

Einführung in Datenbanken

— Übungsblatt 12 (Relationale Algebra) —

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

Einsendeschluss ist Montag, der 20.01.2025, 18⁰⁰.

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).

Datenbank

Die Aufgaben dieses Übungsblattes beziehen sich auf eine Datenbank mit Informationen über Nahrungsergänzungsmittel (Multivitamin-Tabletten und Ähnliches). Das Schema „vit_public“ besteht aus folgenden Tabellen:

- **Stoff_Kategorie**(Kat, Bezeichnung, Sort_Nr)
Z.B. V für Vitamine, M für Mineralstoffe. Die Spalte **Sort_Nr** gibt eine sinnvolle Sortierung dieser Stoffgruppen.
- **Stoff**(Vit, Einheit, Tagesdosis°, Kat→Stoff_Kategorie)
Es gibt einen Eintrag für jeden Wirkstoff (z.B. Vitamin C), der Informationen wie die Mengeneinheit (z.B. mg) und die empfohlene Tagesdosis enthält.
- **Praeparat**(Pid, Name, Hersteller, PZN°, Anz°, Einheit°, Tagesdosis°, Gewicht°, Preis°, glutenfrei°, lactosefrei°)
Informationen über die Tabletten. **Name** und **Hersteller** zusammen sind auch eindeutig. **Anz** ist die Anzahl Tabletten in der Packung, **Gewicht** das Gesamtgewicht aller dieser Tabletten. **Tagesdosis** ist die Anzahl Tabletten, die pro Tag empfohlen werden (meistens 1).
- **Inhalt**(Pid → Praeparat, Vit → Stoff, Menge, Prozent°, Anmerkung°)
Diese Tabelle enthält die Information, welche Menge von welchem Wirkstoff in der an einem Tag zu schluckenden Anzahl von Tabletten enthalten ist. **Prozent** ist der Prozentsatz der für den Wirkstoff empfohlenen Tagesdosis, der mit dieser Menge erreicht wird. Theoretisch ist das redundant, weil ja auch in der Tabelle **Stoff** eine empfohlene Tagesdosis steht. Da es aber auf den Packung meistens angegeben ist, wurde diese Zahl mit erfasst. Die empfohlenen Mengen ändern sich gelegentlich.

Die folgende zusätzliche Tabelle gibt es nur in der PostgreSQL-Version, nicht in RelaX:

- **Zutat**(Pid → Praeparat, Seq, Name, Anmerkung°)
Während **Inhalt** nur die Wirkstoffe beschreibt, stehen hier die genauen chemischen Verbindungen, die in den Tabletten enthalten sind, und es werden auch Hilfsstoffe aufgelistet (eben alle Zutaten). **Seq** speichert die Reihenfolge der Einträge in der Zutatenliste eines Präparats (sie sind nach der Menge angeordnet, aber die genauen Mengenangaben der Zutaten sind unbekannt — soweit es nicht direkt die in **Inhalt** gelisteten Wirkstoffe sind).

Link zu RelaX (für Anfragen in relationaler Algebra):

[<http://dbis-uibk.github.io/relax/calc/gist/aeb39915904003f10caa6e53bbc5f19a>]

Link zum Adminer (für SQL-Anfragen in PostgreSQL):

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

Hinweise zu RelaX

Falls Sie die Daten der Vitamin-DB im für RelaX nötigen Format sehen wollen, können Sie diese Webadresse aufrufen (das ist aber nur interessant, wenn Sie eigene Datenbanken als „Gist“ anlegen wollen):

[<https://gist.github.com/aeb39915904003f10caa6e53bbc5f19a>]

Die Daten sind in einer Art vereinfachtem CSV-Format: Wenn ein Spaltenwert nicht nur ein einzelnes Wort oder eine einzelne Zahl ist, muss er in "... " geschrieben werden.

Unsere übliche Beispiel-Datenbank (STUDENTEN, AUFGABEN, BEWERTUNGEN) gibt es auch:

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

Tipps:

- In relationaler Algebra ist ein natürlicher Verbund einfacher als ein allgemeiner Verbund. Er verknüpft zwei Tabellen über den gleich benannten Spalten. Sie müssen sich also vorher vergewissern, dass das genau die richtigen Spalten sind. Die Syntax (hier bezogen auf die Beispiel-Datenbank der Vorlesung) ist:

STUDENTEN join BEWERTUNGEN.

In der Vorlesung würde der natürliche Verbund so geschrieben:

STUDENTEN ⋈ BEWERTUNGEN

- Ein Beispiel für die Syntax der Selektion ist:

$\sigma_{SID = 101}$ (STUDENTEN).

Leider ist der Index in RelaX nicht speziell markiert. In der Vorlesung ist es:

$\sigma_{SID = 101}$ (STUDENTEN).

- Ein Beispiel einer Projektion ist:

$\pi_{SID, EMAIL}$ (STUDENTEN).

Dies entspricht:

$\pi_{SID, EMAIL}$ (STUDENTEN).

