Prof. Dr. Stefan Brass
PD Dr. Alexander Hinneburg
Institut für Informatik
MLU Halle-Wittenberg

04.11.2024

Einführung in Datenbanken

— Übungsblatt 4 (Relationale Schemata) —

Ihre Lösungen laden Sie bitte in die Übungsplattform in StudIP hoch ([StudIP-Eintrag der Vorlesung], Reiter „Übungsplattform“).

Einsendeschluss ist Montag, der 11.11.2024, 18^{00}.

Hausaufgaben können einzeln oder in Zweier-Gruppen bearbeitet werden. Sie können die Gruppe für jede Aufgabe neu wählen.

Denken Sie daran, dass Sie bei jeder Aufgabe angeben müssen, ob Sie bereit sind, vorzurechnen:

- „`VORRECHNEN:0`“: Ich werde nicht zur Übung kommen.
- „`VORRECHNEN:1`“: Ich möchte diese Aufgabe nicht vorrechnen.
- „`VORRECHNEN:2`“: Ich möchte diese Aufgabe nur ungern vorrechnen.
- „`VORRECHNEN:3`“: Ich kann vorrechnen, lasse aber gern anderen den Vortritt.
- „`VORRECHNEN:4`“: Ich kann problemlos vorrechnen.
- „`VORRECHNEN:5`“: Ich möchte gerne, dass meine Abgabe besprochen wird.

Falls Sie als Gruppe abgeben, muss jedes Gruppenmitglied einzeln die Bereitschaft zum Vorrechnen erklären (`VORRECHNEN1:N` ist der Wert für den Studierenden, der die Aufgabe in die Übungsplattform hochgeladen hat, und `VORRECHNEN2:M` der Wert für den anderen Studierenden).

Vergessen Sie nicht, eventuell verwendete Quellen wie ChatGPT (oder auch die Lösung eines anderen Studierenden) anzugeben (mit der in Kapitel 0, Folien 22 bis 27 beschriebenen Codierung). Ohne Quellenangabe werden „zu ähnliche Lösungen“ als Plagiat behandelt. Mit Quellenangabe werden sie normal korrigiert. In der Klausur müssen Sie aber ähnliche Aufgaben ohne Hilfe lösen! Sie sind erwachsene Menschen und können sich ja denken, dass Sie nur durch eigene Beschäftigung mit den Aufgaben lernen.

Wir würden uns freuen, wenn Sie die für die jeweilige Aufgabe verwendete Zeit in Minuten in der Form „`ZEIT:N`“ angeben würden (bei Gruppenarbeit „`ZEIT1:N`“ und „`ZEIT2:M`“). Diese Angabe ist freiwillig.

Hinweis: SQL-Anfragen mit Syntaxfehlern werden automatisch mit 0 Punkten bewertet! Testen Sie also Ihre Anfragen (z.B. im Adminer).

Aufgabe 1 (6 Punkte)

Loggen Sie sich über das Adminer-Webinterface bei der PostgreSQL-Datenbank für diese Übungen ein:

[https://dbs.informatik.uni-halle.de/edb?pgsql=db&
username=student_gast&db=postgres&ns=komponist_public]

Das Passwort finden Sie in StudIP-Eintrag der Vorlesung, Reiter „Informationen“. Diese Aufgabe bezieht sich auf das Schema „`komponist_public`“. Wir haben auf dem zweiten Übungsblatt schon die Tabelle `KOMPONIST` verwendet. Das Schema dieser Tabelle in der Kurznotation der Vorlesung ist:

`KOMPONIST(`<u>`KNr`</u>`, Name, Vorname°, geboren°, gestorben°).`

In der Variante nur mit ASCII-Zeichen ist es:

```
KOMPONIST(#KNr, Name, Vorname?, geboren?, gestorben?).
```

Man kann darüber streiten, ob es angemessen ist, Nullwerte in der Spalte „`Vorname`“ zu erlauben. In den Daten kommt es nicht vor. Es gibt aber nicht in allen Kulturen Vornamen nach deutschem Muster. Die Datenbank enthält teils auch relativ alte Musikstücke von kaum bekannten Komponisten — wahrscheinlich war befürchtet worden, dass die Trennung von Namen in Vor- und Nachname nicht immer möglich ist.

