Datenbanksysteme I

Wintersemester 2023/24

—

Übungsblatt 3

Abteilung Datenbanken
Institut für Informatik
Universität Leipzig

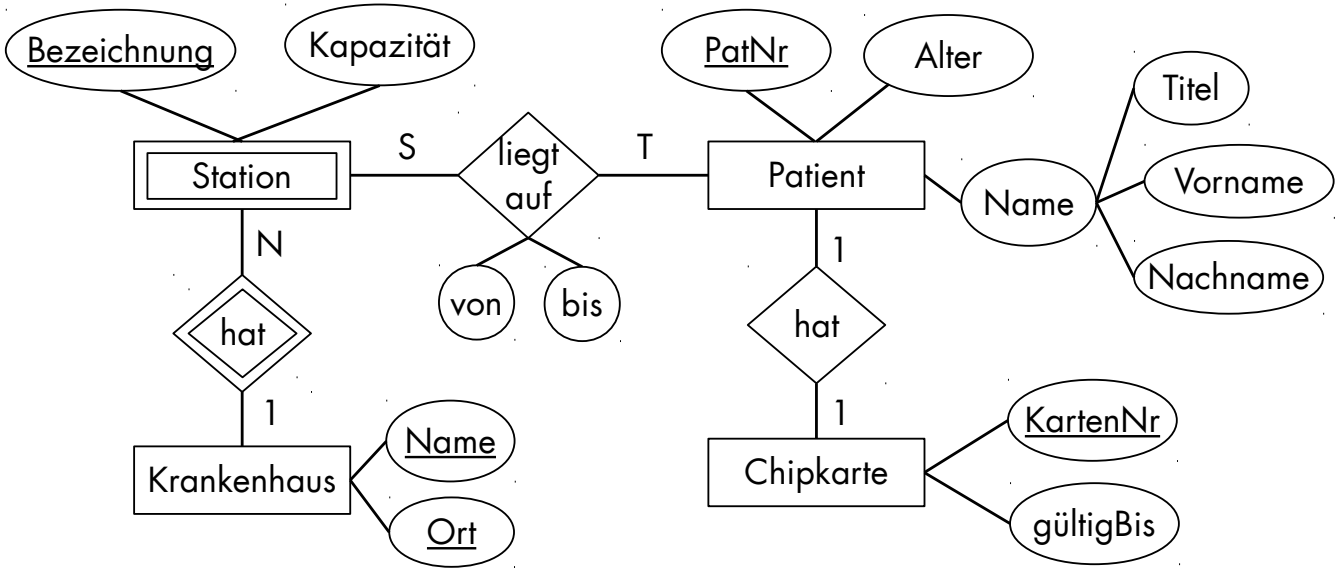
Wiederholung

- Konzeptionelles Schema: Bietet logische Sicht auf die Struktur der Daten
  - Welche Daten sollen in der DB gespeichert werden?
  - Welche Beziehungen bestehen zwischen den Daten?
  - Welche Integritätsbedingungen bestehen?

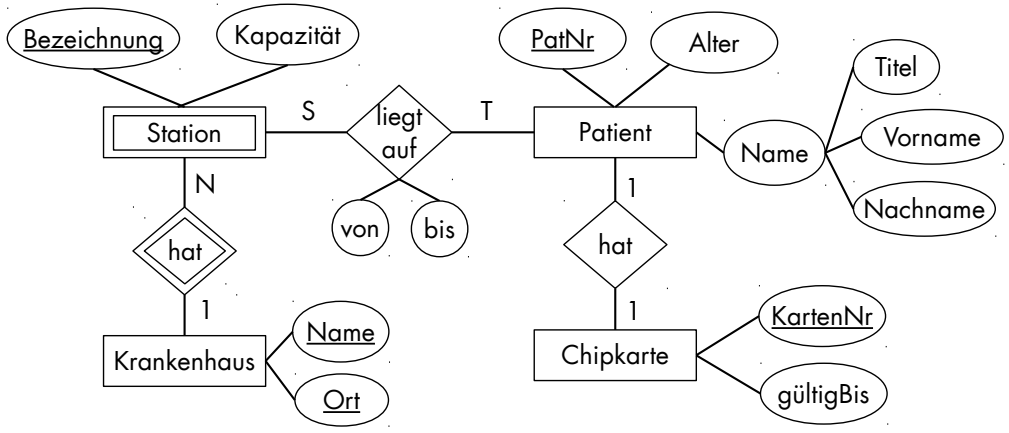
- Modellierung der relevanten Miniwelt: Bestandteil des **DB-Entwurfs**
  - UML-Modell
  - E/R-Modell (Chen)
  - **Relationenmodell** (Codd)

Aufgabe 1: ERM → Relationenmodell

(a) Überführen Sie das nachfolgende ER-Modell in ein relationales Datenbankschema. Kennzeichnen Sie Primär- und Fremdschlüssel.



