

## 2.7 Üben und vertiefen

### Aufgabe 1:

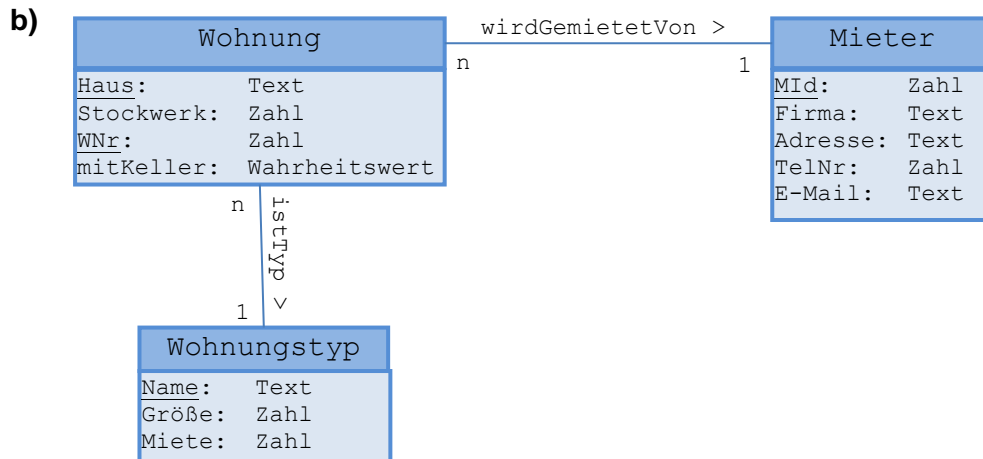
- a) Als Primärschlüssel dient `Haus` und `WNr`, da durch deren Angabe jede Wohnung eindeutig bestimmt wird.
- b) Der Mieter kann über die `WNr` sofort das Stockwerk erkennen.  
Bei einer Suche nach allen Wohnungen in einem bestimmten Stockwerk kann eine entsprechende SQL-Abfrage einfacher geschrieben werden.
- c) Da Wohnungen des gleichen Typs die gleiche Miete sowie auch die gleiche Größe haben, sind diese Informationen redundant.

Ebenso sind die Informationen der Mieter mehrfach gespeichert, wenn diese mehrere Wohnungen angemietet haben.

- d) individuelle Lösungen, z. B.:
  - Die Miete für einen Typ wird geändert, dies aber bei einer Wohnung übersehen.
  - Durch Umbauten ändert sich die Größe einer Wohnung, es wird jedoch nicht der Typ angepasst.
  - Es ändern sich Daten eines Mieters und diese werden nicht in allen entsprechenden Datensätzen geändert.
- e)
  - Da alle Daten der Firma VR7 gelöscht werden, entsteht eine Delete-Anomalie, d.h. bei einer Neuvermietung müssen alle Daten wieder eingetragen werden.
  - In allen Datensätzen mit dem Typ R9 müssen die Mieten erhöht werden. Werden Datensätze übersehen, ergeben sich verschiedene Mieten für gleiche Wohnungstypen, es entsteht eine Update-Anomalie.
  - Die Daten des Kunden können nicht eingetragen werden, da dies nur mit dem Primärschlüsselattributen `Haus` und `WNr` möglich ist. Es entsteht eine Insertion-Anomalie.
  - In allen Datensätzen mit dem Mieter Cx24R müssen die Adressen geändert werden. Werden Datensätze übersehen, können verschiedene Adressen der Firma vorliegen. Es entsteht eine Update-Anomalie.

## Aufgabe 2:

- a) Die redundanten Informationen müssen in neuen Tabellen gespeichert werden. Es muss eine Tabelle **Mieter** und eine Tabelle **Wohnungstyp** erstellt und mit entsprechenden Beziehungen mit **Wohnung** verknüpft werden.



Hinweis:

Der Einfachheit halber wurde PLZ, Ort, Straße und Hausnummer in einem Attribut **Adresse** gespeichert. Für eine vollständige atomare Speicherung müsste die Tabelle **Ort** (PLZ, Name) eingefügt und die **Adresse** dann auf die Attribute **PLZ**, **Straße** und **Hausnummer** aufgeteilt werden.

- c) Da die Miete eines Wohnungstyps und die Adresse eines Mieters nur noch einmal gespeichert sind, kann keine Update-Anomalie mehr auftreten. Es kann nur eine Inkonsistenz zwischen **Stockwerk** und **WNr** auftreten. Möchte man diese auch beseitigen, darf die **WNr** das **Stockwerk** nicht mehr enthalten.