Bei der Spalte „`gestorben`“ braucht man Nullwerte sicher für noch lebende Komponisten. Es ist außerdem auch plausibel, dass man eventuell nicht von jedem Komponisten die Lebensdaten ermitteln kann.

Ihre Aufgabe ist, das Schema der übrigen vier Tabellen dieser Datenbank herauszufinden (`Stueck`, `CD`, `Aufnahme`, `Solist`), und in dieser Kurznotation aufzuschreiben (entsprechend Kap. 6 der Vorlesung). Wählen Sie dafür am besten die ASCII-Variante der Notation und geben Sie das Schema als `.txt`-Datei ab.

Wenn Sie im Adminer auf den Tabellennamen links klicken, wird das Schema der jeweiligen Tabelle angezeigt. Die Datentypen der Spalten sind für diese Aufgabe nicht relevant. Schlüssel und Fremdschlüssel müssen Sie aber angeben. Schlüssel finden Sie unter „Indizes“, was in zweierlei Hinsicht problematisch ist:

- „Index“ ist ein Konzept des internen Schemas. Es wird u.a. benutzt, um einen Schlüssel zu implementieren. Auf Ebene des relationalen Modells interessieren wir uns nur für Schlüssel.
- In dieser Vorlesung und vermutlich der Mehrheit der Lehrbücher wird „Indexe“ als Plural der Datenstruktur benutzt. Dagegen herrscht Einigkeit darüber, dass in $f(x_i, y_j)$ die Zahlen i und j „Indizes“ sind.

Die Groß-/Kleinschreibung von Tabellen- und Spaltennamen ist egal. Der Adminer zeigt die Namen aus dem Systemkatalog von PostgreSQL an, und PostgreSQL konvertiert alle Namen in Kleinbuchstaben (außer bei Verwendung von `"..."`).

Aufgabe 2 (10 Punkte)

Es soll eine Datenbank mit allen Modulen eines Studiengangs erstellt werden, die auch die Voraussetzungen enthält. Ihre Aufgabe ist es, `CREATE TABLE`-Anweisungen für die beiden Tabellen dieser Datenbank zu erstellen.

Die Tabelle „`Module`“ sieht so aus:

Module						
ID	Titel	LP	Art	Fach	Sem	Dauer
INF.00677.09	Objektorientierte Programmierung	5	P		1	1
MAT.02372.02	Mathematik B	15	P		1	2
INF.06483.05	Einführung in Datenbanken	5	P		3	1
INF.06484.03	Datenbank-Programmierung	5	W		4	1
WIW.00388.05	Grundlagen der BWL	5	A	BWL	3	1

- Die Werte in der Spalte `ID` sind Zeichenketten, die immer genau 12 Zeichen lang sind. Diese Spalte ist Primärschlüssel der Tabelle.
- Die Werte in der Spalte `Titel` sind Zeichenketten bis zur Länge 80. Die Werte in dieser Spalte sind ebenfalls eindeutig.
- In die Spalte `LP` werden nicht-negative ganze Zahlen eingetragen. Stellen Sie bitte mit einem `CHECK`-Constraint sicher, dass die Einfügung negativer Zahlen zu einem Fehler führt. Der Wert `0` soll möglich sein.
- In der Spalte „`Art`“ steht einer der folgenden Werte (jeweils ein Zeichen):
  - „`P`“ für ein Pflichtmodul,
  - „`W`“ für ein Wahlpflichtmodul der Informatik (inkl. Bio-/Wirtschaftsinformatik),
  - „`A`“ für ein Modul eines Anwendungsfaches (das in der Spalte „`Fach`“ steht).

