



Übung zur Vorlesung *Grundlagen: Datenbanken* im WS16/17

Harald Lang, Linnea Passing (gdb@in.tum.de)

<http://www-db.in.tum.de/teaching/ws1617/grundlagen/>

Blatt Nr. 08

Tool zum Üben der relationalen Algebra:

<http://db.in.tum.de/people/sites/muehe/ira/>

Tool zum Üben von SQL-Anfragen:

<http://hyper-db.com/interface.html>

Hausaufgabe 1

(Auszug aus der Klausur WiSe/2014)

Gegeben sei das bekannte Uni-Schema sowie folgende zusätzliche Tabelle **GdbPunkte** welche die wöchentliche Übungsteilnahme modelliert:

MatrNr	Gruppe	Woche	Punkte
24002	1	1	1
24002	1	2	0
24002	3	3	1
24002	3	4	0
26120	2	1	0
26120	2	2	0
26120	2	4	2
29555	1	1	2
29555	2	2	0
29555	3	3	2

Gehen Sie davon aus, dass es keine Wochen gibt, in denen kein Student anwesend war (d.h. in diesem Beispiel gab es genau 4 Übungswochen). Weiterhin können Sie davon ausgehen, dass jede Gruppe in jeder Woche mindestens einen Teilnehmer hatte. Betrachten Sie bei jeder Aufgabe nur Studenten, die mindestens an einer Übung teilgenommen haben.

- a) Geben Sie für die Relation **GdbPunkte** einen Kandidatenschlüssel an und begründen Sie Ihre Wahl (max 2 Sätze).

$$\kappa = \{\text{MatrNr}, \text{Gruppe}, \text{Woche}\}$$

Geben Sie jeweils ein SQL-92 Statement an, welches die Frage beantwortet. Ihre Ausgabe muss auch funktionieren, wenn die Ausprägung der Relationen anders ist als hier gezeigt.

- b) Wie viele Punkte wurden insgesamt vergeben? Geben Sie ein Tupel mit einem Attribut mit dem Namen **Punktsumme** aus.

Ergebnis für die Beispielausprägung:	
Punktsumme	8

```
SELECT SUM(Punkte) Punktsumme FROM GdbPunkte;
```

- c) Wie viele Punkte wurden im Durchschnitt pro Woche in den jeweiligen Gruppen vergeben? Geben Sie pro Gruppe ein Tupel mit den Attributen **Gruppe** und **PunkteProWoche** zurück. Sortieren Sie die Ausgabe absteigend nach **PunkteProWoche**.

Ergebnis für die Beispielausprägung:

Gruppe	PunkteProWoche
3	1.25
1	0.75
2	0.5

```
select p.gruppe,
(1.0*sum(p.punkte))/(select count(distinct woche)
from gdbPunkte) as punkteProWoche
from gdbPunkte p
group by p.gruppe
order by punkteProWoche desc
```

alternativ:

```
select p.gruppe,
(1.0*sum(p.punkte))/count(distinct woche) as punkteProWoche
% da jede Gruppe in jeder Woche stattfindet
from gdbPunkte p
group by p.gruppe
order by punkteProWoche desc
```

- d) Welche Studenten haben an mehr als 2 verschiedenen Gruppen teilgenommen? Geben Sie pro Student ein Tupel mit den Attributen **MatrNr**, **Name** und **AnzahlGruppen** aus.

Ergebnis für die Beispielausprägung:

MatrNr	Name	AnzahlGruppen
29555	Feuerbach	3

```
select p.MatrNr, s.Name, count(DISTINCT p.Gruppe) as AnzahlGruppen
from Studenten s, GdbPunkte p
where s.MatrNr = p.MatrNr group by p.matrnr, s.Name
having count(DISTINCT p.Gruppe) > 2;
```

- e) Aus welchen Gruppen hat nie ein Student herausgewechselt?