## Aufgabe 3:

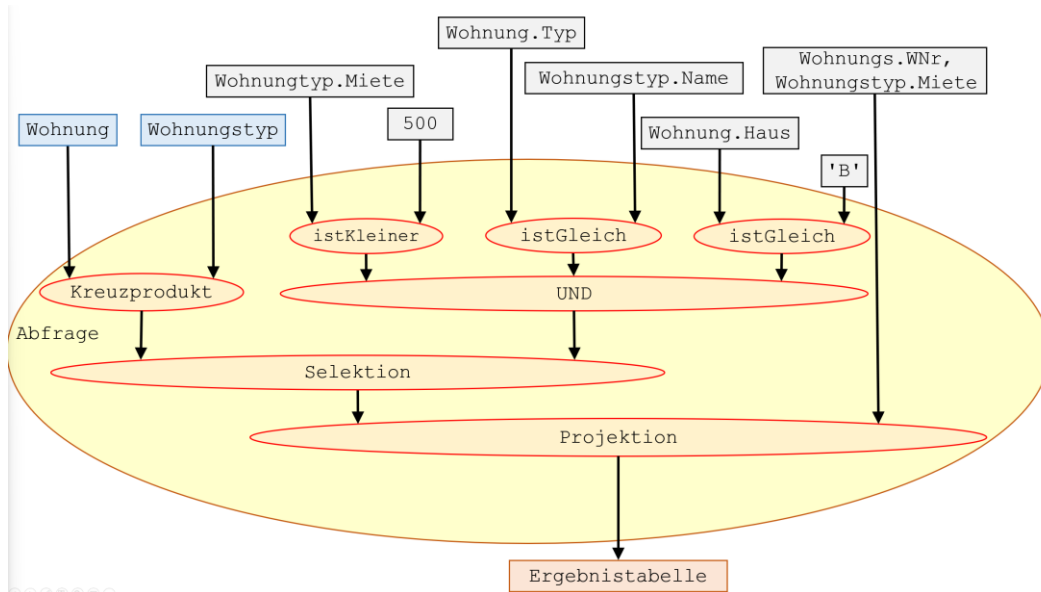
- a) Wohnung(Haus: Text; Stockwerk: Zahl; WNr: Zahl;  
mitKeller: Wahrheitswert; Typ: Text;  
Mieter: Zahl)
- Wohnungstyp(Name: Text; Größe: Zahl; Miete: Zahl)
- Mieter(MId: Zahl; Firma: Text; Adresse: Text;  
TelNr: Zahl; E-Mail: Text)

b)

- Die Eigenschaft **Dusche** oder **Badewanne** müsste in der Tabelle **Wohnungstyp** durch zusätzliche Attribute gespeichert werden, z.B. **Badewanne**: Wahrheitswert und **Dusche**: Wahrheitswert.
- In einer zusätzlichen Tabelle **Haus** können alle Eigenschaften des entsprechenden Hauses gespeichert werden, z.B. **Aufzug**: Wahrheitswert oder **Tiefgarage**: Wahrheitswert.

- Da das Vorhandensein eines Kellerraums nicht vom Wohnungstyp abhängt, wird ein Attribut z.B. Kellergröße: Zahl in die Tabelle Wohnung eingetragen. Ist die Größe des Kellers für alle Wohnungen eines Typs gleich, erfolgt die Eintragung in der Tabelle Wohnungstyp.

c) `SELECT Wohnung.WNr, Wohnungstyp.Miete`  
`FROM Wohnung, Wohnungstyp`  
`WHERE Wohnungstyp.Miete < 500`  
`AND Wohnung.Type = Wohnungstyp.Name`  
`AND Wohnung.Haus = 'B'`



#### Aufgabe 4:

- a) Es sind medizinische Daten über verschiedene Personen anonym gespeichert: Alter, Geschlecht, Body-Mass-Index, Anzahl der Kinder, Raucher oder Nicht-Raucher, Region des Wohnorts und die medizinischen Kosten.

- b) Durch angepasste SQL-Abfragen ähnlich

```
SELECT * (oder COUNT(*))
FROM Medizin
WHERE Geschlecht = 'w' AND Raucher = 1
```

lässt sich der Zusammenhang in einer Vierfeldertafel darstellen:

	weiblich	männlich	Gesamt
Raucher	5	12	17
Nichtraucher	27	22	49
Gesamt	32	34	66

	weiblich	männlich	Gesamt
Raucher	7,6%	18,2%	25,8%
Nichtraucher	40,9%	33,3%	74,2%
Gesamt	48,5%	41,5%	100%

Ergebnis: Der Anteil der Raucher unter den männlichen Personen ist höher.

