

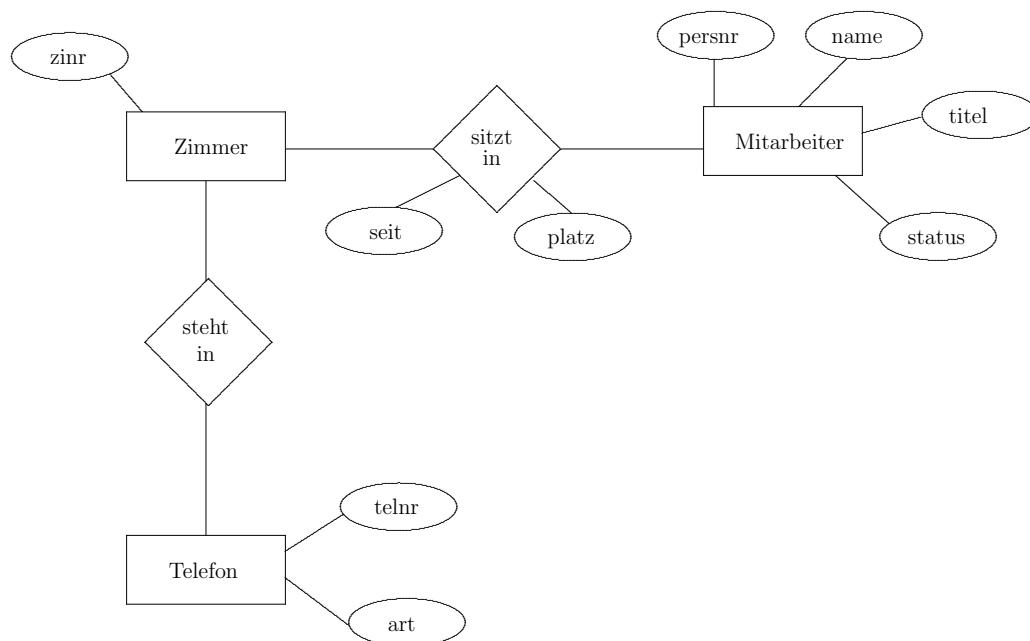
# Übungsaufgaben DB-Modellierung (Lösungen)

**Hinweis:** die jeweils angegebene Lösung ist nur eine von vielen Möglichkeiten!

## 1. Aufgabe:

*In Büroräumen (charakterisiert durch eine Zimmernummer) sitzen seit einem Zeitpunkt Mitarbeiter (Personalnummer, Name, Titel, Status) an einem bestimmten Platz. In den Zimmern sind Telefone (besitzen eine eindeutige Telefonnummer) aufgestellt, die als Hausapparat oder Amtsapparat geschaltet sind.*

### ER-Diagramm:



### Relationen-Schema:

(Primärschlüssel sind unterstrichen)

Zimmer(zinnr)

Mitarbeiter(persnr, name, titel, status)

Telefone(telnr,art,zinr → Zimmer(zinr))  
sitzt\_in(zinr → Zimmer(zinr),persnr → Mitarbeiter(persnr),seit,platz)

