

Einführung in Datenbanken

— Übungsblatt 2 (SQL Syntax, Erste Anfragen) —

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

Einsendeschluss ist Montag, der 28.10.2024, 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 die Komponisten-Tabelle in einer Datenbank mit Informationen über Musik-CDs (klassische Musik).

KOMPONIST				
<u>KNr</u>	Name	Vorname	geboren	gestorben
11	Händel	Georg Friedrich	1685	1759
⋮	⋮	⋮	⋮	⋮

Sie finden Sie im Adminer, Schema „komponist_public“. Die Webadresse ist:

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

Das Passwort wurde in der Übung bekanntgegeben. Sie finden es auch im StudIP-Eintrag der Vorlesung, Reiter „Datenbank“.

Aufgabe 1 (2 Punkte)

Schreiben Sie eine SQL-Anfrage, die den Vornamen des Komponisten „Vivaldi“ bestimmt. Versuchen Sie, sich dabei an die eingeschränkte Syntax aus Kapitel 3 der Vorlesung zu halten.

Das erwartete Ergebnis ist:

vorname
Antonio

Schreiben Sie Ihre Bereitschaft, vorzurechnen, in einen Kommentar (ebenso eventuelle weitere Angaben zu Quellen und Arbeitszeit). Da die Fähigkeit, Kommentare in SQL zu schreiben, wichtig ist, wird ein Punkt abgezogen, wenn die Angabe fehlt. Dies gilt auch für die anderen Aufgaben.

Aufgabe 2 (2 Punkte)

Schreiben Sie eine SQL-Anfrage, die die Lebensdaten von Wolfgang Amadeus Mozart liefert. Dabei ist „Wolfgang Amadeus“ der Vorname und „Mozart“ der Nachname (Spalte „Name“).

Das erwartete Ergebnis ist:

geboren	gestorben
1756	1791

Aufgabe 3 (2 Punkte)

Geben Sie alle Komponisten aus, die zwischen 1500 und 1599 geboren wurden (jeweils einschließlich der Grenzen). Drucken Sie alle Spalten der Tabelle.

Das erwartete Ergebnis ist (die Sortierung könnte anders sein):

knr	name	vorname	geboren	gestorben
13	Monteverdi	Claudio	1567	1643
19	Byrd	William	1543	1623
42	Franck	Melchior	1580	1639
44	Mainerio	Giorgio	1545	1582
47	Da Nola	Giovan Domenico	1510	1592
48	Azzaiolo	Filippo	1530	1569
49	Susato	Tilman	1500	1561

Solange Sie keine bestimmte Sortierung in Ihrer SQL-Anfrage verlangen, darf das DBMS die Ergebniszeilen in irgendeiner Reihenfolge ausgeben (das hängt auch von dem Auswertungsplan ab, für den sich der Optimierer entschieden hat). Bei der Beispiel-Ausgabe entspricht die Komponisten-Nummer **knr** wohl der Speicherreihenfolge, und der Optimierer hat einen einfachen „Full Table Scan“ gewählt. Das kann in einer anderen PostgreSQL-Version oder bei einer größeren Tabelle aber anders sein. Wenn Sie wollen, können Sie

ORDER BY geboren

an die Anfrage anhängen, um eine Sortierung nach dem Geburtsjahr zu bekommen. Wenn Sie für den Vergleich mit der erwarteten Ausgabe eine Sortierung nach der **knr** erzwingen wollen, schreiben Sie entsprechend „ORDER BY knr“.

Aufgabe 4 (2 Punkte)

Ein Studienkollege hat folgende Anfrage geschrieben:

```
SELECT GEBOREN
FROM   komponist
WHERE  vorname = "Peter", geboren > 1800.
```

Leider enthält sie Syntaxfehler. Versuchen Sie zuerst, die Fehler zu finden, ohne die Anfrage auszuführen. Anschließend führen Sie die fehlerhafte Anfrage aus (es ist auch nützlich, sich mit den Reaktionen von PostgreSQL auf Syntaxfehler vertraut zu machen). Dann korrigieren Sie die Anfrage bitte. Geben Sie die korrigierte Anfrage mit Kommentaren zu den Syntaxfehlern ab. Achten Sie darauf, nur echte Syntaxfehler zu nennen, keine Stilfehler.

Aufgabe 5 (2 Punkte)

Welchen Wert gibt folgende Anfrage aus?

```
SELECT char_length('\r\n' 'abc')
```

Die Funktion `char_length` liefert die Länge einer Zeichenkette in Zeichen (es gibt auch `octet_length` für die Länge in Bytes). Sie finden diese Funktionen in der PostgreSQL-Dokumentation:

[<https://www.postgresql.org/docs/current/functions-string.html>]

Überlegen Sie zuerst selbst, bevor Sie die Anfrage ausführen.

Geben Sie eine kurze Erklärung des Ergebnisses ab, und zwar als `.txt`-Datei. Bitte kein Word, und auch kein PDF.

Allgemeine Hinweise:

- Anfragen, die in PostgreSQL nicht ausführbar sind (wegen Syntaxfehlern) werden automatisch mit 0 Punkten bewertet! Eventuelle Anmerkungen müssen als SQL-Kommentar geschrieben werden.
- Abweichungen vom erwarteten Ergebnis im Beispielszustand führen dagegen nicht automatisch zu 0 Punkten (aber schon zu Punktabzug). Sie müssen keine besonderen Tricks machen, um die oben gezeigten Ausgaben zu erreichen (das wäre eine Art von „mogeln“). Anfragen, die zwar im Beispielszustand das gewünschte Ergebnis ausgeben, aber mit der Aufgabenstellung kaum etwas zu tun haben, können auch mit 0 Punkten bewertet werden.
- Bei Unklarheiten und Problemen können Sie Fragen im Forum in StudIP stellen. Bitte posten Sie aber keine Anfragen, die der Lösung nahe kommen. Bei komischen Syntaxfehlern, die Sie sich nicht erklären können, können Sie probieren, dem Dozenten oder dem Übungsleiter eine persönliche EMail mit der Anfrage zu schicken.
- Sie haben natürlich auch die Möglichkeit, in der Übung zu fragen, oder im Anschluss an die Vorlesung, oder im Tutorium jeweils am Montag 12¹⁵ im Multimedia-Pool (Raum 3.02). Der Dozent ist anwesend so lange wie es Fragen gibt.
- Schreiben Sie Ihren Namen (sowie Matrikelnummer etc.) bitte nicht in die Abgaben oder Dateinamen, da wir die Anfragen auch für ein Forschungsprojekt verwenden wollen. Das soll natürlich in anonymisierter Form geschehen, aber wir können nicht garantieren, dass wir identifizierbare Informationen auch im Text der Anfrage in allen Fällen finden und eliminieren.
- Vergessen Sie nicht die Angaben zum Vorrechnen (natürlich als SQL-Kommentar).