

# Datenbanksysteme I

## WS 18/19

### Teillösung Übungsblatt 1-3

Universität Leipzig, Institut für Informatik  
Abteilung Datenbanken  
Prof. Dr. E. Rahm,  
V. Christen, M. Franke

## Aufgabe 2: DBVS vs. Dateisystem

- die Ansicht der Artikelliste soll nach dem Lagerbestand sortiert werden
  - neue Sortieroperation implementieren
  - Verbund eigenständiger Daten
- mehrere Personen für Wareneingang
  - Sicherstellen der Mehrbenutzerfähigkeit bei parallelen Schreibzugriffen
- separates Programm arbeitet auf Kopie der Artikelliste
  - Redundanz (Synchronisieren von Änderungen)

## Aufgabe 2: DBVS vs. Dateisystem (2)

- Ergänzung Artikelliste um Attribute und Erweiterung der Artikelnummerierung
  - Änderungen in allen Programmen, die auf die Daten zugreifen, erforderlich (auch wenn neue Attribute nicht benötigt)
  - Neues Artikelnummernformat muss in allen Dateien eingeführt werden, die auf die Artikelliste verweisen
- Serverausfall
  - Unvollständige Schreiboperationen können zu zerstörten Daten führen; Probleme
- Firmenfusion
  - Es können sehr große Dateien entstehen, welche nicht mit herkömmlichen Programmen verwaltet werden können.

## Aufgabe 4: 3-Ebenen-Architektur

- a) Ein Mitarbeiter verlässt die Abteilung Datenbanken.  
**Kein Modell betroffen.**
- b) Abteilungsleiter sollen zusätzlich zu den Daten der Angestellten ihrer Abteilung auch die Daten der Angestellten anderer Abteilungen, die an einem Projekt ihrer Abteilung mitarbeiten, lesen können.  
**Externes Modell betroffen.**
- c) Der Lesezugriff auf Projekttitel ist zu langsam und wird durch Anlegen eines Zugriffspfades (Index) beschleunigt.  
**Internes Modell betroffen.**
- d) Durch Reorganisation der Gehaltsabrechnung werden Angestellendaten statt in alphabetischer Reihenfolge sortiert nach Gehaltsgruppen benötigt.  
**Externes Modell betroffen.**
- e) Ein Anwendungsprogramm zur Anzeige der Liste aller Projekte mit ihren zugehörigen Angestellten soll hinsichtlich des Datenschutzes keine Adressdaten der Angestellten darstellen.  
**Externes Modell betroffen.**
- f) Eine Abteilung wird aufgelöst und alle ehemaligen Angestellten werden anderen Abteilungen zugewiesen.  
**Kein Modell betroffen.**

## Aufgabe 5b: Transaktionen

1. Aufgrund eines Stromausfalls wurde ein Transaktionsprogramm unterbrochen. Die Änderungen, die durch das Transaktionsprogramm vor dem Abbruch durchgeführt worden waren, gehen nicht verloren.  
**Falsch:** Wegen der Atomarität wird dem Anwendungsprogramm entweder der Zustand vor BOT oder der Zustand nach EOT zugesichert. Zwischenzustände gehen bei Fehlern verloren und werden nicht sichtbar. Daher wird hier der Zustand bei BOT wieder restauriert.
2. Das Datenbanksystem gewährleistet, dass in einer Transaktion nur die aktuellste Version der Daten, die von einer anderen Transaktion gerade modifiziert werden, gelesen werden kann.  
**Falsch:** Wegen der Isolationseigenschaft sind Daten, die von anderen Transaktionen modifiziert werden, nicht zu lesen.
3. Eine Online-Auktionsbörse verwende eine Datenbank zur Verwaltung der Auktionsdaten (z. B. Gebote). Mit den Mitteln des DBVS kann das Abgeben eines Gebotes, welches das bisher höchste Gebot unterschreitet, verhindert werden.  
**Richtig:** Durch Definition von Integritätsbedingungen. Logische Integritätsbedingungen werden durch Transaktion gewahrt (Consistency).
4. Nach dem Absturz des DBS wurden die Änderungen aller erfolgreich abgeschlossenen Transaktionen zurückgesetzt, so dass der DB-Zustand nun konsistent bleibt.  
**Falsch:** Wegen der Dauerhaftigkeit erfolgreich abgeschlossener Transaktionen dürfen deren Änderungen nicht rückgängig gemacht werden.

