

Übung zur Vorlesung *Grundlagen: Datenbanken* im WS21/22

Michael Jungmair, Josef Schmeißer, Moritz Sichert, Lukas Vogel (gdb@in.tum.de)
<https://db.in.tum.de/teaching/ws2122/grundlagen/>

Blatt Nr. 10

Hausaufgabe 1

Fügen Sie die folgenden Tupel in eine anfangs leere erweiterbare Hashtabelle, welche 2 Einträge pro Bucket aufnehmen kann, ein. Dabei soll die Matrikelnummer als Suchschlüssel verwendet werden.

MatrNr	Name
2	Müller
8	Schmidt
19	Fischer
16	Huber
20	Bauer
34	Schneider
30	Wagner

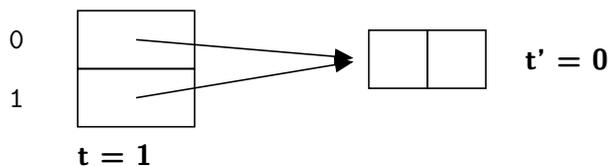
Verwenden Sie als Hashverfahren die inverse binäre Repräsentation der Matrikelnummer, wie in der Vorlesung beschrieben.

Lösung:

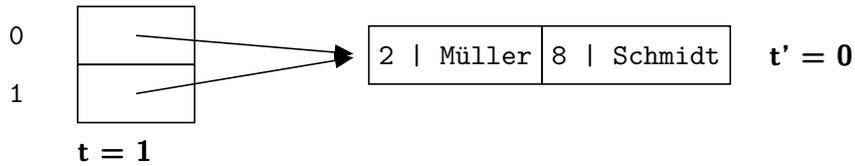
Zunächst errechnen wir den Hashwert für alle Studierende: Die inverse binäre Repräsentation der Matrikelnummer.

MatrNr	Name	Binär	Invers Binär
2	Müller	000010	010000
8	Schmidt	001000	000100
19	Fischer	010011	110010
16	Huber	010000	000010
20	Bauer	010100	001010
34	Schneider	100010	010001
30	Wagner	011110	011110

Zunächst eine leere erweiterbare Hashtabelle:



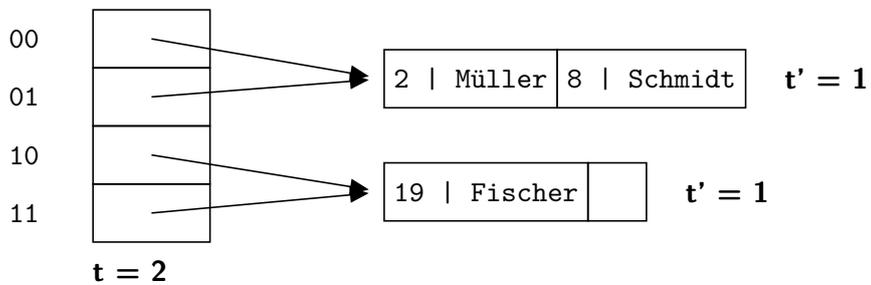
Wir fügen nun die ersten zwei Einträge ein, wonach die Hashtabelle wie folgt aussieht:



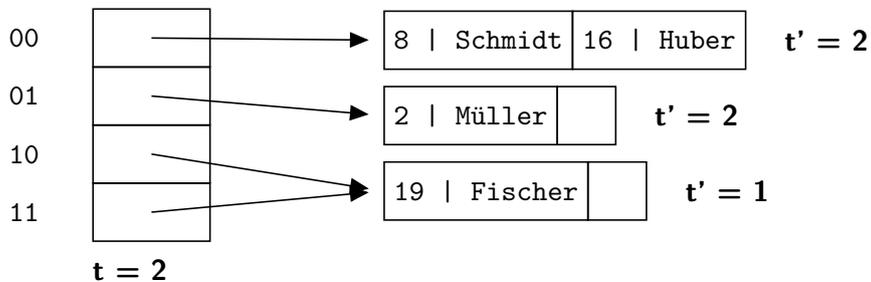
Der nächste Eintrag führt zu einem Überlauf. Da $t' < t$, können wir das Bucket teilen. Wir erhalten somit folgende Hashtabelle:



