

Prof. Dr. Stefan Decker, PD Dr. Ralf Klamma

K. Fidomski, M. Slupczynski, S. Welten

03.05.2021

Datenbanken und Informationssysteme (Sommersemester 2021)

Übung 2 - ER Diagramm

Abgabe bis 10.05.2021 14:00 Uhr

Zu spät eingereichte Übungen werden nicht berücksichtigt.

Wichtige Hinweise

- Bei Nichtbeachtung dieser Hinweise wird die Abgabe mit 0 Punkten bewertet!
- Bitte reichen Sie Ihre Lösung nur in Dreier- oder Vierergruppen ein.
- Achten Sie auch darauf, dass Ihre Gruppe im Moodle korrekt eingerichtet ist.
- Bitte laden Sie Ihre schriftlichen Lösungen ins Moodle als ein zusammenhängendes PDF-Dokument hoch.
- Bitte geben Sie Namen, Matrikelnummern und Moodle-Gruppennummer auf der schriftlichen Lösung an.
- Wird offensichtlich die gleiche Lösung von zwei Gruppen abgegeben, dann erhalten beide Gruppen 0 Punkte.

Die Lösung zu diesem Übungsblatt wird in den Übungen am 10.05 und 12.05 2021 vorgestellt.

Bitte beachten Sie auch die aktuellen Ankündigungen im Moodle-Lernraum zur Vorlesung.

***) bezeichnet Bonusaufgaben.**

Vergessen Sie nicht, alle Gruppenmitglieder einzutragen!

Der Bearbeitungsmodus kann mit Doppelklick gestartet werden und mit Strg+Enter beendet werden.

Anleitung

Aufgaben bearbeiten

An verschiedenen Stellen in diesem Aufgabenblatt finden Sie die folgende Code-Struktur:

```
# YOUR CODE HERE  
raise NotImplementedError()
```

Ihre Lösung sollte immer statt dieser zwei Zeilen eingefügt werden.

Beispiel:

```
from assets.ER import ERDiagram  
g = ERDiagram()
```

```
# YOUR CODE HERE  
raise NotImplementedError()
```

```
g.display()
```

sollte also beispielsweise zu folgendem Code verändert werden:

```
from assets.ER import ERDiagram  
g = ERDiagram()  
  
g.add_node("Hello")  
g.add_attribute("Hello", "World")  
  
g.display()
```

Dies gilt auch für Markdown-Zellen, die mit [Bitte ersetzen Sie diesen Text mit Ihrer Antwort] gekennzeichnet sind.

Hier kann man mit Doppelklick die Zelle editieren und mit Hilfe von [Markdown](#) die Antwort formatieren.

ER-Diagramme zeichnen

In der Vorlesung haben Sie das Relationale Datenmodell kennengelernt. ER-Diagramme bieten ein Modell zur Darstellung von Entitäten und deren Relationen.

ER.py

Zum Erstellen von ER-Diagrammen, stellen wir die Klasse ERDiagram mit folgendem Interface vor:

```
from assets.ER import ERDiagram # Die Datei ER.py muss im assets Ordner neben der .ipynb liegen
g = ERDiagram()
```

Beziehungen zwischen Entity-Typen

Oft stehen zwei oder mehr Entity-Typen in Beziehung zueinander. Beziehungen werden im ER-Diagramm durch Rauten dargestellt und können ebenfalls Attribute haben. Es können außerdem Kardinalitäten angegeben werden um darzustellen, dass n Individuen des einen Entity-Typen mit m Individuen des anderen Entity-Typen in Beziehung stehen. Diese Kardinalitäten werden im ER-Diagramm an die Kanten zwischen Entity-Typ und Relation geschrieben.

Eine genauere Beschreibung dieser Klasse finden Sie im entsprechenden [Entity Relationship Tutorial](#)

Aufgabe 2.1 (ER-Diagramm mit (min,max)-Notation)

(5 Punkte)

Folgende Anforderungen skizzieren ein Informationssystem zur Speicherung von Daten des "Alles für Kinder" Bewertungsportals.

Erstellen Sie ein ER-Diagramm, das die unten genannten Teilanforderungen darstellt.

Verwenden Sie für Kardinalitätsrestriktionen die (min, max) -Notation.

- In diesem System gibt es Benutzer, die Rezensionen schreiben. Ein Benutzer hat einen Namen, eine eindeutige eMail-Adresse und einen Nickname. Zu jedem Benutzer werden die Länder, die er besucht hat, abgespeichert.
- Jede Rezension hat eine eindeutige Nummer, und eine Zusammenfassung. Zu jeder Rezension wird zusätzlich gespeichert an welchem Datum und in welcher Sprache sie geschrieben wurde. Benutzer können mehrere Rezensionen schreiben, jedoch wurde jede Rezension von genau einem Benutzer geschrieben.
- Rezensionen können Kommentare enthalten. Ein Kommentar enthält einen Freitext und ein Datum. Innerhalb der Rezension wird der Kommentar durch eine fortlaufende Nummer eindeutig identifiziert. Kommentare können Antworten auf einen anderen Kommentar sein. Ein Kommentar wird von genau einem Benutzer geschrieben, aber ein Benutzer kann mehrere Kommentare schreiben.
- Jede Rezension bezieht sich immer auf genau einen Ort. Jeder Ort hat einen Namen, eine eindeutige Nummer und eine Adresse bestehend aus Straße, PLZ, Stadt und Land. Außerdem werden die GPS Koordinaten, bestehend aus Breitengrad und Längengrad, von jedem Ort gespeichert.
- Ein Ort kann entweder ein Hotel, Spielplatz, Restaurant oder ein Geschäft sein. Andere Typen werden nicht erwartet. Zu jedem Hotel wird die Anzahl der