### SQL-Anweisungen zum Erzeugen der Tabellen:

```
CREATE TABLE Zimmer
  (zinr VARCHAR(5) NOT NULL CONSTRAINT Zimmer_PK PRIMARY KEY)

CREATE TABLE Mitarbeiter
  (persnr VARCHAR(20) NOT NULL CONSTRAINT Mitarbeiter_PK PRIMARY KEY,
   name VARCHAR(40) NOT NULL,
   titel VARCHAR(20) NOT NULL,
   status VARCHAR(20) NOT NULL
  )

CREATE TABLE Telefone
  (telnr VARCHAR(20) NOT NULL CONSTRAINT Telefone_PK PRIMARY KEY,
   art VARCHAR(4) NOT NULL,
   zinr VARCHAR(5) NOT NULL,
   CONSTRAINT Zimmer_FK FOREIGN KEY(zinr)
     REFERENCES Zimmer
  )

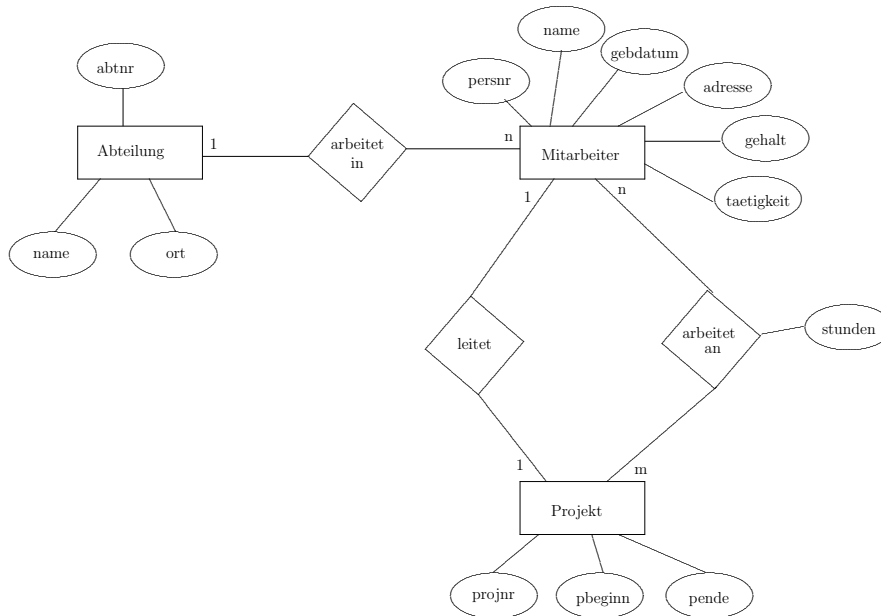
CREATE TABLE sitzt_in
  (zinr VARCHAR(5) NOT NULL,
   persnr VARCHAR(20) NOT NULL,
   seit DATETIME NOT NULL,
   platz INTEGER NOT NULL,
   CONSTRAINT sitzt_in_PK PRIMARY KEY(zinr,persnr),
   CONSTRAINT Z_sitzt_in_FK FOREIGN KEY(zinr)
     REFERENCES Zimmer,
   CONSTRAINT M_sitzt_in_FK FOREIGN KEY(persnr)
     REFERENCES Mitarbeiter
  )
```

### 2. Aufgabe:

*In einer Abteilung (besitzt Abteilungsnummer, Name und Ort) arbeiten Mitarbeiter (charakterisiert durch Personalnummer, Name, Geburtsdatum, Privatadresse, Gehalt, Tätigkeitsbezeichnung), die an Projekten (eindeutige Projektnummer, Projektbeginn und -ende) arbeiten. Jeder Mitarbeiter arbeitet mit einer bestimmten Wochen-*

*stundenzahl an einem bestimmten Projekt und kann an mehreren Projekten gleichzeitig arbeiten. Jedes Projekt wird von einem anderen Mitarbeiter geleitet.*

### ER-Diagramm:



### Relationen-Schema:

(Primärschlüssel sind unterstrichen)

Abteilung(abtnr,name,ort)

Mitarbeiter(persnr,name,gebdatum,adresse,gehalt,taetigkeit,abtnr → Abteilung(abtnr))

Projekt(projnr,pbeginn,pende,leiter → Mitarbeiter(persnr))

arbeitet\_an(persnr → Mitarbeiter(persnr),projnr → Projekt(projnr),Stunden)

