

Aufgabe 2

Gegeben sei der folgende Ausschnitt des Schemas für die Verwaltung der Einnahme von Medikamenten:

```
Person : {[ ID : INTEGER, Name : VARCHAR(255), Wohnort : VARCHAR(255) ]}
Hersteller : {[ ID : INTEGER, Name : VARCHAR(255), Standort : VARCHAR(255), AnzahlMitarbeiter : INTEGER ]}
Medikament : {[ ID : INTEGER Name : VARCHAR(255), Kosten : INTEGER, Wirkstoff : VARCHAR(255), produziert_von : INTEGER ]}
nimmt : {[ Person : INTEGER, Medikament : INTEGER, von : DATE, bis : DATE ]}
hat_Unverträglichkeit_gegen : {[ Person : INTEGER, Medikament : INTEGER ]}
```

```
1 CREATE TABLE Person (
2     ID INTEGER PRIMARY KEY,
3     Name VARCHAR(255),
4     Wohnort VARCHAR(255)
5 );
6
7 CREATE TABLE Hersteller (
8     ID INTEGER PRIMARY KEY,
9     Name VARCHAR(255),
10    Standort VARCHAR(255),
11    AnzahlMitarbeiter INTEGER
12 );
13
14 CREATE TABLE Medikament (
15     ID INTEGER PRIMARY KEY,
16     Name VARCHAR(255),
17     Kosten INTEGER,
18     Wirkstoff VARCHAR(255),
19     produziert_von INTEGER REFERENCES Hersteller(ID)
20 );
21
22 CREATE TABLE nimmt (
23     Person INTEGER,
24     Medikament INTEGER,
25     von DATE,
26     bis DATE,
27     PRIMARY KEY (Person, Medikament, von, bis)
28 );
29
30 CREATE TABLE hat_Unverträglichkeit_gegen (
31     Person INTEGER REFERENCES Person(ID),
32     Medikament INTEGER REFERENCES Medikament(ID)
33 );
34
35 INSERT INTO Person VALUES
36     (1, 'Walter Müller', 'Holzapfelkreuth'),
37     (2, 'Dilara Yildiz', 'Fürth');
38
39 INSERT INTO Hersteller VALUES
40     (1, 'Hexal', 'Holzkirchen', 3700),
41     (2, 'Ratiopharm', 'Ulm', 563);
42
43 INSERT INTO Medikament VALUES
44     (1, 'IbuHexal', 3, 'Ibuprofen', 1),
45     (2, 'Ratio-Paracetamol', 2, 'Paracetamol', 2),
46     (3, 'BudeHexal', 4, 'Budesonid', 1),
47     (4, 'Ratio-Budesonid', 5, 'Budesonid', 2);
48
```

```

49 INSERT INTO nimmt VALUES
50   (1, 1, '2021-07-12', '2021-07-23'),
51   (1, 3, '2021-07-12', '2021-07-23'),
52   (2, 4, '2021-02-13', '2021-03-24');
53
54 INSERT INTO hat_Unverträglichkeit_gegen VALUES
55   (1, 1),
56   (1, 3),
57   (2, 2);

```

Die Tabelle `Person` beschreibt Personen über eine eindeutige ID, deren Namen und Wohnort. Die Tabelle `Medikament` enthält Informationen über Medikamente, unter anderem deren Namen, Kosten, Wirkstoffe und einen Verweis auf den Hersteller dieses Medikaments. Die Tabelle `Hersteller` verwaltet verschiedene Hersteller von Medikamenten. Die Tabelle `hat_Unverträglichkeit_gegen` speichert die IDs von Personen zusammen mit den IDs von Medikamenten, gegen die diese Person eine Unverträglichkeit hat. Die Tabelle `nimmt` hingegen verwaltet die Einnahme der verschiedenen Medikamente und speichert zudem in welchem Zeitraum eine Person welches Medikament genommen hat bzw. nimmt.

Beachten Sie bei der Formulierung der SQL-Anweisungen, dass die Ergebnisrelationen keine Duplikate enthalten dürfen. Sie dürfen geeignete Views definieren.