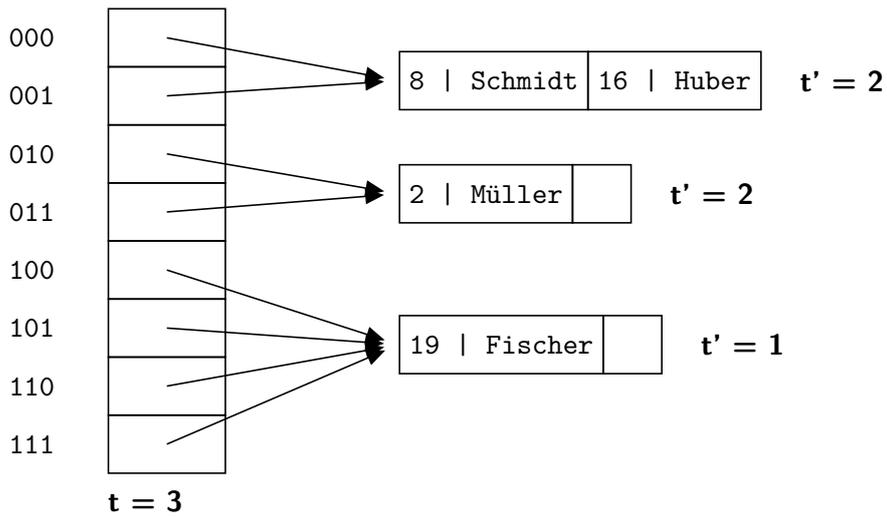
Der nächste Eintrag führt zu einem erneuten Überlauf! Leider in dem Bucket, für das $t = t'$ gilt. Wir müssen also die globale Tiefe erhöhen und erhalten somit folgende Hashtabelle:



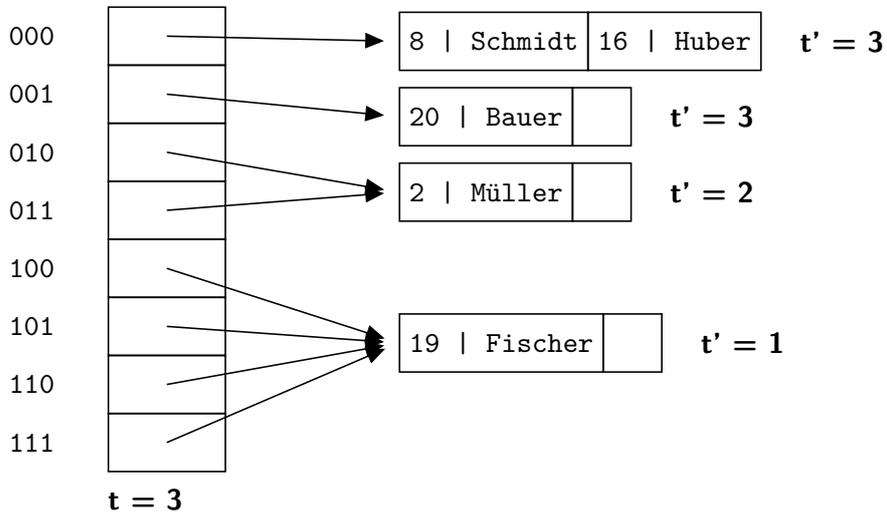
Nun können wir das erste Bucket splitten und unseren neuen Wert eintragen:



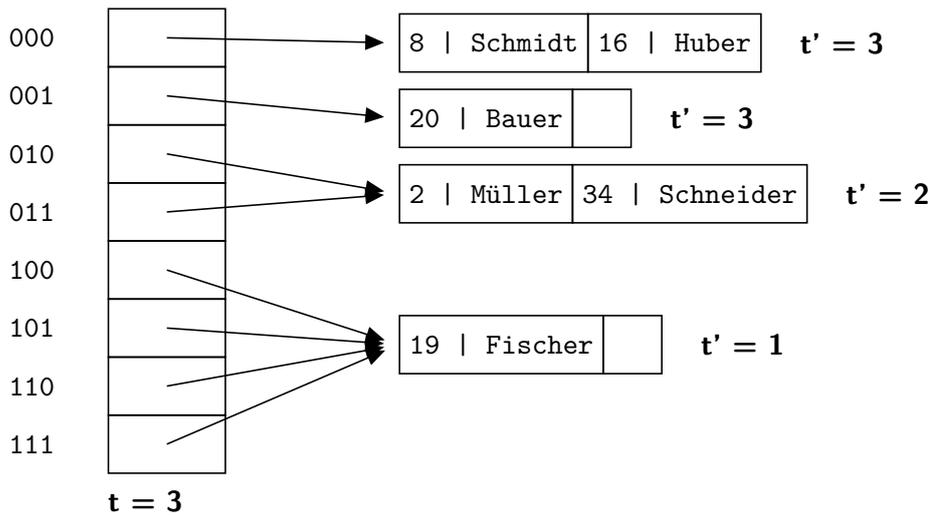
Bauer muss dem obersten Bucket hinzugefügt werden. Da dort kein Platz ist und $t = t'$ gilt, müssen wir erneut zuerst die globale Tiefe erhöhen.