Sterne gespeichert. Zu jedem Spielplatz werden ein oder mehrere Spielgeräte gespeichert. Zu jedem Restaurant wird die Kategorie gespeichert. Zu jedem Geschäft wird der Typ gespeichert.

Hinweis: Attribute werden bei dieser Übung nicht benotet. Gefragt sind 5 Relationen.

WICHTIG:

Sie können die vorgestellten Codebeispiele benutzen, um das Diagramm zu erstellen, jedoch ist die Abgabe **schriftlich**, also muss diese Datei im nachhinein als PDF abgegeben werden.

In [3]:

```
from assets.ER import ERDiagram
g = ERDiagram()

g.neuer_knoten('Benutzer')
g.neuer_knoten('Kommentar', schwach = True)
g.neuer_knoten('Ort')
g.neuer_knoten('Rezension')

# YOUR CODE HERE
g.add_node('Land')

g.add_attribute('Benutzer', 'Name')
g.add_attribute('Benutzer', 'eMail', is_pk=True)
g.add_attribute('Benutzer', 'Nickname')
g.add_relation('Benutzer', 'besucht', 'Land', '(1,n)', '(0,n)')
# Jeder Benutzer hat ein Land besucht (angenommen er wurde auf Erde geboren), aber nicht jedes Land muss von einem Benutzer besucht werden

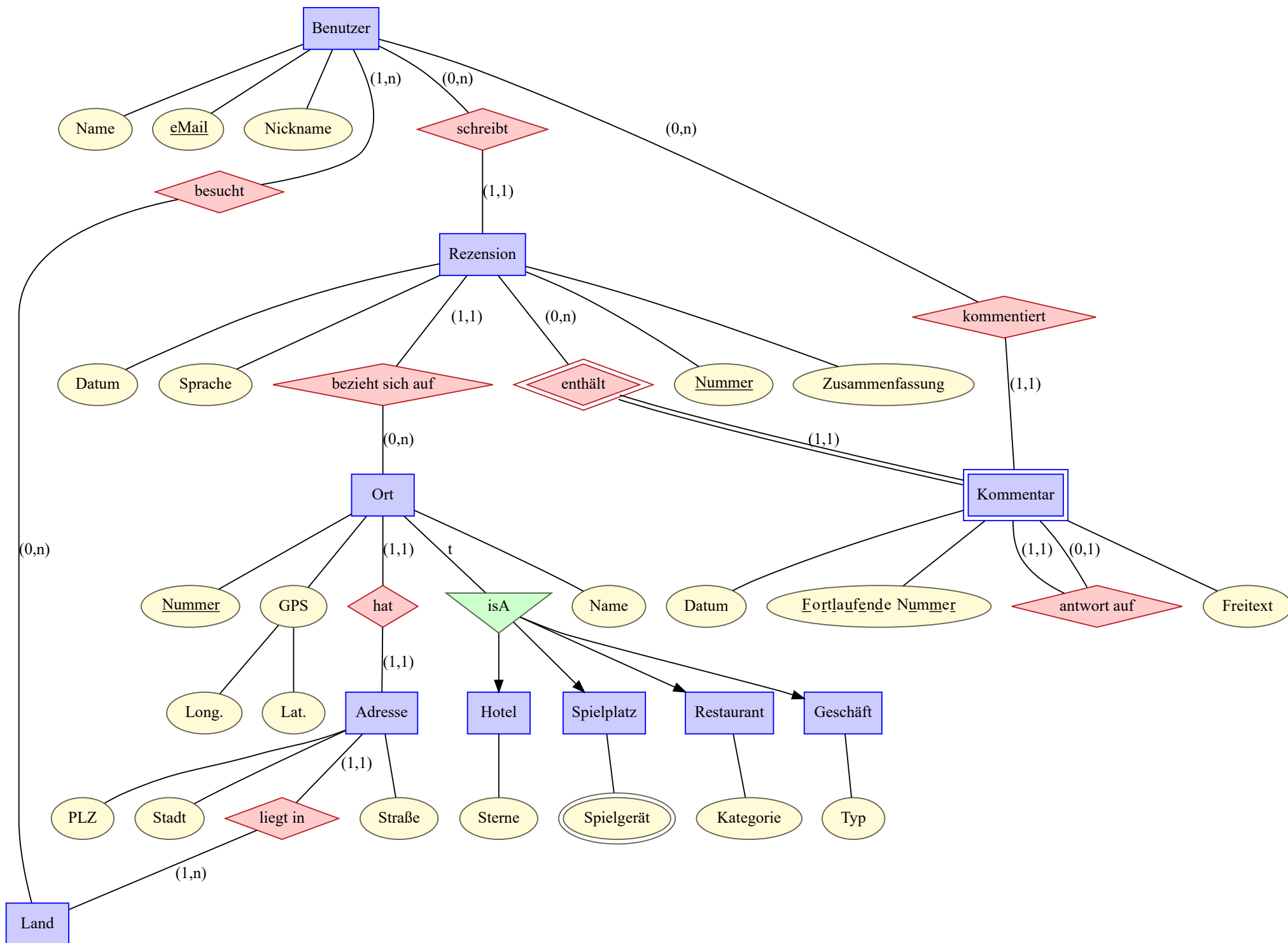
g.add_attribute('Rezension', 'Nummer', is_pk=True)
g.add_attribute('Rezension', 'Zusammenfassung')
g.add_attribute('Rezension', 'Datum')
g.add_attribute('Rezension', 'Sprache')
g.add_relation('Benutzer', 'schreibt', 'Rezension', '(0,n)', '(1,1)')
# Jede Rezension wird von genau einem Benutzer geschrieben, Benutzer könne aber mehrerer (oder auch keine) Rezension(en) schreiben

g.add_attribute('Kommentar', 'Freitext')
g.add_attribute('Kommentar', 'Datum')
g.add_attribute('Kommentar', 'Fortlaufende Nummer', is_pk=True, is_weak=True)
g.add_relation('Kommentar', 'antwort auf', 'Kommentar', '(0,1)', '(1,1)')
g.add_relation('Rezension', 'enthält', 'Kommentar', '(0,n)', '(1,1)', is_weak=True)
g.add_relation('Benutzer', 'kommentiert', 'Kommentar', '(0,n)', '(1,1)')

g.add_relation('Rezension', 'bezieht sich auf', 'Ort', '(1,1)', '(0,n)')
g.add_node('Adresse')
g.add_attribute('Adresse', 'Straße')
g.add_attribute('Adresse', 'PLZ')
g.add_attribute('Adresse', 'Stadt')
g.add_relation('Adresse', 'liegt in', 'Land', '(1,1)', '(1,n)')
g.add_attribute('Ort', 'Name')
g.add_attribute('Ort', 'Nummer', is_pk=True)
g.add_attribute('Ort', 'GPS', composed_of=['Long.', 'Lat.'])
g.add_relation('Ort', 'hat', 'Adresse', '(1,1)', '(1,1)')

g.add_is_a('Ort', ['Hotel', 'Spielplatz', 'Restaurant', 'Geschäft'], 't', is_disjunct=True)
g.add_attribute('Hotel', 'Sterne')
g.add_attribute('Spielplatz', 'Spielgerät', is_multiple=True)
g.add_attribute('Restaurant', 'Kategorie')
g.add_attribute('Geschäft', 'Typ')
```

