

Einführung in Datenbanken

— Übungsblatt 9 (NULL, Aggregationsfunktionen) —

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

Einsendeschluss ist Montag, der 16.12.2024, 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 „EMP-DEPT-Datenbank“ mit Informationen über Angestellte und Abteilungen einer Firma (eine Beispiel-Datenbank von Oracle). Das Schema „empdept_public“ besteht aus folgenden Tabellen:

- Abteilungen:
`dept(deptno, dname, loc)`
- Angestellte:
`emp(empno, ename, job, mgro→emp, hiredate, sal, commo, deptnoo→dept)`
- Gehaltsstufen:
`salgrade(grade, losal, hisal)`

Probieren Sie die SQL-Anfragen wie immer mit PostgreSQL aus, z.B. über die Adminer-Webschnittstelle (diese Testmöglichkeit besteht auch in der Klausur):

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

Das Passwort steht Sie in StudIP-Eintrag der Vorlesung, Reiter „Informationen“.

Aufgabe 1 (4 Punkte)

Gesucht sind Angestellte, die noch keiner Abteilung zugeordnet sind (also einen Nullwert in der Spalte `deptno` haben), die aber einen Vorgesetzten haben, der einer Abteilung zugeordnet ist (man könnte sie dann vielleicht auch dieser Abteilung zuordnen).

Geben Sie folgende Daten aus:

- Nummer und Name des Angestellten ohne Abteilung,
- Nummer und Name des Vorgesetzten,
- Nummer und Name der Abteilung des Vorgesetzten.

Das erwartete Ergebnis ist:

empno	ename	Vorgesetzter Nr.	Name	deptno	dname
7369	SMITH	7902	FORD	20	RESEARCH

Bitte wählen Sie die Ausgabespalten wie im Beispiel.

Aufgabe 2 (4 Punkte)

Einige Angestellten bekommen eine Provision für Verkäufe (der Wert steht in der Spalte `comm`: „commission“). Drucken Sie für jeden Angestellten

- den Namen,
- die Abteilung (den Namen der Abteilung),
- das normale Gehalt (in der Spalte `sal`: „salary“),
- die Provision und
- die Auszahlung als Summe des Gehalts und der Provision.

Für Angestellte, die keine Provision haben (Nullwert in der Spalte „`comm`“) geben Sie bitte „(keine)“ aus. Es könnte dabei das Problem auftreten, dass Zahlen und Zeichenketten in einer Spalte gemischt werden sollen. Sie müssen den Zahlwert also in eine Zeichenkette verwandeln (mittels `CAST` oder durch Konkatenation mit der leeren Zeichenkette).

Sortieren Sie die Ausgabe alphabetisch nach dem Abteilungsnamen und bei gleicher Abteilung absteigend nach der Auszahlung, und, falls dies auch noch gleich ist, alphabetisch nach dem Namen des Angestellten.

Das erwartete Ergebnis ist:

Angestellter	Abteilung	Gehalt	Provision	Auszahlung
KING	ACCOUNTING	5000	(keine)	5000
CLARK	ACCOUNTING	2450	(keine)	2450
MILLER	ACCOUNTING	1300	(keine)	1300
FORD	RESEARCH	3000	(keine)	3000
SCOTT	RESEARCH	3000	(keine)	3000
JONES	RESEARCH	2975	(keine)	2975
ADAMS	RESEARCH	1100	(keine)	1100
BLAKE	SALES	2850	(keine)	2850
MARTIN	SALES	1250	1400	2650
ALLEN	SALES	1600	300	1900
WARD	SALES	1250	500	1750
TURNER	SALES	1500	0	1500
JAMES	SALES	950	(keine)	950

Aufgabe 3 (4 Punkte)

Geben Sie für jeden Beruf (Spalte `job`) das minimale und das maximale Gehalt aus und den Angestellten mit minimalem und maximalem Gehalt. Gehalt ist der Wert in der Spalte `sal` („salary“), die Provision in der Spalte `comm` soll hier nicht berücksichtigt werden.

Die Daten sollen nur für Berufe ausgegeben werden, bei denen es tatsächlich eine Gehaltsspanne gibt, d.h. das minimale Gehalt muss echt kleiner als das maximale Gehalt sein.

Falls mehrere Angestellte das gleiche minimale oder maximale Gehalt haben, ist es in Ordnung, dass es mehrere Ausgabezeilen für einen Beruf gibt (es soll dann jede Kombination eines Angestellten mit minimalem Gehalt und eines Angestellten mit maximalem Gehalt in der Ausgabe erscheinen). Glücklicherweise ist dieser Fall im Beispiel-Zustand selten, es gibt dort nur zwei Verkäufer mit dem gleichen minimalen Gehalt.