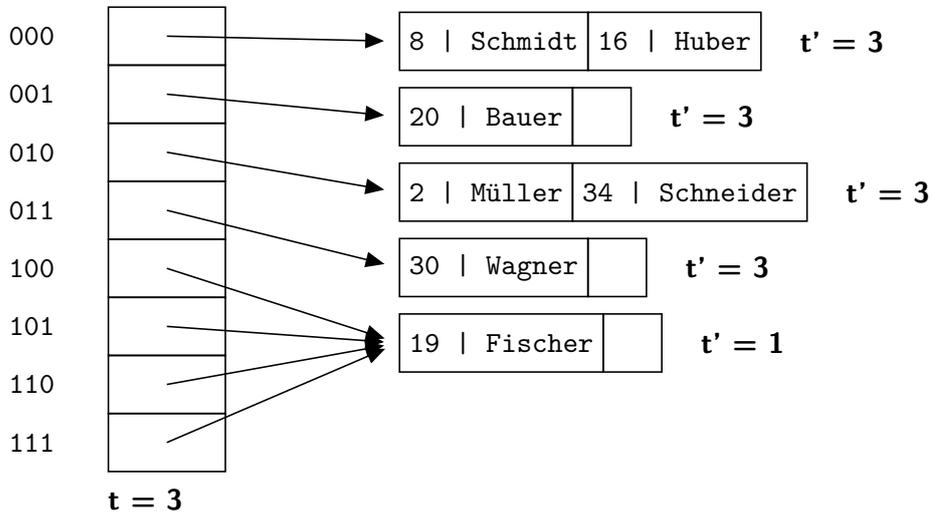
Wir können nun das oberste Bucket splitten, und Bauer hinzufügen:



Schneider passt in ein halbvolleres Bucket:



Für Wagner müssen wir zuletzt noch einmal die lokale Tiefe erhöhen.



Hausaufgabe 2

Fügen Sie in einen anfangs leeren R-Baum mit Knotenkapazität 4 folgende Datenpunkte nacheinander ein:

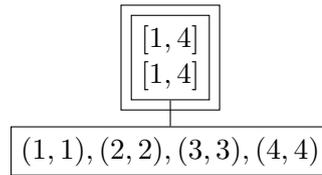
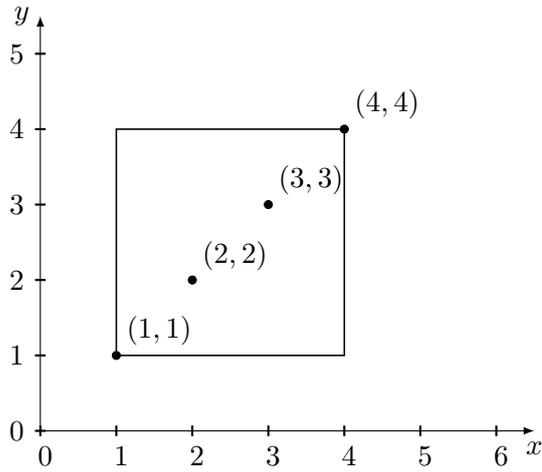
$$(1, 1), (2, 2), (3, 3), (4, 4), (5, 3), (5, 4), (6, 5), (4, 2)$$

Splitten Sie die Knoten dabei so, dass die summierte Fläche der durch den Split entstandenen Boxen möglichst klein ist.

