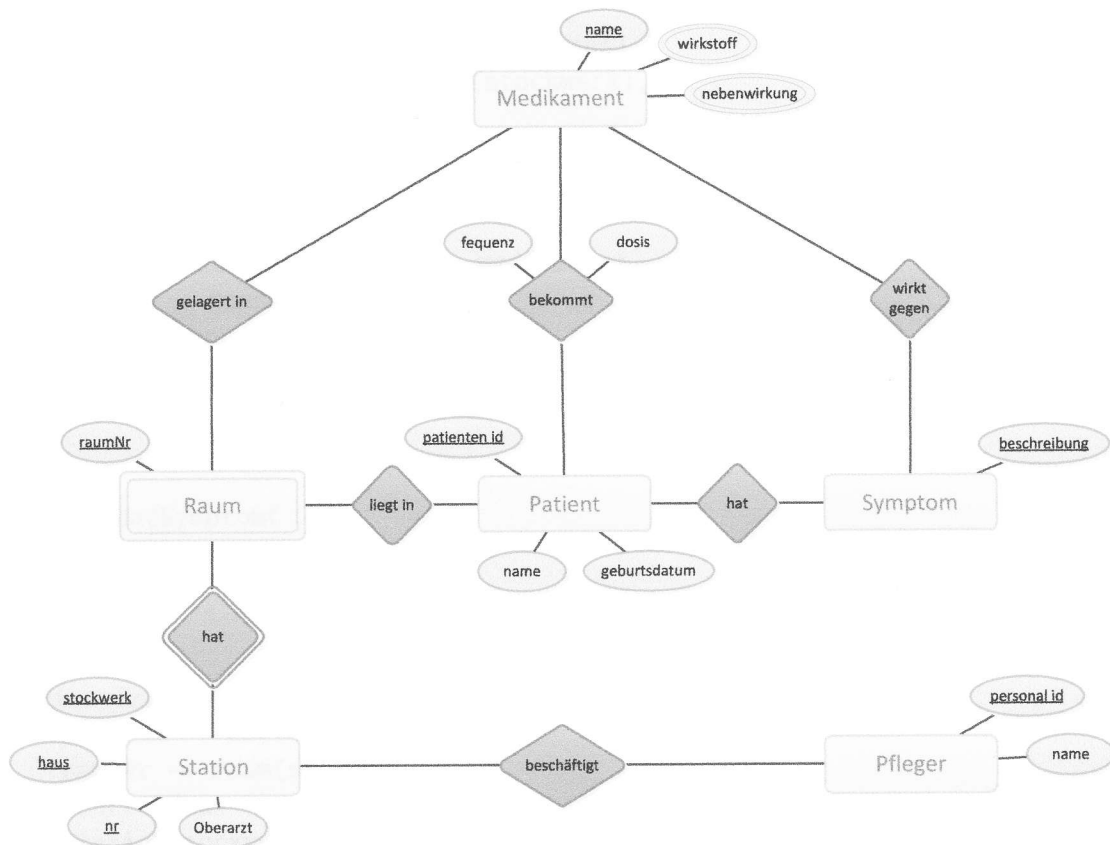


Krankenhausdatenbank - Musterlösung

ER-Modellierung



Relationenschema & DDL

station(haus, stockwerk, nr, oberarzt)

medikament(name)

patient(patientId, name, geburtsdatum)

pfleger(personalId, name)

symptom(beschreibung)

raum(
raumNr,
stationHaus → station(haus),

```

stationStockwerk → station(stockwerk),
stationNr → station(nr)
)

beschaeftigt(stationHaus → station(haus),
stationStockwerk → station(stockwerk),
stationNr → station(nr),
pfleger
)

medikamentInRaum(medikament → medikament,
raumNr → raum(raumNr),
stationHaus → raum(stationHaus),
stationStockwerk → raum(stationStockwerk),
stationNr → raum(stationNr)
)

patientHatSymptom(
patient → patient,
symptom → symptom
)

patientInRaum(
patient → patient,
raumNr → raum(raumNr),
stationHaus → raum(stationHaus),
stationStockwerk → raum(stationStockwerk),
stationNr → raum(stationNr)
)

patientMedikament(
patient → patient,
medikament → medikament,
frequenz,
dosis
)

wirktGegen(
medikament → medikament,
symptom → symptom
)