Aufgabe 1: ERM → Relationenmodell



- **Station**(KName, KOrt, Bezeichnung, Kapazität (KName, KOrt):FS auf Krankenhaus(Name, Ort))
- **Krankenhaus**(Name, Ort)
- **Patient**(PatNr, Alter, KartenNr UNIQUE NOT NULL, KartenNr:FS auf Chipkarte(KartenNr))
- **PatientName**(PatNr,Titel, Vorname, Nachname, PatNr:FS auf Patient(PatNr))
- **Chipkarte**(KartenNr, gültigBis)
- **LiegtAufStation**(PatNr,KName, KOrt, Bezeichnung, von, bis, PatNr:FS auf Patient(PatNr), (KName, KOrt, Bezeichnung):FS auf Station(KName, KOrt, Bezeichnung))

Aufgabe 1: ERM → Relationenmodell

(b) Überführen Sie das relationale Datenbankschema in eine DDL-Spezifikation in SQL.

```
CREATE TABLE Krankenhaus(
    Name            VARCHAR(255)        NOT NULL,
    Ort             VARCHAR(255)        NOT NULL,
    PRIMARY KEY(Name, Ort))

CREATE TABLE Station(
    KName           VARCHAR(255)        NOT NULL,
    KOrt            VARCHAR(255)        NOT NULL,
    Bezeichnung     VARCHAR(255)        NOT NULL,
    Kapazitaet      INT,
    PRIMARY KEY(KName, KOrt, Bezeichnung),
    FOREIGN KEY(KName, KOrt) REFERENCES Krankenhaus(Name, Ort)
        ON DELETE CASCADE ON UPDATE CASCADE)
CREATE TABLE Chipkarte(
    KartenNr        INT                 PRIMARY KEY,
    gueltigBis      TIMESTAMP           NOT NULL)

CREATE TABLE Patient(
    PNR             INT                 PRIMARY KEY,
    Alter           INT                 NOT NULL,
    KartenNr        INT                 UNIQUE NOT NULL,
    FOREIGN KEY(KartenNR) REFERENCES Chipkarte(KartenNr))
```

Aufgabe 1: ERM → Relationenmodell

(b) Überführen Sie das relationale Datenbankschema in eine DDL-Spezifikation in SQL.

```
CREATE TABLE PatientName(
    PatNr           INT                 PRIMARY KEY,
    Titel           VARCHAR(50),
    Vorname         VARCHAR(100)        NOT NULL,
    Nachname        VARCHAR(100),       NOT NULL,
    FOREIGN Key(PatNr) REFERENCES Patient(PatNr))

CREATE TABLE PatientLiegtAuf(
    PatNr           INT                 NOT NULL,
    KName           VARCHAR(255)        NOT NULL,
    KOrt            VARCHAR(255)        NOT NULL,
    Bezeichnung     VARCHAR(255)        NOT NULL,
    von             DATE                NOT NULL,
    bis             DATE                NOT NULL,
    PRIMARY KEY(PatNr, KName, KOrt, Bezeichnung, von),
    FOREIGN KEY(PatNr) REFERENCES Patient(PatNr),
    FOREIGN KEY(KName, KOrt, Bezeichnung) REFERENCES
        Station(KName, KOrt, Bezeichnung)
```

Aufgabe 1: ERM → Relationenmodell

(c) Beurteilen Sie, inwieweit Kardinalitätsrestriktionen im Relationenmodell umgesetzt werden können.

- $1:[0..1]$-Beziehungen
  - **Mitarbeiter**(MNr:PK, ANr:FK REF Ausweis(ANr), ...)
    Ausweis(ANr:PK, gültigBis)
  - **Mitarbeiter**(MNr:PK, ANr:FK REF Ausweis(ANr) UNIQUE NOT NULL, ...)
    Ausweis(ANr:PK, gültigBis)

- $1:[0..N]$- und $1:[1..N]$-Beziehungen
  - **Kunde**(KNr:PK, ...)
    Konto(IBAN:PK, KNr:FK REF Kunde(Knr))
  - **Kunde**(KNr:PK, ...)
    Konto(IBAN:PK, KNr:FK REF Kunde(Knr) NOT NULL)

- $1:1$- und $1:X$-Beziehungen
  - 1:X - Professor hält genau 2 Vorlesungen
  - Nur über **Trigger** realisierbar (Kapitel 8)

Aufgabe 2: Referentielle Integrität

(a) Erläutern Sie den Begriff *Referentielle Integrität*.

