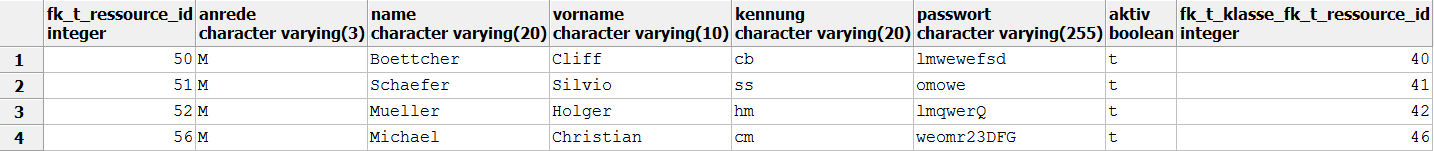
select distinct on (Anrede) Anrede, fk\_t\_ressource\_id

from t\_Lehrer

order by Anrede DESC, fk\_t\_ressource\_id DESC ;

**Ermitteln Sie sämtliche Informationen zu allen Lehrern, deren Ressourcen\_ID sich im geschlossenen Intervall [50, 60] befindet. Zudem müssen Sie den Dienstgrad eines Majors besitzen.**

**Ergebnis:**

****

select \*

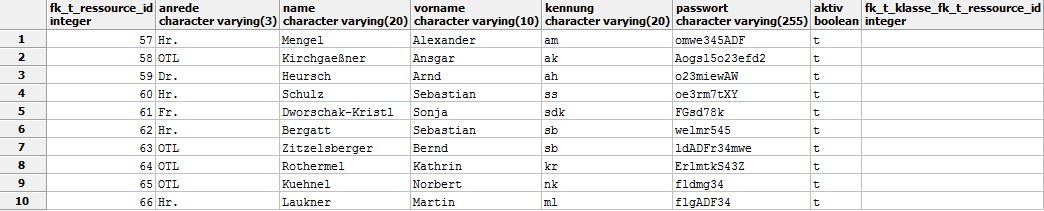
from t\_Lehrer

where fk\_t\_ressource\_id between 50 and 60 and

anrede = 'M';

**Ermitteln Sie sämtliche Informationen zu allen Lehrern, die keine Klassenlehrer sind. Zudem müssen sie entweder den Dienstgrad Oberstleutnant besitzen oder aber Zivilisten sein.**

**Ergebnis:**

****

select \*

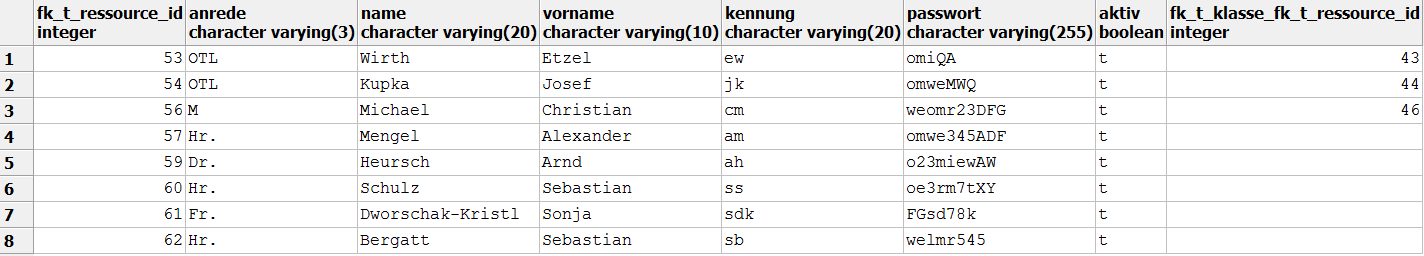
from t\_Lehrer

where fk\_t\_klasse\_fk\_t\_ressource\_id IS NULL and

(anrede = 'Hr.' or anrede = 'Fr.' or anrede = 'OTL' or anrede = 'Dr.');

**Sie kennen aus der technischen Informatik den Logikbaustein XOR, der für die Entweder-Oder-Verknüpfung steht. Ermittlen Sie basierend auf dieser Logik eine Abfrage, die Ihnen sämtliche Informationen zu allen Lehrkräften anzeigt, deren Ressourcen\_ID sich entweder im geschlossenen Intervall [55, 65] befindet, oder die den Dienstgrad Oberstleutnant besitzen.**

Ergebnis:

****

select \*

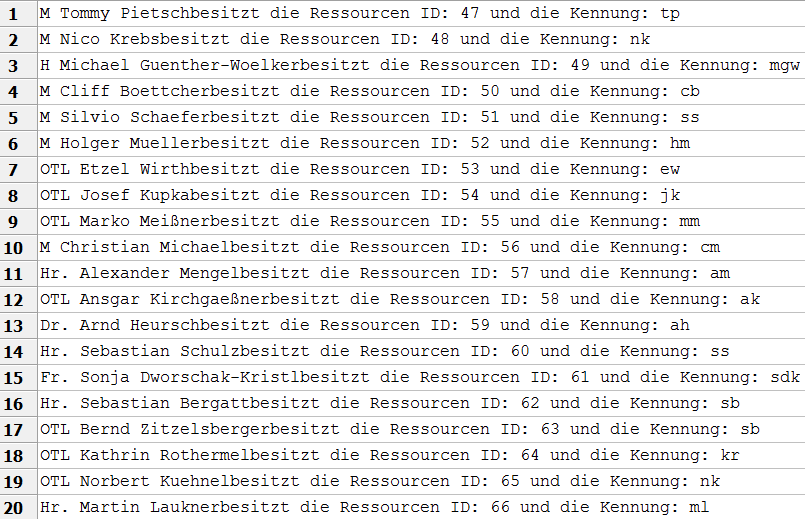
from t\_Lehrer

where (fk\_t\_ressource\_id >= 55 AND fk\_t\_ressource\_id <= 65 AND anrede != 'OTL')

OR

(anrede = 'OTL' AND (fk\_t\_ressource\_id < 55 OR fk\_t\_ressource\_id > 65));

**Erzeugen Sie die nachfolgende Ausgabe:**

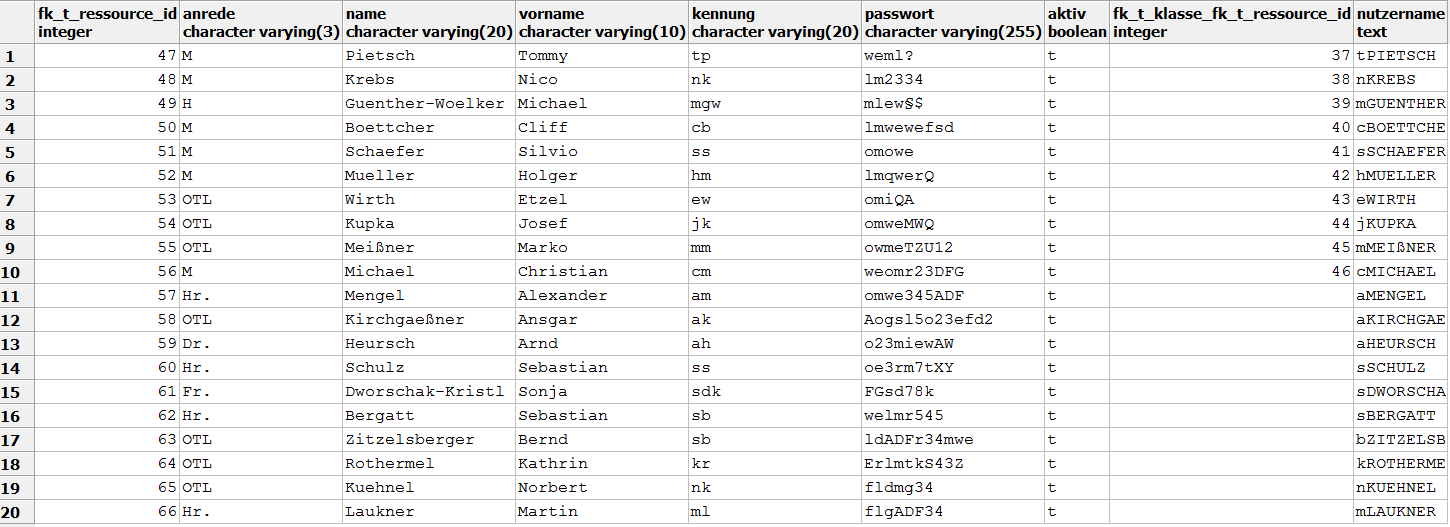
****

**select concat(anrede, ' ',vorname, ' ',name, 'besitzt die Ressourcen ID: ',  
fk\_t\_ressource\_id, ' und die Kennung: ', kennung)**

