

Datenbanksysteme

Übung 1

Sommersemester 2017

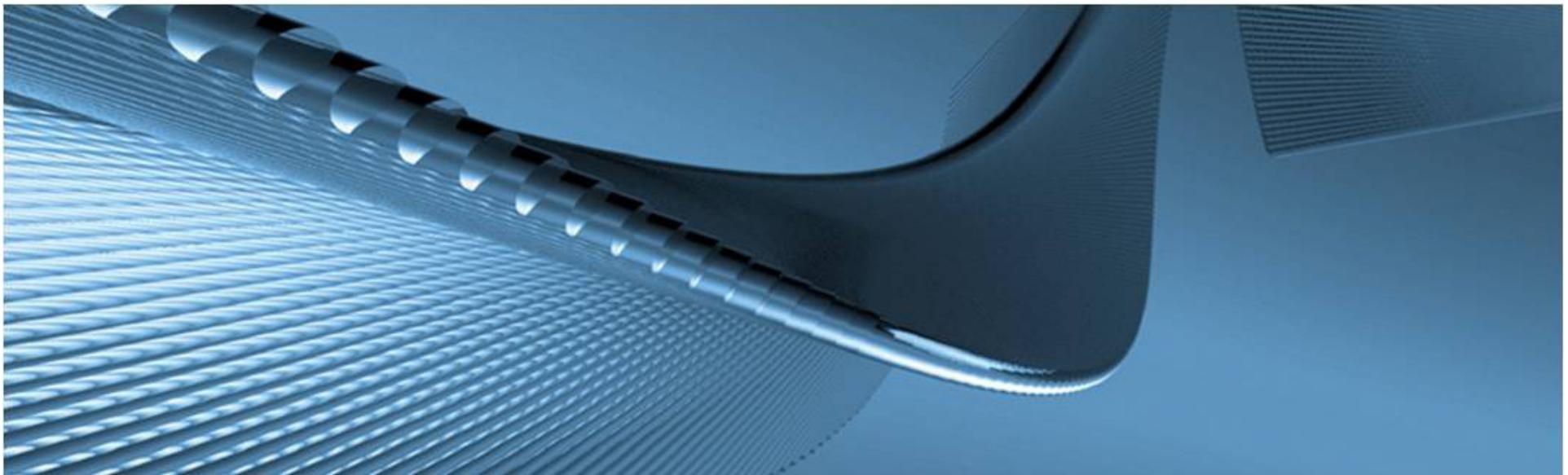
Jutta Mülle

Fakultät für Informatik

IPD, Lehrstuhl Prof. Klemens Böhm

muelle@kit.edu

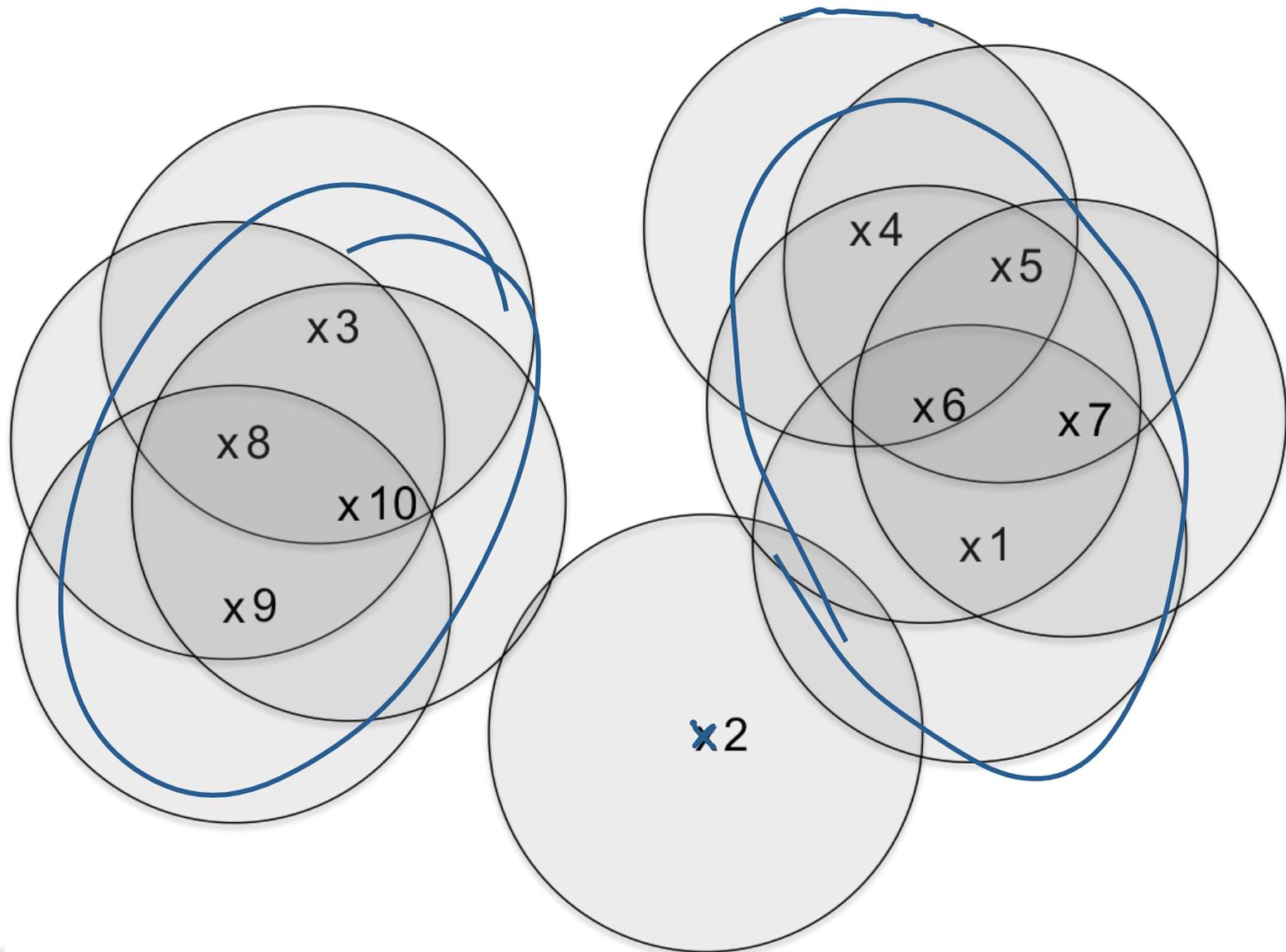
<http://dbis.ipd.kit.edu/>



Aufgabe 1 - DBSCAN

a) In diesem Aufgabenteil geht es um den Algorithmus DBSCAN; verwenden Sie genau die Variante aus der Vorlesung und wenden Sie ihn auf den kleinen Datenbestand gemäß Abbildung 1 an. Um jedes Datenobjekt, gekennzeichnet durch „x“ und benannt mit einer Nummer, ist ein Kreis mit Radius ε eingezeichnet. Der Algorithmus betrachtet das Objekt mit der kleinsten Nummer, das noch nicht angefasst wurde, als nächstes.

Abbildung 1: Datenbestand



DBSCAN (1)

- *Dichte* := Anzahl wasauchimmer pro Volumeneinheit.
- *Objekt ist dicht* := mindestens minPts andere Objekte in Kugel um Objekt mit Radius ϵ (rote Punkte in der Abbildung mit minPts = 3).

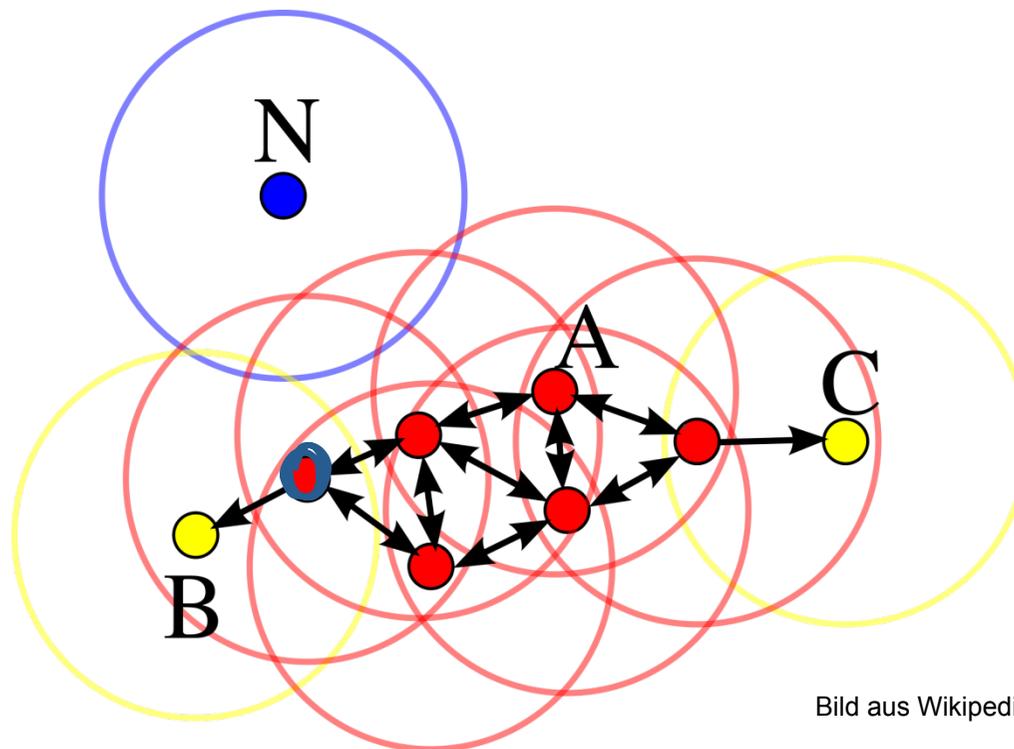


Bild aus Wikipedia

- Motivation
- Index
- kd-Baum, k-NN
- Outlier
- Outlier – Verfahren
- DBSCAN
- Anomalien
- Teilräume
- Motivation
- HiCS

DBSCAN (2)

- *Dichte-erreichbares Objekt* :=
Objekt in ε -Umgebung eines dichten Objekts,
das selbst nicht dicht ist. („Rand des Clusters“)
- Zuordnung dichter Punkte zu Cluster ist deterministisch,
die Dichte-erreichbarer Punkte ist nichtdeterministisch.
(Gelbes Objekt kann unterschiedlichen Clustern
zugeordnet werden.)