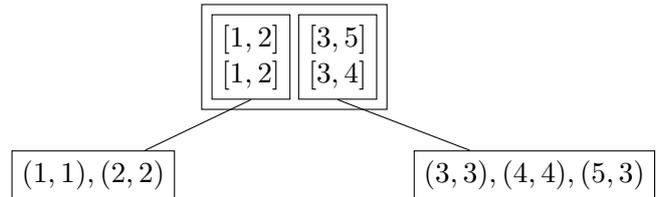
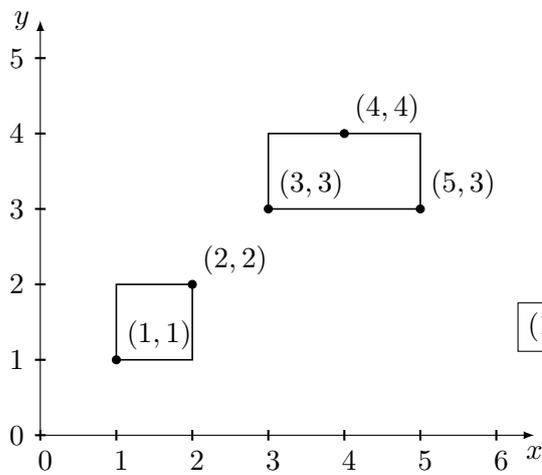
Illustrieren Sie die einzelnen Phasen im Aufbau des R-Baums. Zeichnen Sie hierzu den Baum und den Datenraum unmittelbar vor jedem Split und im Endzustand.

Lösung:

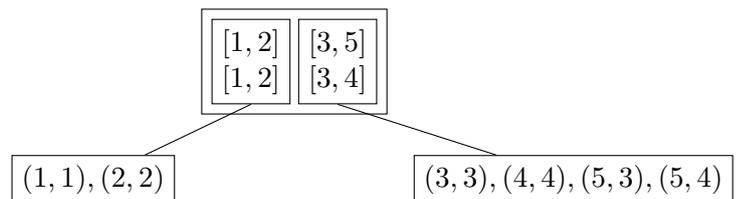
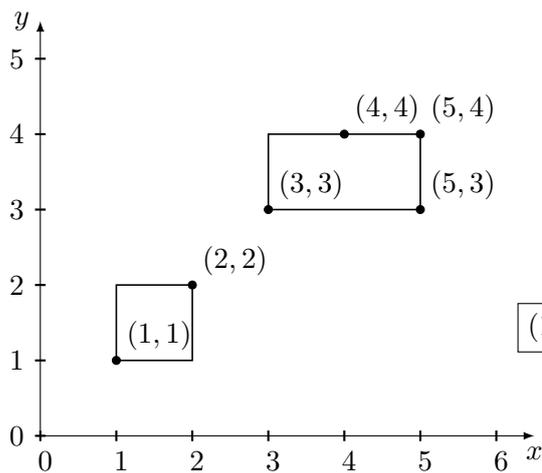
Wir fügen die Werte $(1, 1)$, $(2, 2)$, $(3, 3)$ und $(4, 4)$ in den ersten Knoten ein.



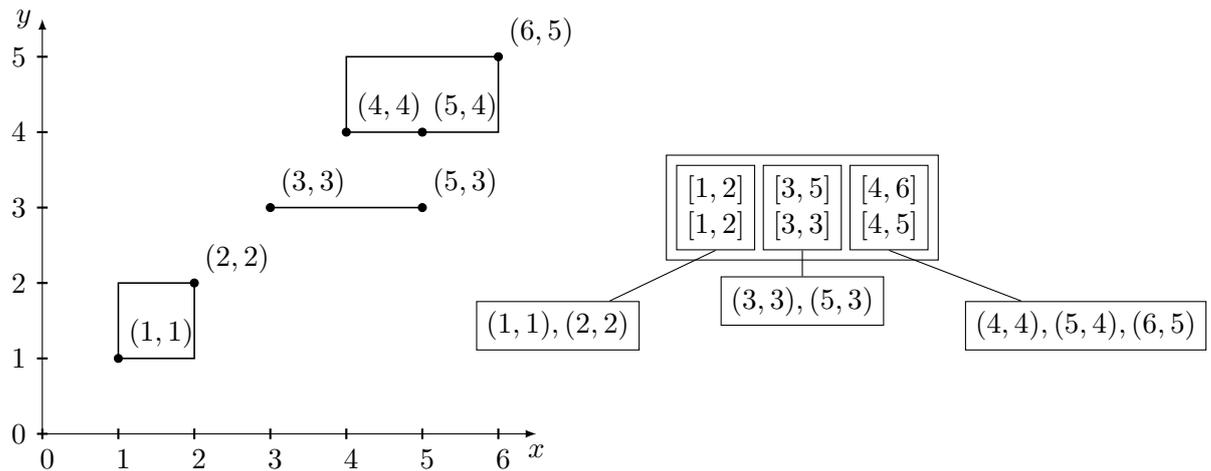
Da der Knoten nun voll ist, müssen wir ihn für den Wert $(5, 3)$ aufspalten. Um die Gesamtfläche gering zu halten, fügen wir den neuen Wert dann in das zweite Blatt ein:



