Datenbanken 2 Praktikum 1

Dokumentation zu Integritätsbedingungen, UDF und SQL-Anfragen

Von: Phillip Urban, Jurek Jesse und Feyza Durmaz

Matrikelnummern: 11131888, 11135480, 11133991

Referentielle Integritätsbedingungen

1. Jeder Dozent muss mindestens einer Vorlesung zugeordnet sein.

Mit dieser Bedingung wollen wir verhindern, dass man keinen Dozenten in der Datenbank erstellen kann, der kein Modul unterrichtet.

1. Wenn ein Angestellter einen Arbeitsplatz besetzt, und der Angestellte aus der DB gelöscht wird, wird der Eintrag mit **ON DELETE SET NULL** „gelöscht“. Der Arbeitsplatz wird von der Fakultät weiterhin angeboten, ist aber im Moment nicht besetzt.
2. Bei Löschen eines Studierenden wird der passende Eintrag in der Relationstabelle mit der Fakultät „Studium“ gelöscht. Jeder aktive Studierende besitzt daher einen Eintrag mit Studiengang und Fakultätsnummer. Um diese Bedingung durchzusetzen wird **ON DELETE CASCADE** genutzt.

UDF Funktionsweise mit Beispielen

Erläuterung: Mit Hilfe unserer UDF können wir die Übersicht des Notenspiegels einzelner Studenten vereinfacht darstellen.

Beispiel:

CREATE OR REPLACE FUNCTION get\_avg\_note ()

RETURNS TABLE (

matnr integer,

name varchar,

avg\_note numeric)

language plpgsql

AS $body$

BEGIN

return query **SELECT** student.matnr, student.name, **AVG**(note)

**AS** note

**FROM** prüfungsbesuch p2

**INNER JOIN** student **ON** student.matnr = p2.pnr

**GROUP BY** student.name, student.matnr

**ORDER BY** note **ASC**;

END; $body$;

Komplexe – und interessante –

SQL-Anfragen

1. Durchschnittliche Note mit Geburtsdatum der Studenten:

**SELECT** student.gebdatum, get\_avg\_note.avg\_Note

**FROM** get\_avg\_note()

**INNER JOIN** student **ON** student.matnr = get\_avg\_note.matnr;

1. Rang des Professors mit Anzahl der gehaltenen Vorlesungen:

**SELECT** professor.rang, **COUNT**(\*) **AS** AnzahlVorlesungen

**FROM** anwesenheit

**INNER JOIN** professor **ON** professor.personalnr = anwesenheit.pnr

**GROUP BY** professor.rang;

1. Anzahl der Studenten in einer Fakultät:

**SELECT** fakultät."name", **COUNT**(\*) **AS** AnzahlStudenten

**FROM** studium

**INNER JOIN** fakultät **ON** fakultät.faknr = studium.faknr

**GROUP BY** fakultät."name";

1. Anzahl von Studenten pro Vorlesung:

**SELECT** vorlesung.titel, **COUNT**(\*) **AS** teilnehmer

**FROM** anwesenheit

**INNER JOIN** vorlesung **ON** vorlesung.vorlnr = anwesenheit.vorlnr

**WHERE** anwesenheit.pnr **IN** (select matnr from student)

**GROUP BY** vorlesung.titel;

1. Durchschnittliche Anzahl der Semester pro Fakultät:

**SELECT** fakultät."name", **AVG**(person.semester) **AS** avgSemesterAnzahl

**FROM** anwesenheit **INNER** **JOIN** studium

**ON** studium.pnr = anwesenheit.pnr

**INNER** **JOIN** fakultät **ON** fakultät.faknr = studium.faknr

**INNER** **JOIN** person **ON** person.pnr = studium.pnr

**WHERE** anwesenheit.pnr **IN**

(**SELECT** matnr **FROM** student)

**GROUP** by fakultät."name"

**ORDER** by fakultät."name";