- (a) Schreiben Sie SQL-Anweisungen, die für die bereits existierende Tabelle `nimmt` alle benötigten Fremdschlüsselconstraints anlegt. Erläutern Sie kurz, warum die Spalten `von` und `bis` Teil des Primärschlüssels sind.

```

1 ALTER TABLE nimmt
2 ADD CONSTRAINT FK_Person
3   FOREIGN KEY (Person) REFERENCES Person(ID),
4 ADD CONSTRAINT FK_Medikament
5   FOREIGN KEY (Medikament) REFERENCES Medikament(ID);

```

- (b) Schreiben Sie eine SQL-Anweisung, welche sowohl den Namen als auch die ID von Personen und Medikamenten ausgibt, bei denen die Person das jeweilige Medikament nimmt.

```

1 SELECT DISTINCT
2   p.ID as Personen_ID,
3   p.Name as Personen_Name,
4   m.ID as Medikamenten_ID,
5   m.Name as Medikamenten_Name
6 FROM Person p, Medikament m, nimmt n
7 WHERE
8   n.Person = p.ID AND
9   n.Medikament = m.ID;

```

- (c) Schreiben Sie eine SQL-Anweisung, welche die ID und den Namen der Medikamente mit den niedrigsten Kosten je Hersteller bestimmt.

```

1 SELECT m.ID, m.Name
2 FROM Medikament m, Medikament n
3 WHERE
4   m.produziert_von = n.produziert_von AND
5   m.Kosten >= n.Kosten

```

```

6 GROUP BY m.ID, m.Name
7 HAVING COUNT(*) <= 1;

```

- (d) Schreiben Sie eine SQL-Anweisung, die die Anzahl aller Personen ermittelt, die ein Medikament genommen haben, gegen welches sie eine Unverträglichkeit entwickelt haben.

```

1 SELECT COUNT(DISTINCT p.ID)
2 FROM
3     Person p,
4     nimmt n,
5     hat_Unverträglichkeit_gegen u
6 WHERE
7     p.ID = n.Person AND
8     u.Medikament = n.Medikament;

```

- (e) Schreiben Sie eine SQL-Anweisung, die die ID und den Namen derjenigen Personen ermittelt, die weder ein Medikament mit dem Wirkstoff Paracetamol noch ein Medikament mit dem Wirkstoff Ibuprofen genommen haben.

```

1 SELECT ID, Name FROM Person
2 EXCEPT
3 SELECT p.ID, p.Name
4 FROM Person p, nimmt n, Medikament m
5 WHERE
6     p.ID = n.Person AND
7     n.Medikament = m.ID AND
8     m.Wirkstoff IN ('Ibuprofen', 'Paracetamol');

```

- (f) Schreiben Sie eine SQL-Anweisung, welche die Herstellernamen und die Anzahl der bekannten Unverträglichkeiten gegen Medikamente dieses Herstellers ermittelt. Das Ergebnis soll aufsteigend nach der Anzahl der bekannten Unverträglichkeiten sortiert werden.

```

1 SELECT p.Name, (
2     SELECT COUNT(h.ID)
3     FROM Hersteller h, Medikament m, hat_Unverträglichkeit_gegen u
4     WHERE
5         m.produziert_von = h.ID AND
6         u.Medikament = m.ID AND
7         h.ID = p.ID
8 ) AS Unverträglichkeiten
9 FROM Hersteller p
10 ORDER BY Unverträglichkeiten ASC;

```

- (g) Formulieren Sie eine Anfrage in relationaler Algebra, welche die Wohnorte aller Personen bestimmt, welche ein Medikament mit dem Wirkstoff Paracetamol nehmen oder genommen haben. Die Lösung kann als Baum- oder als Term-Schreibweise angegeben werden.

$$\pi_{\text{Person.Wohnort}} (\sigma_{\text{Medikament.Wirkstoff} = \text{'Paracetamol'}} (\text{Person} \bowtie_{\text{Person.ID}=\text{nimmt.Person}} \text{nimmt} \bowtie_{\text{nimmt.Medikament}=\text{Medikament.ID}} \text{Medikament}))$$

- (h) Formulieren Sie eine Anfrage in relationaler Algebra, welche die Namen aller Personen bestimmt, die von allen bekannten Herstellern, deren Standort München ist, Medikamente nehmen oder genommen haben. Die Lösung kann als Baum- oder als Term-Schreibweise angegeben werden.

$$\pi_{\text{Person.Name}} \left(\sigma_{\text{Hersteller.Standort} = \text{'München'}} \left((\text{Person} \bowtie_{\text{Person.ID} = \text{nimmt.Person}} \text{nimmt}) \bowtie_{\text{nimmt.Med.}} \right) \right)$$