Motivation
Index
kd-Baum,
k-NN
Outlier
Outlier
– Verfahren
DBSCAN
Anomalien
Teilräume
Motivation
HiCS

DBSCAN Pseudocode (1)

```
■ DBSCAN(D,  $\epsilon$ , MinPts)
  C = 0
  for each unvisited point P in dataset D
    mark P as visited
    N = getNeighbors(P,  $\epsilon$ )
    if sizeof(N) < MinPts
      mark P as NOISE
    else
      C = next cluster
      add P to cluster C
      for P' in N
        if P' is not yet member of any cluster
          recursiveExpandCluster(P', C,  $\epsilon$ , MinPts)
```

Motivation
Index
kd-Baum,
k-NN
Outlier
Outlier
– Verfahren
DBSCAN
Anomalien
Teilräume
Motivation
HiCS

DBSCAN Pseudocode (2)

Vorlesungsfolie Kap2-37

- recursiveExpandCluster(P, C, ϵ , MinPts)
 - add P to cluster C
 - if P is not visited
 - mark P as visited
 - N = getNeighbors(P, ϵ)
 - if sizeof(N) \geq MinPts
 - for P' in N
 - if P' is not yet member of any cluster
 - recursiveExpandCluster(P', C, ϵ , MinPts)

Motivation
Index
kd-Baum,
k-NN
Outlier
Outlier
– Verfahren
DBSCAN
Anomalien
Teilräume
Motivation
HiCS

Aufgabe 1b - DBSCAN

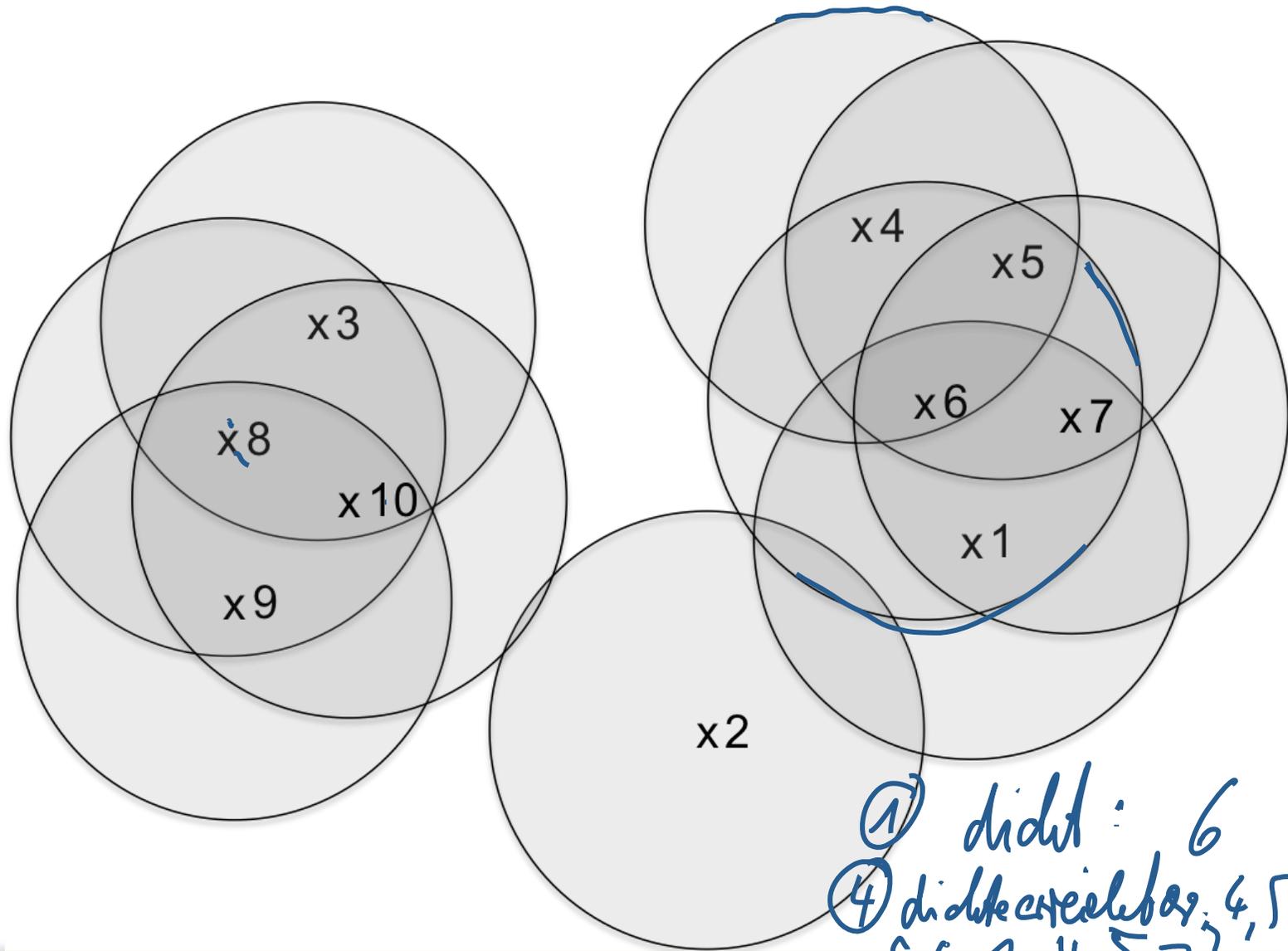
b) Sei $\text{MinPts} = 4$.

- Wieviele Objekte sind dicht?
- Wieviele Objekte sind Dichte-erreichbar?
- Wie genau sieht das Clustering-Resultat mit DBSCAN aus?

Schreiben Sie das Resultat wie folgt hin:

" $C1 = \{X, Y, \dots\}$ $C2 = \{U, V, \dots\}$...". Dabei sind U, V, X, Y usw. Nummern von Objekten. Die Reihenfolge der Cluster soll die sein, in der DBSCAN sie generiert. Bitte lassen Sie Noise weg, falls DBSCAN eine derartige Ausgabe generieren würde.

Abbildung 1: Datenbestand



① dicht: 6
④ dichteste über: 4, 5, 7
 $C_n = \{6, 7, 4, 5, 7\}$

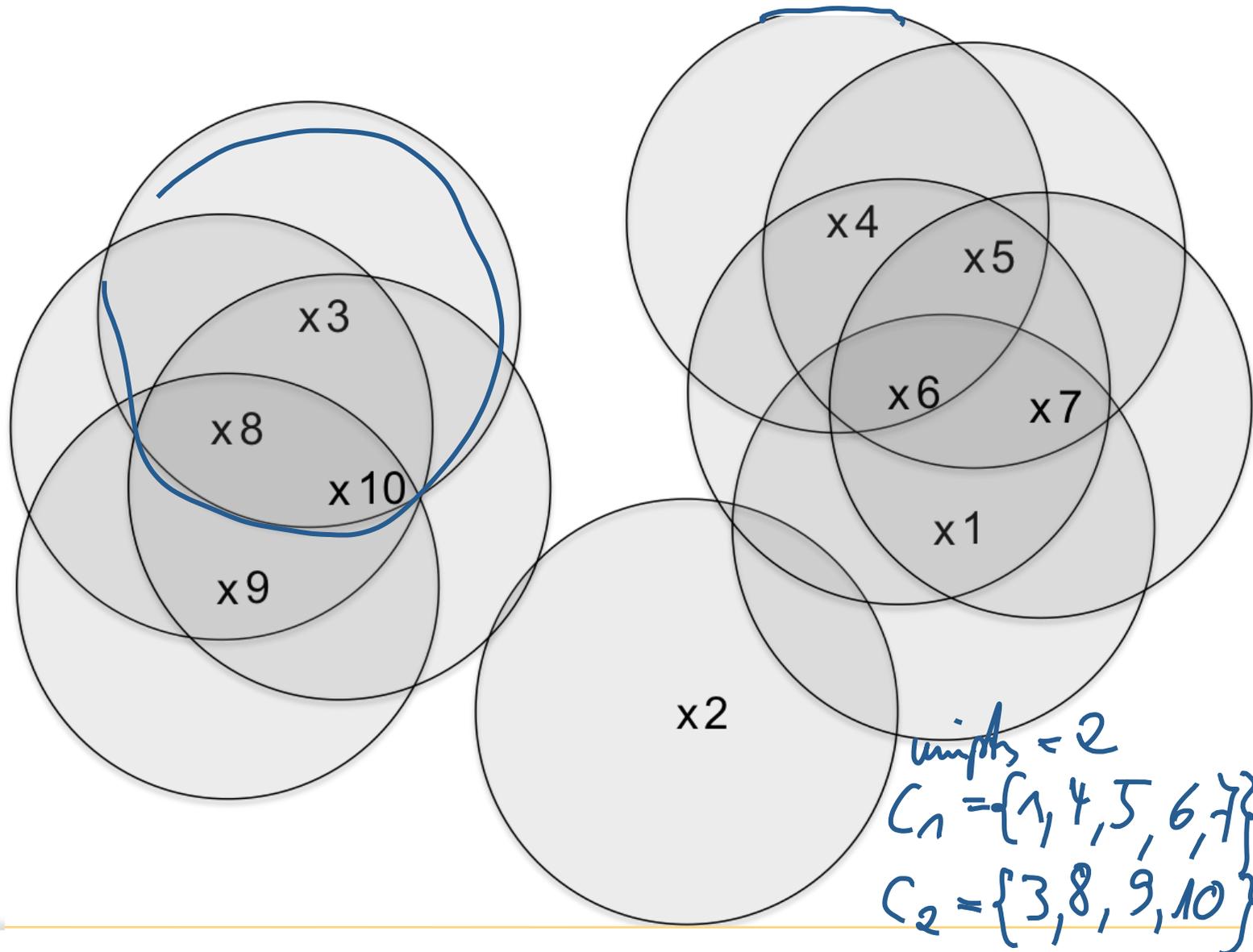
Aufgabe 1b: DBSCAN

- Sei $\text{MinPts} = 4$. Wieviele Objekte sind dicht? Wieviele Objekte sind Dichte-erreichbar?
- Objekt 6 ist dicht, d.h. ein Objekt ist dicht.
- 1, 4, 5 und 7 sind Dichte-erreichbar, d.h. vier Objekte sind Dichte-erreichbar.

Aufgabe 1b: DBSCAN

- Wie genau sieht das Clustering-Resultat mit DBSCAN aus? Schreiben Sie das Resultat wie folgt hin:
"C1 = {X, Y, ...} C2 = {U, V, ... } ...". Dabei sind U, V, X, Y usw. Nummern von Objekten. Die Reihenfolge der Cluster soll die sein, in der DBSCAN sie generiert. Bitte lassen Sie Noise weg, falls DBSCAN eine derartige Ausgabe generieren würde.

Abbildung 1: Datenbestand



Aufgabe 1b: DBSCAN

- Wie genau sieht das Clustering-Resultat mit DBSCAN aus? Schreiben Sie das Resultat wie folgt hin:
"C1 = {X, Y,...} C2 = {U, V, ... } ...". Dabei sind U, V, X, Y usw. Nummern von Objekten. Die Reihenfolge der Cluster soll die sein, in der DBSCAN sie generiert. Bitte lassen Sie Noise weg, falls DBSCAN eine derartige Ausgabe generieren würde.
 - $C_1 = \{1, 4, 5, 6, 7\}$

Aufgabe 1c - DBSCAN

c) Sei $\text{MinPts} = 2$.

- Wieviele Objekte sind dicht?
- Wieviele Objekte sind Dichte-erreichbar?
- Wie genau sieht das Clustering-Resultat mit DBSCAN aus?

Schreiben Sie das Resultat wie folgt hin:

" $C1 = \{X, Y, \dots\}$ $C2 = \{U, V, \dots\}$...". Dabei sind U, V, X, Y usw. Nummern von Objekten. Die Reihenfolge der Cluster soll die sein, in der DBSCAN sie generiert. Bitte lassen Sie Noise weg, falls DBSCAN eine derartige Ausgabe generieren würde.