- Beziehungen werden im RM durch Fremdschlüssel realisiert
- Fremdschlüssel
  - Attribut, das in Bezug auf den Primärschlüssel einer anderen (oder derselben) Relation definiert ist (gleicher Wertebereich)
- **Referentielle Integrität**:
  - Zu jedem Fremdschlüssel in R gibt es einen Primärschlüssel in S, sodass für jeden Wert der Attribute Wertgleichheit vorliegt.

  $=$ Zu jedem Wert (ungleich NULL) eines Fremdschlüsselattributs in Relation R muss ein gleicher Wert des Primärschlüssels in irgendeinem Tupel von Relation S vorhanden sein

  $\rightarrow$ Zu jedem Zeitpunkt, d. h. auch nach Änderungsoperationen

Aufgabe 2: Referentielle Integrität

(b) Ergänzen Sie die Fremdschlüssel-Definitionen im relationalen Schema um geeignete Lösch-Regeln, welches folgendes sicherstellen:

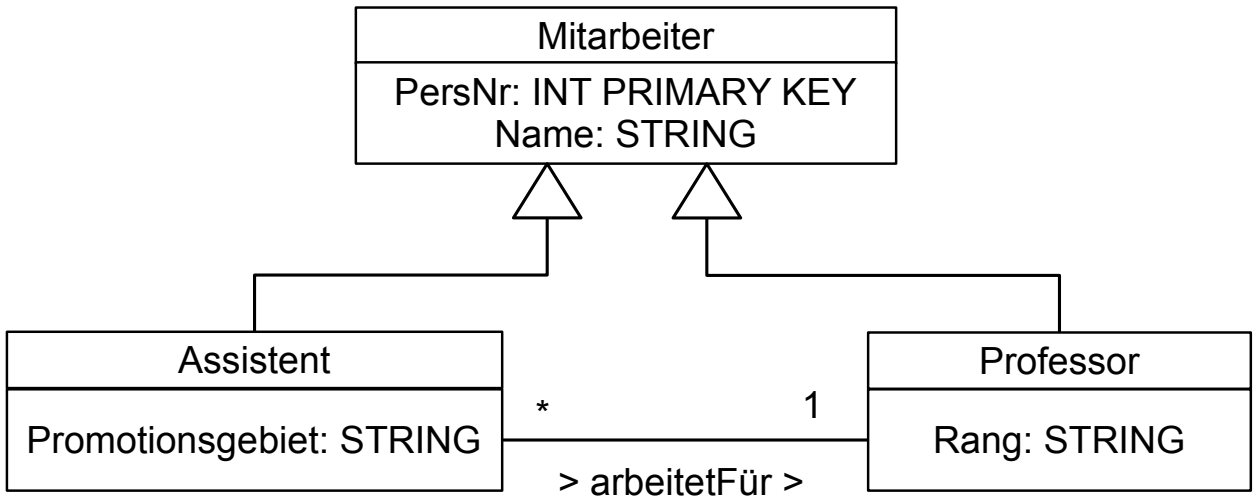
1. Scheidet ein Angestellter aus (d. h. wird er gelöscht), so wird auch die Information gelöscht, an welchen Projekten er mitgearbeitet hat.
2. Wird ein Projekt gelöscht, so auch alle Informationen, welche Mitarbeiter mit wie vielen Stunden dort gearbeitet haben.
3. Das Löschen eines Projektleiters soll zurückgewiesen werden.

```
CREATE TABLE Angestellter(ANr INT PRIMARY KEY, AName VARCHAR)

CREATE TABLE Projekt(
    PNr INT PRIMARY KEY, PName VARCHAR(30),
    Projektleiter INT NOT NULL,
    FOREIGN KEY (Projektleiter) REFERENCES Angestellter(ANr)
    ON DELETE NO ACTION
)

CREATE TABLE Mitarbeit(
    ANr INT, PNr INT, Arbeitsstd INT, PRIMARY KEY(ANr, PNr),
    FOREIGN KEY (ANr) REFERENCES Angestellter
    ON DELETE CASCADE,
    FOREIGN KEY (PNr) REFERENCES Projekt
    ON DELETE CASCADE
)
```