### SQL-Anweisungen zum Erzeugen der Tabellen:

```

CREATE TABLE Abteilung
(
    abtnr INTEGER NOT NULL CONSTRAINT Abteilung_PK PRIMARY KEY,
    name VARCHAR(20) NOT NULL,
    ort VARCHAR(20) NOT NULL
)
    
```

```

CREATE TABLE Mitarbeiter
(
    persnr VARCHAR(20) NOT NULL CONSTRAINT Mitarbeiter_PK PRIMARY KEY,
    
```

```

    name VARCHAR(40) NOT NULL,
    gebdatum DATETIME NOT NULL,
    adresse VARCHAR(80) NOT NULL,
    gehalt MONEY NOT NULL,
    taetigkeit VARCHAR(20) NOT NULL,
    abtnr INTEGER,
    CONSTRAINT Abt_Mitarb_FK FOREIGN KEY(abtnr)
        REFERENCES Abteilung
)

CREATE TABLE Projekt
(projnr VARCHAR(20) NOT NULL CONSTRAINT Projekt_PK PRIMARY KEY,
pbeginn DATETIME NOT NULL,
pende DATETIME NOT NULL,
leiter VARCHAR(20),
CONSTRAINT Leiter_FK FOREIGN KEY(leiter)
    REFERENCES Mitarbeiter(persnr)
)

CREATE TABLE arbeitet_an
(persnr VARCHAR(20) NOT NULL,
projnr VARCHAR(20) NOT NULL,
stunden INTEGER NOT NULL,
CONSTRAINT arbeitet_an_PK PRIMARY KEY(persnr,projnr),
CONSTRAINT m_arbeitet_FK FOREIGN KEY(persnr)
    REFERENCES Mitarbeiter,
CONSTRAINT p_arbeitet_FK FOREIGN KEY(projnr)
    REFERENCES Projekt
)

```

### 3. Aufgabe:

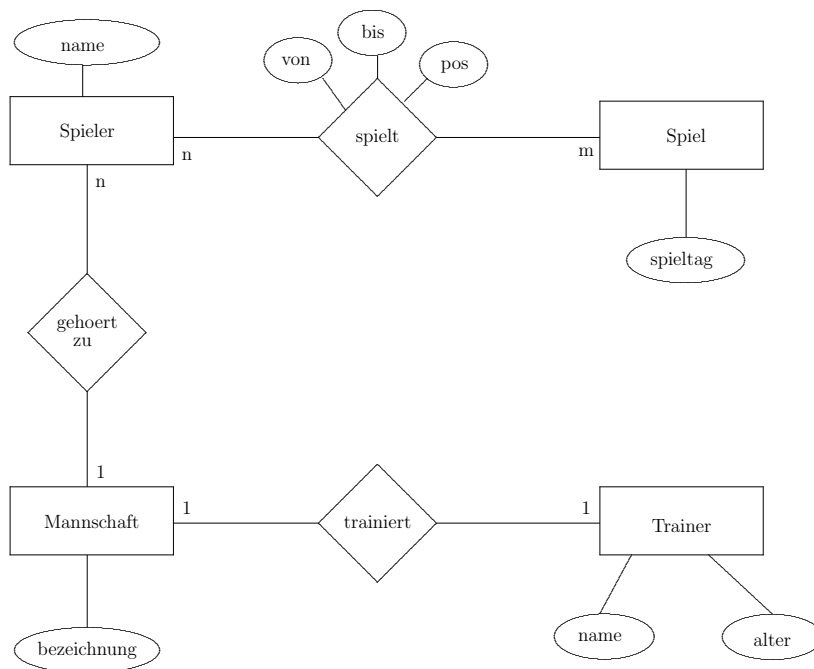
*In einer Fußballdatenbank sollen für alle Spieler (charakterisiert durch einen Namen) folgende Fakten gespeichert werden:*

- *in welchem Spiel (Spieltag) im Einsatz*
- *wie lange (von,bis) im Einsatz*
- *auf welcher Position im Einsatz*
- *zu welcher Mannschaft gehörig*

*Von der Mannschaft sind von Interesse:*

- *Name der Mannschaft*
- *Name des Trainers*
- *Alter des Trainers*

### ER-Diagramm:



### Relationen-Schema:

(Primärschlüssel sind unterstrichen)

Spieler(name,mannschaft → Mannschaft(bezeichnung))  
 Spiel(spieltag)  
 Mannschaft(bezeichnung,tName → Trainer(name))  
 Trainer(name,alter)  
 spielt(name → Spieler(name),spieltag → Spiel(spieltag),von,bis,pos)

