

Einführung in Datenbanken

— Übungsblatt 12 (Mengenoperation in SQL) —

Ihre Lösungen laden Sie bitte in die Übungsplattform in StudIP hoch ([StudIP-Eintrag der Vorlesung], Reiter „LTI-Tool“, dann auf „Anwendung starten“).

Einsendeschluss ist Montag, der 22.01.2024, 18⁰⁰.

Datenbank für Aufgabe 1 bis 3:

Wählen Sie im Adminer das Schema „resultate_public“:

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

Dies ist die Beispiel-Datenbank von Folie 13-76:

- `Resultate_1(STUDENT, H, Z, E)`
- `Resultate_2(STUDENT, ATYP, PROZENT)`

Die Ergebnisse von Hausaufgaben (H), Zwischenklausur (Z) und Endklausur (E) sind hier zusammengefasst. Es ist jeweils der erreichte Anteil der Gesamtpunktzahl gespeichert (in Prozent). Dabei sind zwei verschiedene Darstellungen möglich:

Resultate_1			
STUDENT	H	Z	E
Ann Lloyd	80	90	95
Jim Ford	95	60	75

Resultate_2		
STUDENT	ATYP	PROZENT
Ann Lloyd	H	80
Ann Lloyd	Z	90
Ann Lloyd	E	95
Jim Ford	H	95
Jim Ford	Z	60
Jim Ford	E	75

Die Aufgaben 1 bis 3 sind in SQL zu lösen. Selbstverständlich wäre es auch eine gute Übung, sie zusätzlich in relationaler Algebra zu lösen. Unter folgender Webadresse können Sie Relax mit dieser Datenbank öffnen:

[<http://dbis-uibk.github.io/relax/calc/gist/e0a76d711d449d67914a2b188b7ea137>]

Ihre Lösungen in relationaler Algebra können Sie allerdings nicht abgeben (und bringen demgemäß auch keine Punkte).

Aufgabe 1 (4 Punkte)

Schreiben Sie eine Anfrage in SQL, die nur die Tabelle `Resultate_1` verwendet, und als Ergebnis den Inhalt der Tabelle `Resultate_2` liefert. Das erwartete Ergebnis ist also die Tabelle `Resultate_2` (ohne Tabellenüberschrift). Die Groß-/Kleinschreibung der Spaltennamen ist nicht wichtig. Eine spezielle Sortierung der Zeilen ist auch nicht verlangt.

Anmerkung: Mit Ihrer Anfrage könnte man also `Resultate_2` als Sicht (virtuelle Tabelle) definieren, wenn `Resultate_1` als gespeicherte Tabelle vorliegt (oder als vorher definierte Sicht). Das wäre vermutlich die praktische Anwendung Ihrer Anfrage: Man stellt die Daten den Nutzern in zwei verschiedenen Repräsentationen zur Verfügung, die beide für bestimmte Zwecke praktisch sind, und das, ohne die Daten tatsächlich redundant abzuspeichern.

In dieser Aufgabe brauchen Sie selbstverständlich keine Sicht zu definieren. Der Standard-Account „`student_gast`“ hat ja keine Schreibrechte. In Ihrem persönlichen Datenbank-Account könnten Sie es dagegen ausprobieren, wenn Sie wollen. Dazu müssten Sie die Tabellen installieren, indem Sie folgendes Skript ausführen:

[<https://users.informatik.uni-halle.de/~brass/db23/sql/resultate.sql>]

Die Sicht muss natürlich anders heißen als die vorhandenen Tabellen, also z.B. `Res2`. Für die Hausaufgabe geben Sie aber bitte nur die verlangte Anfrage ab, kein `CREATE VIEW`.

Aufgabe 2 (4 Punkte)

Schreiben Sie nun eine Anfrage in SQL, die die umgekehrte Umstrukturierung vornimmt: Diese Anfrage soll nur die Tabelle `Resultate_2` verwenden, und die Tabelle `Resultate_1` als Ergebnis liefern (natürlich wieder mit beliebiger Groß-/Kleinschreibung der Spaltennamen und ohne vorgegebene Sortierung).