  Schreiben Sie bitte einen `CHECK`-Constraint, der sicherstellt, dass man keine anderen Werte in die Spalte einfügen kann. Sie können dafür auch die logische Verknüpfung `OR` („oder“) verwenden.
- Die Spalte „`Fach`“ soll Zeichenketten bis zur Länge 60 erlauben, und kann aber auch einen Nullwert enthalten. Sie bekommen einen Bonuspunkt, wenn Sie mit einem `CHECK`-Constraint sicherstellen, dass die Spalte genau dann einen Nullwert enthält, wenn in der Spalte „`Art`“ nicht der Wert „`A`“ steht. D.h. die Spalte soll nur für Anwendungsfach-Module verwendet werden. Sie können mit den Bedingungen „`Fach IS NULL`“ und „`Fach IS NOT NULL`“ die Spalte auf den Nullwert testen.
- In der Spalte „`Sem`“ steht das empfohlene Semester, eine positive ganze Zahl (der DB-Entwurf ist etwas vereinfacht, da es auch mehrere Semester zur Auswahl geben kann — das ist in dieser Tabellenstruktur nicht möglich). Die Spalte kann auch einen Nullwert enthalten, wenn das Modul z.B. nur ganz unregelmäßig angeboten wird.
- Die Spalte „`Dauer`“ gibt die Dauer des Moduls in Semestern an. Erlaubte Werte sind nur `1` und `2`. Schreiben Sie wieder einen entsprechenden `CEHCK`-Constraint.

Die Tabelle „`Voraussetzungen`“ sieht so aus:

Voraussetzungen		
zuerst	danach	Studienleistung
INF.00677.09	INF.06483.05	X
INF.00677.09	INF.06484.03	
INF.06483.05	INF.06484.03	

Diese Beispiel-Zeilen bedeuten:

- Die Studienleistung von „Objektorientierte Programmierung“ ist Voraussetzung für „Einführung in Datenbanken“.
- Das Modul „Objektorientierte Programmierung“ (also mit Prüfung) ist Vorausetzung für „Datenbank-Programmierung“.
- Das Modul „Einführung in Datenbanken“ (wieder Studienleistung und Prüfung) ist Voraussetzung für „Datenbank-Programmierung“.

Schreiben Sie eine `CREATE TABLE`-Anweisung für diese Tabelle:

- Die Spalten „`zuerst`“ und „`danach`“ sind jeweils ein Fremdschlüssel, der die Tabelle „`Module`“ referenziert.
- Die Kombination aus „`zuerst`“ und „`danach`“ ist Primärschlüssel dieser Tabelle.
- In der Spalte „`Studienleistung`“ sind nur die folgenden beiden Werte erlaubt:
  - „`X`“, wenn nur die Studienleistung vorausgesetzt wird, und
  - „ “ (ein Leerzeichen) für den Normalfall, dass das ganze Modul abgeschlossen sein muss.

Alle drei Spalten erlauben keine Nullwerte.

Geben Sie eine Textdatei ab mit Endung `.sql`, die die beiden `CREATE TABLE`-Anweisungen enthält.

Sie haben per EMail Benutzernamen und Passwort für eine Datenbank im Adminer bekommen, in der Sie auch eigene Tabellen anlegen können. Der Benutzername ist gleichzeitig auch Name der Datenbank (Sie müssen ihn also in die Eingabefelder „`Username`“ und „`Database`“ beim Adminer eingeben). Sie müssen zunächst ein eigenes Schema anlegen, da Sie für das Standard-Schema `public` nicht das Recht haben, darin Tabellen anzulegen. Beim Adminer bekommen Sie einen Link „`Create Schema`“ angezeigt, wenn links kein Schema ausgewählt ist. Sie können aber auch den SQL-Befehl „`CREATE SCHEMA xyz`“ eingeben. Wählen Sie dann dieses Schema in der Auswahlbox links.

Anschließend können Sie Ihre `CREATE TABLE`-Statements testen. Wenn Sie wollen, können Sie Ihre Tabellen anschließend mit den Beispieldaten füllen, indem Sie die folgende Datei mit „Import“ im Adminer ausführen (oder den Inhalt mit Copy&Paste ins SQL-Fenster einfügen):

[https://users.informatik.uni-halle.de/~brass/db24/homework/h4_insert.sql]