**from t\_Lehrer;**

**Lassen Sie sich sämtliche Informationen zu allen Lehrkräften ausgeben. Erzeugen Sie zudem mittels der Abfrage eine neue Spalte *nutzername*, mit den abgebildeten Inhalten und der Formattierung:**

**Ergebnis:**

****

**select \*, concat(lower(left(vorname,1)), upper(name)) as Nutzername**

**from t\_Lehrer;**

**Ermitteln Sie die Anzahl sämtlicher Ressourcen\_Ids, die auf Orte sowie Lehrer entfallen.**

Ergebnis:



select count(\*) as "Anzahl aller Ressourcen IDs von Orten und Lehrern: "

from t\_ressource

where id in (select fk\_t\_ressource\_id from t\_Ort) OR

id in (select fk\_t\_ressource\_id from t\_Lehrer);

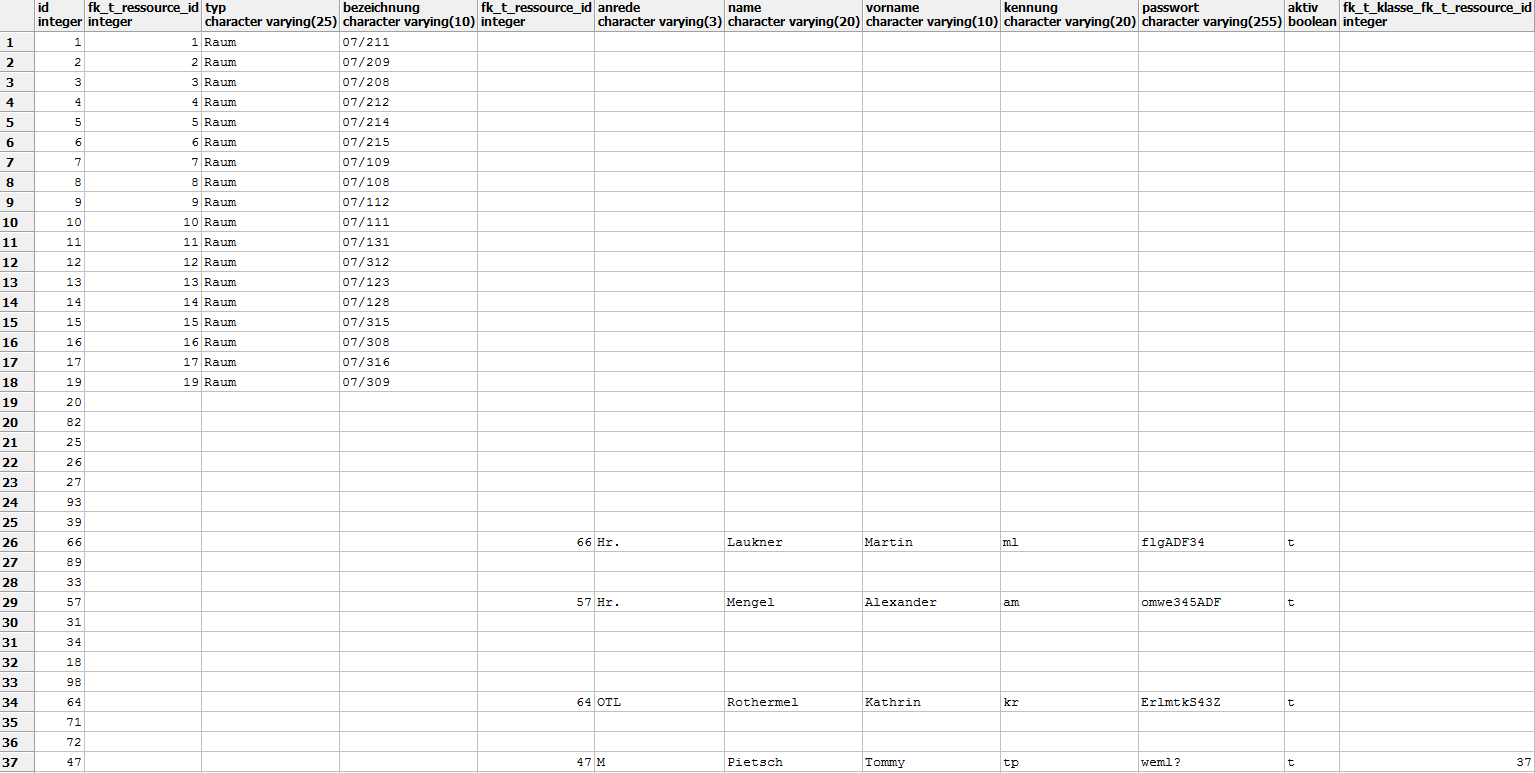
*Alternativ:*

select (select count(fk\_t\_ressource\_id) from t\_Lehrer) +

(select count(fk\_t\_ressource\_id) from t\_Ort)