Für die Kosten muss ein Wert festgelegt werden, der sinnvoll ist. Mit Hilfe von Aggregationen kann der Durchschnitt bestimmt werden, dieser liegt bei ca. 7138. Ansonsten kann mit verschiedenen Werten experimentiert werden.

```
SELECT *                               (oder COUNT(*))
FROM   Medizin
WHERE  Kosten > 7138 AND Raucher = 1
```

Die Bedingung für Kosten und Raucher werden entsprechend angepasst, um die Einträge der Vierfeldertafel zu bestimmen.

	bis 7138	mehr als 7138	Gesamt
Raucher	1	16	17
Nichtraucher	49	0	49
Gesamt	50	16	66

Ergebnis: Die Kosten für Raucher sind deutlich höher als für Nichtraucher.

Alternativ können mit der Aggregationsfunktion direkt die durchschnittlichen Kosten für Raucher und Nichtraucher bestimmt werden:

```
SELECT AVG(Kosten)
FROM   Medizin
WHERE  Raucher = 1          (bzw. Raucher = 0)
```

Ergebnis:

Die durchschnittlichen Kosten betragen für Raucher 24370 und für Nichtraucher 1160.

Für das Alter in Jahren muss ein Wert festgelegt werden, der sinnvoll ist. Mit Hilfe von Aggregationen kann das Durchschnittsalter bestimmt werden, dieses liegt bei 36,7727. Ansonsten kann mit verschiedenen Werten experimentiert werden.

```
SELECT *                               (oder COUNT(*))
FROM   Medizin
WHERE  Jahre > 36.7727 AND Raucher = 1
```

Die Bedingung für Kosten und Raucher werden entsprechend angepasst um die Einträge der Vierfeldertafel zu bestimmen.

	<= 36.7727	> 36.7727	Gesamt		jung	alt	Gesamt
Raucher	5	12	17	Raucher	7,6%	18,2%	25,8%
Nichtraucher	23	26	49	Nichtraucher	34,8%	39,4%	74,2%
Gesamt	28	38	66	Gesamt	42,4%	57,6%	100%

Ergebnis: Der Anteil der Raucher unter den älteren Personen ist größer.

Alternativ können mit der Aggregationsfunktion direkt die Durchschnittsalter für Raucher und Nichtraucher bestimmt werden:

```
SELECT AVG(Jahre)
FROM   Medizin
WHERE  Raucher = 1          (bzw. Raucher = 0)
```

Ergebnis:

Die durchschnittlichen Kosten betragen für Raucher 36,3 und für Nichtraucher 36,9.

Hier ergibt sich aufgrund der Statistik ein anderes Bild, da in der Vierfeldertafel der Abstand vom Durchschnittsalter bei den Rauchern und Nichtrauchern nicht mit einberechnet wird.

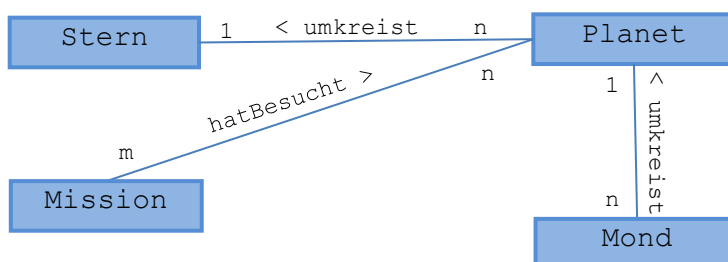
- c) Generell besteht für alle Bereich der Medizin ein Interesse an diesen Zusammenhängen, sei es bei Versicherungen, um Beiträge zu berechnen, Gesundheitsvorsorge, um entsprechende Personengruppen durch Werbekampagnen anzusprechen, als Grundlage für neue Gesetze oder auch Werbekampagnen von Firmen in diesem Bereich.
- d) individuelle Lösungen, z. B.:  
Wird nur nach Raucher und Nichtraucher unterschieden, ergeben sich für weibliche bzw. männliche Personen dadurch geringere bzw. höhere Krankenkassenbeiträge.  
Bei Berücksichtigung des Alters kann dies ebenfalls zu höheren bzw. geringeren Beiträgen führen.
- e) individuelle Lösungen, z. B.:  
Falsche oder manipulierte Datensätze führen in der Konsequenz zu falschen Entscheidungen, so dass Beiträge oder auch gesetzliche Entscheidungen nicht der Ursache entsprechend und zum Nachteil der betroffenen Personengruppen führen.

