

GDB Aufgabenblatt 4

Abgegeben von Timon Back, Fabian Behrendt, Nicolai Stäger

Aufgabe 1: Relationenalgebra

1.A

$$\pi_{\text{Jahresgehalt}} \left(\text{Job} \bowtie_{\text{JNR}=\text{Job}} \text{Bewerbung} \bowtie_{\text{PNR}=\text{Bewerber}} \left(\pi_{\text{PNR}} \left(\sigma_{\text{Geburtsdatum} \geq 1980-01-01} (\text{Person}) \right) \right) \right)$$

1.B

$$\pi_{\text{Titel}, \text{Jahresgehalt}}$$

$$\left(\text{Job} \bowtie_{\text{JNR}=\text{Job}} \text{Bewerbung} \bowtie_{\text{Bewerber}=\text{PNR}} \left(\text{Person} \bowtie_{\text{Heimat}=\text{LNR}} \left(\pi_{\text{LNR}} \left(\sigma_{\text{Name}=\text{Schweiz}} (\text{Land}) \right) \right) \right) \right)$$

1.C

$$\pi_{\text{Vorname}, \text{Nachname}} (\text{Person} \bowtie_{\text{PNR} < > \text{Bewerber}} \text{Bewerbung})$$

1.D

Das Geburtsdatum aller Personen, deren Bewerbung von einer Person bearbeitet wurde, die nach dem 31.12.1994 geboren wurde.

Aufgabe 2: Schemadefinition

2.A

Definieren Sie das angegebene Schema mithilfe von Befehlen der SQL DDL (Data Definition Language).

```
CREATE TABLE Buch (
    Titel          varchar(50) NOT NULL PRIMARY KEY,
    Erscheinungsjahr int(4) NOT NULL,
    Seitenzahl     int(4) NOT NULL
                    CHECK(0 <= Seitenzahl AND Seitenzahl <= 4000),
    Verlag         varchar(50) NOT NULL
);
```

```
CREATE TABLE Person (
    PID            int PRIMARY KEY NOT NULL,
    Vorname        varchar(50) NOT NULL,
    Nachname       varchar(50) NOT NULL UNIQUE,
    Lieblingsbuch  varchar(50),
    CONSTRAINT fk_lieblingsbuch
        FOREIGN KEY (Lieblingsbuch)
        REFERENCES Buch (Titel)
);
```

```
CREATE TABLE Schreibt (
    Autor          int,
    Buch           varchar(50),
    CONSTRAINT pk_schreibt PRIMARY KEY (Autor, Buch),
```

```

        CONSTRAINT fk_autor FOREIGN KEY (Autor) REFERENCES Person (PID),
        CONSTRAINT fk_buch FOREIGN KEY (Buch) REFERENCES Buch (Titel)
    );

CREATE TABLE Begutachtet (
    Lektor          int,
    Buch            varchar(50),
    CONSTRAINT pk_begutachtet PRIMARY KEY (Lektor, Buch),
    CONSTRAINT fk_lektor FOREIGN KEY (Lektor) REFERENCES Person (PID),
    CONSTRAINT fk_buch2 FOREIGN KEY (Buch) REFERENCES Buch (Titel)
);

```

2.B

Erklären Sie knapp, was es für Transaktionen bedeutet, dass in MySQL die referentielle Integrität von Fremdschlüsseln nicht verzögert am Ende der Transaktion (deferred) geprüft werden kann, sondern stets direkt.

Durch die direkte Prüfung müssen die Constraints sofort erfüllbar sein und nicht mit Abschluss der Transaktion. So kann es zu Problemen beim Erzeugen von Tabellen (nächste Frage), aber auch beim Löschen von Tabellen und Daten kommen, da dort die Reihenfolge eingehalten wird. Es können nur Tabellen gelöscht werden, auf die keine Constraint Referenzen zeigen. Das bedeutet auch, dass Einträge die auf sich selbst zeigen oder in einem Zyklus stehen nicht gelöscht werden können.

