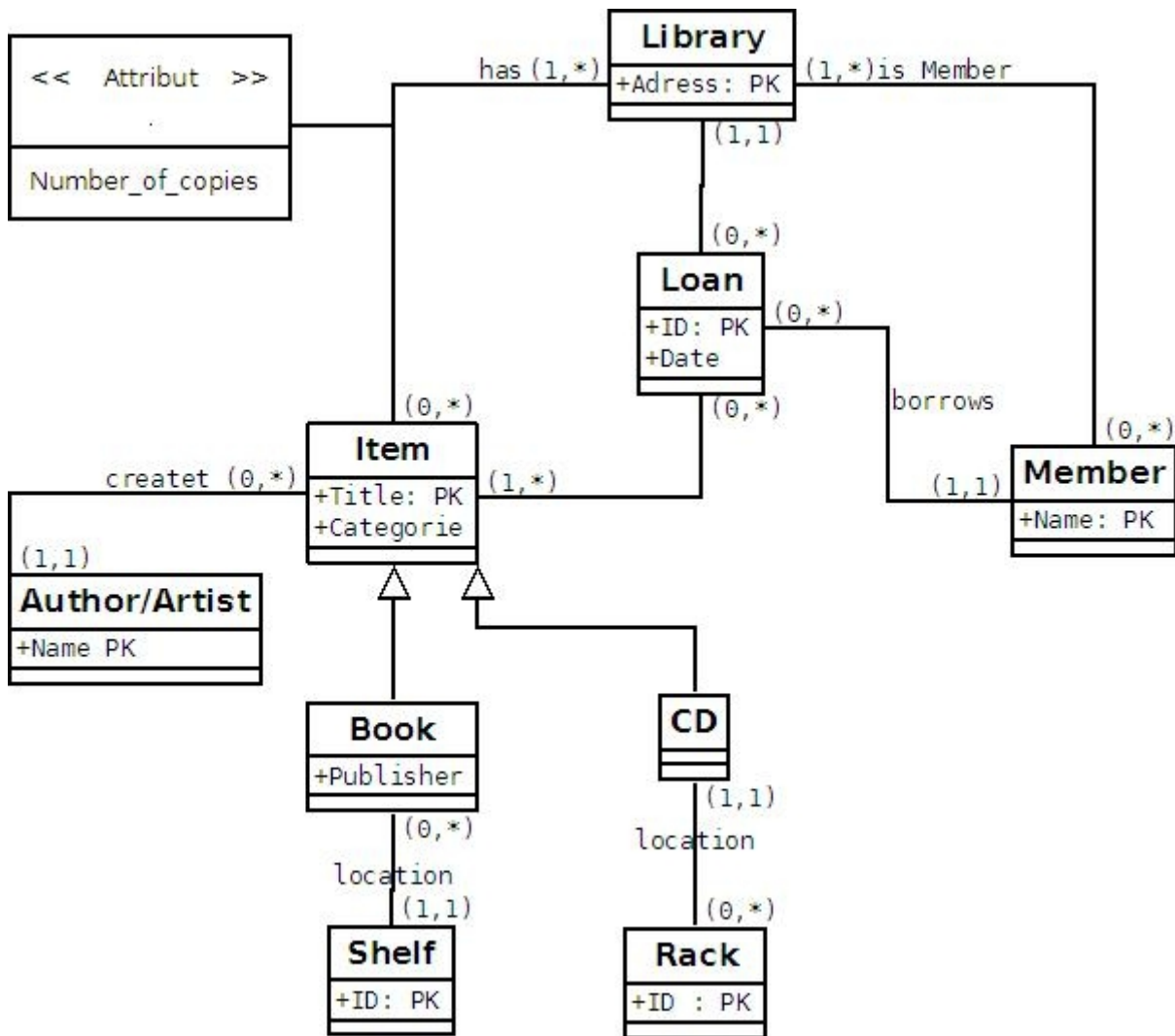


Aufgabe 1

UML-Modell:

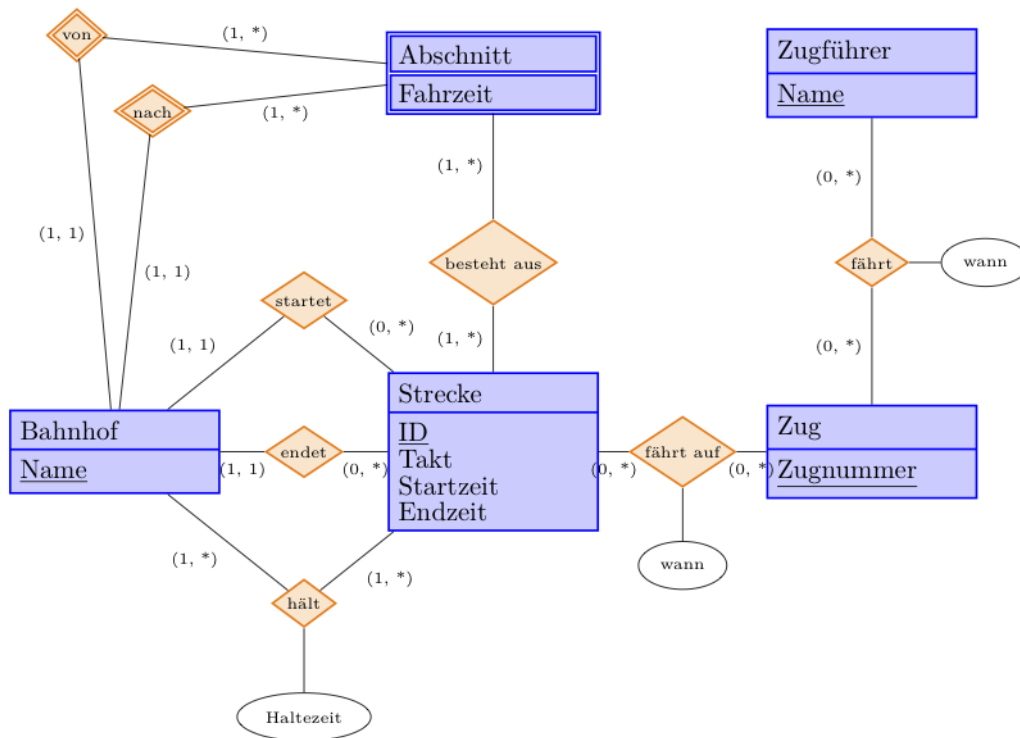


Folgende Integritätsbedingungen müssen überprüft werden:

- Um Mitglied zu werden muss man älter als 16 Jahre sein.
- Die Angabe „number of books and CDs in a library“ muss aus der Anzahl der Items in einer Library errechnet werden.
- Überprüfung ob die maximale Ausleihzeit überschritten wurde.
- Berechnung der nachzuzahlenden Gebühren.
- Dauer der „loan period“. Da diese überall gleich ist, wurde sie nicht als Attribut für „Library“ angelegt.

Aufgabe 2

a)



Als zusätzliche Integritätsbedingungen gelten die üblichen Beschränkungen – bei den Fahr- und Haltezeiten muss es sich um positive Werte handeln, und die Startzeit muss vor der Endzeit einer Strecke liegen.

Außerdem müssen die Streckenabschnitte konsistent sein, d.h. sie müssen genau einmal den Start- und Endbahnhof enthalten (es sei denn, die Strecke ist rund, siehe Ringbahn, in welchem Fall Startgleich Endbahnhof ist) und aus konsekutiven Streckenabschnitten bestehen.

Die Attribute der „fährt auf“- sowie der „fährt“-Relation müssen konsistent sein und es darf keine Überschneidungen geben – ein Zug darf zur gleichen Zeit nur von einem Zugführer gefahren werden, und kein Zug darf gleichzeitig auf mehreren Strecken verkehren.

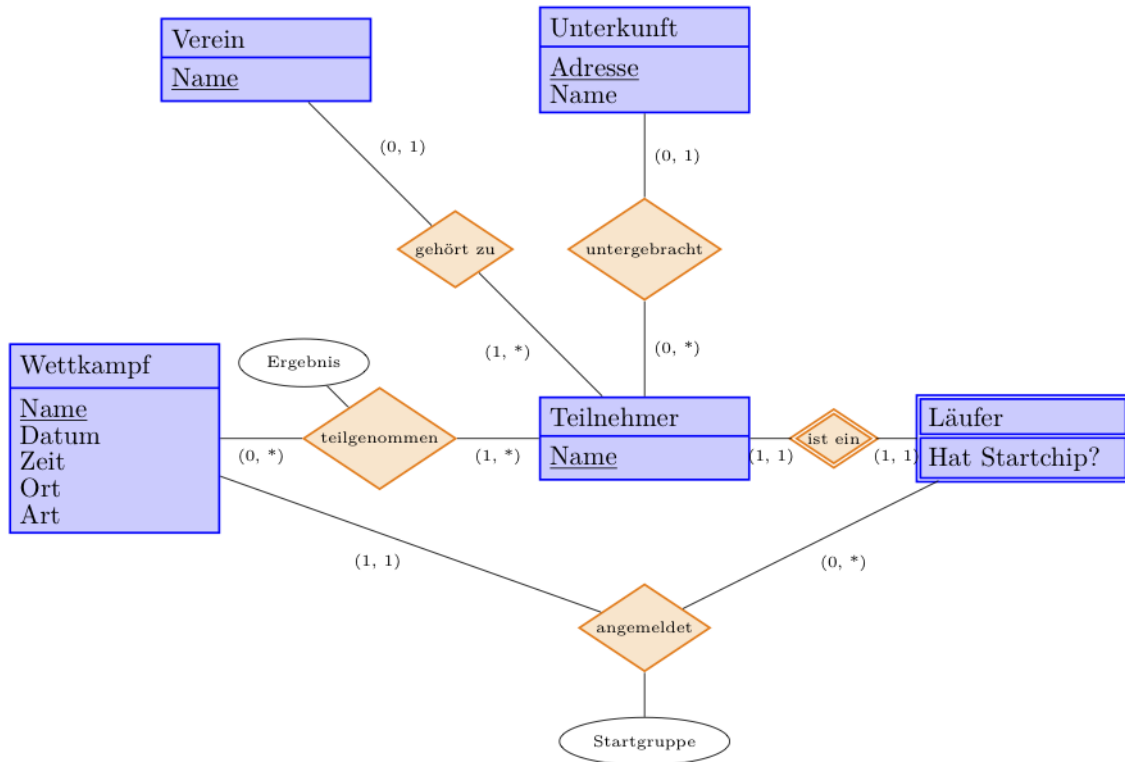
b)