## Aufgabe 5d: Sicherung IB durch AWP

### **Nachteile:**

- AWP müssen Fehlererkennung und Fehlerbehandlung durchführen (Komplexität)
- Informationssystem von der Korrektheit der AWP abhängig
- korrekte Datenzugriffe nicht ohne AWP möglich
- Hinzufügen neuer IB erfordert Änderung aller AWP

### **Vorteile:**

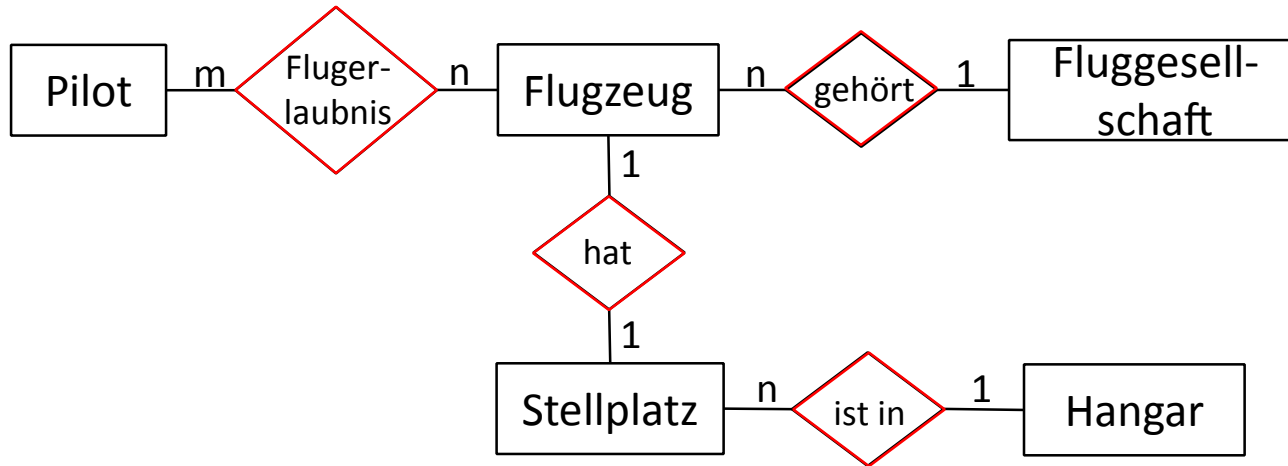
- weniger Overhead im Falle vieler zur Laufzeit zu überprüfender Integritätsbedingungen
- IB modellierbar, die mit dem bei DBMS schwer oder nicht umgesetzt werden können (temporale IB bei Zeitreihen)

## Aufgabe 2b: Schlüsselkandidaten

Name	Mobilfunknummer	Festnetznummer der WG
Vorname	Matrikelnummer	Universität
Geburtsdatum	Email-Adresse	Homepage

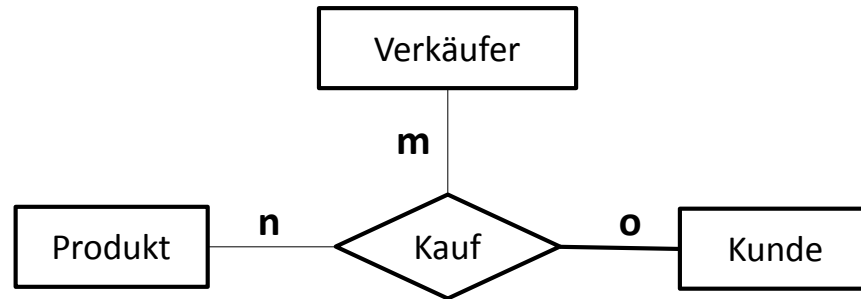
- Sichere Schlüsselkandidaten
  - Matrikelnummer
  - Handy-Nr (vorausgesetzt jeder Student hat ein eigenes Mobiltelefon)
  - Email-Adresse (vorausgesetzt jedem Student wird eine bei der Immatrikulation vergeben)
- Mögliche Schlüsselkandidaten
  - Name + Vorname + Geburtsdatum ?
- Kein Schlüsselkandidat
  - Matrikelnummer + Name (Minimalität!)