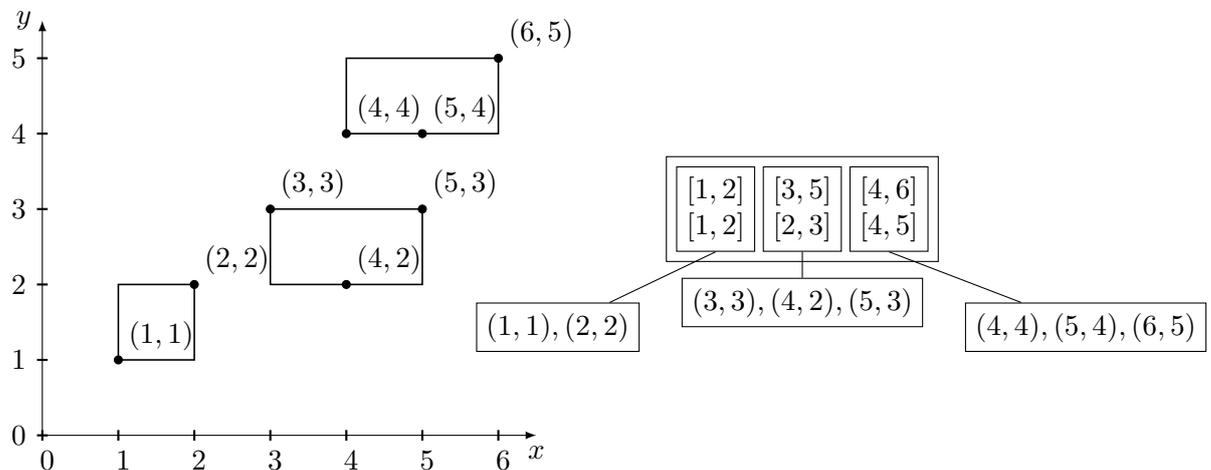
In dieses Blatt passt auch noch der Wert $(5, 4)$:



Vor Einfügen des Wertes $(6, 5)$ müssen wir nun den zweiten Knoten splitten. Um die Größe nach dem Split minimal zu halten, ordnen wir die Werte $(3, 3)$ und $(5, 3)$, sowie $(4, 4)$ und $(5, 4)$ jeweils einer Box mit der Fläche 0 zu und fügen schließlich $(6, 5)$ in die obere Box ein.



Für den Wert $(4, 2)$ gibt es zwei Möglichkeiten: Wir können den Wert entweder in die zweite Box mit der Größe 0, oder in die erste Box mit der Größe 1 einfügen. In beiden Fällen vergrößert sich die Gesamtfläche um 2. Wir entscheiden uns hier für die erste Option, die Alternative ist aber genauso richtig.



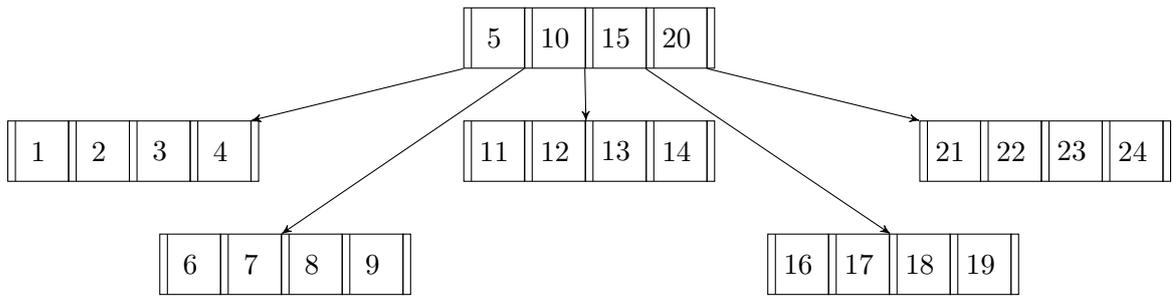
NB: Bei R-Bäumen ist es nicht immer eindeutig, wo neue Werte eingefügt werden müssen und wie man Knoten aufspaltet!

