

# DAB1 – Praktikum 10: Lösungen

## SQL DDL und DML

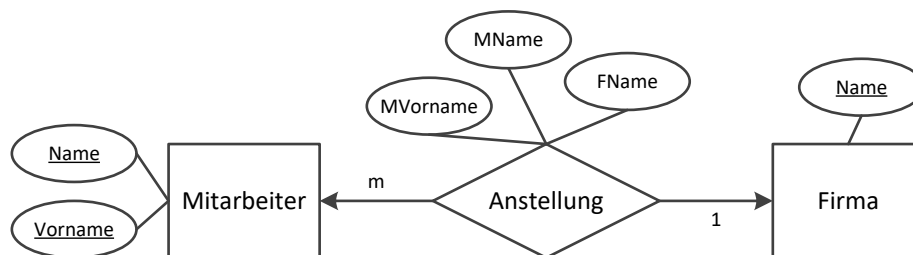
### Vorbereitung

Sie haben bereits ein lokales DBMS in Betrieb und können darauf SQL-Anweisungen ausführen. Erstellen Sie eine neue Datenbankinstanz, auf welcher Sie die folgenden Aufgaben bearbeiten. Die Aufgaben bauen aufeinander auf und sollen dementsprechend in der gegebenen Reihenfolge gelöst werden. *Speichern Sie alle erstellten SQL-Anweisungen in einer Datei ab, damit Sie sie in späteren Aufgaben wieder verwenden können.*

### Aufgabe 1

Schreiben Sie SQL-Anweisungen für sämtliche benötigte CREATE TABLE-Anweisungen für Mitarbeiter, welche höchstens bei einer Firma arbeiten (siehe Beispiel aus der Vorlesung). Firmen stellen naturgemäß mehrere Mitarbeiter an. Mitarbeiter haben Vor- und Nachnamen, Firmen jeweils einen Namen. Definieren Sie hierzu auch Primärschlüssel: Kombination beider Namen auf Mitarbeiter, Name auf Firma. Stellen Sie die Kardinalitäten mit Schlüsseln sicher. Führen Sie die SQL Statements auf Ihrem DBMS aus. Bereinigen Sie allfällige Fehler, bis das Schema in Tabellenform auf der Datenbank vorliegt.

Lösung:



```
CREATE TABLE Mitarbeiter (  
  Vorname varchar(100) NOT NULL,  
  Nachname varchar(100) NOT NULL,  
  
  CONSTRAINT PK_Mitarbeiter PRIMARY KEY (Vorname, Nachname)  
);
```

```
CREATE TABLE Firma (  
  Name varchar(100) NOT NULL,  
  
  CONSTRAINT PK_Firma PRIMARY KEY (Name)  
);
```

```
CREATE TABLE Anstellung (
  MVorname      varchar(100) NOT NULL,
  MNachname     varchar(100) NOT NULL,
  FName        varchar(100) NOT NULL,

  CONSTRAINT PK_Anstellung PRIMARY KEY (MVorname, MNachname),
  CONSTRAINT FK_Mitarbeiter FOREIGN KEY (MVorname, MNachname)
    REFERENCES Mitarbeiter (Vorname, Nachname),
  CONSTRAINT FK_Firma FOREIGN KEY (FName)
    REFERENCES Firma (Name)
);
```

**Bemerkung:** Statt eines Primärschlüssels PK\_Anstellung könnte auch einfach ein Schlüssel CONSTRAINT UK\_Anstellung UNIQUE (MVorname, MName) definiert werden. Es entspricht jedoch gängigem Vorgehen, bei jeder Tabelle einen Primärschlüssel zu definieren.

## Aufgabe 2

Fügen Sie der Tabelle Firma ein Attribut hinzu, um das Gründungsjahr mit einem geeigneten Datentyp zu erfassen. Es soll zusätzlich erfasst werden, welchen Jahreslohn ein Mitarbeiter bei einer Firma erhält. Beide neuen Attribute sollen bei der Dateneingabe immer Werte erhalten.

- 1) Auf welchem Element wird der Jahreslohn am sinnvollsten erfasst und warum?
- 2) Schreiben Sie die notwendigen SQL-Anweisungen (ALTER TABLE) und führen Sie diese aus.