# Aufgabe 3a: Abbildungstypen



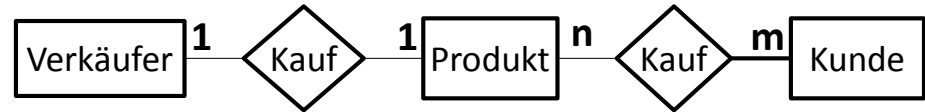
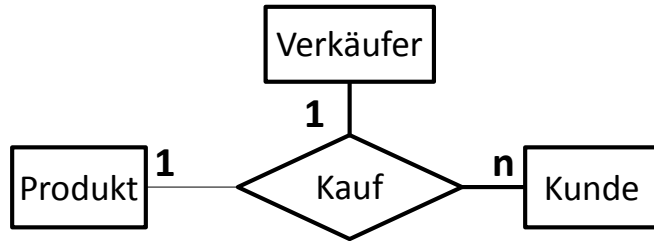


## Aufgabe 3b: N-stellige Relationship-Mengen



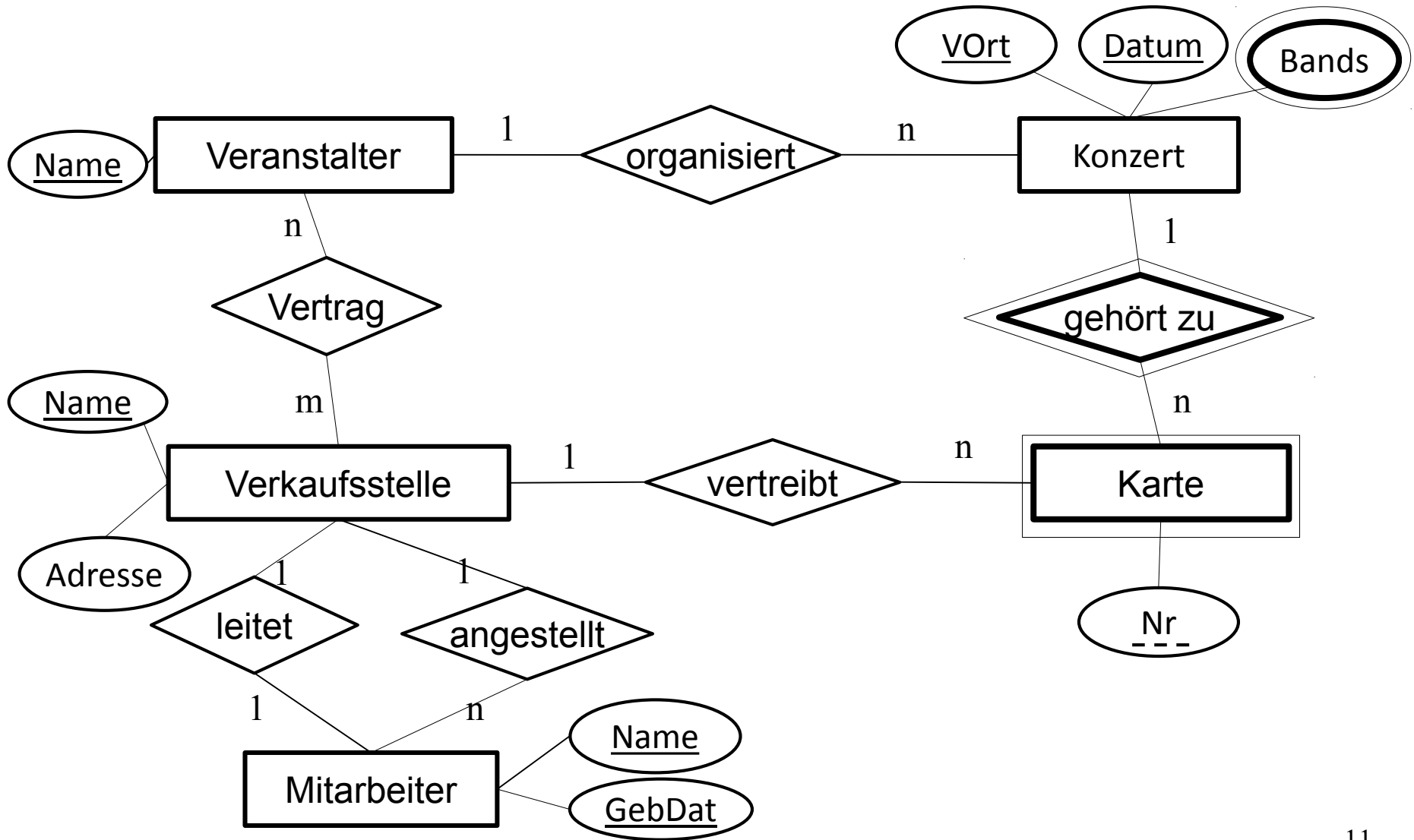
- Verkäufer  $\times$  Produkt  $\rightarrow$  o Kunden  
*„Für ein konkretes Paar (Verkäufer, Produkt) gibt es o Kunden.“*  
 *(„Ein Verkäufer verkauft das gleiche Produkt an verschiedene Kunden.“ )*
- Kunde  $\times$  Produkt  $\rightarrow$  m Verkäufer  
*„Für ein konkretes Paar (Kunde, Produkt) gibt es m Verkäufer.“*  
 *(„Ein Kunde kauft das gleiche Produkt bei verschiedenen Verkäufern. “ )*
- Kunde  $\times$  Verkäufer  $\rightarrow$  n Produkte  
*„Für ein konkretes Paar (Kunde, Verkäufer) gibt es n Produkte.“*  
 *(„Ein Kunde kauft beim gleichen Verkäufer verschiedene Produkte. “ )*