Das erwartete Ergebnis ist:

Beruf	von	bis	Min. Gehalt	Max. Gehalt
CLERK	800	1300	SMITH	MILLER
MANAGER	2450	2975	CLARK	JONES
SALESMAN	1250	1600	MARTIN	ALLEN
SALESMAN	1250	1600	WARD	ALLEN

Wählen Sie die Spaltenüberschriften bitte wie im Beispiel gezeigt. Eine spezielle Sortierung ist nicht verlangt. Wenn Sie sich den Vergleich mit der Beispiel-Ausgabe erleichtern wollen, sortieren Sie nach Beruf, Min. Gehalt und Max. Gehalt.

Hinweis: Erwartet wird eine Lösung unter Verwendung von `NOT EXISTS`. Es steht Ihnen aber frei, auch Aggregationsfunktionen einzusetzen.

Aufgabe 4 (4 Punkte)

Geben Sie alle Angestellten mit Beruf und Abteilungs-Namen aus. Die Abteilungen sollen alphabetisch sortiert sein. Innerhalb jeder Abteilung soll

- zuerst der „PRESIDENT“ aufgelistet werden (falls in dieser Abteilung vorhanden),
- dann der Manager,
- dann der Clerk und
- dann alle anderen Angestellten.

In jedem Abschnitt (Präsident, Manager, Clerk, andere Berufe) sollen die Angestellten alphabetisch nach ihren Namen sortiert werden. Natürlich könnte eine Abteilung auch mehr als einen Manager oder Clerk haben.

Das erwartete Ergebnis ist:

Abteilung	Beruf	Angestellter
ACCOUNTING	PRESIDENT	KING
ACCOUNTING	MANAGER	CLARK
ACCOUNTING	CLERK	MILLER
RESEARCH	MANAGER	JONES
RESEARCH	CLERK	ADAMS
RESEARCH	ANALYST	FORD
RESEARCH	ANALYST	SCOTT
SALES	MANAGER	BLAKE
SALES	CLERK	JAMES
SALES	SALESMAN	ALLEN
SALES	SALESMAN	MARTIN
SALES	SALESMAN	TURNER
SALES	SALESMAN	WARD

Hinweis: Sie können (bei PostgreSQL) unter ORDER BY Terme (Wertausdrücke) verwenden, nicht nur einzelne Spalten. Sie können also nach berechneten Werten sortieren. Im Skript werden Wertausdrücke mit Fallunterscheidungen besprochen.

Es gibt aber auch eine andere Lösung, die wie beim Logikrätsel mit einer WITH-Hilfstabelle und VALUES arbeitet. Sie dürfen dafür voraussetzen, dass PRESIDENT, MANAGER, CLERK, ANALYST und SALESMAN die einzigen Berufe in der Firma sind.

Aufgabe 5 (4 Punkte)

Betrachtet seien alle Angestellten mit Untergebenen (in der Spalte `mgr` ist die `empno` des direkten Vorgesetzten eingetragen). Berechnen Sie für diese Angestellten:

- Anzahl dieser Angestellten (`anz`),
- das minimale, das maximale und das durchschnittliche Gehalt aus der Spalte `sal` (`min_sal`, `max_sal`, `avg_sal`),
- die Anzahl verschiedener Jobs (`anz_jobs`). Z.B. soll auch dann, wenn zwei Angestellte den Job `MANAGER` haben, der Beruf `MANAGER` nur ein Mal gezählt werden.

Nennen Sie die Ausgabespalten, wie in Klammern gezeigt. Runden Sie das Durchschnittsgehalt durch Aufruf der Funktion `ROUND(...)`. Die Provision in der Spalte `comm` brauchen Sie nicht zu berücksichtigen.

Ihre Anfrage muss genau eine Ergebniszeile mit allen diesen Daten liefern (wie immer bei Aggregationsanfragen ohne `GROUP BY`).

Die erwartete Antwort ist:

<code>anz</code>	<code>min_sal</code>	<code>max_sal</code>	<code>avg_sal</code>	<code>anz_jobs</code>
6	2450	5000	3213	3

1 Datensatz

Falls es etwas hilft: Die Angestellten mit Untergebenen sind im Beispielzustand:

<code>empno</code>	<code>ename</code>	<code>job</code>	<code>mgr</code>	<code>hiredate</code>	<code>sal</code>	<code>comm</code>	<code>deptno</code>
7566	JONES	MANAGER	7839	1981-04-02	2975	NULL	20
7698	BLAKE	MANAGER	7839	1981-05-01	2850	NULL	30
7782	CLARK	MANAGER	7839	1981-06-09	2450	NULL	10
7788	SCOTT	ANALYST	7566	1982-12-09	3000	NULL	20
7839	KING	PRESIDENT	NULL	1981-11-17	5000	NULL	10
7902	FORD	ANALYST	7566	1981-12-03	3000	NULL	20

Diese Tabelle brauchen Sie natürlich nicht auszugeben.