Aufgabe 3: Generalisierung im Relationenmodell

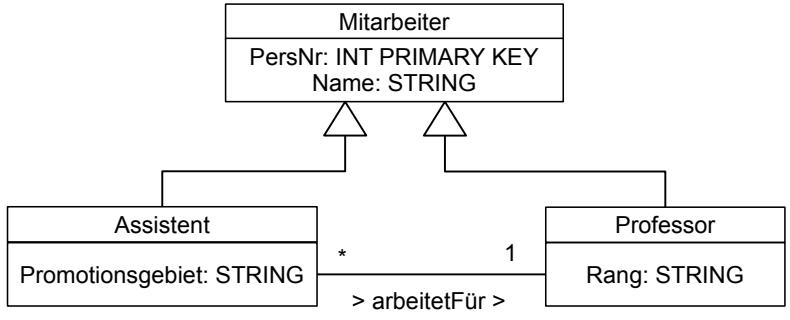
Geben Sie für die angegebene Generalisierungsbeziehung alle in der Vorlesung vorgestellten Varianten zur Überführung in das Relationenmodell an. Verwenden Sie dafür folgende Instanzen der Klassen Assistent, Professor und Mitarbeiter:

- Professor: { PersNr: 123, Name: Rahm, Rang: C4 }
- Assistent: { PersNr: 1234, Name: Christen, Promotionsgebiet: Ontologies, arbfür:123}
- Assistent: { PersNr: 1235, Name: Franke, Promotionsgebiet: Privacy, arbfür:123}
- Mitarbeiter: { PersNr: 1236, Name: Hesse }



Aufgabe 3: Generalisierung im Relationenmodell

- Professor: { PersNr: 123, Name: Rahm, Rang: C4 }
- Assistent: { PersNr: 1234, Name: Christen, Promotionsgebiet: Ontologies, arbfür:123}
- Assistent: { PersNr: 1235, Name: Franke, Promotionsgebiet: Privacy, arbfür:123}
- Mitarbeiter: { PersNr: 1236, Name: Hesse }



Vertikale Partitionierung

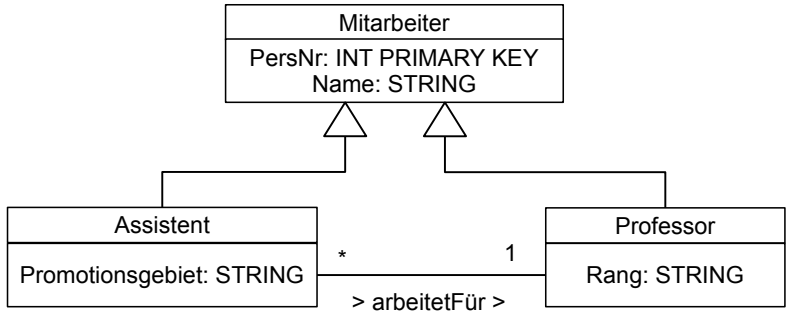
Mitarbeiter	
PersNr	**Name**
123	Rahm
1234	Christen
1235	Franke
1236	Hesse

Professor	
PersNr	**Rang**
123	C4

Assistent		
PersNr	**Pr.Gebiet**	**arb.Für**
1234	Ontologies	123
1235	Privacy	123

Aufgabe 3: Generalisierung im Relationenmodell

- Professor: { PersNr: 123, Name: Rahm, Rang: C4 }
- Assistent: { PersNr: 1234, Name: Christen, Promotionsgebiet: Ontologies, arbfür:123}
- Assistent: { PersNr: 1235, Name: Franke, Promotionsgebiet: Privacy, arbfür:123}
- Mitarbeiter: { PersNr: 1236, Name: Hesse }



Horizontale Partitionierung

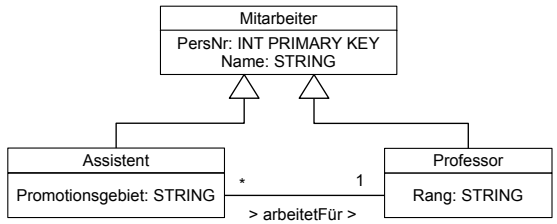
Mitarbeiter	
PersNr	**Name**
1236	Hesse

Professor		
PersNr	**Name**	**Rang**
123	Rahm	C4

Assistent			
PersNr	**Name**	**Pr.Gebiet**	**arb.Für**
1234	Christen	Ontologies	123
1235	Franke	Privacy	123

Aufgabe 3: Generalisierung im Relationenmodell

- Professor: { PersNr: 123, Name: Rahm, Rang: C4 }
- Assistent: { PersNr: 1234, Name: Christen, Promotionsgebiet: Ontologies, arbfür:123}
- Assistent: { PersNr: 1235, Name: Franke, Promotionsgebiet: Privacy, arbfür:123}
- Mitarbeiter: { PersNr: 1236, Name: Hesse }



Volle Redundanz

Mitarbeiter	
PersNr	**Name**
123	Rahm
1234	Christen
1235	Franke
1236	Hesse

Professor		
PersNr	**Name**	**Rang**
123	Rahm	C4

Assistent			
PersNr	**Name**	**Pr.Gebiet**	**arb.Für**
1234	Christen	Ontologies	123
1235	Franke	Privacy	123