### SQL-Anweisungen zum Erzeugen der Tabellen:

```
CREATE TABLE Spiel
  (spieltag INTEGER NOT NULL CONSTRAINT Spiel_PK PRIMARY KEY)
```

```

CREATE TABLE Trainer
  (name VARCHAR(20) NOT NULL CONSTRAINT Trainer_PK PRIMARY KEY)

CREATE TABLE Mannschaft
  (bezeichnung VARCHAR(20) NOT NULL CONSTRAINT M_PK PRIMARY KEY,
   tName VARCHAR(20),
   CONSTRAINT M_Trainer_FK FOREIGN KEY(tName)
     REFERENCES Trainer(name)
  )

CREATE TABLE Spieler
  (name VARCHAR(20) NOT NULL CONSTRAINT Spieler_PK PRIMARY KEY,
   gehoert_zu VARCHAR(20),
   CONSTRAINT Sp_M_FK FOREIGN KEY(gehoert_zu)
     REFERENCES Mannschaft(bezeichnung)
  )

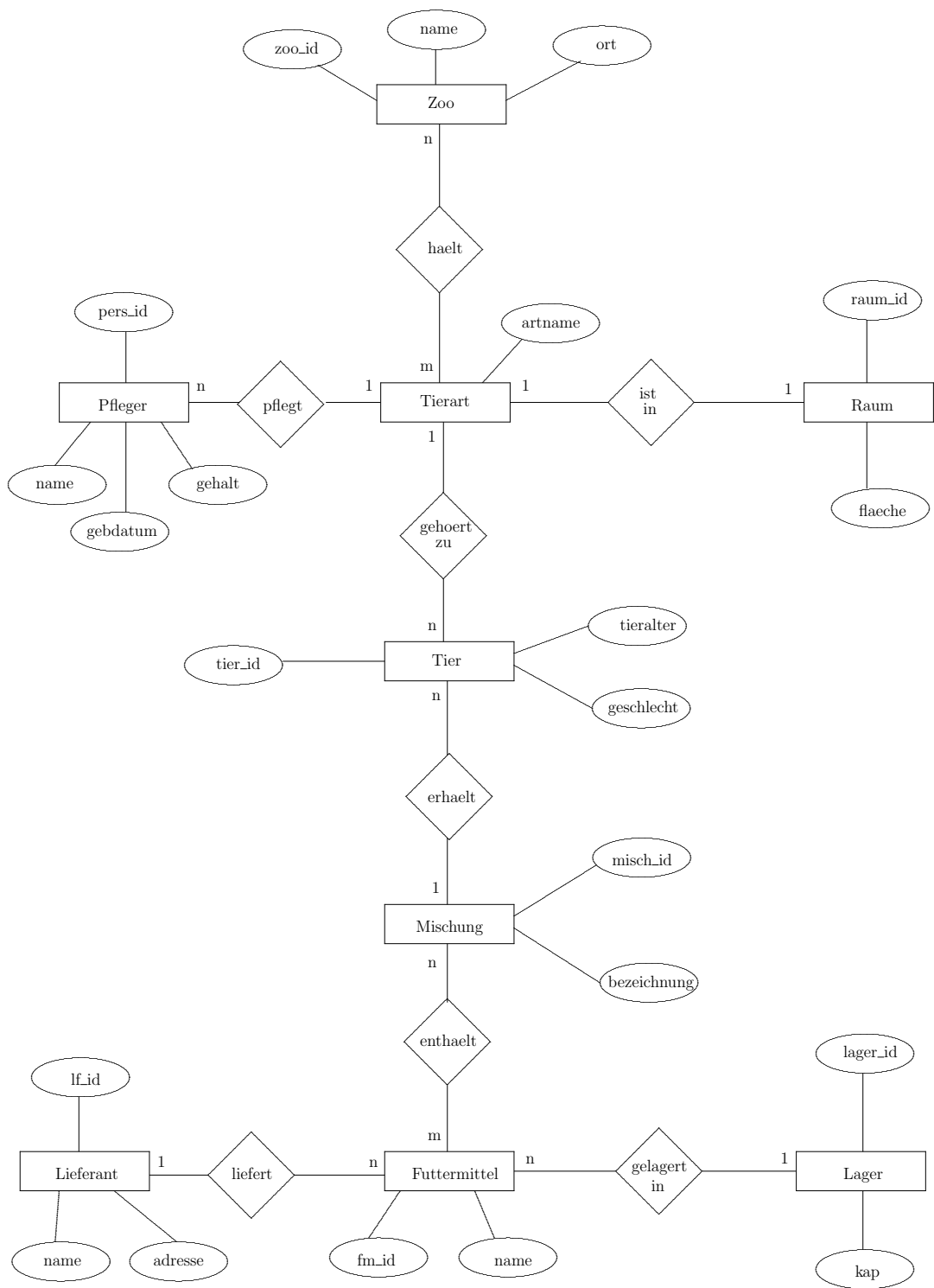
CREATE TABLE spielt
  (name VARCHAR(20) NOT NULL,
   spieltag INTEGER NOT NULL,
   von INTEGER NOT NULL,
   bis INTEGER NOT NULL,
   pos VARCHAR(20) NOT NULL,
   CONSTRAINT spielt_PK PRIMARY KEY(name,spieltag),
   CONSTRAINT sp_sp_FK FOREIGN KEY(name) REFERENCES Spieler,
   CONSTRAINT sp_st_FK FOREIGN KEY(spieltag) REFERENCES Spiel
  )