## Aufgabe 3c: N-stellige Relationship-Mengen

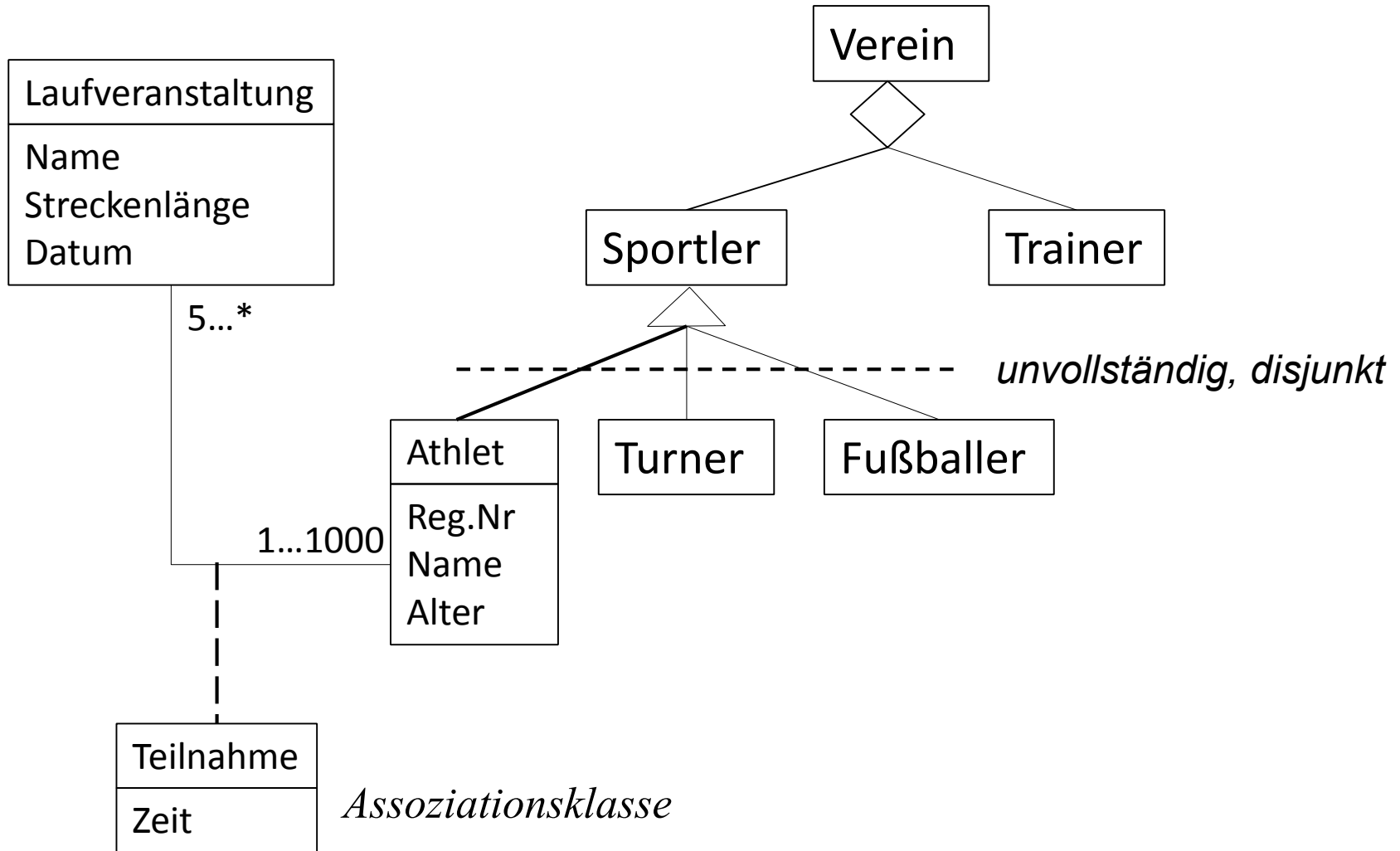


- Verkäufer × Produkt → n Kunden
- Kunde × Produkt → 1 Verkäufer
- Kunde × Verkäufer → 1 Produkte
- Produkt → m Kunden
  - damit Verkäufer × Produkt → m Kunden
- Produkt → 1 Verkäufer
  - damit Kunde × Produkt → 1 Verkäufer
- Verkäufer → 1 Produkt
  - damit Kunde × Verkäufer → 1 Produkt

# Aufgabe 4: ER-Diagramm

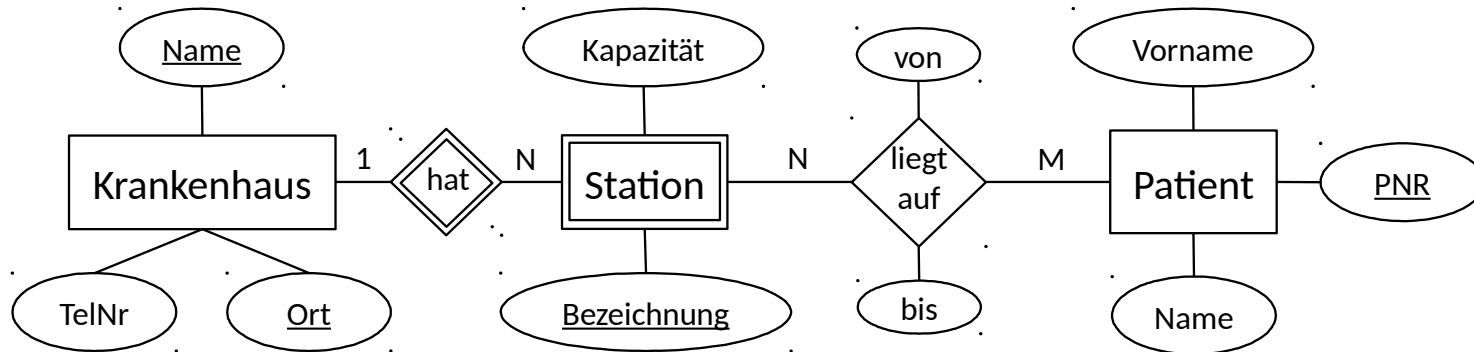


# Aufgabe 5a-c: UML



## Aufgabe 2: ERM → Relationenmodell

Überführen Sie das nachfolgende ER-Modell in ein relationales Schema.  
Kennzeichnen Sie Primär- und Fremdschlüssel.



**Patient**(PNR, Vorname, Nachname)

**Krankenhaus**(Name, Ort, TelNr)

**Station**(KName, KOrt, Bezeichnung, Kapazität)

FK (Kname, KOrt) auf Krankenhaus

**liegtAuf**(PNR, KName, KOrt, Bezeichnung, von, bis)

FK (PNR) auf Patient

FK (Kname, KOrt) auf Krankenhaus

FK (Bezeichnung) auf Station

# Aufgabe 3: Relationenmodell → ERM

Kauf						
<u>KundenNr</u>	FName	<u>ArtikelNr</u>	<u>Zeitstempel</u>	Menge	Preis	Lieferadresse

