

Aufgabe 1: SQL Basics: Fußball-DB

(1 P.)

In dieser Aufgabe formulieren Sie SQL-Anfragen am Beispiel einer “Fußball-Datenbank”, die Vereine, Spiele, Spieler und Einsätze (Daten gelten für eine Saison) enthält.

VEREIN (VName, Ort, Praesident)

SPIEL (Heim, Gast, Resultat, Zuschauer, Termin, Spieltag)

SPIELER (SpNr, Name, Vorname, Verein, Gehalt, GebOrt)

EINSATZ (Heim, Gast, SpNr, VonMin, BisMin, Tore, Karte)

Folgende Fremdschlüssel-Beziehungen gelten:

- SPIELER.Verein ist Fremdschlüssel auf VEREIN.VName
- SPIEL.Heim und SPIEL.Gast sind jeweils Fremdschlüssel auf VEREIN.VName
- (EINSATZ.Heim, EINSATZ.Gast) ist Fremdschlüssel auf (SPIEL.Heim, SPIEL.Gast)
- EINSATZ.SpNr ist Fremdschlüssel auf SPIELER.SpNr

Formulieren Sie die Antworten auf folgende Anfragen in SQL. Wählen Sie für die Rückgabespalten sinnvolle Werte.

- Alle Spieler, die bei einem Verein in Kaiserslautern (KL) spielen, aufsteigend nach Gehalt sortiert.
- Alle Spieler, die bisher noch keine Tore geschossen haben.
- Die Gesamttore der Saison je Verein, absteigend sortiert.
- Diejenigen Spieler, die der FCK in der Saison mehr als 3 Mal eingesetzt hat.
- Wie heißen die Präsidenten von Vereinen, die zuhause mindestens ein Spiel mit 2:0 gewonnen haben? Listen Sie jeden Präsidenten nur einmal im Ergebnis auf.

Aufgabe 2: SQL anwenden mit PostgreSQL: Uni-DB

(1 P.)

Für diese Aufgabe ist es erforderlich, dass Sie PostgreSQL auf Ihrem Rechner installieren (alternativ ist PostgreSQL auf tux{1,3,5,6,7,8} des SCI installiert) und den vorgegebenen Daten-Dump des aus der Vorlesung bekannten Uni-Schemas (Link dazu auf der VL-Webseite) in die Datenbank laden. Wenn Sie PostgreSQL am SCI benutzen möchten, schreiben Sie zuvor eine Email an Manuel Hoffmann unter Angabe Ihres SCI-Benutzernamen. Falls Sie keinen SCI-Account besitzen, dann geben Sie Ihren RHRK-Accountnamen an.

Alle für diese Aufgabe erforderlichen Anfragen können und sollen Sie auf den gegebenen Daten ausprobieren.

- Führen Sie folgende Anfragen auf der Uni-Datenbank aus und geben Sie die Resultate an. Erklären Sie außerdem, was die Anfragen bedeuten. Eine ASCII-Datei mit den Anfragen finden Sie auf der VL-Webseite.

i)

```
SELECT DISTINCT s.MatrNr
FROM Studenten AS s, Professoren AS p, Assistenten AS a,
    hoeren AS h, Vorlesungen AS v
WHERE s.MatrNr=h.MatrNr AND v.VorlNr=h.VorlNr
    AND v.gelesenVon=p.PersNr AND a.Boss=p.PersNr AND a.Name='Platon'
    AND NOT EXISTS (
    SELECT *
    FROM pruefen AS pr
    WHERE pr.MatrNr=s.MatrNr AND pr.VorlNr=v.VorlNr
        AND pr.PersNr=p.PersNr
    );
```

ii)

```
SELECT p.Name, AVG(s.Semester)
FROM Studenten AS s, hoeren AS h, Vorlesungen AS v, Professoren AS p
WHERE s.MatrNr=h.MatrNr AND v.VorlNr=h.VorlNr AND p.PersNr=v.gelesenVon
GROUP BY p.Name
HAVING COUNT(s.MatrNr)>2;
```