### Aufgabe 5:

a)

- `Sonne` ist keine Klasse, sondern ein Objekt der Klasse `Stern`. Manchmal wird `Sonne` auch als Synonym für `Stern` benutzt, dann enthält das Diagramm zwei Klassen, die dasselbe beschreiben.
- Die Beziehung `umkreist` zwischen `Mond` und `Sonne` ist falsch: Ein Mond umkreist definitionsgemäß einen Planeten.
- Bei der Beziehung `umkreist` zwischen `Mond` und `Planet` handelt es sich um eine 1:n-Beziehung.
- Bei der Beziehung `umkreist` zwischen `Planet` und `Stern` fehlen die Kardinalitäten. Es handelt sich auch hier um eine 1:n-Beziehung (siehe dazu auch Aufgabe 5f).

b)



- c) Die Namen von Himmelskörpern werden von der Internationale Astronomische Union (IAU) vergeben. Diese werden so gewählt, dass sie eindeutig sind, daher sind sie als Primärschlüssel geeignet.  
In einem Verein kommen Personen zusammen, die vorher schon unabhängig voneinander Vor- und Nachnamen bekommen haben. Daher kann es unterschiedliche Personen mit gleichen Namen geben, diese sind also nicht ausreichend als Primärschlüssel. Es müssen Attribute hinzugefügt oder ein künstlicher Schlüssel festgelegt werden.
- d) Die drei Größen Halbachse, Umlaufzeit und Bahngeschwindigkeit hängen über physikalische Formeln zusammen, sodass die Speicherung aller drei redundant wäre und zu Inkonsistenzen führen kann.
- e) In der Tabelle `Himmelskörper` müsste hinzugefügt werden:
- ein Attribut `umkreist` als Fremdschlüssel für den umkreisten Zentralkörper
  - ein Attribut `Typ`, um Sterne und Monde unterscheiden zu können
- Damit können jetzt auch Kleinplaneten und Asteroiden gespeichert werden (entsprechender Eintrag bei `Typ`).
- f) Die Kardinalität der Beziehung `umkreist` zwischen `Himmelskörper` und sich selbst muss auf n:m geändert werden, da bei einem Mehrfachsternsystem mehrere Sonnen umkreist werden. Zudem kann im neuen Modell auch der Fall „Stern umkreist Stern“ (in Mehrfachsystemen) erfasst werden.
- g) Im anfänglichen Modell können nur Planeten in die Beziehungstabelle `hatBesucht` eingetragen werden, ein Besuch eines Mondes oder Asteroiden ist hier nicht speicherbar.

Im zweiten Modell werden die besuchten Objekte in der Beziehungstabelle `hatBesucht` anhand z.B. des Missionsnamens und des Himmelskörpernamens gespeichert, zusätzlich wäre vermutlich noch ein Attribut `Zeitpunkt` sinnvoll um die Reihenfolge des Anflugs zu notieren.

hatBesucht		
Mission	Himmelskörper	Datum*
New Horizons	APL	2006-06-13
New Horizons	Jupiter	2007-02-28
New Horizons	Pluto	2015-07-14
New Horizons	Charon	2015-07-14
New Horizons	Styx	2015-07-14
New Horizons	Hydra	2015-07-14
New Horizons	Nix	2015-07-14
New Horizons	Kerberos	2015-07-14
New Horizons	Arrokoth*	2019-01-01

Hinweis:

Zusätzliche Informationen wurden im Internet recherchiert, was als zusätzliche Aufgabe auch bei anderen Missionen gestellt werden kann.

```

h) SELECT H1.Name
    FROM Himmelskörper H1, Himmelskörper H2
    WHERE H1.umkreist = H2.Name
          AND H2.Typ = 'Stern'
          AND H1.Typ = 'Planet'
          AND H2.Name <> 'Sonne'

SELECT Name
    FROM Himmelskörper
    WHERE umkreist = 'Jupiter' AND Typ = 'Mond'

SELECT H1.Name
    FROM Himmelskörper H1, Himmelskörper H2
    WHERE H1.umkreist = H2.Name
          AND H2.Typ = 'Stern'
          AND H1.Typ = 'Stern'

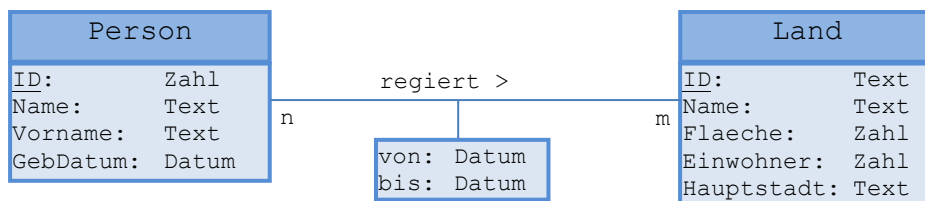
```