- Beachten Sie, dass die Selektion stärker bindet als der Join. Sie müssen ggf. Klammern setzen.
- Denken Sie daran, dass die Groß-/Kleinschreibung der Spalten- und Tabellennamen bei RelaX wichtig ist.

Aufgabe 1 (4 Punkte)

Schreiben Sie folgende Anfrage in relationaler Algebra (in der Syntax von RelaX): Welche Präparate enthalten mindestens 7 mg Zink? Sie können voraussetzen, dass die Einheit für Zink „mg“ ist, und brauchen das nicht zu testen. Geben Sie Name und Hersteller des Präparates und die Menge an Zink aus.

Das erwartete Ergebnis ist:

Praeparat.Name	Praeparat.Hersteller	Inhalt.Menge
'A-Z Komplett'	'Abtei'	10
'Kardiodrink'	'MensSana'	10

RelaX gibt die Tabellennamen in den Spaltennamen mit aus, was nicht der Vorlesung entspricht. Sie können die Spalten aber einfach über ihren Namen ansprechen, solange der Name eindeutig einer Tabelle zugeordnet werden kann (wie in SQL).

Bitte geben Sie die Anfrage als .txt-Datei ohne die griechischen Buchstaben ab (so dass sie leicht getestet werden kann). Anfragen, die in RelaX einen Syntaxfehler geben, werden mit 0 Punkten bewertet.

Aufgabe 2 (4 Punkte)

Schreiben Sie folgende Anfrage in relationaler Algebra (in der Syntax von RelaX):

Geben Sie alle Präparate aus, die sowohl Lutein enthalten, als auch einen Inhaltsstoff mit einer Anmerkung, in der „Beta-Carotin“ als Substring vorkommt (Vitamin A ist öfters teilweise als Provitamin Beta-Carotin enthalten. Das steht dann in der Anmerkung. Sie brauchen `Vit = 'A'` nicht zu testen.). Die beiden Stoffe sollen sich gegenseitig in der Aufnahme behindern.

Geben Sie Name und Hersteller des Präparates aus.

RelaX versteht auch LIKE-Bedingungen. Sie könnten für diese Anfrage z.B. `intersect` verwenden oder auch mit

`(rho A Inhalt) x (rho B Inhalt)`

ein kartesisches Produkt von zwei Kopien der Relation erzeugen (die Klammern sind eigentlich nicht nötig). Statt eines kartesischen Produktes können Sie auch gern gleich einen Join verwenden (die Klammern sind wieder überflüssig):

`(rho A Inhalt) join A.Pid=B.Pid (rho B Inhalt)`

Das erwartete Ergebnis ist:

Praeparat.Name	Praeparat.Hersteller
'Kardiodrink'	'MensSana'

Aufgabe 3 (4 Punkte)

Schreiben Sie die Anfrage von Aufgabe 2 auch in SQL.

Drucken Sie also Name und Hersteller von allen Präparaten, die sowohl Lutein enthalten, als auch einen Inhaltsstoff mit einer Bemerkung, die den Substring „Beta-Carotin“ enthält.

Aufgabe 4 (4 Punkte)

Schreiben Sie folgende Anfrage in relationaler Algebra (in der Syntax von RelaX):

Geben Sie für alle Präparate Name, Hersteller und den Gehalt an Zink aus. Falls das Präparat kein Zink enthält (d.h. kein passender Eintrag in `Inhalt`), soll der Wert 0 ausgegeben werden.

Das erwartete Ergebnis ist:

Praeparat.Name	Praeparat.Hersteller	Inhalt.Menge
'A-Z Vital'	'gesundleben'	5
'Centrum'	'Pfizer'	5
'A-Z Komplett'	'Abtei'	10
'A-Z Complete'	'Doppelherz aktiv'	5
'Kardiodrink'	'MensSana'	10
'Mineraldrink'	'MensSana'	5
'Magnesium Verla 300'	'Verla-Pharm Arzneimittel'	0

Der genaue Name der letzten Spalte ist nicht vorgeschrieben. Nützliche Operationen könnten z.B. die Vereinigung `union` und die Mengendifferenz `-` sein.

Bei `union` müssen auch die Spaltennamen übereinstimmen (nicht aber der Relationen-Präfix). Sie können z.B. `Menge<-0` in der Projektion `pi` verwenden, um eine Spalte `Menge` mit dem konstanten Wert 0 zu erstellen. In RelaX ist der Pfeil eigentlich andersherum, aber es versteht auch diese Variante, die mit der Vorlesung übereinstimmt.

Im Prinzip hätte RelaX auch die von SQL bekannte Funktion `coalesce`, aber Sie brauchen das hier vermutlich nicht. Falls Sie diese Funktion doch nutzen wollen, beachten Sie, dass die Spalte beim Aufruf solcher Funktionen explizit benannt werden muss mittels „<-“.

Aufgabe 5 (4 Punkte)

Schreiben Sie die gleiche Anfrage in SQL, d.h. geben Sie für alle Präparate den Gehalt an Zink aus, inklusive dem Wert 0.