- Alle Objekte außer Objekt 2 sind dicht, d.h. neun Objekte sind dicht.
- Kein Objekt ist Dichte-erreichbar.
- $C1 = \{1, 4, 5, 6, 7\}$ $C2 = \{3, 8, 9, 10\}$

Aufgabe 1d: DBSCAN

d) Für den Fall, dass $\text{MinPts} = 1$, lässt sich stets eine Formel angeben für die Anzahl der Dichte-erreichbaren Objekte. Schreiben Sie diese Formel hin. Erklären Sie ggf. neue Parameter. Erklären Sie kurz in natürlicher Sprache, warum Ihre Formel richtig ist.

- Die Anzahl Dichte-erreichbarer Punkte in diesem Fall ist stets 0.
- Wenn die ε -Kugel um ein Objekt kein anderes Objekt enthält, ist dieses Objekt nicht Dichte-erreichbar. Es ist nämlich in keiner Kugel um ein anderes Objekt enthalten. Wenn jene Kugel ein anderes Objekt enthält, sind beide Objekte gemäß Definition dicht.

Aufgabe 2 – Modellierung mit SQL-DDL

Es soll ein Informationssystem für die Fußballweltmeisterschaft entwickelt werden. Für die Umsetzung wird ein SQL-basiertes Datenbanksystem gewählt.

- a) Überlegen Sie zunächst, welche Entitäten im Rahmen des Szenarios von Bedeutung sind und wie diese zueinander in Beziehung stehen. *foreign keys*
- b) Welche Informationsbedürfnisse können Sie sich in diesem Kontext vorstellen? Finden Sie einige typische Anfragen an das System. Ergänzen Sie gegebenenfalls die zu modellierenden Entitäten.
- c) Welche Integritätsbedingungen wären in diesem Szenario sinnvoll?
- d) Setzen Sie die Modellierung, ausgehend von Ihren Überlegungen, direkt in SQL-DDL um.

Aufgabe 2 – Modellierung mit SQL-DDL

Es soll ein Informationssystem für die Fußballweltmeisterschaft entwickelt werden. Für die Umsetzung wird ein SQL-basiertes Datenbanksystem gewählt.

		Ergebniss	Tipp	
	Deutschland - Portugal		:	:
Mo, 16.06.2014, Salvador 18:00 Uhr - ARD				
	Ghana - USA		:	:
Di, 17.06.2014, Natal 00:00 Uhr - ARD				
	Deutschland - Ghana		:	:
Sa, 21.06.2014, Fortaleza 21:00 Uhr - ARD				
	USA - Portugal		:	:
Mo, 23.06.2014, Manaus 00:00 Uhr - ZDF				
	USA - Deutschland		:	:
Do, 26.06.2014, Recife 18:00 Uhr - ZDF				
	Portugal - Ghana		:	:
Do, 26.06.2014, Brasilia 18:00 Uhr - ZDF				
Abschlusstabelle			Punkte	Tore

GRUPPE G



Spielplan der Fußball-WM 12. Juni bis 13. Juli 2014

GRUPPE A

Do, 12.06.2014, São Paulo 22:00 Uhr - ZDF	Mexiko - Kamerun	0	0
Fr, 13.06.2014, Natal 18:00 Uhr - ZDF	Brasilien - Mexiko	0	0
Di, 17.06.2014, Fortaleza 21:00 Uhr - ZDF	Kamerun - Kroatien	0	0
Do, 19.06.2014, Manaus 00:00 Uhr - ARD	Kamerun - Brasilien	0	0
Mo, 23.06.2014, Brasília 22:00 Uhr - ARD	Kroatien - Mexiko	0	0
Mo, 23.06.2014, Recife 22:00 Uhr - ARD			

Absteiger: Punkte: Tore:

GRUPPE B

Fr, 13.06.2014, Salvador 21:00 Uhr - ZDF	Spanien - Niederlande	0	0
Sa, 14.06.2014, Curitiba 00:00 Uhr - ZDF	Chile - Australien	0	0
Mi, 18.06.2014, Porto Alegre 18:00 Uhr - ARD	Australien - Niederlande	0	0
Mi, 18.06.2014, Rio de Janeiro 21:00 Uhr - ARD	Spanien - Chile	0	0
Mo, 23.06.2014, Curitiba 18:00 Uhr - ARD	Australien - Spanien	0	0
Mo, 23.06.2014, São Paulo 18:00 Uhr - ARD	Niederlande - Chile	0	0
Mo, 23.06.2014, São Paulo 18:00 Uhr - ARD			

Absteiger: Punkte: Tore:

GRUPPE C

Sa, 14.06.2014, Belo Horizonte 18:00 Uhr - ARD	Kolumbien - Griechenland	0	0
So, 15.06.2014, Recife 03:00 Uhr - ARD	Ellenbeinküste - Japan	0	0
Do, 19.06.2014, Brasília 18:00 Uhr - ZDF	Kolumbien - Ellenbeinküste	0	0
Fr, 20.06.2014, Natal 00:00 Uhr - ZDF	Japan - Griechenland	0	0
Di, 24.06.2014, Curitiba 22:00 Uhr - ZDF	Japan - Kolumbien	0	0
Di, 24.06.2014, Fortaleza 22:00 Uhr - ZDF	Griechenland - Ellenbeinküste	0	0
Di, 24.06.2014, Fortaleza 22:00 Uhr - ZDF			

Absteiger: Punkte: Tore:

GRUPPE D

Sa, 14.06.2014, Fortaleza 21:00 Uhr - ARD	Uruguay - Costa Rica	0	0
So, 15.06.2014, Manaus 00:00 Uhr - ARD	England - Italien	0	0
Do, 19.06.2014, São Paulo 21:00 Uhr - ZDF	Uruguay - England	0	0
Fr, 20.06.2014, Recife 18:00 Uhr - ARD	Italien - Costa Rica	0	0
Di, 24.06.2014, Natal 18:00 Uhr - ZDF	Italien - Uruguay	0	0
Di, 24.06.2014, Belo Horizonte 18:00 Uhr - ZDF	Costa Rica - England	0	0
Di, 24.06.2014, Belo Horizonte 18:00 Uhr - ZDF			

Absteiger: Punkte: Tore:

ACHTELFINALE 1

Sieger Gruppe A vs. Zweiter Gruppe B

Sa, 28.06.2014, Belo Horizonte | 18:00 Uhr

ACHTELFINALE 2

Sieger Gruppe C vs. Zweiter Gruppe D

Sa, 28.06.2014, Rio de Janeiro | 22:00 Uhr

VIERTELFINALE 2

Sieger AF 1 vs. Sieger AF 2

Fr, 04.07.2014, Fortaleza | 22:00 Uhr

HALBFINALE 1

Sieger VF 2 vs. Sieger VF 1

Di, 08.07.2014, Belo Horizonte | 22:00 Uhr

FINALE

Sieger HF 1 vs. Sieger HF 2

Sa, 13.07.2014, Rio de Janeiro | 21:00 Uhr

HALBFINALE 2

Sieger VF 4 vs. Sieger VF 3

Mi, 09.07.2014, São Paulo | 22:00 Uhr

VIERTELFINALE 4

Sieger AF 3 vs. Sieger AF 4

Sa, 05.07.2014, Salvador | 22:00 Uhr

ACHTELFINALE 3

Sieger Gruppe B vs. Zweiter Gruppe A

So, 29.06.2014, Fortaleza | 18:00 Uhr

ACHTELFINALE 4

Sieger Gruppe D vs. Zweiter Gruppe C

So, 29.06.2014, Recife | 22:00 Uhr

SPIEL UM PLATZ 3

Verlierer HF 1 vs. Verlierer HF 2

Sa, 12.07.2014, Brasília | 22:00 Uhr

ACHTELFINALE 5

Sieger Gruppe E vs. Zweiter Gruppe F

Mo, 30.06.2014, Brasília | 18:00 Uhr

ACHTELFINALE 6

Sieger Gruppe G vs. Zweiter Gruppe H

Mi, 30.06.2014, Porto Alegre | 22:00 Uhr

VIERTELFINALE 1

Sieger AF 5 vs. Sieger AF 6

Fr, 04.07.2014, Rio de Janeiro | 18:00 Uhr

VIERTELFINALE 3

Sieger AF 7 vs. Sieger AF 8

Sa, 05.07.2014, Brasília | 18:00 Uhr

ACHTELFINALE 7

Sieger Gruppe F vs. Zweiter Gruppe E

Di, 01.07.2014, São Paulo | 18:00 Uhr

ACHTELFINALE 8

Sieger Gruppe H vs. Zweiter Gruppe G

Di, 01.07.2014, Salvador | 22:00 Uhr

GRUPPE E

So, 15.06.2014, Brasília 18:00 Uhr - ZDF	Frankreich - Honduras	0	0
So, 15.06.2014, Porto Alegre 21:00 Uhr - ZDF	Schweiz - Frankreich	0	0
Fr, 20.06.2014, Salvador 21:00 Uhr - ARD	Honduras - Ecuador	0	0
Sa, 21.06.2014, Curitiba 00:00 Uhr - ARD	Honduras - Schweiz	0	0
Mi, 25.06.2014, Manaus 22:00 Uhr - ARD	Ecuador - Frankreich	0	0
Mi, 25.06.2014, Rio de Janeiro 22:00 Uhr - ARD			

Absteiger: Punkte:

GRUPPE F

Mo, 16.06.2014, Rio de Janeiro 00:00 Uhr - ZDF	Argentinien - Bosnien-Herz.	0	0
Mo, 16.06.2014, Curitiba 21:00 Uhr - ARD	Iran - Nigeria	0	0
Sa, 21.06.2014, Belo Horizonte 18:00 Uhr - ARD	Argentinien - Iran	0	0
So, 22.06.2014, Curitiba 00:00 Uhr - ARD	Nigeria - Bosnien-Herz.	0	0
Mi, 25.06.2014, Porto Alegre 18:00 Uhr - ARD	Nigeria - Argentinien	0	0
Mi, 25.06.2014, Salvador 18:00 Uhr - ARD	Bosnien-Herz. - Iran	0	0
Mi, 25.06.2014, Salvador 18:00 Uhr - ARD			

Absteiger: Punkte:

GRUPPE G

Mo, 16.06.2014, Salvador 18:00 Uhr - ARD	Deutschland - Portugal	0	0
Di, 17.06.2014, Natal 00:00 Uhr - ARD	Ghana - USA	0	0
Sa, 21.06.2014, Fortaleza 21:00 Uhr - ARD	Deutschland - Ghana	0	0
Mo, 23.06.2014, Manaus 00:00 Uhr - ZDF	USA - Portugal	0	0
Di, 24.06.2014, Recife 18:00 Uhr - ZDF	USA - Deutschland	0	0
Di, 24.06.2014, Recife 18:00 Uhr - ZDF	Portugal - Ghana	0	0
Di, 24.06.2014, Brasília 18:00 Uhr - ZDF			

Absteiger: Punkte:

GRUPPE H

Di, 17.06.2014, Belo Horizonte 18:00 Uhr - ZDF	Belgien - Algerien	0	0
Mi, 18.06.2014, Curitiba 00:00 Uhr - ZDF	Russland - Südkorea	0	0
So, 22.06.2014, Rio de Janeiro 18:00 Uhr - ZDF	Belgien - Russland	0	0
So, 22.06.2014, Porto Alegre 21:00 Uhr - ZDF	Südkorea - Algerien	0	0
Di, 24.06.2014, São Paulo 22:00 Uhr - ZDF	Südkorea - Belgien	0	0
Di, 24.06.2014, Curitiba 22:00 Uhr - ZDF	Algerien - Russland	0	0
Di, 24.06.2014, Curitiba 22:00 Uhr - ZDF			

Absteiger: Punkte:



Aufgabe 2 – Modellierung mit SQL-DDL

a) Überlegen Sie zunächst, welche Entitäten im Rahmen des Szenarios von Bedeutung sind und wie diese zueinander in Beziehung stehen.