### Aufgabe 6:

a) In den Tabellen sind Informationen über Personen, Länder und die Beziehungen zwischen den Personen und Ländern gespeichert.

b) Tab1: Land  
 Tab2: Person  
 Tab3: regiert

c)



```

Land(ID: Text; Name: Text; Flaeche: Zahl;
     Einwohner: Zahl; Hauptstadt: Text)
Person(ID: Zahl; Name: Text; Vorname: Text;
       GebDatum: Datum)
regiert(ID: Zahl; Land: Text; von: Datum; bis: Datum)
  
```

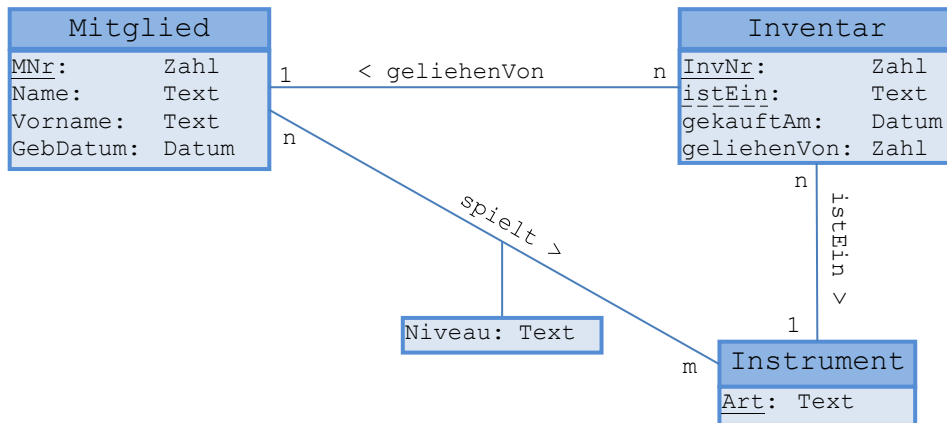
### Aufgabe 7:

- Die Tabelle Kredit enthält anonyme Datensätze über verschiedene Personen, deren Besitz und Kreditverhalten.
- Anhand der Angaben kann ein Kreditinstitut Entscheidungsregeln festlegen, die für zukünftige Kreditvergaben im Allgemeinen gelten, z.B. keine Kredite an Personen mit zwei Jobs o.Ä.
- Sollten die Daten in Bezug zu Personen gebracht werden, kann dadurch beim Verkauf bzw. Abschluss von bestimmten Verträgen diese Datensätze zu negativen Entscheidungen verführen, d.h. höhere Preise beim Verkauf bzw. höhere Beiträge oder kein Abschluss bei Verträgen.

**Aufgabe 8:**

- a) Die Tabelle `Inventar` kann durch die 1:n-Beziehung `geliehenVon` mit `Mitglied` verknüpft werden. So kann der Leiter feststellen, wer welches Instrument ausgeliehen hat. Die Tabelle `Instrument` kann durch eine n:m-Beziehung z.B. namens `spielt` mit `Mitglied` verknüpft werden. Wird in der Beziehungstabelle dann noch das `Niveau` gespeichert, werden alle Voraussetzungen des Musikkapellenleiters erfüllt.

b)



- c) `Mitglied`(MNr: Zahl; Name: Text; Vorname: Text; GebDatum: Datum)  
`Inventar`(InvNr: Zahl; Art: Text; gekauftAm: Datum; geliehenVon: Zahl)  
`Instrument`(Art: Text)  
`spielt`(MNr: Zahl; spieltEin: Text; Niveau: Text)
- d) Als `Instrument` kann Gesang eingeführt werden, auch wenn dann die Bezeichnung der Beziehung `spielt` im üblichen Sprachgebrauch nicht mehr so passend ist.
- e)
- ```

SELECT Mitglied.Name, Mitglied.Vorname
FROM   Mitglied, spielt
WHERE  Mitglied.MNr = spielt.MNr
      AND spielt.spieltEin = 'Trompete'
      AND (spielt.Niveau = 'mittelmäßig'
          OR spielt.Niveau = 'gut'
          OR spielt.Niveau = 'sehr gut')
    
```
- f) Wird `Niveau` nicht als Text sondern als Zahl (z. B. in Schulnoten) gespeichert, ist die ODER-Verknüpfung nicht mehr nötig.
- ```

SELECT Mitglied.Name, Mitglied.Vorname
FROM   Mitglied, spielt
WHERE  Mitglied.MNr = spielt.MNr
      AND spielt.spieltEin = 'Trompete'
      AND spielt.Niveau <= 3
    
```

Hinweis: Die Note 3 entspricht dabei dem Wort mittelmäßig.