**Ermitteln Sie sämtliche Ressourcen IDs, sowie die zugehörigen weiteren Informationen, falls es sich bei den Ressourcen um Orte oder Lehrer handelt.**

Ergebnis (Ausschnitt):



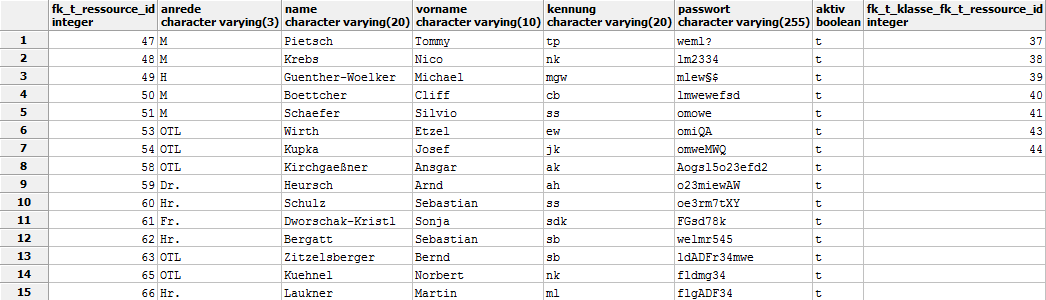
select \*

from (t\_ressource left join t\_ort as t1 on t1.fk\_t\_ressource\_id = id)

left join t\_Lehrer as t2 on t2.fk\_t\_ressource\_id = id;

**Ermitteln Sie Informationen zu allen Lehrer, in deren Nachnamen der Buchstabe ‚m‘ bzw. ‚M‘ nicht vorkommt.**

Ergebnis:



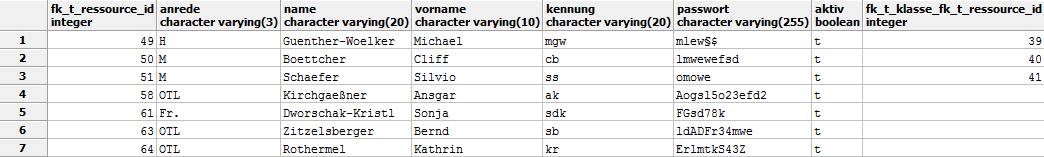
select \*

from t\_Lehrer

where name !~\* 'm';

**Ermitteln Sie Informationen zu allen Lehrer, deren Nachnamen wenigstens acht Zeichen umfassen.**

Ergebnis:



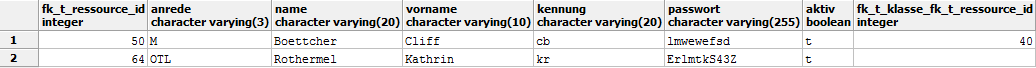
select \*

from t\_Lehrer

where name ~\* '.{8}';

**Ermitteln Sie Informationen zu allen Lehrer, deren Nachnamen aus genau 9 Zeichen bestehen.**

Ergebnis:



select \*

from t\_Lehrer

where name ~\* '^.{9}$';

**Ermitteln Sie Informationen zu allen Lehrer, deren Nachnamen aus wenigstens sieben Zeichen besteht, es können auch mehr sein. Es ist sicherzustellen, dass sämtliche Zeichen Buchstaben des Alphabets inklusive Umlauten sowie Eszett sein müssen.**

Ergebnis:



select \*

from t\_Lehrer

where name ~\* '^[a-zäöüß]{7}[a-zäöüß]\*$';

**Ermitteln Sie Informationen zu dem Lehrer, dessen mit dem MD5-Algorithmus gehashtes Passwort *3a29c08f700c1e95242dd458c28321ab* lautet.**

Ergebnis:



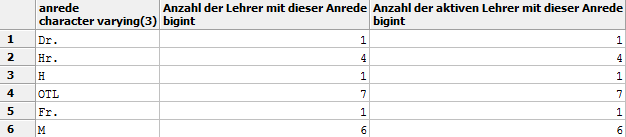
select \*, md5(passwort)

from t\_Lehrer

where md5(passwort) = '3a29c08f700c1e95242dd458c28321ab';

**Ermitteln Sie für alle Lehrer, basierend auf ihrer jeweiligen Anrede, wie viele Lehrer mit dieser Anrede aktuell in der DB existieren sowie wie viele davon aktuell im aktiven Einsatz an der Fachschule sind.**

Ergebnis:



select anrede, count(\*) as "Anzahl der Lehrer mit dieser Anrede",

count(aktiv) as "Anzahl der aktiven Lehrer mit dieser Anrede"

from t\_Lehrer

group by anrede;

**Ermitteln Sie die prozentualen Anteile, die die Lehrer, die Klassen, die Materialien sowie die Orte gemessen an den gesamten Ressourcen (also alle tatsächliche existierenden Ressourcen\_IDs) ausmachen.**

Ergebnis:



select (select count(\*) from t\_Lehrer)::float / count(\*) as "Prozentualer Anteil der Lehrer an den gesamten Ressourcen",

(select count(\*) from t\_Klasse)::float / count(\*) as "Prozentualer Anteil der Klassen an den gesamten Ressourcen",

(select count(\*) from t\_Material)::float / count(\*) as "Prozentualer Anteil des Materials an den gesamten Ressourcen",

(select count(\*) from t\_Ort)::float / count(\*) as "Prozentualer Anteil der Orte an den gesamten Ressourcen"

from t\_ressource;

**Wie das obige Resultat zeigt, ergeben die Anteile der Lehrer, Klassen, Materialien sowie Orte zusammengenommen weniger als 1 bzw 100%. Modifizieren Sie Ihre obige Abfrage so, dass die prozentualen Anteile der Lehrer, Klassen, Materialen und Orte gegenüber der Summe der Lehrer, Klassen, Materialien und Orte angezeigt wird. So sollte auch die Summe der prozentualen Anteile 1 bzw. 100% ergeben. Runden Sie auf zwei Nachkommastellen.**

Ergebnis:



select round((select count(\*) from t\_Lehrer)::numeric(4,2) /   
(select (select count(\*) from t\_Lehrer) + (select count(\*) from t\_Klasse) + (select count(\*) from t\_Material) + (select count(\*) from t\_Ort) ), 2)  
as "Prozentualer Anteil der Lehrer an den gesamten Ressourcen",  
  
round((select count(\*) from t\_Klasse)::numeric(4,2) /   
 (select (select count(\*) from t\_Lehrer) + (select count(\*) from t\_Klasse) + (select count(\*) from t\_Material) + (select count(\*) from t\_Ort) ), 2)

"Prozentualer Anteil der Klassen an den gesamten Ressourcen",

round((select count(\*) from t\_Material)::numeric(4,2) /   
(select (select count(\*) from t\_Lehrer) + (select count(\*) from t\_Klasse) + (select count(\*) from t\_Material) + (select count(\*) from t\_Ort) ), 2)

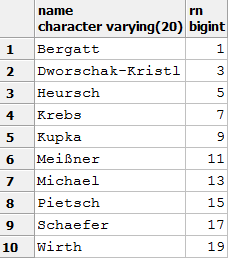
"Prozentualer Anteil des Materials an den gesamten Ressourcen",

round((select count(\*) from t\_Ort)::numeric(4,2) /   
 (select (select count(\*) from t\_Lehrer) + (select count(\*) from t\_Klasse) + (select count(\*) from t\_Material) + (select count(\*) from t\_Ort) ), 2)

"Prozentualer Anteil der Orte an den gesamten Ressourcen"

**Erstellen Sie eine Abfrage, in der die Nachnamen der Lehrer alphabetisch aufsteigend sortiert werden sollen. Es soll allerdings lediglich jeder zweite, „ungerade“ Nachname ausgegeben werden. Also der 1. Name, danach der 3. Name usw… Wir nehmen für diese Aufgabe an, dass sich alle Nachnamen voneinander unterscheiden. Hinweis: Modulo-Operator.**

Ergebnis:



select name, rn

from (select a.name, (select count(\*) from t\_Lehrer b where b.name <= a.name) rn

from t\_Lehrer a

) x

where x.rn % 2 = 1

order by name ASC;