- Mannschaft
- Spieler
- Spiel
- Gruppe
- Stadion
- Tribünen
- Trainer
- Schiedsrichter
- Verwarnungen
- Erzielte Tore

Aufgabe 2 – Modellierung mit SQL-DDL

- b) Welche Informationsbedürfnisse können Sie sich in diesem Kontext vorstellen? Finden Sie einige typische Anfragen an das System. Ergänzen Sie gegebenenfalls die zu modellierenden Entitäten.

Aufgabe 2 – Modellierung mit SQL-DDL

b) Welche Informationsbedürfnisse können Sie sich in diesem Kontext vorstellen? Finden Sie einige typische Anfragen an das System. Ergänzen Sie gegebenenfalls die zu modellierenden Entitäten.

- Welche Mannschaft ist nach der Gruppenphase weiter?
- Welcher Spieler führt die Torschützenliste an?
- Wann spielt Deutschland gegen USA?
- In welchem Stadion spielt Deutschland gegen USA?
- Wieviele Zuschauer können Spiel X maximal sehen?
Wieviele Zuschauer hatte Spiel X tatsächlich?
- Wieviele Stürmer hat Spanien im Kader?
- Welche Spieler sind für das nächste Spiel gesperrt?
- Welche Mannschaft hat die WM 2014 gewonnen?

Aufgabe 2 – Modellierung mit SQL-DDL

c) Welche Integritätsbedingungen wären in diesem Szenario sinnvoll?

Pro Team ^{max.} 1 Team im Spiel pro Tag.
Pro Team max. 20 Spieler erlaubt.

- Mannschaften dürfen nicht gegen sich selbst spielen.
- Jeder Spieler muss zu genau einer Mannschaft gehören.
- Der Kader einer Mannschaft besteht aus 20 Spielern.
- Die tatsächliche Zuschauerzahl in einem Stadion darf die maximale Kapazität nicht überschreiten.

Aufgabe 2 – Modellierung mit SQL-DDL

- d) Setzen Sie die Modellierung, ausgehend von Ihren Überlegungen, direkt in SQL-DDL um.

Vorgehensweise:

- 1. Primary Key und essentielle Attribute bestimmen.*
- 2. Fremdschlüssel einführen.*
- 3. Informationsbedürfnisse überprüfen, ob beantwortbar, z.B. „Welche Mannschaft ist nach der Gruppenphase weiter?“*
- 4. Integritätsbedingungen spezifizieren/ergänzen.*

Aufgabe 2 – Modellierung mit SQL-DDL

- d) Setzen Sie die Modellierung, ausgehend von Ihren Überlegungen, direkt in SQL-DDL um.

```
CREATE TABLE Mannschaft (  
    mannschafts_id INTEGER NOT NULL,  
    name            VARCHAR (200),  
  
    PRIMARY KEY (mannschafts_id)  
);
```

Aufgabe 2 – Modellierung mit SQL-DDL

- d) Setzen Sie die Modellierung, ausgehend von Ihren Überlegungen, direkt in SQL-DDL um.

```
CREATE DOMAIN GRUPPE AS CHAR(1)
CHECK (VALUE IN ('A', 'B', 'C', 'D', 'E', 'F', 'G', 'H'));
```

```
CREATE TABLE Mannschaft (
    mannschafts_id INTEGER NOT NULL,
    name            VARCHAR (200),
    gruppe          GRUPPE NOT NULL,
    punkte_in_gruppe INTEGER DEFAULT 0 NOT NULL,
    PRIMARY KEY (mannschafts_id)
);
```

Aufgabe 2 – Modellierung mit SQL-DDL

- d) Setzen Sie die Modellierung, ausgehend von Ihren Überlegungen, direkt in SQL-DDL um.

```
CREATE TABLE Spieler (  
    spieler_id    INTEGER NOT NULL,  
    name          VARCHAR (200) NOT NULL,  
  
    PRIMARY KEY (spieler_id)  
);
```

Aufgabe 2 – Modellierung mit SQL-DDL

d) Setzen Sie die Modellierung, ausgehend von Ihren Überlegungen, direkt in SQL-DDL um.

```
CREATE DOMAIN SPIELPOSITION AS VARCHAR(60)
CHECK (VALUE IN (,Tor', ,Abwehr', ,Mittelfeld', ,Sturm'));
CREATE TABLE Spieler ( Philip Lahm
    spieler_id      INTEGER NOT NULL,
    name            VARCHAR (200) NOT NULL,
    gesperrt       BOOLEAN,
    tore           INTEGER,
    spielposition   SPIELPOSITION,
    mannschafts_id INTEGER, #5
    PRIMARY KEY (spieler_id)
    FOREIGN KEY (mannschafts_id) REFERENCES
    Mannschaft(mannschafts_id) Attribut
);
```

Aufgabe 2 – Modellierung mit SQL-DDL

d) Setzen Sie die Modellierung, ausgehend von Ihren Überlegungen, direkt in SQL-DDL um.

```
CREATE DOMAIN SPIELPOSITION AS VARCHAR(60)
CHECK (VALUE IN ('Tor', 'Abwehr', 'Mittelfeld', 'Sturm'));
CREATE TABLE Spieler (
    spieler_id      INTEGER NOT NULL,
    name            VARCHAR (200) NOT NULL,
    gesperrt       BOOLEAN,
    tore            INTEGER, Default NOT NULL,
    spielposition  SPIELPOSITION,
    mannschafts_id INTEGER,
    PRIMARY KEY (spieler_id)
    FOREIGN KEY (mannschafts_id) REFERENCES
        Mannschaft(mannschafts_id)
```

So korrekt?

Aufgabe 2 – Modellierung mit SQL-DDL

d) Setzen Sie die Modellierung, ausgehend von Ihren Überlegungen, direkt in SQL-DDL um.

```
CREATE DOMAIN SPIELPOSITION AS VARCHAR(60)
CHECK (VALUE IN (,Tor', ,Abwehr', ,Mittelfeld', ,Sturm'));
CREATE TABLE Spieler (
    spieler_id      INTEGER NOT NULL,
    name            VARCHAR (200) NOT NULL,
    gesperrt       BOOLEAN DEFAULT FALSE NOT NULL,
    tore           INTEGER DEFAULT 0 NOT NULL,
    spielposition  SPIELPOSITION,
    mannschafts_id INTEGER NOT NULL,
    PRIMARY KEY (spieler_id)
    FOREIGN KEY (mannschafts_id) REFERENCES
                    Mannschaft(mannschafts_id)
```

Aufgabe 2 – Modellierung mit SQL-DDL

- d) Setzen Sie die Modellierung, ausgehend von Ihren Überlegungen, direkt in SQL-DDL um.

```
CREATE TABLE Stadion (  
    stadion_id      INTEGER NOT NULL,  
    max_zuschauer  INTEGER NOT NULL,  
    ort             VARCHAR(200),  
    PRIMARY KEY (stadion_id)  
);
```

Aufgabe 2 – Modellierung mit SQL-DDL

d) Setzen Sie die Modellierung, ausgehend von Ihren Überlegungen, direkt in SQL-DDL um.

```
CREATE DOMAIN SPIELRUNDE AS VARCHAR(50)
CHECK (VALUE IN (,Vorrunde', ,Viertelfinale', ,Halbfinale', ,Finale'));
```

```
CREATE TABLE Spiel (
```

```
    spiel_id          INTEGER NOT NULL,
    Mannschaft_a_id   INTEGER NOT NULL,
    Mannschaft_b_id   INTEGER NOT NULL,
    sieger_id         INTEGER,
    tore_a            INTEGER DEFAULT 0,
    tore_b            INTEGER DEFAULT 0,
    datum             TIMESTAMP,
    spielrunde        SPIELRUNDE,
    zuschauer         INTEGER DEFAULT 0 NOT NULL,
    stadion_id        INTEGER NOT NULL,
```

```
    → PRIMARY KEY (spiel_id)
    FOREIGN KEY (mannschaft_a_id) REFERENCES Mannschaft(mannschafts_id),
    FOREIGN KEY (mannschaft_b_id) REFERENCES Mannschaft(mannschafts_id),
    FOREIGN KEY (sieger_id) REFERENCES Mannschaft(mannschafts_id),
    FOREIGN KEY (stadion_id) REFERENCES Stadion(stadion_id)
```

```
);
```

So korrekt?

*„Mannschaft darf nicht
gegen sich selbst spielen“*

Aufgabe 2 – Modellierung mit SQL-DDL

d) Setzen Sie die Modellierung, ausgehend von Ihren Überlegungen, direkt in SQL-DDL um.

```
CREATE DOMAIN SPIELRUNDE AS VARCHAR(50)
CHECK (VALUE IN (,Vorrunde', ,Viertelfinale', ,Halbfinale', ,Finale'));
CREATE TABLE Spiel (
    spiel_id          INTEGER NOT NULL,
    Mannschaft_a_id  INTEGER NOT NULL,
    Mannschaft_b_id  INTEGER NOT NULL,
    sieger_id        INTEGER,
    tore_a           INTEGER DEFAULT 0,
    tore_b           INTEGER DEFAULT 0,
    datum            TIMESTAMP,
    spielrunde       SPIELRUNDE,
    zuschauer        INTEGER DEFAULT 0 NOT NULL,
    stadion_id       INTEGER NOT NULL,
    PRIMARY KEY (spiel_id)
    FOREIGN KEY (mannschaft_a_id) REFERENCES Mannschaft(mannschafts_id),
    FOREIGN KEY (mannschaft_b_id) REFERENCES Mannschaft(mannschafts_id),
    FOREIGN KEY (sieger_id) REFERENCES Mannschaft(mannschafts_id),
    FOREIGN KEY (stadion_id) REFERENCES Stadion(stadion_id),
    CHECK (mannschaft_a_id != mannschaft_b_id) ←
```

Aufgabe 2 – Modellierung mit SQL-DDL

Diskussion der Ergebnisse

Entitäten umgesetzt:

- Mannschaft
- Spieler
- Spiel
- Gruppe
- Stadion
- Tribünen
- Trainer
- Schiedsrichter
- Verwarnungen
- Erzielte Tore

Aufgabe 2 – Modellierung mit SQL-DDL

Diskussion der Ergebnisse

Informationsbedürfnisse umgesetzt:

- Welche Mannschaft ist nach der Gruppenphase weiter?
- Welcher Spieler führt die Torschützenliste an?
- Wann spielt Deutschland gegen USA?
- In welchem Stadion spielt Deutschland gegen USA?
- Wieviele Zuschauer können Spiel X maximal sehen?
Wieviele Zuschauer hatte Spiel X tatsächlich?
- Wie viele Stürmer hat Spanien im Kader?
- Welche Spieler sind für das nächste Spiel gesperrt?
- Welche Mannschaft hat die WM 2014 gewonnen?

Aufgabe 2 – Modellierung mit SQL-DDL

Diskussion der Ergebnisse

Integritätsbedingungen umgesetzt:

- Mannschaften dürfen nicht gegen sich selbst spielen.
- Jeder Spieler muss zu genau einer Mannschaft gehören.
- Der Kader einer Mannschaft besteht aus 20 Spielern.
- Die tatsächliche Zuschauerzahl in einem Stadion darf die maximale Kapazität nicht überschreiten.