Erläutern Sie, was passieren würde, wenn Buch das Feld Editor erhalten würde, welches ein Fremdschlüssel auf Person.PID ist. Was müsste man bei der Definition des Schemas in SQL DDL beachten?

Das würde bedeuten, dass das Schema in einem Zyklus steht. Um die Tabelle Buch zu erzeugen, muss erst das Feld Person.PID (Autor) vorhanden sein; um die Tabelle Person zu erzeugen, muss erst das Feld Buch.Titel (Lieblingsbuch) vorhanden sein.

Dies lässt sich einfach umsetzen in dem erst im Nachhinein die Constraint hinzugefügt wird (ALTER TABLE ...)

2.C

Buch

```

INSERT INTO Buch (Titel, Erscheinungsjahr, Seitenzahl, Verlag)
VALUES ('Schall und Wahn', 1929, 304, 'Diogenes');
INSERT INTO Buch (Titel, Erscheinungsjahr, Seitenzahl, Verlag)
VALUES ('Als ich im Sterben lag', 1930, 173, 'Diogenes');
INSERT INTO Buch (Titel, Erscheinungsjahr, Seitenzahl, Verlag)
VALUES ('Hundert Jahre Einsamkeit', 1967, 480, 'Fischer');
INSERT INTO Buch (Titel, Erscheinungsjahr, Seitenzahl, Verlag)
VALUES ('Der Fremde', 1942, 160, 'rororo');
INSERT INTO Buch (Titel, Erscheinungsjahr, Seitenzahl, Verlag)
VALUES ('Krieg und Frieden', 1869, 1536, 'Anaconda');
INSERT INTO Buch (Titel, Erscheinungsjahr, Seitenzahl, Verlag)
VALUES ('Anna Karenina', 1878, 991, 'Anaconda');
INSERT INTO Buch (Titel, Erscheinungsjahr, Seitenzahl, Verlag)
VALUES ('Schuld und Sühne', 1866, 752, 'Deutscher Taschenbuch
Verlag');
INSERT INTO Buch (Titel, Erscheinungsjahr, Seitenzahl, Verlag)
VALUES ('Requiem für einen Traum', 1978, 316, 'Rowohlt');
INSERT INTO Buch (Titel, Erscheinungsjahr, Seitenzahl, Verlag)

```

```
VALUES ('Der Talisman', 1984, 714, 'Heyne');
```

Person

```
INSERT INTO Person (PID, Vorname, Nachname, Lieblingsbuch)
VALUES (1, 'Leo', 'Tolstoi', 'Schuld und Sühne');
INSERT INTO Person (PID, Vorname, Nachname, Lieblingsbuch)
VALUES (2, 'Fjodor', 'Dostojewski', 'Krieg und Frieden');
INSERT INTO Person (PID, Vorname, Nachname, Lieblingsbuch)
VALUES (3, 'Hubert', 'Selby', 'Der Fremde');
INSERT INTO Person (PID, Vorname, Nachname, Lieblingsbuch)
VALUES (4, 'Albert', 'Camus', 'Schuld und Sühne');
INSERT INTO Person (PID, Vorname, Nachname, Lieblingsbuch)
VALUES (5, 'William', 'Faulkner', 'Schuld und Sühne');
INSERT INTO Person (PID, Vorname, Nachname, Lieblingsbuch)
VALUES (6, 'Stephen', 'King', 'Hundert Jahre Einsamkeit');
INSERT INTO Person (PID, Vorname, Nachname, Lieblingsbuch)
VALUES (7, 'Peter', 'Straub', 'Schall und Wahn');
INSERT INTO Person (PID, Vorname, Nachname, Lieblingsbuch)
VALUES (8, 'Gabriel', 'García Márquez', 'Requiem für einen Traum');
```