**Lösung:**

- 1) Der Jahreslohn sollte in der Beziehung zwischen Mitarbeitern und Firmen erfasst werden. Logisch gesehen ist dies ein Attribut der Anstellung, nicht eines Mitarbeiters an sich. Zusätzlich ist dies auch für zukünftige Änderungen sinnvoll, da zum Beispiel die Einschränkung gelockert werden kann, dass Mitarbeiter nur bei einer Firma tätig sind. Dann müssen auch unterschiedliche Gehälter je Anstellung erfasst werden können.
- 2) 

```
ALTER TABLE Firma ADD Gründungsjahr integer NOT NULL;
ALTER TABLE Anstellung ADD Jahreslohn decimal(10,2) NOT NULL;
```

  
**Bemerkung:** Es sind auch andere Datentypen möglich.

## Aufgabe 3

Die Adressen von Mitarbeitern sollen nun auch erfasst werden. Schreiben Sie SQL-Anweisungen für das Hinzufügen von Attributen für PLZ, Ort, Strasse und Hausnummer hinzu. Überlegen Sie, für welche Attribute es sinnvoll ist, nicht immer einen Wert bei der Eingabe zu erzwingen. Überlegen Sie die Konsequenzen Ihrer Wahl der Datentypen. Führen Sie die SQL-Anweisungen auf Ihrem DBMS aus.

**Lösung:**

```
ALTER TABLE Mitarbeiter ADD PLZ int NOT NULL;
ALTER TABLE Mitarbeiter ADD Ort varchar(100) NOT NULL;
ALTER TABLE Mitarbeiter ADD Strasse varchar(100) NOT NULL;
ALTER TABLE Mitarbeiter ADD HausNr varchar(8) NOT NULL DEFAULT '';
```

Bemerkung: NULL ist (leider) default in SQL. In der Praxis würde die Hausnummer vermutlich als „nullable“ definiert werden. Bei der Lösung hier wird stattdessen ein ‚Leerstring‘ als Platzhalter für nicht vorhandene Hausnummern benutzt.

## Aufgabe 4

Fügen Sie nun Tupel in Ihre Tabellen ein (INSERT). Erstellen Sie SQL-Anweisungen für:

- 3 Firmen
- 5 Mitarbeiter, verteilt auf verschiedene Firmen

Führen Sie die SQL-Anweisungen aus.

Lösung:

```
-- Firmen
INSERT INTO Firma (Firmaname, Gründungsjahr) VALUES ('Migros', 1925);
INSERT INTO Firma (Firmaname, Gründungsjahr) VALUES ('Coop', 1890);
INSERT INTO Firma (Firmaname, Gründungsjahr) VALUES ('Volg', 1886);

-- Mitarbeiter
INSERT INTO Mitarbeiter (Nachname, Vorname, PLZ, Ort, Strasse, HausNr)
VALUES ('Müller', 'Heinz', 8400, 'Winterthur', 'Badgasse', '12');
INSERT INTO Mitarbeiter (Nachname, Vorname, PLZ, Ort, Strasse, HausNr)
VALUES ('Meier', 'Hans', 8400, 'Winterthur', 'Neustadtgasse', '9');
INSERT INTO Mitarbeiter (Nachname, Vorname, PLZ, Ort, Strasse, HausNr)
VALUES ('Schmid', 'Peter', 8400, 'Winterthur', 'Marktgasse', '120');
INSERT INTO Mitarbeiter (Nachname, Vorname, PLZ, Ort, Strasse, HausNr)
VALUES ('Kunz', 'Ulrich', 8400, 'Winterthur', 'Wülflingerstrasse', '27b');
INSERT INTO Mitarbeiter (Nachname, Vorname, PLZ, Ort, Strasse, HausNr)
VALUES ('Tanner', 'Roger', 8400, 'Winterthur', 'Römerstrasse', '113');

INSERT INTO Anstellung (Nachname, Vorname, FirmaName, Jahreslohn)
VALUES ('Müller', 'Heinz', 'Coop', 70000);
INSERT INTO Anstellung (Nachname, Vorname, FirmaName, Jahreslohn)
VALUES ('Meier', 'Hans', 'Migros', 72000);
INSERT INTO Anstellung (Nachname, Vorname, FirmaName, Jahreslohn)
VALUES ('Schmid', 'Peter', 'Volg', 69000);
INSERT INTO Anstellung (Nachname, Vorname, FirmaName, Jahreslohn)
VALUES ('Kunz', 'Ulrich', 'Volg', 68000);
INSERT INTO Anstellung (Nachname, Vorname, FirmaName, Jahreslohn)
VALUES ('Tanner', 'Roger', 'Coop', 74000);
```

## Aufgabe 5

Die Telefonnummern von Mitarbeitern sollen zusätzlich erfasst werden. Es wird verlangt, dass zwingend eine Nummer vorhanden sein muss (NOT NULL).