```

```
CREATE TABLE station(
```

```
1 haus CHAR(1),  
2 stockwerk INTEGER,  
3 nr INTEGER,  
4 oberarzt VARCHAR(255),  
5 PRIMARY KEY(haus, stockwerk, nr)  
6 )
```

```
1 CREATE TABLE medikament(  
2 name VARCHAR(255),  
3 PRIMARY KEY(name)  
4 )
```

```
1 CREATE TABLE patient(  
2 patientId INTEGER,  
3 name VARCHAR(255),  
4 geburtsdatum DATE,  
5 PRIMARY KEY(patientId)  
6 )
```

```
1 CREATE TABLE pfleger(  
2 personalId INTEGER,  
3 name VARCHAR(255),  
4 PRIMARY KEY(personalId)  
5 )
```

```
1 CREATE TABLE symptom(  
2 beschreibung VARCHAR(255),  
3 PRIMARY KEY(beschreibung)  
4 )
```

```
1 CREATE TABLE raum(  
2 raumNr INTEGER,  
3 stationHaus CHAR(1),  
4 stationStockwerk INTEGER,  
5 stationNr INTEGER,  
6 FOREIGN KEY(stationHaus, stationStockwerk, stationNr) REFERENCES station,  
7 PRIMARY KEY(raumNr, stationHaus, stationStockwerk, stationNr)  
8 )
```

```

1 CREATE TABLE beschaeftigt(
2     stationHaus CHAR(1),
3     stationStockwerk INTEGER,
4     stationNr INTEGER,
5     pfleger INTEGER,
6     FOREIGN KEY(stationHaus, stationStockwerk, stationNr) REFERENCES station,
7     FOREIGN KEY(pfleger) REFERENCES pfleger
8 )

```

```

1 CREATE TABLE medikamentInRaum(
2     medikament VARCHAR(255),
3     raumNr INTEGER,
4     stationHaus CHAR(1),
5     stationStockwerk INTEGER,
6     stationNr INTEGER,
7     FOREIGN KEY(medikament) REFERENCES medikament,
8     FOREIGN KEY(raumNr, stationHaus, stationStockwerk, stationNr)
9     REFERENCES raum
10 )

```

```

1 CREATE TABLE patientHatSymptom(
2     patient INTEGER,
3     symptom VARCHAR(255),
4     FOREIGN KEY(patient) REFERENCES patient,
5     FOREIGN KEY(symptom) REFERENCES symptom
6 )

```

```

1 CREATE TABLE patientInRaum(
2     patient INTEGER,
3     raumNr INTEGER,
4     stationHaus CHAR(1),
5     stationStockwerk INTEGER,
6     stationNr INTEGER,
7     FOREIGN KEY(patient) REFERENCES patient,
8     FOREIGN KEY(raumNr, stationHaus, stationStockwerk, stationNr) REFERENCES
9     raum
10 )

```

```

1 CREATE TABLE patientMedikament(
2     patient INTEGER,
3     medikament VARCHAR(255),
4     frequenz VARCHAR(255),
5     dosis VARCHAR(255),
6     FOREIGN KEY(patient) REFERENCES patient,

```

```

7 FOREIGN KEY(medikament) REFERENCES medikament
8 )

```

```

1 CREATE TABLE wirktGegen(
2   medikament VARCHAR(255),
3   symptom VARCHAR(255),
4   FOREIGN KEY(medikament) REFERENCES medikament,
5   FOREIGN KEY(symptom) REFERENCES symptom
6 )