Aufgabe 3 (4 Punkte)

Schreiben Sie nun eine Anfrage, um das Ergebnis von Lösungen zu Aufgabe 1 zu testen. Dazu sei folgende „Lösung“ gegeben:

```
WITH LOESUNG(STUDENT, ATYP, PROZENT)
AS (SELECT DISTINCT STUDENT, 'H' AS ATYP, 80 AS PROZENT
    FROM Resultate_1)
...
```

Das Ergebnis ist natürlich falsch:

STUDENT	ATYP	PROZENT
Ann Lloyd	H	80
Jim Ford	H	80

Ihre Aufgabe ist nun, die „symmetrische Differenz“ mit dem korrekten Ergebnis, also `Resultate_2` zu berechnen. Das Ergebnis soll also folgende Zeilen enthalten:

- Zeilen von `LOESUNG`, die nicht in `Resultate_2` vorkommen („falsche Zeilen“) und
- Zeilen von `Resultate_2`, die nicht in `LOESUNG` vorkommen („fehlende Zeilen“).

In einer zusätzlichen Spalte `DIFF` sollen die falschen Zeilen mit „-“ und die fehlenden Zeilen mit „+“ markiert werden:

STUDENT	ATYP	PROZENT	DIFF
Jim Ford	H	80	-
Ann Lloyd	Z	90	+
Ann Lloyd	E	95	+
Jim Ford	H	95	+
Jim Ford	Z	60	+
Jim Ford	E	75	+

Zur Vereinfachung können Sie davon ausgehen, dass die zu testende Anfrage keine Duplikate liefert (und auch das korrekte Ergebnis keine enthält).

Sortierung und die Groß-/Kleinschreibung der Spaltenüberschriften sind egal.

Datenbank für Aufgabe 4:

Aufgabe 4 ist in relationaler Algebra zu schreiben und bezieht sich auf unsere „Studenten-Aufgaben-Datenbank“:

- STUDENTEN(SID, VORNAME, NACHNAME, EMAIL^o)
- AUFGABEN(ATYP, ANR, THEMA, MAXPT)
- BEWERTUNGEN(SID → STUDENTEN, (ATYP, ANR) → AUFGABEN, PUNKTE)

Probieren Sie Ihre Lösung mit dem aus der Übung bekannten System Relax aus:

[<http://dbis-uibk.github.io/relax/calc/gist/8dc2652578ee12ae756a234c4cf21b3f>]

Vergessen Sie nicht, dass die Tabellen- und Spaltennamen in Relax case-sensitiv sind und in unserer Datenbank groß geschrieben werden.

Aufgabe 4 (4 Punkte)

Geben Sie Vorname und Nachname aller Studierenden aus, die für irgendeine Aufgabe mit dem Thema „SQL“ mindestens 90% der vollen Punktzahl bekommen haben.

Das erwartete Ergebnis ist:

STUDENTEN.VORNAME	STUDENTEN.NACHNAME
'Michael'	'Grau'

Datenbank für Aufgabe 5 und 6:

Die folgenden Aufgaben sind wieder in SQL zu lösen. Sie beziehen sich auf die inzwischen bekannte Datenbank mit Informationen über Nahrungsergänzungsmittel (Multivitamin-Tabletten und Ähnliches). Das Schema „vit_public“ besteht aus folgenden Tabellen:

- Stoff_Kategorie(Kat, Bezeichnung, Sort_Nr)
- Stoff(Vit, Einheit, Tagesdosis°, Kat→Stoff_Kategorie)
- Praeparat(Pid, Name, Hersteller, PZN°, Anz°, Einheit°, Tagesdosis°, Gewicht°, Preis°, glutenfrei°, lactosefrei°)
- Inhalt(Pid → Praeparat, Vit → Stoff, Menge, Prozent°, Anmerkung°)
- Zutat(Pid → Praeparat, Seq, Name, Anmerkung°)

Auf Übungsblatt 9 gab es einige zusätzliche Informationen.

Link zum Adminer (für den Test Ihrer Anfragen):

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

Aufgabe 5 (3 Punkte)

Geben Sie für alle Präparate in der Datenbank die PID und den Namen, sowie den Gehalt an Zink aus. Dabei sollen auch Präparate mit aufgelistet werden, die gar kein Zink enthalten, dann natürlich mit der Menge 0.

Obwohl es unterschiedliche Lösungsmöglichkeiten gibt, ist bei dieser Aufgabe verlangt, dass Sie einen Outer Join verwenden. Abgaben ohne Outer Join werden mit 0 Punkten bewertet. Das erwartete Ergebnis ist:

pid	name	menge
1	A-Z Vital	5.0
2	Centrum	5.0
3	A-Z Komplett	10.0
4	A-Z Complete	5.0
5	Kardiodrink	10.0
6	Mineraldrink	5.0
7	Magnesium Verla 300	0

Aufgabe 6 (3 Punkte)

Lösen Sie nun die gleiche Aufgabe noch einmal ohne Outer Join.