

Aufgabenblatt 10 - Lars Groeber

Aufgabe 1

a)

Stellt eine Relation zwischen einem Bild und einer Person dar. Ein Bild hat eine ID und eine Angabe über Lizenz und Preis. Eine Person hat eine ID und eine Angabe über den Vor- und Nachnamen. Ein Bild kann beliebig viele Personen zeigen, eine Person kann von beliebig vielen Bildern gezeigt werden.

Bild (ID, Lizenz, Preis)

Person (ID, VName, NName)

Bild_zeigt (ID_Bild, ID_Person)

b)

Stellt eine Relation zwischen einem Bild und einer Person dar. Ein Bild hat eine ID und eine Angabe über Lizenz und Preis. Eine Person hat eine ID und eine Angabe über den Vor- und Nachnamen. Ein Bild wurde genau einmal erstellt von beliebig vielen Personen, dabei wird auch der Aufnahmezeitpunkt angegeben.

Bild (ID, Lizenz, Preis, Erstellt von, Aufnahmezeitpunkt)

Person (ID, VName, NName)

c)

Stellt eine Relation zwischen einem Bild und einem System dar. Ein Bild hat eine ID und eine Angabe über Lizenz und Preis. Ein System hat eine ID und einen Ort. Ein Bild kann in keinem oder genau einem System gespeichert sein, während ein System beliebig viele Bilder speichern kann.

Bild (ID, Lizenz, Preis)

Systeme (ID, Ort)

gespeichert_in(Bild_ID, System_ID)

d)

Stellt eine relation zwischen einem System und einem BackupStorage dar. Ein BackupStorage ist dabei ein System mit der zusätzlichen Angabe über die Kapazität. Ein System hat eine ID und einen Ort. Ein System muss in genau zwei BackupStorages "gebackupt" sein, in einem BackupStorage können beliebig viele Systeme gebackupt werden.

Systeme (ID, Ort, Backup1, Backup2)

BackupStorage(ID, Ort, Kapazität)

Aufgabe 2

```
DROP DATABASE IF EXISTS Blatt10;
CREATE DATABASE Blatt10;

USE Blatt10;

### a)

# Person
CREATE TABLE Person (
```

```

ID      INT      NOT NULL PRIMARY KEY AUTO_INCREMENT,
VName  VARCHAR(255) NULL,
NName  VARCHAR(255) NULL
);

# Bild
CREATE TABLE Bild (
  ID      INT      NOT NULL PRIMARY KEY AUTO_INCREMENT,
  Lizenz  VARCHAR(255) NULL,
  Preis   DECIMAL   NULL,
  Erstellt_von INT    NOT NULL,
  Aufnahmezeitpunkt DATETIME NULL,
  FOREIGN KEY fk_bild_person (Erstellt_von) REFERENCES Person (ID)
);

# zeigt Relation
CREATE TABLE Bild_zeigt (
  ID_Bild  INT NOT NULL,
  ID_Person INT NOT NULL,
  PRIMARY KEY (ID_Bild, ID_Person),
  FOREIGN KEY fk_bild_zeigt_bild (ID_Bild) REFERENCES Bild (ID),
  FOREIGN KEY fk_bild_zeigt_person (ID_Person) REFERENCES Person (ID)
);

# BackupStorage
CREATE TABLE BackupStorage (
  ID      INT      NOT NULL PRIMARY KEY AUTO_INCREMENT,
  Ort      VARCHAR(255) NULL,
  Kapazität DECIMAL NOT NULL
);

# System
CREATE TABLE Systeme (
  ID      INT      NOT NULL PRIMARY KEY AUTO_INCREMENT,
  Ort      VARCHAR(255) NULL,
  Backup1 INT      NOT NULL,
  Backup2 INT      NOT NULL,
  FOREIGN KEY fk_system_backup1 (Backup1) REFERENCES BackupStorage (ID),
  FOREIGN KEY fk_system_backup2 (Backup2) REFERENCES BackupStorage (ID)
);

# gespeichert_in
CREATE TABLE Gespeichert_in (
  Bild_ID  INT NOT NULL,
  System_ID INT NOT NULL,
  PRIMARY KEY (Bild_ID, System_ID),
  FOREIGN KEY fk_gespeichert_in_bild (Bild_ID) REFERENCES Bild (ID),
  FOREIGN KEY fk_gespeichert_in_system (System_ID) REFERENCES Systeme (ID)
);

### b)

INSERT INTO Person (VName, NName)
VALUES
  ("Max", "Meier"), ("Lars", "Groeber"), ("Max", "Mustermann"), ("Alexa", "Home");

INSERT INTO Bild (Lizenz, Preis, Erstellt_von, Aufnahmezeitpunkt)
VALUES
  ("CC", 200, 1, NOW()), ("CC", 2000, 1, NOW()), ("CC", 500, 2, NOW()), ("CC", 200, 3, NOW());

INSERT INTO Bild_zeigt (ID_Bild, ID_Person) VALUES (1, 1), (1, 2), (2, 2), (3, 3);
INSERT INTO BackupStorage (Ort, Kapazität) VALUES
  ("Berlin", 50), ("Frankfurt", 20), ("München", 20), ("Köln", 100);
INSERT INTO Systeme (Ort, Backup1, Backup2)
VALUES
  ("Berlin", 1, 2), ("Frankfurt", 2, 1), ("München", 3, 4), ("Köln", 3, 1);
INSERT INTO Gespeichert_in (Bild_ID, System_ID) VALUES (1, 1), (1, 2), (2, 2), (3, 3);

```

```
### c)

SELECT
    P.VName AS Vorname,
    P.NName AS Nachname,
    B.ID    AS Bild_ID,
    B.Preis
FROM Bild B
    JOIN Person P ON B.Erstellt_von = P.ID
ORDER BY P.NName, P.VName, B.Preis;

### d)

SELECT
    B.ID AS Bild_ID,
    S.ID AS System_ID,
    Backup1,
    Backup2
FROM Bild B
    JOIN Gespeichert_in Gi ON B.ID = Gi.Bild_ID
    JOIN Systeme S ON Gi.System_ID = S.ID;
```

Aufgabe 3

i)

$A\{a_1, a_2, a_3\}$

$B\{b_1, b_2\}$

$C\{c_1, c_2\}$

$r\{\langle a_1, b_1, c_1 \rangle, \langle a_2, b_2, c_2 \rangle, \langle a_3, b_1, c_1 \rangle, \langle a_1, b_2, c_2 \rangle\}$

ii)

Funktioniert nicht, da wir (siehe i)) genau vier Verknüpfungen (wegen der (2,2) bei C) brauchen, mit einem A aber nur maximal 2 erstellen können.