

Übung zur Vorlesung *Datenbanksysteme I* im SS 20

Gabriel Haas (gabriel.haas@uni-jena.de)

dbis1.github.io/courses/ss20/db1/

Blatt Nr. 11

Abgabe auf moodle.uni-jena.de. Die Bedingungen für den Notenbonus finden Sie auf der Vorlesungswebsite.

Tool zum Üben von SQL-Anfragen: <https://hyper-db.com/interface.html>.

Hausaufgabe 1

- Was ist ein Equi-Join?
- Bei welchen Join-Prädikaten ($<$, $=$, $>$) kann man sinnvoll einen Hashjoin einsetzen?
- Gegeben die Relation $Profs = \{\underline{PersNr}, Name\}$ und $Raeume = \{\underline{PersNr}, RaumNr\}$.
 - Skizzieren Sie eine geschickte Möglichkeit, den Equi-Join $Profs \bowtie_{Profs.PersNr < Raeume.PersNr} Raeume$ durchzuführen.
 - In welchem Fall wäre selbst ein Ausdruck wie

$$Profs \bowtie_{Profs.PersNr < Raeume.PersNr} Raeume$$

effizient auswertbar?

- Der Student Maier hat einen Algorithmus gefunden, der den Ausdruck $A \times B$ in einer Laufzeit von $O(|A|)$ materialisiert. Was sagen Sie Herrn Maier?

Hausaufgabe 2

Gegeben sei die folgende Relation **Zehnkampf** mit Athletennamen und den von ihnen erreichten Punkten im Zehnkampf:

Name	Punkte
Eaton	8869
Suarez	8523
Behrenbruch	8126
Hardee	8671
...	...

- Ermitteln Sie die Goldmedaillengewinner in SQL. (Eine Goldmedaille bekommen alle Athleten, für die gilt: es gibt niemand besseren (also mit mehr Punkten).)
- Ermitteln Sie die Silbermedaillengewinner in SQL. (Eine Silbermedaille bekommen alle, für die gilt: es gibt genau eine/n bessere/n.)

HINWEIS: Beachten Sie, dass die Relation **Zehnkampf** in der oben genannten Webschnittstelle nicht existiert. Verwenden Sie die folgende Syntax um temporäre Relationen zu erzeugen:

```

with zehnkampf(name,punkte) as (
  values
    ('Eaton', 8869),
    ('Suarez', 8523),
    ('Behrenbruch', 8126),
    ('Hardee', 8671),
    ('Sebrle', 8869)
)
select * from zehnkampf order by punkte desc

```

Hausaufgabe 3

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

StudentenGF : $\{\{\underline{MatrNr} : integer, Name : varchar(20), Semester : integer,$
Geschlecht : char, *Fakultaet* : varchar(20)]}

ProfessorenF : $\{\{\underline{PersNr} : integer, Name : varchar(20), Rang : char(2),$
Raum : integer, *Fakultaet* : varchar(20)]}

Die erweiterten Tabellen sind auch in der Webschnittstelle angelegt.

- a) Ermitteln Sie den Männeranteil an den verschiedenen Fakultäten in SQL!
- b) Ermitteln Sie in SQL die Studenten, die alle Vorlesungen ihrer Fakultät hören. Geben Sie zwei Lösungen an, höchstens eine davon darf auf Abzählen basieren.