

CREATE TABLE

In Kapitel 9 stehen die zur dortigen Fallstudie »Anmeldewesen einer VHS« gehörenden CREATE-TABLE-Befehle.

Damit Sie das selber ausprobieren können, sind sie hier zum Herunterladen und Kopieren aufgeführt. Getestet wurden sie mit PostgreSQL.

Beachten Sie bitte, dass die automatische Primärschlüsselvergabe bei MySQL mit AUTO_INCREMENT erfolgt. Das müssten Sie ändern. Außerdem ist dort die Angabe DEFAULT CURRENT_DATE beim Attribut Anmeldedatum der Tabelle Anmeldungen nicht zulässig. Sie sollten das also löschen.

Und falls Ihr DBMS den Datentyp TIME nicht unterstützt, den Sie beim Attribut Uhrzeit der Tabelle Angebote benötigen, können Sie sich damit behelfen, dass Sie dieses Attribut als CHAR(5) definieren, vielleicht mit der CHECK-Klausel `REGEXP_LIKE(Uhrzeit, '([0-1][0-9]|2[0-3]):[0-5][0-9]')`.

```
CREATE TABLE Mitarbeiter
( PersonalNr    INTEGER GENERATED ALWAYS AS IDENTITY,
  Vorname       VARCHAR(20) NOT NULL,
  Name          VARCHAR(30) NOT NULL,
  Status        CHAR(8)
              CHECK(Status IN ('Vollzeit', 'Teilzeit', 'Honorar')),
  Telefon       VARCHAR(12),
  EMail         VARCHAR(30),
PRIMARY KEY (PersonalNr)
);
```

```
CREATE TABLE Sparten
( Kürzel        CHAR(5),
  Bezeichnung   VARCHAR(30) NOT NULL,
  Leiter        INTEGER,
PRIMARY KEY (Kürzel),
FOREIGN KEY (Leiter) REFERENCES Mitarbeiter (PersonalNr)
);
```

```
CREATE TABLE Kurse
( KursNr        NUMERIC(6),
  Bezeichnung   VARCHAR(30) NOT NULL,
  Sparte        CHAR(5),
PRIMARY KEY (KursNr),
FOREIGN KEY (Sparte) REFERENCES Sparten(Kürzel)
);
```

```
CREATE TABLE Semester
( Semesterkürzel CHAR(6),
  Bezeichnung VARCHAR(20) NOT NULL,
  VON DATE,
  BIS DATE,
PRIMARY KEY (Semesterkürzel)
);

CREATE TABLE Angebote
( KursNr NUMERIC(6) NOT NULL,
  Semesterkürzel CHAR(6) NOT NULL,
  Dozent INTEGER NOT NULL,
  Wochentag CHAR(2)
CHECK(Wochentag IN ('Mo', 'Di', 'Mi', 'Do', 'Fr', 'Sa', 'So')),
  Uhrzeit TIME,
  Dauer NUMERIC(3) CHECK (Dauer >=45),
  Normalpreis NUMERIC(5,2),
  Ermässigt NUMERIC(5,2),
CHECK(Ermässigt <= Normalpreis),
PRIMARY KEY (KursNr, Semesterkürzel),
FOREIGN KEY (KursNr) REFERENCES Kurse(KursNr),
FOREIGN KEY (Semesterkürzel) REFERENCES
      Semester(Semesterkürzel),
FOREIGN KEY (Dozent) REFERENCES Mitarbeiter(PersonalNr)
);

CREATE TABLE Teilnehmer
( TeilnehmerNr INTEGER GENERATED ALWAYS AS IDENTITY,
  Vorname VARCHAR(20) NOT NULL,
  Name VARCHAR(30) NOT NULL,
  Strasse VARCHAR(30),
  PLZ CHAR(5),
  Ort VARCHAR(30),
  Telefon CHAR(12),
  EMail VARCHAR(30),
  Geburtstag DATE NOT NULL,
  Ermässigung CHAR(1)
CHECK(Ermässigung='j' OR Ermässigung='n'),
PRIMARY KEY (TeilnehmerNr)
);

CREATE TABLE Anmeldungen
```

```
( AnmeldeNr      INTEGER GENERATED ALWAYS AS IDENTITY,
  TeilnehmerNr    INTEGER NOT NULL,
  KursNr         NUMERIC(6) NOT NULL,
  Semesterkürzel CHAR(6) NOT NULL,
  Anmeldedatum   DATE DEFAULT CURRENT_DATE,
PRIMARY KEY (AnmeldeNr),
FOREIGN KEY (TeilnehmerNr) REFERENCES Teilnehmer(TeilnehmerNr),
FOREIGN KEY (KursNr, Semesterkürzel)
      REFERENCES Angebote(KursNr, Semesterkürzel)
);
```