```

#### 4. Aufgabe:

Mehrere in ganz Deutschland verteilte Zoos sollen mittels einer Datenbank verwaltet werden. Jeder Zoo (eindeutige Identifikationsnummer, Name und Ort) hat mehrere Tierarten (bezeichnet durch ihren Namen). Diese Tierarten werden von mehreren Pflegern (Personalnummer, Name, Geburtsdatum, Gehalt) gepflegt und befinden sich in jeweils einem ihnen zugeteilten Raum (Raum-Nummer, Fläche). Als Raum werden hier auch Außengehege bezeichnet. Von jeder Tierart gibt es einige Exemplare (eindeutige Tier-Nummer, Alter, Geschlecht). Jedes Tier bekommt eine spezielle Futtermischung (eindeutige Nummer, Bezeichnung), welche aus verschiedenen Futtermitteln (eindeutige Nummer, Name) hergestellt wird. Die Futtermittel werden von Lieferanten (eindeutige Lieferanten-Nummer, Name, Adresse) geliefert und in Lagern (eindeutige Lager-Nummer, Kapazität) aufbewahrt.

**ER-Diagramm:**



### Relationen-Schema:

(Primärschlüssel sind unterstrichen)

Zoo(zoo\_id,name,ort)  
Tierart(artname,raum\_id → Raum(raum\_id))  
Pfleger(pers\_id,name,gebdatum,gehalt,artname → Tierart(artname))  
Raum(raum\_id,flaeche)  
Tier(tier\_id,tieralter,geschlecht,artname → Tierart(artname),misch\_id → Mischung(misch\_id))  
Mischung(misch\_id,bezeichnung)  
Futtermittel(fm\_id,name,lf\_id → Lieferant(lf\_id),lager\_id → Lager(lager\_id))  
Lieferant(lf\_id,name,adresse)  
Lager(lager\_id,kap)  
Mischung\_enthaelt(misch\_id → Mischung(misch\_id),fm\_id → Futtermittel(fm\_id))  
Arten\_im\_Zoo(zoo\_id → Zoo(zoo\_id),artname → Tierart(artname))