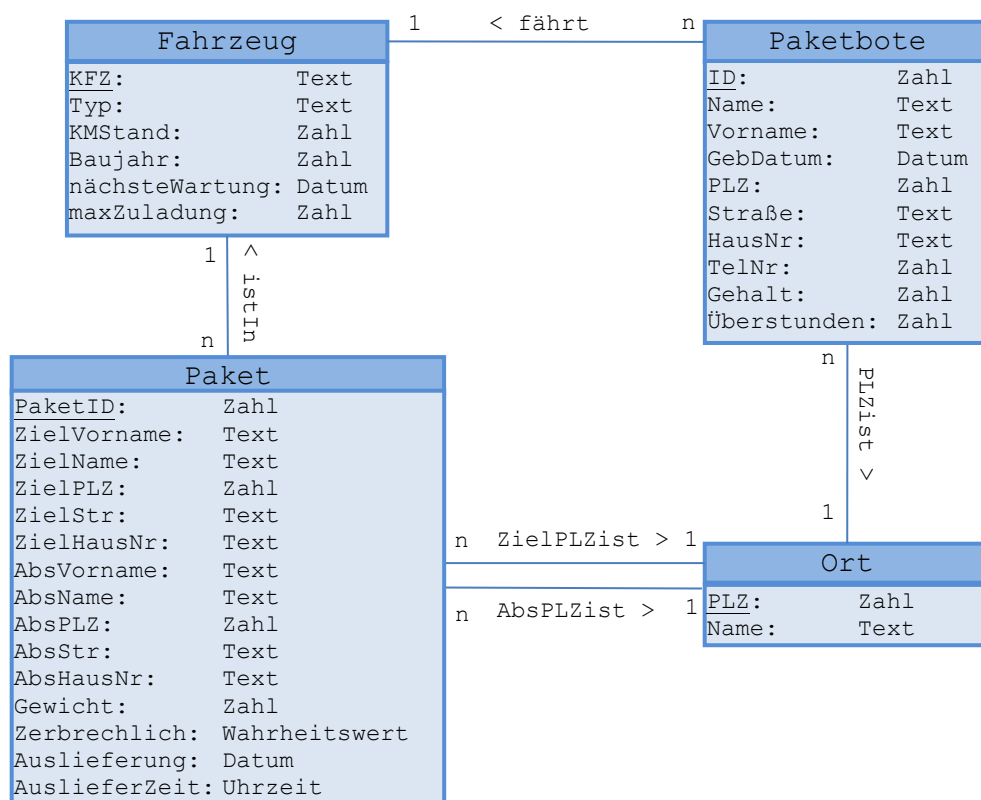
- g) `SELECT Mitglied.Name, Mitglied.Vorname,  
Inventar.InvNr, Inventar.Art  
FROM Mitglied, Inventar  
WHERE Mitglied.MNr = Inventar.geliehenVon`
- h) Die Speicherung des Alters ist unnötig, da dies mit Hilfe des Kauf- und des aktuellen Datums berechnet werden kann.

### Aufgabe 9:

individuelle Lösungen, z. B.:

- a) Klassen: Fahrzeug, Paketbote, Paket  
Beziehungen: fährt (Paketbote – Fahrzeug), istIn (Paket, Fahrzeug)

b)



- c) Fahrzeug (KFZ: Text; Typ: Text; KMStand: Zahl;  
Baujahr: Zahl; nächsteWartung: Datum;  
maxZuladung: Zahl)

Paket (PaketID: Zahl; istIn: Text;  
ZielVorname: Text; ZielName: Text;  
ZielPLZ: Zahl; ZielStr: Text; ZielHausNr: Text;  
AbsVorname: Text; AbsName: Text; AbsPLZ: Zahl;  
AbsStr: Text; AbsHausNr: Text; Gewicht: Zahl;  
Zerbrechlich: Wahrheitswert; Auslieferung: Datum;  
AuslieferZeit: Uhrzeit)

Der Fremdschlüssel `istIn` bezieht sich auf den Primärschlüssel `KFZ` in der Klasse `Fahrzeug`.