- 1) Fügen Sie der Mitarbeitertabelle eine entsprechende Spalte hinzu. Schreiben Sie dazu eine entsprechende SQL-Anweisung und führen Sie diese aus.
- 2) Was geschieht, wenn Sie bei der neuen Spalte „DEFAULT ...“ nicht ebenfalls spezifizieren? Warum?

Lösung:

- 1) `ALTER TABLE Mitarbeiter ADD TelNr varchar(12) NOT NULL DEFAULT '';`
- 2) Da bereits Datensätze für Mitarbeiter vorhanden sind, werden auch diese um eine neue Spalte erweitert. Wenn nun „NOT NULL“ definiert ist, aber kein „DEFAULT“, wird automatisch NULL als Wert in den neuen Spalten der bestehenden Datensätze eingefügt. Dies verletzt in diesem Moment jedoch sofort die eben definierte Einschränkung.

## Aufgabe 6

Ändern Sie nun die Adresse eines Mitarbeiters (UPDATE). Verwenden Sie dazu eine neue Adresse ohne Hausnummer.

Lösung:

```
UPDATE
  Mitarbeiter
SET
  PLZ      = '8401',
  Ort      = 'Winterthur',
  Strasse  = 'Im Lee',
  HausNr   = ''
WHERE
  Nachname = 'Müller' AND Vorname = 'Heinz';
```

## Aufgabe 7

Ändern Sie den Namen einer Ihrer erfassten Firmen (UPDATE).

- 1) Erstellen Sie ein passendes SQL Statement und führen Sie es aus. Was passiert und warum?
- 2) Was ist notwendig, um dieses Update durchführen zu können? Nehmen Sie die Änderungen vor und führen Sie das UPDATE-Statement aus.

Lösung:

- 1) `UPDATE Firma SET Name='Migrolino' WHERE Name='Migros';` (Beispiel)

Das DBMS meldet einen Fehler wegen der Verletzung einer Schlüsselbedingung. Grund ist die referentielle Integrität, welche gewahrt wird. In der Tabelle Anstellung (bzw. Äquivalent der Studierenden) ist die Firma ebenfalls eingetragen, würde aber durch das Update nicht ebenfalls geändert, womit die Referenz zerstört wird.

- 2) `ALTER TABLE Anstellung`  
`DROP CONSTRAINT fk_firma;`

```
ALTER TABLE Anstellung
ADD CONSTRAINT fk_firma FOREIGN KEY (FName)
REFERENCES Firma(Name) ON UPDATE CASCADE;
-- Dann erneut UPDATE...
```

## Aufgabe 8

*Stellen Sie sicher, dass Sie alle bisherigen SQL-Anweisungen noch in einer Datei zur Verfügung haben, um das Schema wiederherzustellen. Alternativ können Sie versuchen, von Ihrem DBMS ein Script für das Schema zu exportieren (ob und wie das funktioniert ist abhängig vom DBMS).*

- 1) Schreiben Sie ein SQL Statement zum Löschen der Mitarbeitertabelle (DROP TABLE) und führen Sie es aus. Warum geht dies nicht ohne weiteres? Ergänzen Sie das Statement um den notwendigen Zusatz, damit die Tabelle gelöscht werden kann.

Lösung:

- 1) `DROP TABLE Mitarbeiter;`

Das geht nicht direkt, da noch Anstellungen vorhanden sind. Grund ist wiederum die referentielle Integrität, welche gewahrt wird. Je nach System muss folgendes gemacht werden:

```
DROP TABLE Mitarbeiter CASCADE;
```

oder

```
DROP TABLE Anstellung;  
DROP TABLE Mitarbeiter;
```