### SQL-Anweisungen zum Erzeugen der Tabellen:

```
CREATE TABLE Zoo
(
  zoo_id INTEGER NOT NULL CONSTRAINT zoo_PK PRIMARY KEY,
  name VARCHAR(20) NOT NULL,
  ort VARCHAR(20) NOT NULL
)

CREATE TABLE Raum
(
  raum_id INTEGER NOT NULL CONSTRAINT raum_PK PRIMARY KEY,
  flaeche INTEGER NOT NULL
)

CREATE TABLE Tierart
(
  artname VARCHAR(20) NOT NULL CONSTRAINT art_PK PRIMARY KEY,
  raum_id INTEGER NOT NULL,
  CONSTRAINT art_raum_FK FOREIGN KEY(raum_id) REFERENCES Raum
)
```



```
CREATE TABLE Pfleger
(pers_id INTEGER NOT NULL CONSTRAINT pfleger_PK PRIMARY KEY,
 name VARCHAR(40) NOT NULL,
 gebdatum DATETIME NOT NULL,
 gehalt MONEY NOT NULL,
 artname VARCHAR(20) NOT NULL,
 CONSTRAINT pfleger_art FOREIGN KEY(artname) REFERENCES Tierart
)
```

```
CREATE TABLE Mischung
(misch_id INTEGER NOT NULL CONSTRAINT misch_PK PRIMARY KEY,
 bezeichnung VARCHAR(20) NOT NULL
)
```

```
CREATE TABLE Tier
(tier_id INTEGER NOT NULL CONSTRAINT tier_PK PRIMARY KEY,
 tieralter INTEGER NOT NULL,
 geschlecht CHAR(1) NOT NULL,
 artname VARCHAR(20) NOT NULL,
 misch_id INTEGER NOT NULL,
 CONSTRAINT tier_art_FK FOREIGN KEY(artname)
 REFERENCES Tierart,