Aufgabe 3 – Modellierung mit SQL-DDL

Verwenden Sie ein Datenbasisschema mit vier Relationen: Lieferant, Produkt, Kunde und Verträge. Sowohl die Lieferant- als auch die Kunde-Relation haben die Attribute Id, Name und Adresse. Ein Id ist eine Zahl mit neun Ziffern. Produkt hat eine Teilnummer (ein Integer zwischen 1 und 999999) und Name. Jedes Tupel in der Verträge-Relation beschreibt einen Vertrag zwischen einem Lieferanten und einem Kunden für ein bestimmtes Produkt mit einer bestimmten Anzahl und einem Preis.

- a) Benutzen Sie SQL-DDL, um das Schema für diese Relationen zu spezifizieren; geben Sie auch die SQL-Domänen an.
- b) Definieren Sie passende Integritätsbedingungen (Primär-, Kandidaten- und Fremd-Schlüssel).
- c) Geben Sie eine weitere Integritätsbedingung in natürlicher Sprache an und ergänzen Sie, soweit möglich ohne SQL-Query-Kenntnisse, das Schema.

Aufgabe 3a) – Lieferant und Kunde

Aufgabe 3a) – Lieferant und Kunde

```
CREATE DOMAIN ID_Domain AS INTEGER
    CHECK (VALUE >= 1 AND VALUE < 1000000000);

CREATE DOMAIN TID_Domain AS INTEGER
    CHECK (VALUE >= 1 AND VALUE < 1000000);

CREATE TABLE Lieferant (
    id ID_Domain NOT NULL,
    name VARCHAR(200),
    adresse VARCHAR(200),
    PRIMARY KEY (id) );

CREATE TABLE Kunde (
    id ID_Domain NOT NULL,
    name VARCHAR(200),
    adresse VARCHAR(200),
    PRIMARY KEY (id) );
```

Aufgabe 3a) – Produkt und Verträge

```
CREATE TABLE Produkt (  
    teilenummer TID_Domain NOT NULL,  
    name VARCHAR(200),  
    PRIMARY KEY (teilenummer) );
```

```
CREATE TABLE Vertraege (  
    vertrags_nr ID_Domain NOT NULL,  
    kunde ID_Domain NOT NULL,  
    lieferant ID_Domain NOT NULL,  
    produkt ID_Domain NOT NULL,  
    anzahl INTEGER,  
    preis DECIMAL (6,2),  
    PRIMARY KEY (vertrags_nr) );
```

Aufgabe 3b) – Integritätsbedingungen

Primärschlüssel bereits in 3a) angegeben. Fremdschlüssel sind in der Relation Verträge. Überlegen Sie das Verhalten bei Löschen/Ändern der durch Fremdschlüssel referenzierten Tupel.

```
CREATE TABLE Vertraege (  
    vertrags_nr ID_Domain NOT NULL,  
    kunde ID_Domain NOT NULL,  
    lieferant ID_Domain NOT NULL,  
    produkt TID_Domain,  
    anzahl INTEGER,  
    preis DECIMAL (6,2),  
    PRIMARY KEY (vertrags_nr),  
    FOREIGN KEY (kunde) REFERENCES Kunde (id)  
        ON DELETE CASCADE ON UPDATE CASCADE,  
    FOREIGN KEY (lieferant) REFERENCES Lieferant (id)  
        ON DELETE NO ACTION ON UPDATE CASCADE,  
    FOREIGN KEY (produkt) REFERENCES Produkt (teilenummer)  
        ON DELETE SET NULL ON UPDATE CASCADE );
```

Aufgabe 3c) – weitere Integritätsbedingungen

Geben Sie eine weitere Integritätsbedingung in natürlicher Sprache an und ergänzen Sie, soweit möglich ohne SQL-Query-Kenntnisse, das Schema.

- Die anzahl in Verträge muss größer Null sein.
- Es muss mindestens ein Kunde in der Datenbasis geben.
- Die Adresse muss eine gültige Postleitzahl enthalten.
- Die Anzahl Verträge eines Lieferanten darf nicht größer als 400 sein.

Aufgabe 4: ER-Modellierung

- Krankenhausszenario -

- **Vorgehensweise**
 - 1) Entity-Typen und Attribute identifizieren
 - 2) Beziehungstypen und deren Attribute finden
 - 3) Kardinalitäten festlegen
- **Hilfreich:**
 - Szenario komplett durchlesen
 - Entitäten und Beziehungen im Text markieren
- **Kontrolle zum Schluss:**
 - Alle Entitäten und Beziehungen modelliert?
 - Jede Kante mit Kardinalität versehen?

Aufgabe 4: ER-Modellierung

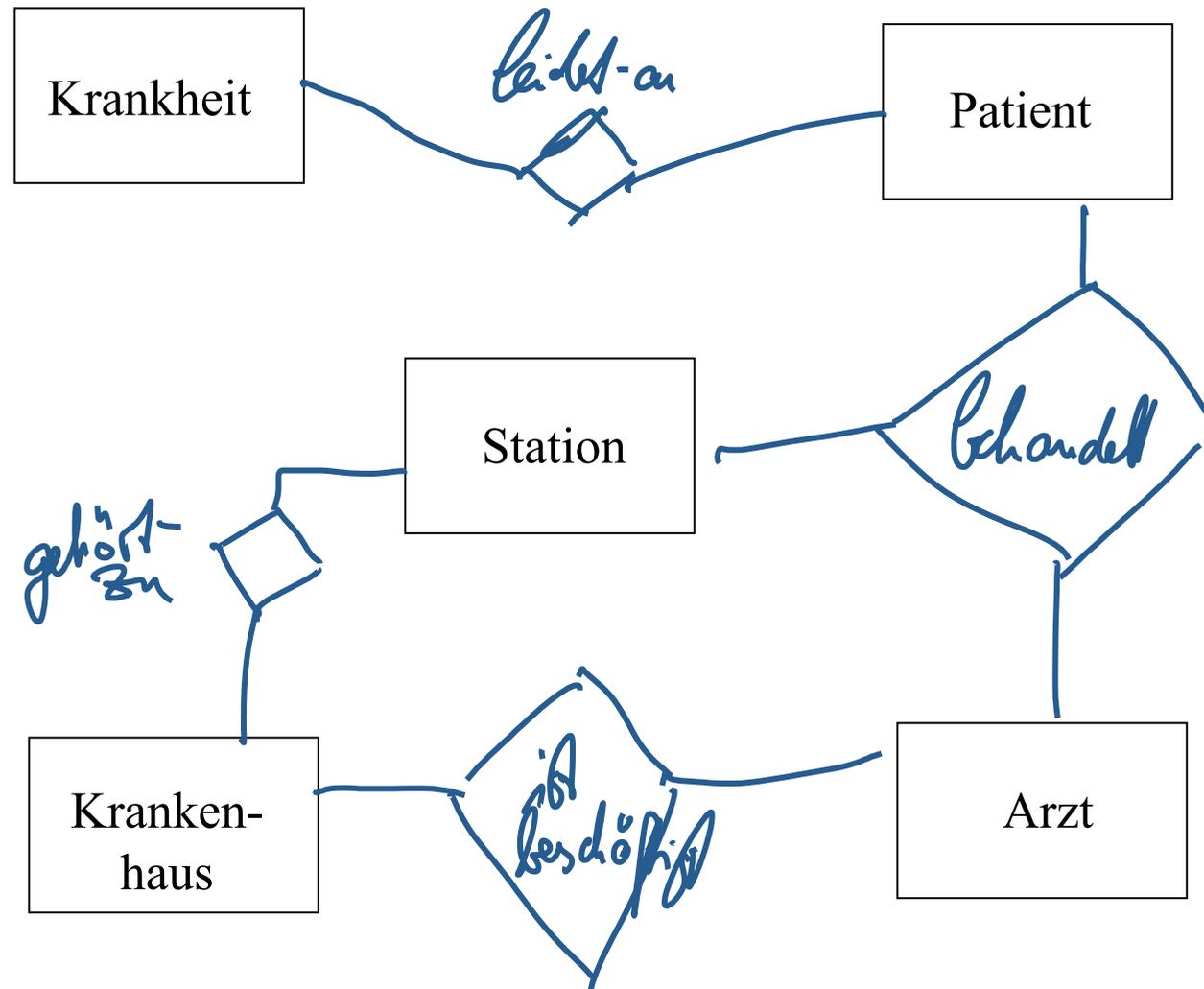
Schritt 1 – Entitytypen und Attribute

Entitytypen

- Ein Patient leidet an einer oder mehreren Krankheiten;
- es gibt jedoch Krankheiten, an denen niemand leidet.
- Jeder Patient wird genau auf einer Station behandelt.
- Ein Patient wird von einem oder mehreren Ärzten behandelt;
- jeder Arzt behandelt mindestens 10 und höchstens 50 Patienten, die auf verschiedenen Stationen liegen können.
- Eine Station kann leer sein,
- Jede Station gehört zu exakt einem Krankenhaus, und
- jedes Krankenhaus hat mindestens 5 Stationen.
- An einem Krankenhaus sind zwischen 5 und 200 Ärzten angestellt,
- Ein Arzt ist an mindestens einem und an höchstens 3 Krankenhäusern beschäftigt.

