

Einführung in Datenbanken

— Übungsblatt 3 (Datentypen in SQL) —

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

Einsendeschluss ist Montag, der 04.11.2024, 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

Diese Aufgabe bezieht sich auf die Tabelle **emp** (von „employees“, Angestellte einer Firma) der Datenbank im Schema **empdept_public**. Die EMP-DEPT-Datenbank ist eine bekannte Beispiel-Datenbank von Oracle (für PostgreSQL wurde sie aber etwas modifiziert). Die Tabelle „dept“ enthält die Abteilungen der Firma („departments“). Der Link zum Adminer ist:

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

Die Tabelle **emp** hat folgende Spalten:

- **empno**: Angestellten-Nummer (identifiziert den Angestellten eindeutig).
- **ename**: Name des Angestellten.
- **job**: Beruf des Angestellten.
- **mgr**: Angestellten-Nummer des direkten Vorgesetzten („manager“).
- **hiredate**: Datum der Einstellung.
- **sal**: Gehalt des Angestellten („salary“).
- **comm**: Provision (nur für Verkäufer) („commission“).
- **deptno**: Abteilung des Angestellten.

Der Beispiel-Inhalt der Tabelle ist:

emp							
empno	ename	job	mgr	hiredate	sal	comm	deptno
7369	SMITH	CLERK	7902	1980-12-17	800	NULL	NULL
7499	ALLEN	SALESMAN	7698	1981-02-20	1600	300	30
7521	WARD	SALESMAN	7698	1981-02-22	1250	500	30
7566	JONES	MANAGER	7839	1981-04-02	2975	NULL	20
7654	MARTIN	SALESMAN	7698	1981-09-28	1250	1400	30
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
7844	TURNER	SALESMAN	7698	1981-09-08	1500	0	30
7876	ADAMS	CLERK	7788	1983-01-12	1100	NULL	20
7900	JAMES	CLERK	7698	1981-12-03	950	NULL	30
7902	FORD	ANALYST	7566	1981-12-03	3000	NULL	20
7934	MILLER	CLERK	7782	1982-01-23	1300	NULL	10

Aufgabe 1 (5 Punkte)

Lesen Sie das Kapitel 5: „SQL: Datentypen“ der Folien zu Ende. Notieren Sie sich ggf. Fragen und stellen Sie diese in der nächsten Übung (oder der Vorlesung).

Schauen Sie sich die folgenden fünf Wertausdrücke an und überlegen Sie sich, was das Ergebnis sein sollte. Anschließend testen Sie das Ergebnis von PostgreSQL im Adminer. Geben Sie eine `.txt`-Datei ab mit den fünf Werten und jeweils einer kurzen Erklärung (1–2 Zeilen pro Teilaufgabe sollten reichen).

- a) `SELECT ROUND(1e9*CAST(0.10 AS REAL))`
- b) `SELECT SIN(0.5 * PI())`
- c) `SELECT LOG(100)`
- d) `SELECT UPPER(SUBSTRING('abcdef' FROM 2 FOR 3)) || '.'`
- e) `SELECT POSITION('de' IN TRIM(' abcdef'))`

Aufgabe 2 (3 Punkte)

Angenommen, Sie wissen, dass das durchschnittliche Gehalt in der Firma 2073.21 € ist (später lernen wir, wie man das mit SQL berechnen kann). Geben Sie für jeden Angestellten seinen Angestellten-Nummer (`empno`), seinen Namen (`ename`), sein Gehalt (`sal`) und die prozentuale Abweichung seines Gehalts bezogen auf den Durchschnittswert 2073.21 aus. Negative Werte stehen für ein unterdurchschnittliches Gehalt, positive Werte für ein überdurchschnittliches Gehalt. Runden Sie den Prozentsatz bitte auf eine ganze Zahl.

Das erwartete Ergebnis ist (mit einem beliebigen Namen für die berechnete Spalte):

empno	ename	sal	Abweichung in %
7369	SMITH	800	-61
7499	ALLEN	1600	-23
7521	WARD	1250	-40
7566	JONES	2975	43
7654	MARTIN	1250	-40
7698	BLAKE	2850	37
7782	CLARK	2450	18
7788	SCOTT	3000	45
7839	KING	5000	141
7844	TURNER	1500	-28
7876	ADAMS	1100	-47
7900	JAMES	950	-54
7902	FORD	3000	45
7934	MILLER	1300	-37

Aufgabe 3 (3 Punkte)

Das Management hat festgestellt, dass für den Monat der Einstellung (`hiredate`) immer ein volles Gehalt gezahlt worden ist. Es möchte eine Aufstellung mit dem korrigierten Einstellungsdatum (der erste Tag des jeweiligen Monats) und den fehlenden Arbeitstagen in diesem Monat (wenn jemand z.B. am 5. des Monats eingestellt wurde, fehlen 4 Arbeitstage). Drucken Sie Angestelltennummer und Angestelltennamen, das tatsächliche Einstellungsdatum, das korrigierte Einstellungsdatum und die fehlenden Tage. Die Daten sollen im Format `DD.MM.YYYY` gedruckt werden (wie es in Deutschland üblich ist).

Das erwartete Ergebnis ist (die Spaltennamen und die Sortierung dürfen anders sein):

empno	ename	hiredate	korrigiert	fehlende Tage
7369	SMITH	17.12.1980	01.12.1980	16
7499	ALLEN	20.02.1981	01.02.1981	19
7521	WARD	22.02.1981	01.02.1981	21
7566	JONES	02.04.1981	01.04.1981	1
7654	MARTIN	28.09.1981	01.09.1981	27
7698	BLAKE	01.05.1981	01.05.1981	0
7782	CLARK	09.06.1981	01.06.1981	8
7788	SCOTT	09.12.1982	01.12.1982	8
7839	KING	17.11.1981	01.11.1981	16
7844	TURNER	08.09.1981	01.09.1981	7
7876	ADAMS	12.01.1983	01.01.1983	11
7900	JAMES	03.12.1981	01.12.1981	2
7902	FORD	03.12.1981	01.12.1981	2
7934	MILLER	23.01.1982	01.01.1982	22

Hinweis: Bei Bedarf können Sie `CAST(x AS INTEGER)` verwenden, um einen `NUMERIC`-Wert *x* in den Datentyp `INTEGER` zu konvertieren. Sie können nur ganze Zahlen auf Datumswerte addieren oder davon subtrahieren. `EXTRACT(...)` liefert aber den Typ `NUMERIC`. Den Typ eines Ausdrucks *e* können Sie in PostgreSQL mit `pg_typeof(e)` bestimmen.

Die Datentyp-Funktionen von PostgreSQL finden Sie in Kapitel 9 des Handbuchs:

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

Die Aufgabe lässt sich mit den im Skript vorgestellten Funktionen lösen.