 CONSTRAINT tier_misch_FK FOREIGN KEY(misch_id)
 REFERENCES Mischung
)
```

```
CREATE TABLE Lieferant
(lf_id INTEGER NOT NULL CONSTRAINT lf_PK PRIMARY KEY,
 name VARCHAR(40) NOT NULL,
 adresse VARCHAR(80) NOT NULL
)
```

```
CREATE TABLE Lager
(lager_id INTEGER NOT NULL CONSTRAINT lager_PK PRIMARY KEY,
 kap INTEGER NOT NULL
)
```

```

CREATE TABLE Futtermittel
(fm_id INTEGER NOT NULL CONSTRAINT fm_PK PRIMARY KEY,
name VARCHAR(20) NOT NULL,
lf_id INTEGER NOT NULL,
lager_id INTEGER NOT NULL,
CONSTRAINT fm_lf_FK FOREIGN KEY(lf_id) REFERENCES Lieferant,
CONSTRAINT fm_lager_FK FOREIGN KEY(lager_id) REFERENCES Lager
)

CREATE TABLE Mischung_enthaelt
(misch_id INTEGER NOT NULL,
fm_id INTEGER NOT NULL,
CONSTRAINT me_PK PRIMARY KEY(misch_id, fm_id),
CONSTRAINT me_misch_FK FOREIGN KEY(misch_id) REFERENCES Mischung,
CONSTRAINT me_fm_FK FOREIGN KEY(fm_id) REFERENCES Futtermittel
)

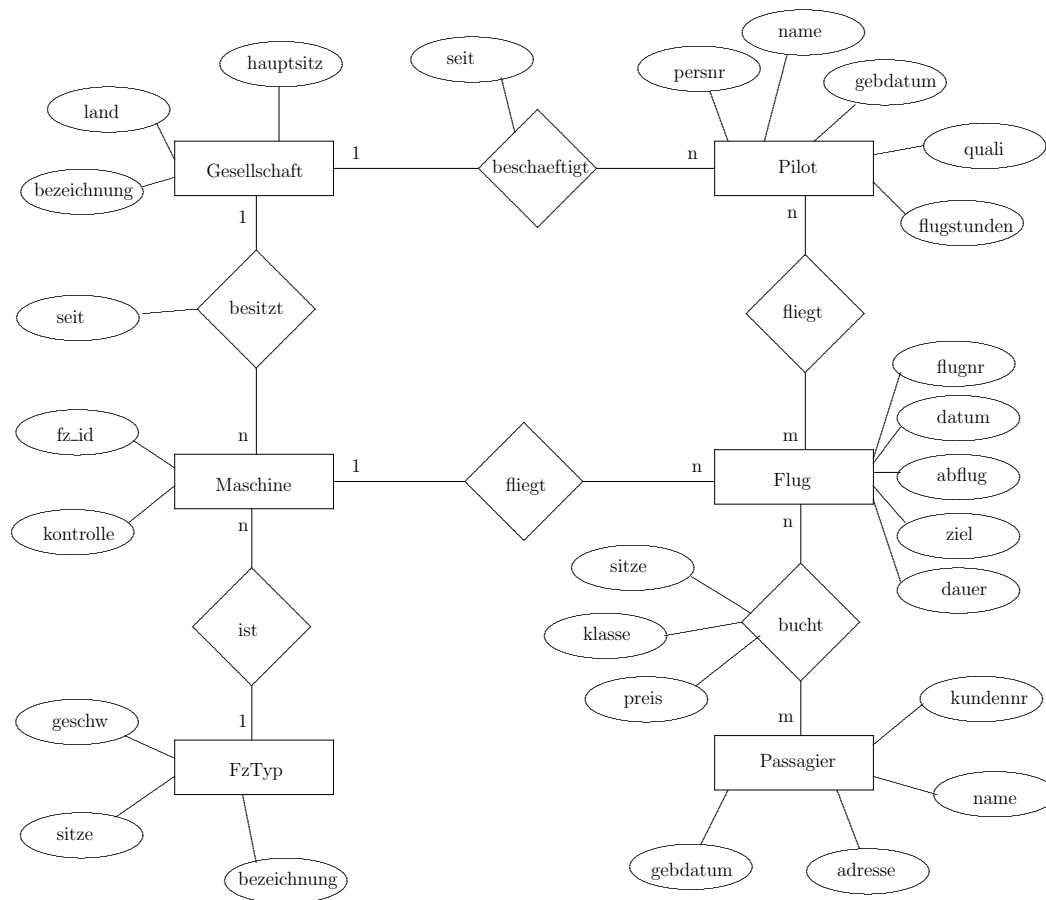
CREATE TABLE Arten_im_Zoo
(zoo_id INTEGER NOT NULL,
artname VARCHAR(20) NOT NULL,
CONSTRAINT aiz_PK PRIMARY KEY(zoo_id, artname),
CONSTRAINT aiz_zoo_FK FOREIGN KEY(zoo_id) REFERENCES Zoo,
CONSTRAINT aiz_art_FK FOREIGN KEY(artname) REFERENCES Tierart
)