iii)

```
SELECT p.Name,
    (v1.Titel || ' (' || v1.VorlNr || ')') as vorlA,
    (v2.Titel || ' (' || v2.VorlNr || ')') as vorlB,
    (v3.Titel || ' (' || v3.VorlNr || ')') as vorlC
FROM Professoren p, Vorlesungen v1, Vorlesungen v2, Vorlesungen v3,
    Voraussetzen va1, Voraussetzen va2
WHERE p.PersNr=v2.gelesenVon
    AND v1.VorlNr=va1.Vorgaenger AND va1.Nachfolger=v2.VorlNr
    AND va1.Nachfolger = va2.Vorgaenger
    AND va2.Nachfolger=v3.VorlNr;
```

iv)

```
SELECT s.MatrNr, s.semester, noteWSum.note/swsGes.sumSws AS noteAvg
FROM Studenten s, (
    SELECT p.MatrNr AS MatrNr, sum(v.sws) AS sumSws
    FROM Vorlesungen v, pruefen p
    WHERE p.VorlNr=v.VorlNr
    GROUP BY p.MatrNr
) AS swsGes,
(
    SELECT noteW.MatrNr AS MatrNr, sum(noteW.note) AS Note
    FROM (
        SELECT p.MatrNr AS MatrNr, p.note*v.sws AS Note
        FROM pruefen p, Vorlesungen v
        WHERE p.VorlNr=v.VorlNr
    ) AS noteW
    GROUP BY noteW.MatrNr
) AS noteWSum
WHERE swsGes.MatrNr=s.MatrNr and noteWSum.MatrNr=s.MatrNr
ORDER BY Note ASC;
```

- b) Formulieren Sie die folgenden Anfragen in SQL und geben Sie zusätzlich die Resultate aus der Uni-DB an.
- Welche Studenten (Name, Matrikelnummer) haben schon einmal eine Vorlesung gehört, die sich mit Ethik (oder einer spezielleren Wissenschaft) beschäftigt und die von einem C4-Professor gehalten wird, dessen Büro sich im 2. Geschoss ($200 \leq \text{Raumnummer} < 300$) befindet?
 - Gibt es einen Assistenten, dessen Fachgebiet mit “Gott” zu tun hat, der für Professor Augustinus oder Professor Sokrates arbeitet, und dessen Boss nur genau eine Vorlesung hält? Geben Sie “Ja” zurück, falls ein solches Tupel existiert und ein leeres Resultat sonst. Anmerkung: Mit `select distinct 'Ja' as antwort from` können Sie eine Ja Antwort generieren.

Aufgabe 3: Relationale Algebra

(1 P.)

- a) Angenommen die Relationen R und S haben n bzw. m Tupel mit $n, m \geq 1$. Geben Sie die minimale und maximale Größe des Resultates der folgenden algebraischen Ausdrücke an. Gehen Sie dabei bei allen Operatoren von Mengensemantiken aus.
- $\sigma_p(R)$ mit p als beliebiges Prädikat
 - $\pi_A(R)$ mit A als beliebige gültige und nichtleerer Attributmenge
 - $R \times S$
 - $R \bowtie_p S$ mit p als beliebiges Join-Prädikat
- b) Drücken Sie $R \bowtie S$ allein durch Kreuzprodukt, Selektion, Projektion und Mengenoperationen aus. Die Attribute einer Relation R können Sie in Ihrer Lösung als $Attr(R)$ bezeichnen.
- c) Die folgenden Ausdrücke sind korrekt für Mengen, aber einige von ihnen sind inkorrekt für Bags. Geben Sie für die Ausdrücke, die für Bags inkorrekt sind, jeweils ein Gegenbeispiel an. Beschreiben Sie für Ausdrücke, die auch für Bags korrekt sind, warum dies der Fall ist.
- $(R \cup S) = (S \cup R)$
 - $\sigma_{C \vee D}(R) = \sigma_C(R) \cup \sigma_D(R)$
 - $(S \cap T) \setminus R = S \cap (T \setminus R)$
 - $\pi_L(R \cup S) = \pi_L(R) \cup \pi_L(S)$