Der Fremdschlüssel `ZielPLZ` bezieht sich auf den Primärschlüssel `PLZ` in der Klasse `Ort`.

Der Fremdschlüssel `AbsPLZ` bezieht sich auf den Primärschlüssel `PLZ` in der Klasse `Ort`.

```
Paketbote(ID: Zahl; Name: Text; Vorname: Text;
          GebDatum: Datum; PLZ: Zahl; Straße: Text;
          HausNr: Text; TelNr: Zahl; Gehalt: Zahl;
          Überstunden: Zahl; fährt: Text)
```

Der Fremdschlüssel `PLZ` bezieht sich auf den Primärschlüssel `PLZ` in der Klasse `Ort`.

Der Fremdschlüssel `fährt` bezieht sich auf den Primärschlüssel `KFZ` in der Klasse `Fahrzeug`.

```
Ort(PLZ: Zahl; Name: Text)
```

**d) Klasse Fahrzeug:**

```
('N-UN 1'; 'WMB 12'; 15343; 2022; 2023-01-01; 1500)
('N-UN 2'; 'WMB 8'; 35227; 2021; 2023-08-04; 1250)
```

**Klasse Paketbote:**

```
(1; 'Wansind'; 'Rainer'; 2000-04-01; 91054;
  'Wohnstraße'; '5'; 0911123xx; 1995; 27; 'N-UN 1')
(2; 'Flotte'; 'Liesel'; 1998-12-03; 91050;
  'Hauptstraße'; '42'; 091193xx; 2195; 54; 'N-UN 2')
```

**Klasse Paket:**

```
(5954939; 'N-UN 1'; 'Bernd'; 'Baguette'; 91052;
  'Am Teich'; '12a'; 'Rudi'; 'Rabe'; 96052;
  'Baumweg'; '23'; 4.5; WAHR; 2022-10-08; 10:00)
(5954940; 'N-UN 1'; 'Adi'; 'Dusel'; 81023;
  'Burgstraße'; '4c'; 'Donald'; 'Ente'; 90053;
  'Seitenstraße'; '111'; 13.5; FALSCH; 2022-10-09; 9:00)
```

**Klasse Ort:**

```
(96052; 'Bamberg')
(91054; 'Erlangen')
```

- e)** Die Gesamtmasse aller Pakete eines Fahrzeugs kann mit Hilfe der einzelnen Datensätze der Pakete berechnet werden und sollte nicht extra gespeichert werden, da so redundante Daten entstehen und evtl. Inkonsistenzen.

- f)**
- ```
SELECT KFZ
FROM Fahrzeug
WHERE nächsteWartung > 2022-31-12
      AND nächsteWartung < 2024-01-01
```

**Hinweis:** Hier für das Jahr 2023.

```
SELECT Name, Vorname, Überstunden
FROM   Paketbote
WHERE  Überstunden > 10
```

**Bestimmung des Fahrers :**

```
SELECT Name, Vorname
FROM   Paketbote
WHERE  fährt = "DC-AC 42"
```

**Bestimmung der Ladung**

```
SELECT PaketID
FROM   Paket
WHERE  istIn = 'DC-AC 42'
```

**Hinweis:** Die Abfragen sind besonders kurz, wenn das KFZ als Primärschlüssel der Fahrzeuge gewählt wird. Ansonsten muss ein Join über die Tabellen `Paketbote` und `Fahrzeug` bzw. `Paket` und `Fahrzeug` erstellt werden.