Schreibt

```
INSERT INTO Schreibt (Autor, Buch) VALUES (1, 'Krieg und Frieden');
INSERT INTO Schreibt (Autor, Buch) VALUES (1, 'Anna Karenina');
INSERT INTO Schreibt (Autor, Buch) VALUES (2, 'Schuld und Sühne');
INSERT INTO Schreibt (Autor, Buch) VALUES (3, 'Requiem für einen Traum');
INSERT INTO Schreibt (Autor, Buch) VALUES (4, 'Der Fremde');
INSERT INTO Schreibt (Autor, Buch) VALUES (5, 'Schall und Wahn');
INSERT INTO Schreibt (Autor, Buch) VALUES (5, 'Als ich im Sterben lag');
INSERT INTO Schreibt (Autor, Buch) VALUES (6, 'Der Talisman');
INSERT INTO Schreibt (Autor, Buch) VALUES (7, 'Der Talisman');
INSERT INTO Schreibt (Autor, Buch) VALUES (8, 'Hundert Jahre Einsamkeit');
```

Begutachtet

```
INSERT INTO Begutachtet (Lektor, Buch)
VALUES (2, 'Anna Karenina');
INSERT INTO Begutachtet (Lektor, Buch)
VALUES (1, 'Schuld und Sühne');
INSERT INTO Begutachtet (Lektor, Buch)
VALUES (8, 'Requiem für einen Traum');
INSERT INTO Begutachtet (Lektor, Buch)
VALUES (6, 'Requiem für einen Traum');
INSERT INTO Begutachtet (Lektor, Buch)
VALUES (5, 'Der Fremde');
INSERT INTO Begutachtet (Lektor, Buch)
VALUES (4, 'Als ich im Sterben lag');
INSERT INTO Begutachtet (Lektor, Buch)
VALUES (2, 'Krieg und Frieden');
INSERT INTO Begutachtet (Lektor, Buch)
VALUES (7, 'Hundert Jahre Einsamkeit');
```

2.D

```
DELETE FROM Begutachtet WHERE Lektor IN
(SELECT PID FROM Person WHERE Vorname = 'Peter');
DELETE FROM Schreibt WHERE Autor IN
(SELECT PID FROM Person WHERE Vorname = 'Peter');
DELETE FROM Person WHERE Vorname = 'Peter';

DROP TABLE Begutachtet;
DROP TABLE Schreibt;
DROP TABLE Person;
DROP TABLE Buch;
```

Aufgabe 3: SQL

3.A

Die PNR sowie der Nachname aller Sachbearbeiter nebst der Anzahl der von ihnen jeweils bearbeiteten Bewerbungen.

```
SELECT PNR, Nachname, COUNT(P.PNR)
  FROM Person P, Bewerbung B
 WHERE B.Sachbearbeiter = P.PNR
GROUP BY PNR, Nachname;
```

3.B

Die PNR aller Sachbearbeiter, die mehr als 2 Bewerbungen bearbeitet haben.

```
SELECT P.PNR
  FROM Person P, Bewerbung B
 WHERE B.Sachbearbeiter = P.PNR
HAVING COUNT(P.PNR) > 2;
```

3.C

Die Vornamen aller Personen, die denselben Nachnamen haben wie der Sachbearbeiter einer ihrer Bewerbungen.

```
SELECT P.Vorname
  FROM Person P, Bewerbung B
 WHERE B.Bewerber = P.PNR
    AND B.Sachbearbeiter IN
    (SELECT PNR
      FROM Person Q
     WHERE Q.Nachname = P.Nachname);
```