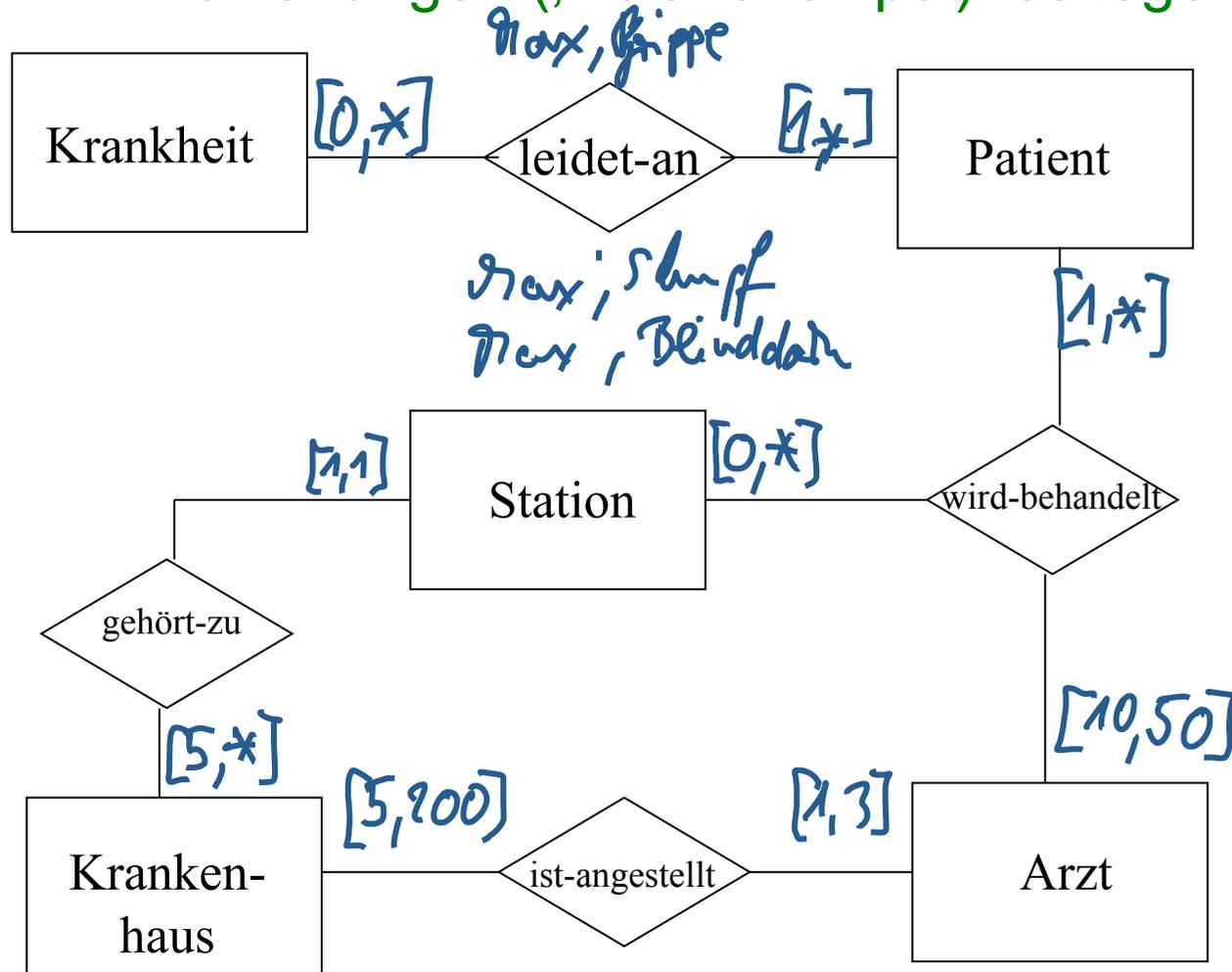
Aufgabe 4a): ER-Modellierung

- 1. Schritt: Entitäten bestimmen, *hier*: ohne Attribute



Aufgabe 4a): ER-Modellierung

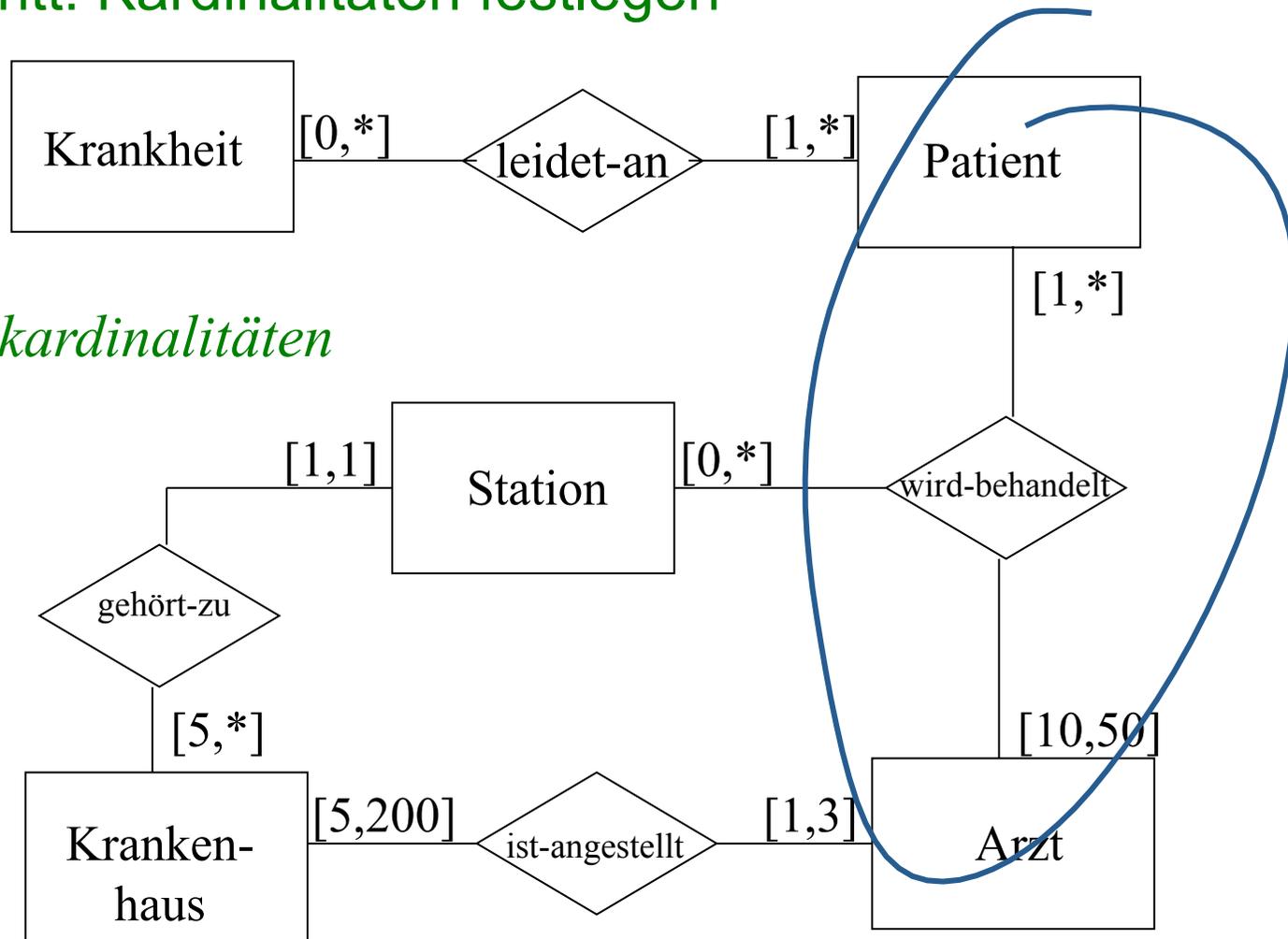
- 2. Schritt: Beziehungen („Relationships“) festlegen.



Aufgabe 4a): ER-Modellierung

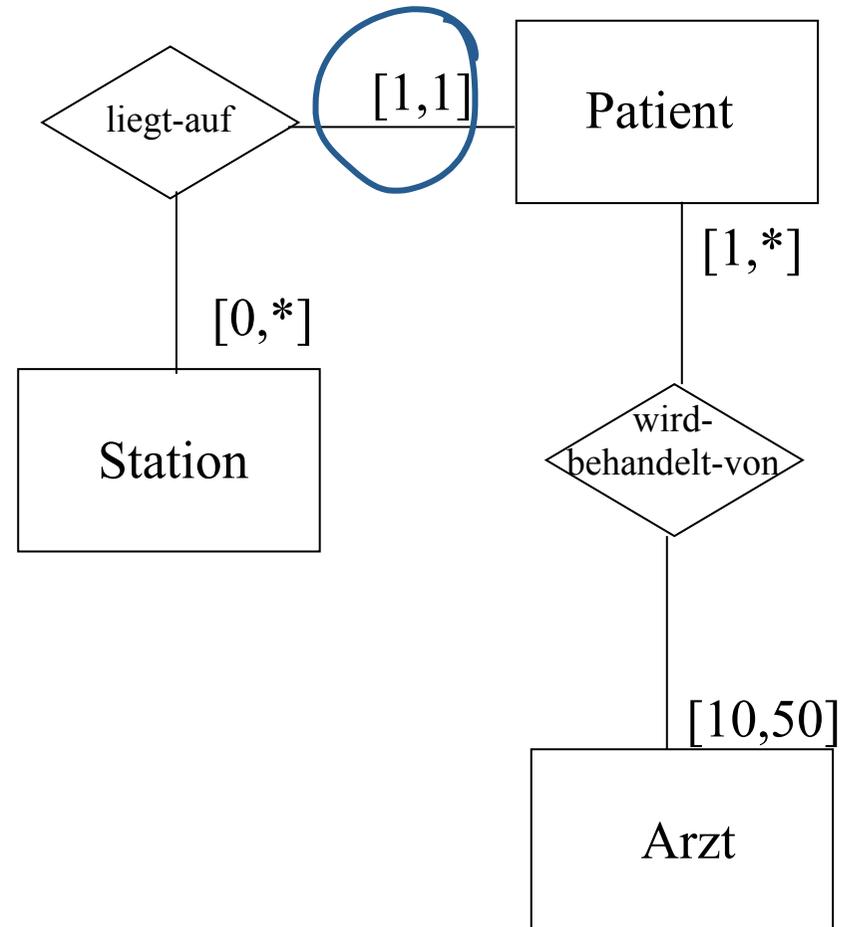
- 3. Schritt: Kardinalitäten festlegen

Teilnehmerkardinalitäten



Aufgabe 4b): ER-Modellierung - nicht darstellbare Eigenschaften

- Nicht im ER-Diagramm umgesetzt werden konnte z.B. die Aussage, dass ein Patient genau auf einer Station liegt.
- An Stelle der dreistelligen Beziehung ‚wird-behandelt‘ könnten hier auf Grund dieser speziellen Bedingung für die Beziehung zwischen Patient und Station auch zwei zweistellige Beziehungen (eine zwischen Patient und Station mit der Teilnehmerkardinalität $[1,1]$ bei Patient und eine zwischen Arzt und Patient mit den gleichen Teilnehmerkardinalitäten wie im dreistelligen Fall) verwendet werden.



Aufgabe 4b): ER-Modellierung

- nicht darstellbare Eigenschaften

- Eine dreistellige Beziehung lässt sich allerdings im Allgemeinen **nicht** mit der gleichen Bedeutung und Ausdrucksmächtigkeit auch durch zweistellige Beziehungen ersetzen.
- Wenn im Beispiel die Einschränkung, dass ein Patient nur auf einer Station liegt, aufgehoben wird, dann ist relevant, auf welcher Station die Behandlung des Patienten durch den Arzt erfolgt und dann wäre auch eine Dreier-Beziehung unumgänglich.