- Bahnhof (Name)
- Strecke (Bezeichner, Takt, Startzeit, Endzeit, startet.Name, endet.Name)
- Abschnitt (von.Name, nach.Name)
- Zug (Zugnummer)
- Zugführer (Name)
- hält (Bahnhof.Name, Strecke.ID, Haltezeit)
- besteht_aus (Strecke.ID, von.Name, nach.Name)
- fährt_auf (Zug.Zugnummer, Strecke.ID, wann)

- fährt (Zugführer.Name, Zug.Zugnummer, wann)

Aufgabe 3

a)



Es muss zusätzlich berechnet werden, in welcher Startgruppe sich ein Läufer befindet. Dafür werden die Ergebnisse der „teilgenommen“-Relation verwendet – wenn der entsprechende Wettkampf nicht länger als ein Jahr zurückliegt.

Wettkämpfe **dürfen** zur selben Zeit am selben Ort ausgetragen werden (es gibt also Überschneidungen).

b)

- Verein (Name)
- Unterkunft (Name, Adresse)
- Wettkampf (Name, Datum, Zeit, Ort, Art)
- Teilnehmer (Name)
- Läufer (Teilnehmer.Name, Hat_Startchip, Wettkampf.Name, Startgruppe, Unterkunft.Adresse, Verein.Name)
- teilgenommen (Wettkampf.Name, Teilnehmer.Name, Ergebnis)

Aufgabe 4

a)

$$\forall (s_0, t_0): |(s_0, t_0, p)| \leq 1, \text{ mit } s_0 \in \text{Student}, t_0 \in \text{Thema}, p \in \text{Professor}$$

Studenten können ein Thema bei null oder einem Professor betreuen lassen.

$$\forall (t_0, p_0): 0 \leq |(t_0, p_0, s)|, \text{ mit } t_0 \in \text{Thema}, p_0 \in \text{Professor}, s \in \text{Student}$$

Ein Thema bei einem Professor wird von null oder mehr Studenten bearbeitet.

$$\forall (s_0, p_0): |(s_0, p_0, t)| \leq 1, \text{ mit } s_0 \in \text{Student}, p_0 \in \text{Professor}, t \in \text{Thema}$$

Ein Student bearbeitet bei einem Professor null oder ein Thema.

b)

$$\forall (s_0): |(s_0, t)| \leq 1, \text{ mit } s_0 \in \text{Student}, t \in \text{Thema}$$

Ein Student bearbeitet maximal ein Thema.

$$\forall (s_0): |(s_0, p)| \leq 1, \text{ mit } s_0 \in \text{Student}, p \in \text{Professor}$$

Ein Student hat maximal einen betreuenden Professor.

$$\forall (p_0): 0 \leq |(p_0, t)|, \text{ mit } p_0 \in \text{Professor}, t \in \text{Thema}$$

Ein Professor kann null oder mehr Themen betreuen.

$$\forall (p_0): 0 \leq |(p_0, s)|, \text{ mit } p_0 \in \text{Professor}, s \in \text{Student}$$

Ein Professor kann null oder mehr Studenten betreuen.

$$\forall (t_0): 0 \leq |(t_0, s)|, \text{ mit } t_0 \in \text{Thema}, s \in \text{Student}$$

Ein Thema kann von null oder mehr Studenten bearbeitet werden.

$$\forall (t_0): 0 \leq |(t_0, p)|, \text{ mit } t_0 \in \text{Thema}, p \in \text{Professor}$$

Ein Thema kann von null oder mehr Professoren gestellt werden.

Mit diesem konzeptionellen Modell lässt sich nicht ausdrücken, dass ein Thema bei einem Professor von einem Studenten bearbeitet wird.

c)

Ein Student kann null oder eine Arbeit schreiben.

Ein Professor kann null oder mehr Arbeiten vergeben.

Ein Thema kann in null oder mehr Arbeiten bearbeitet werden.

Eine Arbeit kann von einem Studenten bei einem Professor in einem Themengebiet geschrieben werden.

Das Modell drückt genau dasselbe aus wie in Aufgabe a).

Der Umstand dass mehrere Studenten ein Thema bei einem Professor bearbeiten können wird in diesem Modell dadurch ausgedrückt, dass es mehrere Objekte „Arbeit“ gibt, welche sich nur in „Student“ unterscheiden, somit wird dasselbe wie in a) ausgedrückt.