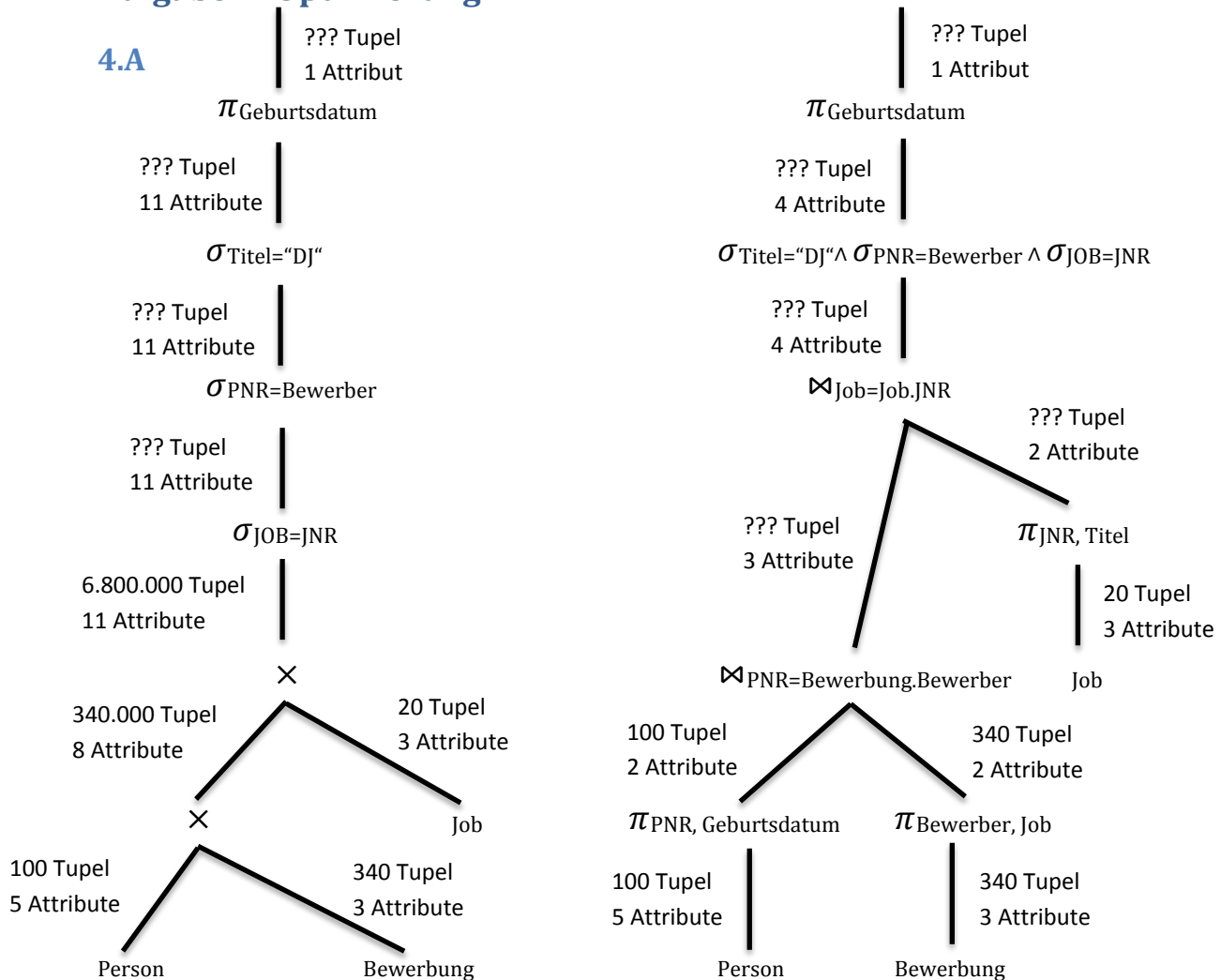
3.D

Die PNR, Vornamen und Nachnamen aller Personen, die bisher kein Sachbearbeiter einer Bewerbung waren.

```
SELECT PNR, Vorname, Nachname
  FROM Person P
 WHERE P.PNR NOT IN ( SELECT Sachbearbeiter FROM Bewerbung);
```

Aufgabe 4: Optimierung

4.A



4.B

Bewerten Sie den Operatorbaum mit den Kardinalitäten der Zwischenergebnisse.

Es werden folgende Kardinalitäten angenommen:

$\text{Card}(\text{Job}) = 20$, $\text{Card}(\text{Person}) = 100$, $\text{Card}(\text{Bewerbung}) = 340$

Es gibt insgesamt 17 verschiedene Job-Titel.

Lösung: siehe 4.A / rechte Zeichnung