Datenbanksysteme

Übung 1

Sommersemester 2017

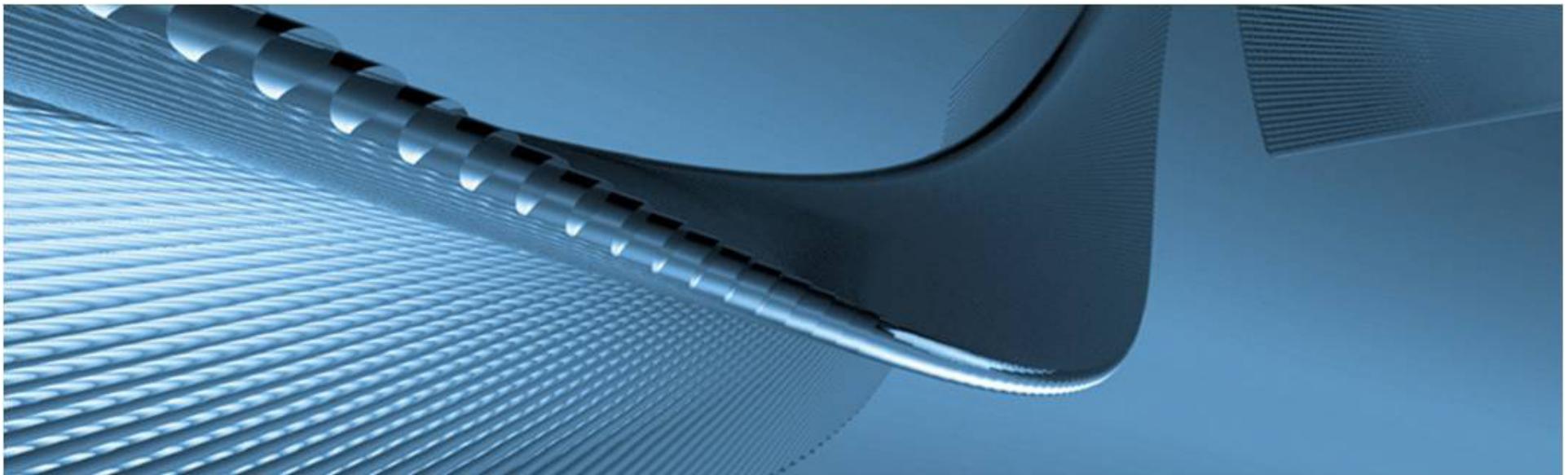
Jutta Mülle

Fakultät für Informatik

IPD, Lehrstuhl Prof. Klemens Böhm

muelle@kit.edu

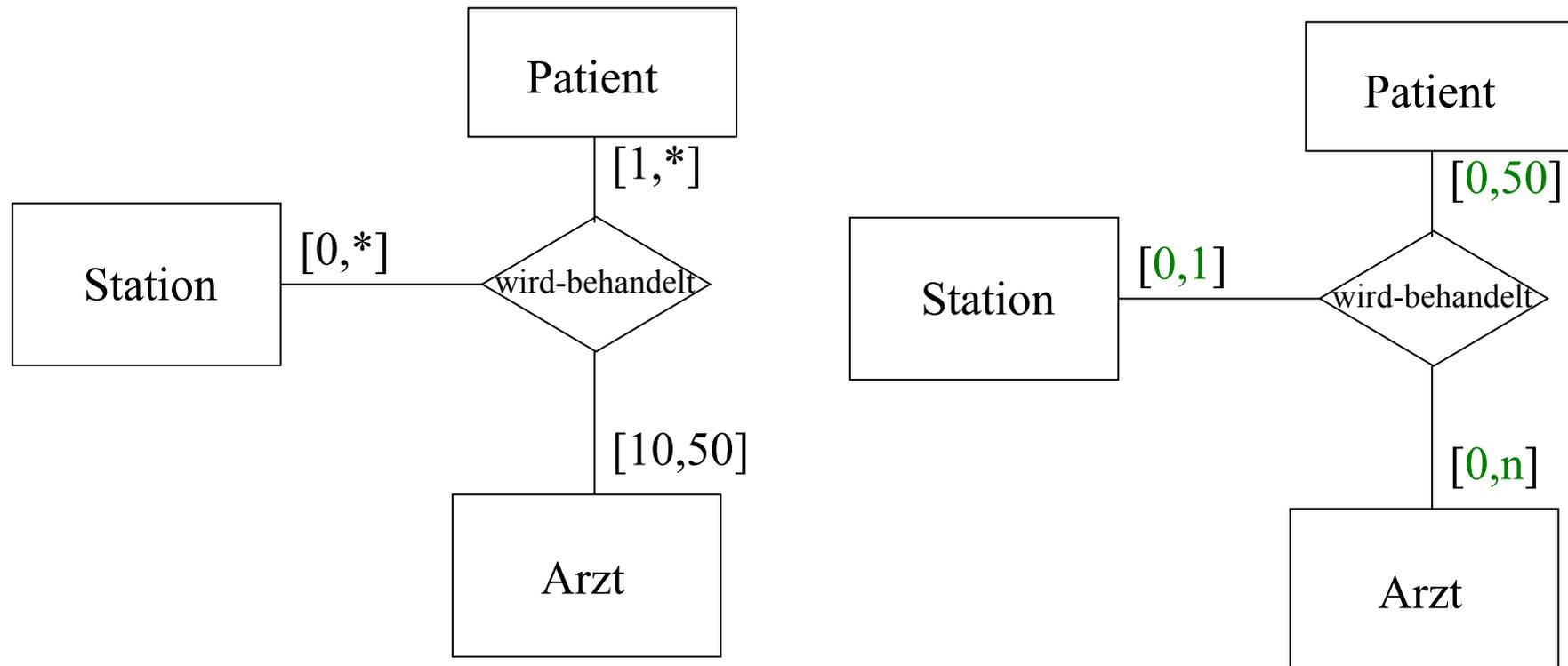
<http://dbis.ipd.kit.edu/>



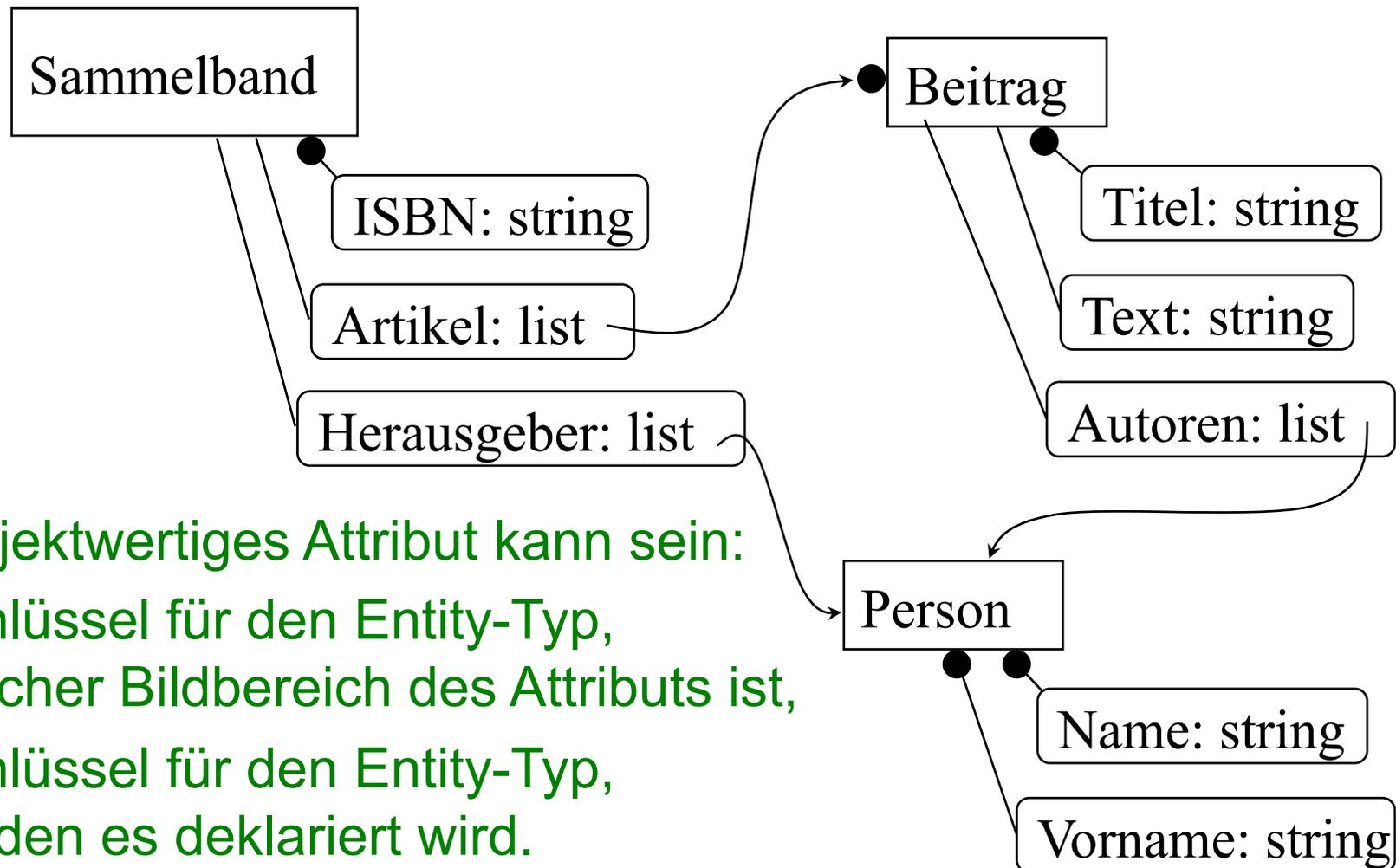
Aufgabe 4c): ER-Modellierung

- Standard- vs. Teilnehmerkardinalitäten

- Im Falle der dreistelligen Modellierung: Ist die Maximalzahl 50 für Patienten, die ein bestimmter Arzt behandelt, formulierbar?
- Bisher: Teilnehmerkardinalitäten.
- Was ist bei Verwendung von **Standardkardinalitäten**?



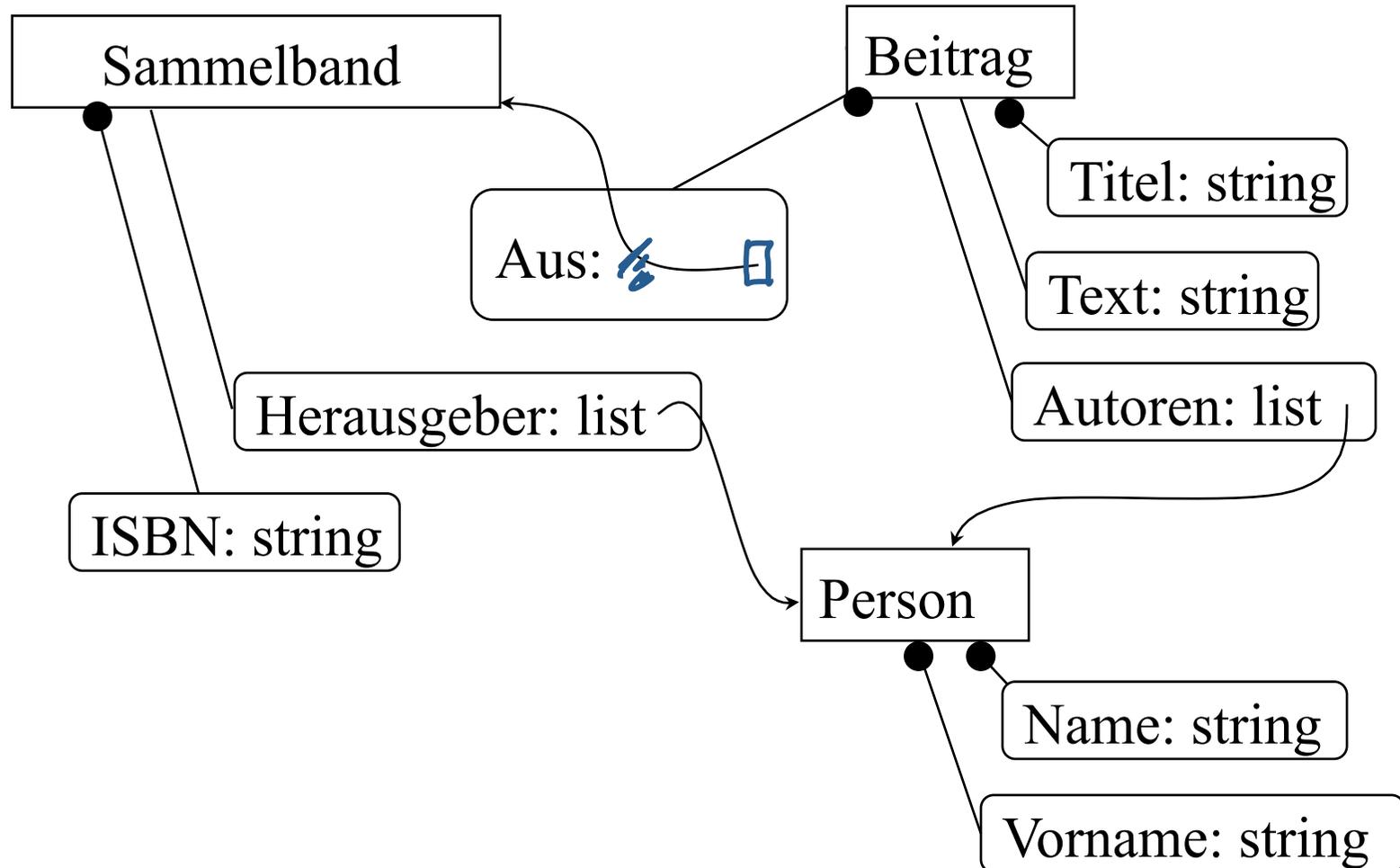
Objektwertige Attribute im EER-Modell und erweitertes Schlüsselkonzept (1)



