

# Einführung in Datenbanken

## — Übungsblatt 3 (Datentypen, CREATE TABLE) —

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 06.11.2023, 18<sup>00</sup>.**

Hausaufgaben müssen einzeln bearbeitet werden. „Zu ähnliche Lösungen“ führen automatisch zu 0 Punkten für alle Beteiligten. Das gilt auch dann, wenn Sie nicht direkt abgeschrieben haben, sondern nur zufällig die gleiche Quelle benutzt haben (ohne diese zu nennen).

Wir haben aber vereinbart, dass bei korrekter Nennung der Quelle (als SQL-Kommentar im vorgeschriebenen Format) selbst identische Abgaben akzeptiert werden (ohne Punktabzug).

### Aufgabe 1 (3 Punkte)

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](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.

Schreiben Sie eine SQL-Anfrage, die folgende Spalten ausgibt:

- die Angestelltennummer `empno`,
- den Namen des Angestellten `ename`,
- das Einstellungsdatum `hiredate` (ohne spezielle Formatierung),
- das Einstellungsdatum im Format 'DD.MM.YYYY' (nutzen Sie die Funktion `TO_CHAR`, siehe [<https://www.postgresql.org/docs/current/functions-formatting.html>])
- das Datum für das einjährige Jubiläum des Angestellten, also ein Jahr nach dem Einstellungsdatum (Sie brauchen keine Schaltjahre zu beachten und können das Datum im Standard-Format ausgeben),
- die Anzahl Tage von der Firmengründung am 01.12.1980 bis zur Einstellung (man kann zwei Datumswerte von einander subtrahieren, und bekommt die Anzahl Tage).

Das erwartete Ergebnis ist:

<code>empno</code>	<code>ename</code>	<code>hiredate</code>	<code>datum</code>	<code>jubilaeum</code>	<code>anz_tage</code>
7369	SMITH	1980-12-17	17.12.1980	1981-12-17	16
7499	ALLEN	1981-02-20	20.02.1981	1982-02-20	81
7521	WARD	1981-02-22	22.02.1981	1982-02-22	83
7566	JONES	1981-04-02	02.04.1981	1982-04-02	122
7654	MARTIN	1981-09-28	28.09.1981	1982-09-28	301
7698	BLAKE	1981-05-01	01.05.1981	1982-05-01	151
7782	CLARK	1981-06-09	09.06.1981	1982-06-09	190
7788	SCOTT	1982-12-09	09.12.1982	1983-12-09	738
7839	KING	1981-11-17	17.11.1981	1982-11-17	351
7844	TURNER	1981-09-08	08.09.1981	1982-09-08	281
7876	ADAMS	1983-01-12	12.01.1983	1984-01-12	772
7900	JAMES	1981-12-03	03.12.1981	1982-12-03	367
7902	FORD	1981-12-03	03.12.1981	1982-12-03	367
7934	MILLER	1982-01-23	23.01.1982	1983-01-23	418

## Aufgabe 2 (2 Punkte)

Was ist das Ergebnis der folgenden Anfrage?

```
SELECT  ename, SUBSTRING(ename FROM 2 FOR 1) || '.' AS x
FROM    emp
WHERE   ename LIKE '%I__'
```

Versuchen Sie das Ergebnis zuerst selbst zu bestimmen (alle Namen finden sich in der obigen Tabelle) und probieren Sie Anfrage erst danach aus. Geben Sie das Anfrage-Ergebnis als `.csv`-Datei ab (mit den Spaltennamen als Kopfzeile).

## Aufgabe 3 (5 Punkte)

Sie haben per EMail Benutzernamen und Passwort für eine Datenbank im Adminer bekommen, in der Sie auch eigene Tabellen anlegen können. Der Benutzername ist gleichzeitig auch Name der Datenbank (Sie müssen ihn also in die Eingabefelder „Username“ und „Database“ beim Adminer eingeben).

Sie müssen zunächst ein eigenes Schema anlegen, da Sie für das Standard-Schema `public` nicht das Recht haben, darin Tabellen anzulegen. Beim Adminer bekommen Sie einen Link „Create Schema“ angezeigt, wenn links kein Schema ausgewählt ist. Sie können aber auch den SQL-Befehl „CREATE SCHEMA xyz“ eingeben. Wählen Sie dann dieses Schema in der Auswahlbox links.

Nun sollen Sie eine Tabelle `aktienkurse` für Aktienkurse anlegen. Sie soll sechs Spalten haben:

- Datum,
- Aktie (Zeichenkette mit bis zu 30 Zeichen),
- Eröffnungskurs
- Schlusskurs
- Tageshoch
- Tagestief

Stellen Sie sicher, dass für jede Aktie und jeden Tag nur eine Zeile eingetragen werden kann. Der Schlusskurs kann einen Nullwert enthalten (so können auch Kurse für den aktuellen Tag eingetragen werden, wenn der Handel noch läuft). Alle anderen Kurse seien immer definiert (diese Spalten sollen also keine Nullwerte erlauben). Die Aktienkurse sollen mit zwei Nachkommastellen gespeichert werden (also in Euro und Cent). Es soll eine Aktie zum Preis von 470000 € geben (die teuerste Aktie der Welt). Sie müssen bei den Kursen also sechs Stellen vor dem Komma erlauben.

Folgende Daten sollen eingefügt werden können (testen Sie diese Anweisungen):

- `INSERT INTO AKTIENKURSE  
VALUES('2023-10-26', 'Bayer', 40.64, 40.72, 41.19, 40.41);`
- `INSERT INTO AKTIENKURSE  
VALUES('2023-10-26', 'Siemens Energy', 11.06, 7.05, 10.55, 6.37);`
- `INSERT INTO AKTIENKURSE  
VALUES('2023-10-30', 'Siemens Energy', 8.71, NULL, 8.86, 7.62);`

Die Kurse stammen von folgenden Webseiten (gerundet auf ganze Cent):

- [[https://www.finanzen.net/historische-kurse/siemens\\_energy](https://www.finanzen.net/historische-kurse/siemens_energy)]
- [<https://www.finanzen.net/historische-kurse/bayer>]

Geben Sie das `CREATE TABLE` Statement ab (in einer `.sql`-Datei).