```

Relationale Algebra

$\pi_{sHaus, sStock, sNr}(\sigma_{numPat/numPfle > 10}(\rho_{(sHaus, sStock, sNr, numPat)}(stationHaus, stationStockwerk, stationNr) \bowtie count(*) (patientInRaum) \bowtie_{sHaus, sStock, sNr} \rho_{(sHaus, sStock, sNr, numPfle)}(stationHaus, stationStockwerk, stationNr) \bowtie count(*) (beschaeftigt) \bowtie_{sHaus, sStock, sNr} \dots))$
pro station wie viele Patienten
pro station wie viele Pfleger

(symptom) \ ($\pi_{symptom}$ wirktGegen)

SQL

```

1 WITH symptomNumPat AS (
2   SELECT symptom, COUNT(*) AS numPat FROM patientHatSymptom
3   GROUP BY symptom
4 )
5 SELECT symptom FROM symptomNumPat
6 WHERE numPat = (
7   SELECT MAX(numPat) FROM symptomNumPat
8 )

```

```

1 SELECT name from patient JOIN (
2   SELECT patientId FROM patient JOIN (
3     SELECT patient, COUNT(*) AS numMedikament FROM patientMedikament
4     GROUP BY patient
5   ) AS numMed
6   ON patient.patientId = numMed.patient
7   WHERE numMedikament > 5
8 )
9 INTERSECT

```

```

10 SELECT patientId FROM patient
11 WHERE date_part('year', age(geburtsdatum)) > 60
12 ) AS p2
13 ON patient.patientId = p2.patientId

```

Trigger

Zunächst einen Trigger erstellen, der beim Einfügen von patientInRaum prüft, ob ein Tupel in medikamentInRaum existiert:

```

1 CREATE TRIGGER keinPatientInLagerraum
2 BEFORE INSERT ON patientInRaum
3 REFERENCING NEW ROW AS n
4 FOR EACH STATEMENT
5 IF EXISTS(
6     SELECT * FROM medikamentInRaum AS m
7     WHERE n.raumNr = m.raumNr
8           AND n.stationHaus = m.stationHaus
9           AND n.stationStockwerk = m.stationStockwerk
10          AND n.stationNr = m.stationNr
11 )
12 RAISE EXCEPTION('In dem Raum werden bereits Medikamente gelagert')
13 END IF;

```

Anschließend wird ein Trigger erstellen, der beim Einfügen von medikamentInRaum prüft, ob ein Tupel in patientInRaum existiert.

```

1 CREATE TRIGGER keinPatientInLagerraum
2 BEFORE INSERT ON medikamentInRaum
3 REFERENCING NEW ROW AS n
4 FOR EACH ROW
5 IF EXISTS(
6     SELECT * FROM patientInRaum AS p
7     WHERE n.raumNr = p.raumNr
8           AND n.stationHaus = p.stationHaus
9           AND n.stationStockwerk = p.stationStockwerk
10          AND n.stationNr = p.stationNr
11 )
12 RAISE EXCEPTION('In dem Raum liegen bereits Patienten')
13 END IF;

```

Für den zweiten Trigger wird zunächst die in der Aufgabe beschriebene Relation erstellt, mit der eine Standardbehandlung für ein Symptom eingetragen werden kann:

```

1 CREATE TABLE standardbehandlung(
2     symptom VARCHAR(255),

```

```
3  medikament VARCHAR(255),  
4  dosis VARCHAR(255),  
5  frequenz VARCHAR(255),  
6  FOREIGN KEY symptom REFERENCES symptom,  
7  FOREIGN KEY medikament REFERENCES medikament  
8  )
```

Anschließend wird der geforderte Trigger erstellt:

```
1 CREATE TRIGGER standardbehandlungEinleiten  
2 AFTER INSERT ON patienthatsymptom  
3 REFERENCING NEW ROW AS n  
4 FOR EACH ROW  
5 BEGIN  
6     INSERT INTO patientMedikament (  
7         SELECT n.patient, medikament, frequenz, dosis FROM standardbehandlung  
8         WHERE n.symptom = standardbehandlung.symptom  
9     )  
10 END
```