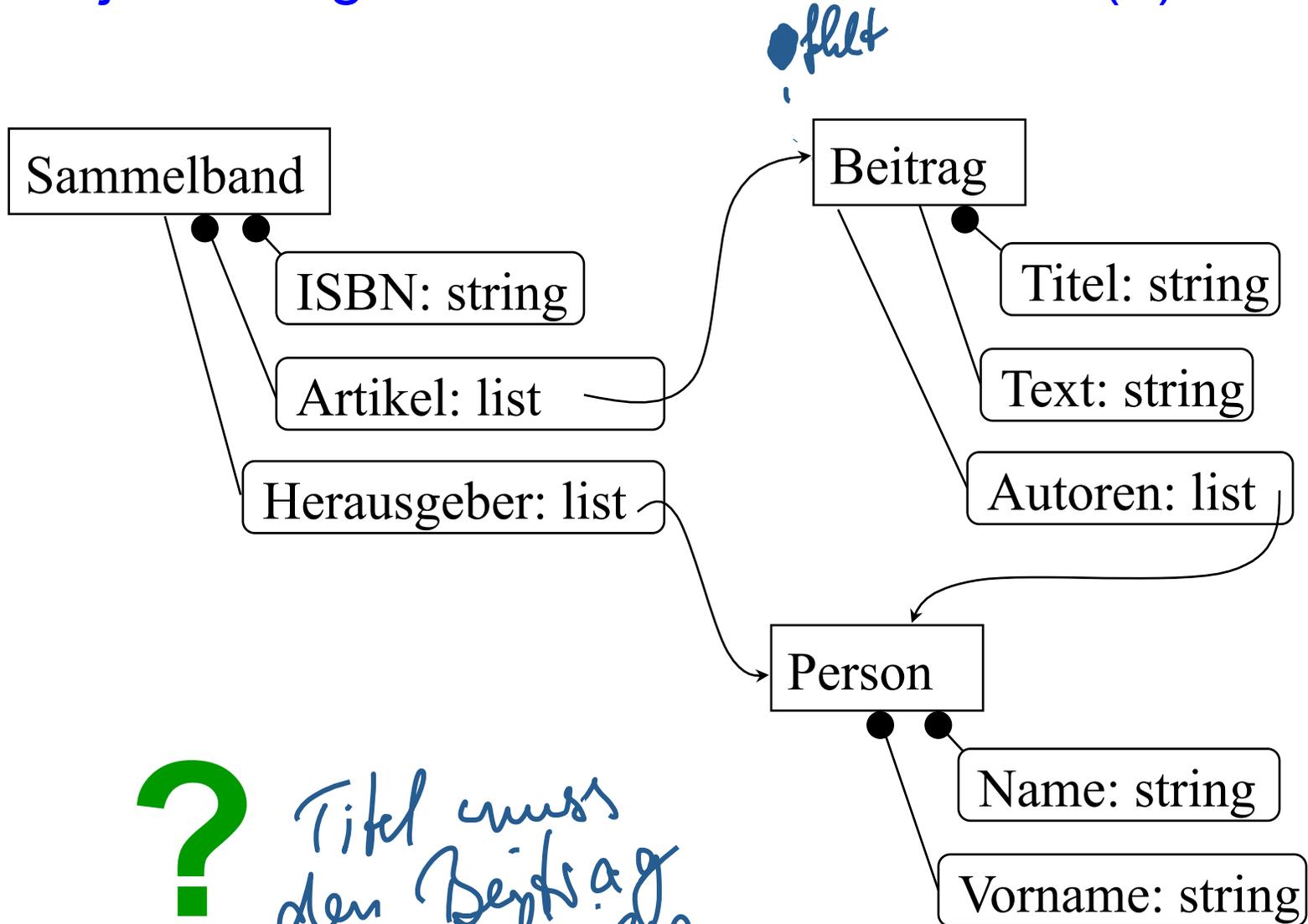
- Ein objektwertiges Attribut kann sein:
 - Schlüssel für den Entity-Typ, welcher Bildbereich des Attributs ist,
 - Schlüssel für den Entity-Typ, für den es deklariert wird.
- Nicht durch Unterstreichen darstellbar.

Objektwertige Attribute im EER-Modell (2)

Objektwertiges Attribut als **Schlüsselattribut**:

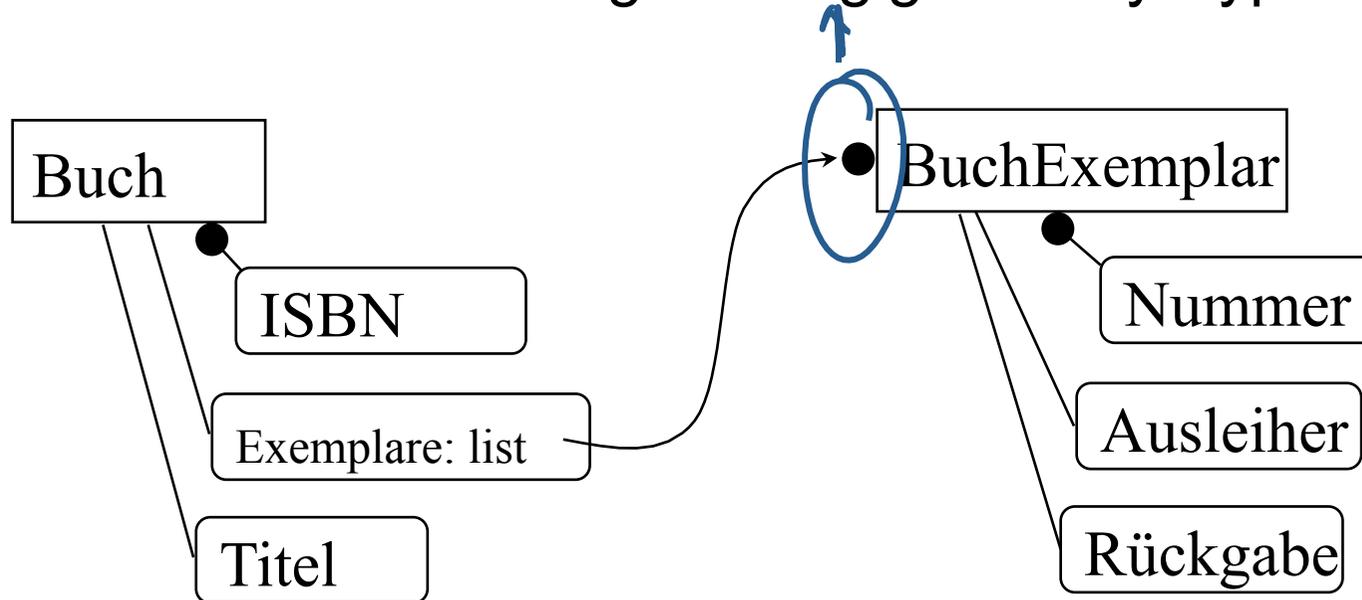


Objektwertige Attribute im EER-Modell (3)



Objektwertige Attribute im EER-Modell (4) – Erweitertes Schlüsselkonzept

- Einsatz zur Modellierung abhängiger Entity-Typen:

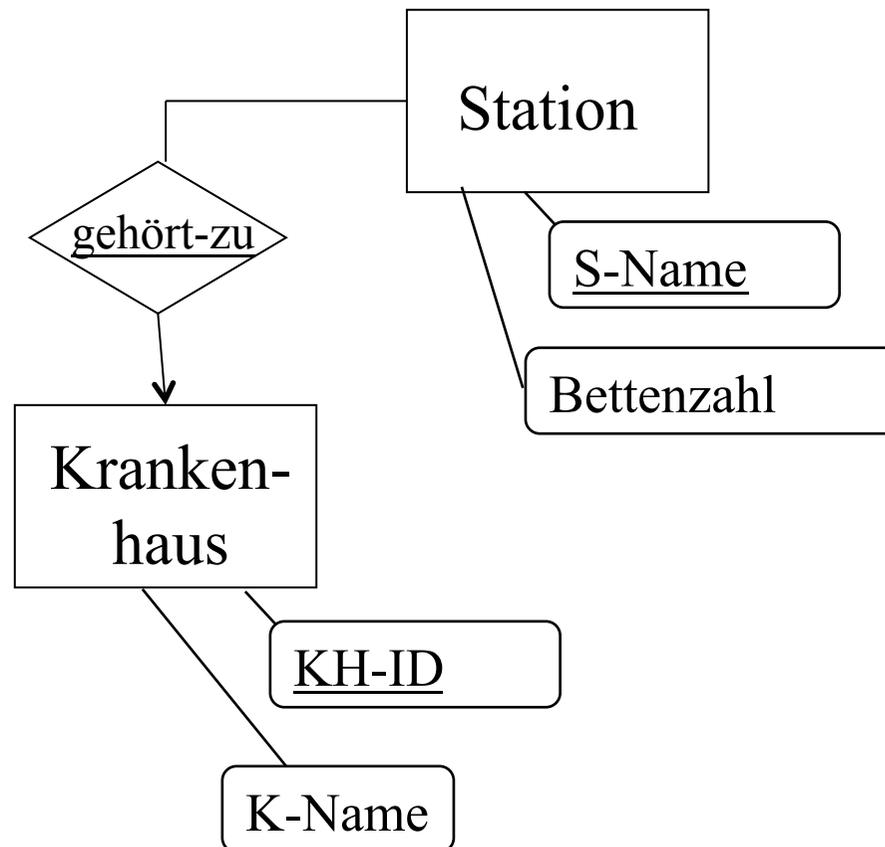


Aufgabe 4d): ER-Modellierung

- Schlüssel in EER: erweitertes Konzept

1. Hinzufügen von Attributen
2. Explizite Modellierung der Abhängigkeit des Entitätstyps Station. „Eine Station kann nur bestehen (existieren), wenn es das Krankenhaus gibt.“

ER



Aufgabe 4d): ER-Modellierung

Schlüssel in EER

- Explizite Modellierung des abhängigen Entitätstyps Station mit objektwertigen Attributen und erweitertem Schlüsselkonzept in EER.
- „Eine Station kann nur bestehen (existieren), wenn es das Krankenhaus gibt.“

EER