```

## 5. Aufgabe:

Eine Fluggesellschaft (eindeutige Bezeichnung, Land, Hauptsitz) besitzt seit einem bestimmten Datum Flugzeuge (eindeutige Flugzeug-Nummer, Datum der letzten Kontrolle) unterschiedlichen Typs (Typbezeichnung, Anzahl Sitze, Geschwindigkeit). Jede Fluggesellschaft beschäftigt seit einem bestimmten Datum eine Menge von Piloten (Personal-Nummer, Name, Geburtsdatum, Qualifikation, Flugstunden). Passagiere (Kundennummer, Name, Adresse, Geburtsdatum) haben Flüge (Flug-Nummer, Datum, Abflugort, Zielflughafen, Dauer) gebucht (Anzahl Sitze, Klasse, Preis). Ein Flug wird von mehreren Piloten und einem bestimmten Flugzeug realisiert.

### ER-Diagramm:



### Relationen-Schema:

(Primärschlüssel sind unterstrichen)

Gesellschaft(bezeichnung, land, hauptsitz)

Pilot(persnr, name, gebdatum, quali, flugstunden, beschäftigt\_bei → Gesellschaft(bezeichnung))

Maschine(fz\_id,kontrolle,typ → FzTyp(bezeichnung),gehört → Gesellschaft(bezeichnung))  
 FzTyp(bezeichnung,sitze,geschw)  
 Flug(flugnr,datum,abflug,ziel,dauer,flugzeug → Maschine(fz\_id))  
 Passagier(kundennr,name,gebdatum,adresse)  
 Pilot\_fliegt(persnr → Pilot(persnr),flugnr → Flug(flugnr))  
 Buchung(flugnr → Flug(flugnr),kundennr → Passagier(kundennr),sitze,klasse,preis)

### SQL-Anweisungen zum Erzeugen der Tabellen:

```

CREATE TABLE Gesellschaft
  (bezeichnung VARCHAR(20) NOT NULL CONSTRAINT gesell_PK PRIMARY KEY,
   land VARCHAR(20) NOT NULL,
   hauptsitz VARCHAR(20) NOT NULL
  )

```

```

CREATE TABLE Pilot
  (persnr INTEGER NOT NULL CONSTRAINT pilot_PK PRIMARY KEY,
   name VARCHAR(40) NOT NULL,
   gebdatum DATETIME NOT NULL,
   quali VARCHAR(20) NOT NULL,
   flugstunden INTEGER NOT NULL,
   beschaeftigt_bei VARCHAR(20),
   CONSTRAINT pilot_gesell_FK FOREIGN KEY(beschaeftigt_bei)
     REFERENCES Gesellschaft(bezeichnung)
  )

```

```

CREATE TABLE FzTyp
  (bezeichnung VARCHAR(20) NOT NULL CONSTRAINT Fz_PK PRIMARY KEY,
   sitze INTEGER NOT NULL,
   geschw INTEGER NOT NULL
  )

```

```

CREATE TABLE Maschine
  (fz_id INTEGER NOT NULL CONSTRAINT masch_PK PRIMARY KEY,
   kontrolle DATETIME NOT NULL,
   typ VARCHAR(20) NOT NULL,
   gehoert VARCHAR(20),
   CONSTRAINT masch_typ_FK FOREIGN KEY(typ)
     REFERENCES FzTyp(bezeichnung),
   CONSTRAINT masch_gesell_FK FOREIGN KEY(gehört)
     REFERENCES Gesellschaft(bezeichnung)
  )

```

```

CREATE TABLE Flug
(flugnr VARCHAR(10) NOT NULL CONSTRAINT flug_PK PRIMARY KEY,
 datum DATETIME NOT NULL,
 abflug VARCHAR(20) NOT NULL,
 ziel VARCHAR(20) NOT NULL,
 dauer INTEGER NOT NULL,
 flugzeug INTEGER NOT NULL,
 CONSTRAINT flug_fz_FK FOREIGN KEY(flugzeug)
 REFERENCES Maschine(fz_id)
)

```

```

CREATE TABLE Passagier
(kundennr INTEGER NOT NULL CONSTRAINT pass_PK PRIMARY KEY,
 name VARCHAR(40) NOT NULL,
 gebdatum DATETIME NOT NULL,
 adresse VARCHAR(80) NOT NULL
)

```

```

CREATE TABLE Pilot_fliegt
(persnr INTEGER NOT NULL,
 flugnr VARCHAR(10) NOT NULL,
 CONSTRAINT Pf_PK PRIMARY KEY(persnr,flugnr),
 CONSTRAINT Pf_Pilot_FK FOREIGN KEY(persnr)
 REFERENCES Pilot,
 CONSTRAINT Pf_Flug_FK FOREIGN KEY(flugnr)
 REFERENCES Flug
)

```

```

CREATE TABLE Buchung
(flugnr VARCHAR(10) NOT NULL,
 kundennr INTEGER NOT NULL,
 sitze INTEGER NOT NULL,
 klasse INTEGER NOT NULL,
 preis MONEY NOT NULL,
 CONSTRAINT buchung_PK PRIMARY KEY(flugnr,kundennr),
 CONSTRAINT buch_fl_FK FOREIGN KEY(flugnr)
 REFERENCES Flug,
 CONSTRAINT buch_pass_FK FOREIGN KEY(kundennr)
 REFERENCES Passagier
)

```