Ergebnis für die Beispielausprägung:

Gruppe
3

```
with verlasseneGruppen as (
select DISTINCT p1.Gruppe as Gruppe
from GdbPunkte p1
where exists (select *
from GdbPunkte p2
where p1.MatrNr = p2.MatrNr
and p1.Woche < p2.Woche
and p1.Gruppe <> p2.Gruppe)
)
select DISTINCT p.Gruppe
from GdbPunkte p
where not exists (select * from verlasseneGruppen v where v.Gruppe = p.Gruppe)
```

f) In welchen Gruppen hat nie ein Student gefehlt?

Ergebnis für die Beispielausprägung:

Gruppe

keine Ausgabetafel

```
with
GruppeWochen as (
  select distinct gruppe, (select count(distinct woche) from gdbpunkte) as wochen
  from gdbpunkte),
nichtDa as (
  select s.gruppe, s.matrnr
  from gdbpunkte s, GruppeWochen gw
  where s.gruppe=gw.gruppe
  group by s.gruppe,s.matrnr,gw.wochen
  having count(*)<gw.wochen)
```

```
select distinct gruppe
  from gdbpunkte p where gruppe not in (select distinct gruppe from nichtDa)
```

alternativ ohne with:

```
select distinct gruppe
  from gdbpunkte p1
  where not exists(select * from gdbpunkte p2
                  where p1.gruppe=p2.gruppe
                  and exists(select * from gdbpunkte p3
                            where p1.gruppe=p3.gruppe
                            and not exists(select * from gdbpunkte p4
                                           where p1.gruppe=p4.gruppe
                                           and p4.matrnr=p2.matrnr
                                           and p4.woche=p3.woche
                                           )
                            )
                  )
  )
```

Hausaufgabe 2

Gegeben seien folgende Informationen:

- Krankenhäuser bestehen aus Stationen. Zu jedem Krankenhaus ist dessen Adresse (wodurch es identifiziert werden kann) und die Anzahl an Betten bekannt. Stationen besitzen einen Namen, der nur innerhalb eines Krankenhauses eindeutig ist.
- Stationen bestehen wiederum aus Zimmern, welche nummeriert sind. Eine solche Nummer ist innerhalb einer Station eindeutig.
- Jedes Krankenhaus beschäftigt Personal. Dabei wird festgehalten, welches Gehalt bezahlt wird. Personal kann in verschiedenen Krankenhäusern beschäftigt sein.
- Personal ist durch eine Personal-Nummer gekennzeichnet und kann unter anderem in Ärzte und Krankenpfleger unterteilt werden. Zu einem Arzt ist sein Fachbereich bekannt und von welchen Krankenpflegern er Vorgesetzter ist. Kein Krankenpfleger kann zugleich Arzt sein, aber durchaus mehrere Vorgesetzte Ärzte haben.

- Eine Station kann von mehreren Ärzten geleitet werden. Ein Arzt kann ebenso mehrere Stationen leiten. Ausserdem ist bekannt, ob und in welchem Zimmer ein Arzt sein Büro hat. Kein Arzt muss sich sein Büro mit einem anderen Arzt teilen.
 - Personal arbeitet auf Stationen in Schichten. Eine Schicht kann über Datum und Zeitraum eindeutig identifiziert werden. Eine Person kann in einer Schicht auf nur einer Station arbeiten.
- a) Erstellen Sie für das oben genannte Szenario ein geeignetes ER-Diagramm. Verwenden Sie dabei – wenn angebracht – das Prinzip der Spezialisierung. Kennzeichnen Sie die Primärschlüssel der Entitätstypen, totale Teilnahmen und schwache Entitätstypen. Zeichnen Sie die Funktionalitäten der Beziehungstypen in das Diagramm ein.
- b) Überführen Sie Ihr in Aufgabe a) erstelltes Modell in ein verfeinertes relationales Schema. Kennzeichnen Sie die Schlüssel durch Unterschreiben. Datentypen müssen *nicht* angegeben werden.

Lösung:

n/a