**Aufgabe10:**

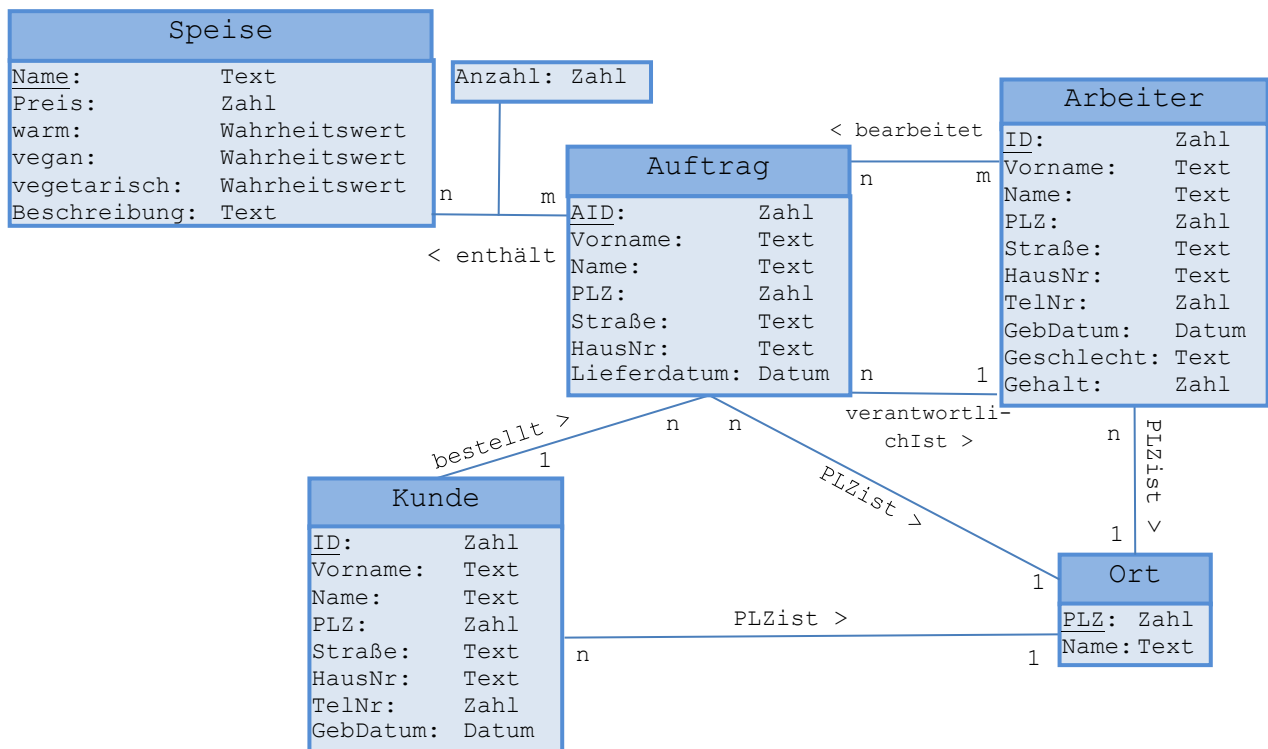
individuelle Lösungen, z. B.:

**a) mögliche Attribute:**

Arbeiter: Name, Vorname, PLZ, Straße, HausNr, TelNr, Gehalt, GebDatum, Geschlecht  
Kunde: Name, Vorname, PLZ, Straße, HausNr, TelNr, GebDatum  
Speise: Name, Preis, warm, vegan, vegetarisch, Beschreibung  
Auftrag: Auftragsnummer, verantwortlich, Name, Vorname, PLZ, Straße, HausNr

Bestellungen und Bearbeitungen müssen in zusätzlichen Beziehungstabellen umgesetzt werden.

b)



- c) **Speise** (Name: Text; Preis: Zahl; warm: Wahrheitswert; vegan: Wahrheitswert; vegetarisch: Wahrheitswert; Beschreibung: Text)
- Kunde** (ID: Zahl; Vorname: Text; Name: Text; PLZ: Zahl; Straße: Text; HausNr: Text; TelNr: Zahl; GebDatum: Datum)
- Arbeiter** (ID: Zahl; Vorname: Text; Name: Text; PLZ: Zahl; Straße: Text; HausNr: Text; TelNr: Zahl; GebDatum: Datum; Geschlecht: Text; Gehalt: Zahl)
- Ort** (PLZ: Zahl; Name: Text)
- Auftrag** (AID: Zahl; Vorname: Text; Name: Text; PLZ: Zahl; Straße: Text; HausNr: Text; Lieferdatum: Datum; bestelltID: Zahl; verantwortlich: Zahl)
- enthält** (AID: Zahl; Speisename: Text; Anzahl: Zahl)
- bearbeitet** (AID: Zahl; ArbeiterID: Text)

d) siehe Datei 2-7\_Aufgabe10d\_Loesung.odt

- e)
- ```

SELECT Auftrag.AID
FROM Auftrag, Kunde
WHERE Auftrag.bestellt = Kunde.ID
      AND Kunde.Name = 'Müller'
      AND Kunde.Vorname = 'Hans'

SELECT Speise.Name, enthält.Anzahl
FROM Auftrag, Kunde, enthält, Speise
WHERE Auftrag.bestellt = Kunde.ID
      AND Auftrag.AID = enthält.AID
      AND enthält.Speisename = Speise.Name
  
```