Hausaufgabe 3

Geben Sie eine Permutation der Zahlen 1 bis 24 an, so dass beim Einfügen dieser Zahlenfolge in einen (anfänglich leeren) B-Baum mit Grad $k = 2$ ein Baum minimaler Höhe entsteht. Skizzieren Sie den finalen Baum.

Lösung:

Damit der Baum eine minimale Höhe erreicht, muss der Füllgrad aller Knoten möglichst hoch sein. Bei $k = 2$ passen vier Elemente in jeden Knoten. Bei 24 einzufügenden Zahlen bedeutet das also, dass ein minimaler Baum insgesamt aus $\frac{24}{4} = 6$ Knoten besteht, die alle komplett gefüllt sind. Aufgrund der Invarianzen eines B-Baums muss dies ein Baum der Höhe 2 sein. Der einzig mögliche minimale Baum ist der folgende:



Damit die Zahlen 5, 10, 15 und 20 in die Wurzel geschrieben werden, muss die Einfügereihenfolge so gewählt werden, dass beim Teilen eines Knotens wegen Überlauf genau diese vier Zahlen als mittlere Elemente verwendet werden. Außerdem muss darauf geachtet werden, dass die Knoten nach dem Aufteilen wieder voll gefüllt werden.

Beispielsweise kann beim Einfügen mit den Zahlen 1, 2, 5, 6 und 7 in dieser Reihenfolge angefangen werden, wodurch die 5 dann in die Wurzel geschoben wird. Um die 10 in die Wurzel zu schreiben, können dann die Zahlen 10, 11, 12 eingefügt werden. Für die 15 dementsprechend die Zahlen 15, 16, 17 sowie für die 20 die Zahlen 20, 21, 22. Die übrigen Zahlen können dann in beliebiger Reihenfolge eingefügt werden.

Eine mögliche Permutation der Einfügereihenfolge, um einen Baum minimaler Höhe zu erhalten ist also: 1, 2, 5, 6, 7, 10, 11, 12, 15, 16, 17, 20, 21, 22, 3, 4, 8, 9, 13, 14, 18, 19, 23, 24.

Hausaufgabe 4

Klausuraufgabe aus der Erstklausur WiSe 2020/21.

Gegeben sei ein erweitertes Universitätsschema mit den folgenden zusätzlichen Relationen *StudentenGF* und *ProfessorenF*:

StudentenGF : {[MatrNr : integer, Name : varchar(20), Semester : integer, Geschlecht : char, Fakultaet : varchar(20)]}

ProfessorenF : {[PersNr : integer, Name : varchar(20), Rang : char(2), Raum : integer, Fakultaet : varchar(20)]}

Die erweiterten Tabellen sind auch in der Webschnittstelle angelegt.

Finden Sie das „einfachste“ Nebenfach. Das „einfachste“ Nebenfach ist die Fakultät, deren Fremdprüfungen die beste Durchschnittsnote haben. Eine Prüfung ist eine Fremdprüfung, wenn sie von einem Prüfling von einer anderen Fakultät als der des/der Prüfers/Prüferin abgelegt wurde. Fakultäten ohne Fremdprüfungen müssen nicht beachtet werden.

Lösung:

```
with
Fakultaeten(name) as (
  select s.Fakultaet as name from StudentenGF s
  union
  select p.Fakultaet as name from ProfessorenF p
),
fremdprüfungenSchnitt(Fakultaet, Schnitt) as (
  select profs.Fakultaet, avg(p.Note)
```

```
from pruefen p, StudentenGF s, ProfessorenF profs
where p.Matrnr = s.Matrnr
and profs.PersNr = p.persnr
and profs.Fakultaet <> s.Fakultaet
group by profs.Fakultaet
)
```

```
select f.name, fn.Schnitt
from fremdpruefungenSchnitt fn, Fakultaeten f
where fn.Fakultaet = f.name
and Schnitt = (select min(Schnitt) from fremdpruefungenSchnitt);
```

oder mit where not exists...