Kunde	
<u>KundenNr</u>	Name

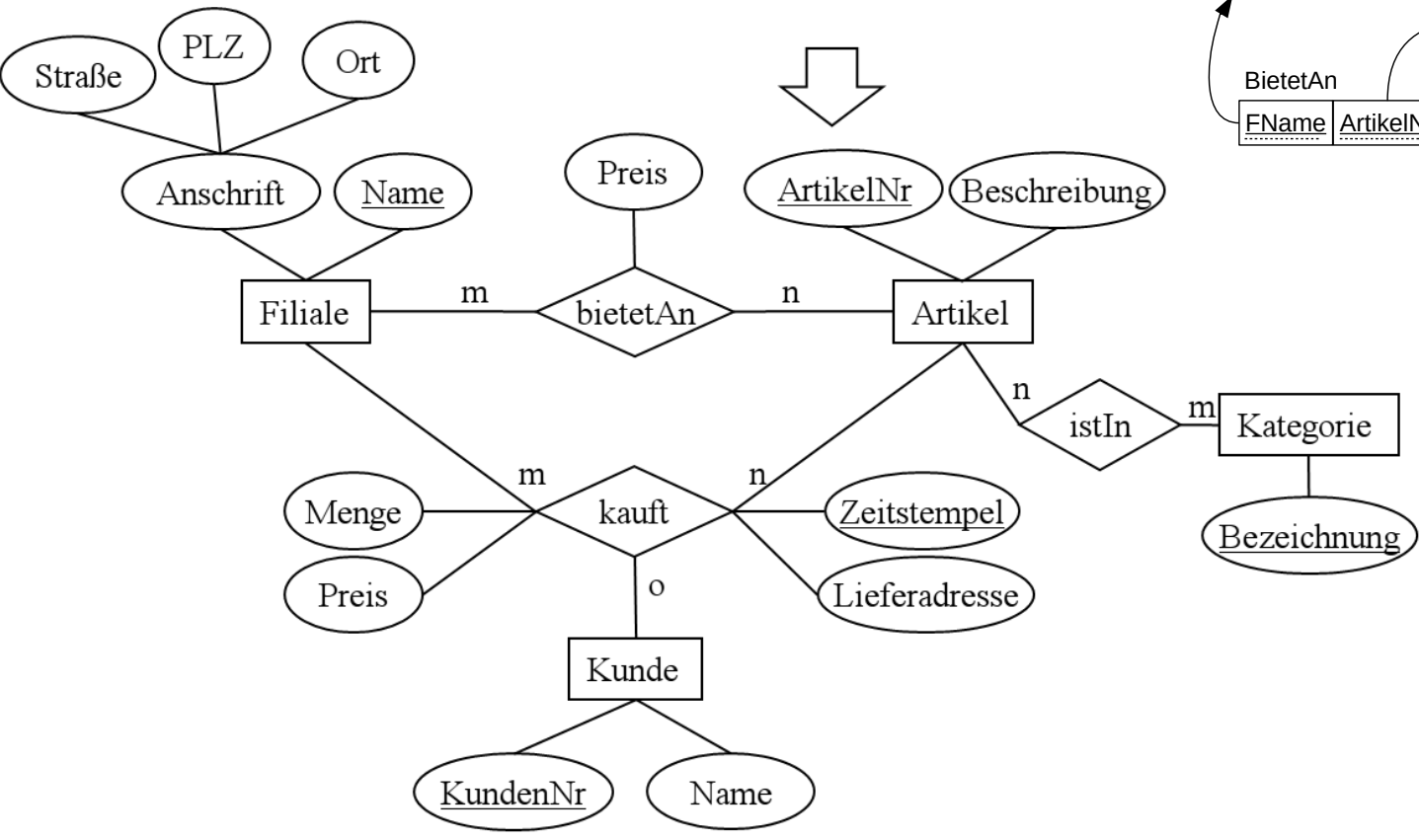
Filiale			
<u>Name</u>	Straße	PLZ	Ort

Artikel	
<u>ArtikelNr</u>	Beschreibung

Produktkat.
<u>Bezeichnung</u>

BietetAn		
<u>FName</u>	<u>ArtikelNr</u>	Preis

IstInKategorie	
<u>ArtikelNr</u>	<u>Bezeichnung</u>



## Aufgabe 4: Referentielle Integrität

```
CREATE TABLE Angestellter (  
  ANr int PRIMARY KEY,  
  AName varchar(30)  
)
```

```
CREATE TABLE Projekt(  
  PNr int PRIMARY KEY,  
  PNAME varchar(30),  
  Projektleiter int NOT NULL,  
  FOREIGN KEY (Projektleiter) REFERENCES Angestellter  
    ON DELETE OF Angestellter NO ACTION  
)
```

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

- 1) Scheidet ein Angestellter aus (d. h. wird er gelöscht), so wird auch die Information gelöscht, in 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.

