

💡 Structured Query Language

Bei **SQL** (Structured Query Language) handelt es sich um eine Datenbanksprache, die von mehr oder weniger allen gängigen Datenbanksystemen verstanden wird – beispielsweise MySQL, PostgreSQL, MariaDB oder SQLite. Auch wenn die grundlegenden Keywords überall gleich sind, gibt es teilweise verschiedene "Dialekte".

Um eine Tabelle zu erstellen, verwendet man in SQL den Befehl `CREATE TABLE`. Dieser folgt dem folgenden Schema:

```
1 | CREATE TABLE tabellenname(spalte1 DATENTYP, spalte2 DATENTYP, ...);
```

Beispielsweise erstellt der folgende Befehl eine Mitarbeiter-Tabelle:

```
1 | CREATE TABLE Mitarbeiter(id INT, name VARCHAR(200), geburtsdatum DATE,
   |      gehalt FLOAT);
```

💡 Datentypen in SQL

Viele der Datentypen sind ähnlich zu den in Python verwendeten Typen. In vielen Fällen unterscheidet sich jedoch das dazugehörige Keyword:

- `CHAR(size)`: eine Zeichenkette fester Länge (z.B. Sozialversicherungsnummer)
- `VARCHAR(size)`: eine Zeichenkette variabler Länge (z.B. Name), jedoch maximal `size` Zeichen
- `BOOL`: Boole'scher Wert
- `INT`: Ganzzahlige Werte
- `FLOAT` bzw. `DOUBLE`: Gleitkommazahlen
- `DATE`: Ein Datum im Format `YYYY-MM-DD`

Eine vollständige Liste der Datentypen findet sich online¹.

Versehentlich erstellte oder nicht mehr benötigte Tabellen können mit dem Befehl `DROP TABLE` gelöscht werden:

```
1 | DROP TABLE mitarbeiter;
```

¹https://www.w3schools.com/sql/sql_datatypes.asp

Aufgabe 1: Tabellen erstellen

Auf dem Arbeitsblatt “Einführung Datenbanken” haben Sie sich bereits Gedanken gemacht, wie Sie Produkte und Kunden als Tabelle speichern könnten.

a) Geben Sie jeweils den SQL-Befehl zur Erstellung der zugehörigen Tabellen an.

Null-Werte verhindern

In der Informatik werden leere (z. B. nicht gesetzte) Werte auch als Null-Werte bezeichnet. In SQL kann dies vermieden werden, indem man nach dem Datentyp das Keyword `NOT NULL` angibt. So wird erzwungen, dass der Wert gesetzt werden muss.

Primärschlüssel

Um sicherzustellen, dass jeder Datensatz in der Tabelle eindeutig identifizierbar ist (nicht wie beim Beispiel *Thomas Müller*), sollte bei jeder Tabelle ein Attribut eindeutig pro Datensatz sein. Dieses Attribut nennt man **Primärschlüssel**. In SQL lässt sich das Attribut durch Einfügen des Keywords `PRIMARY KEY` nach dem Datentyp festlegen.

b) Geben Sie für die Tabellen einen passenden Primärschlüssel an. Verändern Sie die SQL-Befehle zur Erstellung der Tabellen um sinnvolle Primärschlüssel und `NOT NULL`-Einschränkungen.



Aufgabe 2: Datensätze einfügen

Nachdem wir eine Tabelle erstellt haben, können wir nun einzelne Datensätze einfügen. Dies geschieht mit dem Keyword `INSERT` nach folgendem Schema:

```
1 | INSERT INTO tabellenname(spalte1, spalte2, ...) VALUES (wert1, wert2, ...);
```

Beispielsweise legen die folgenden Befehle eine Tabelle "Mitarbeiter" und einen Datensatz in dieser an:

```
1 | INSERT INTO mitarbeiter(id, name, geburtsdatum, gehalt) VALUES (0001, 'Thomas Müller', '1980-03-25', 4250.00);
```

a) Legen Sie für die beiden Tabellen aus Aufgabe 1 mehrere Datensätze an. Geben Sie jeweils den zugehörigen SQL-Befehl an.

b) Was passiert, wenn Sie einen Datensatz mit fehlerhaften Werten anlegen wollen? Wie sieht es aus, wenn Sie den Primärschlüssel doppelt vergeben?