g.display()



Aufgabe 2.2 (ER-Diagramm Spezialisierungen)

(5 Punkte)

Entwerfen Sie zu der folgenden textuellen Beschreibung ein Entity-Relationship-Modell.

Verwenden Sie zur Beschreibung der Kardinalitäten die (min, max) -Notation.

- Entwicklerstudios produzieren Computerspiele. Dies sind zum Beispiel FPS, RPG oder RTS Spiele. Einzelne Spiele können auch in mehrere dieser Kategorien fallen. Jedes Spiel wird nur von einem Entwicklerstudio produziert. Entwicklerstudios produzieren mindestens 1, maximal jedoch 20 verschiedene Spiele.
- Entwicklerstudios werden als Indie- oder als AAA-Spielhersteller kategorisiert, sie können jedoch auch beides sein. Andere Kategorien gibt es nicht.
- Computerspiele können Teil einer Serie sein. Dabei hat jedes Spiel beliebig viele Vorgänger und Nachfolger.
- Ein FPS Computerspiel hat verschiedene Spielmodi. Diese können zum Beispiel "King of the Hill", "Capture the Flag" oder "Domination" sein. Aus technischen Gründen können diese Spielmodi nur für FPS Computerspiele verwendet werden.
- Ein RPG Computerspiel kann entweder ein MMO-RPG sein, oder ein Single-Player RPG. Andere RPGs sind nicht vorgesehen

Hinweis: Es gibt keine Schlüssel oder Attribute in diesem Diagramm

```
In [4]: from assets.ER import ERDiagram
g = ERDiagram()

g.neuer_knoten('Computerspiel')
g.neuer_knoten('Entwicklerstudio')

# YOUR CODE HERE
g.add_is_a('Computerspiel', ['FPS', 'RPG', 'RTS'], 'p', is_disjunct=False)
g.add_relation('Entwicklerstudio', 'produziert', 'Computerspiel', '(1,20)', '(1,1)')

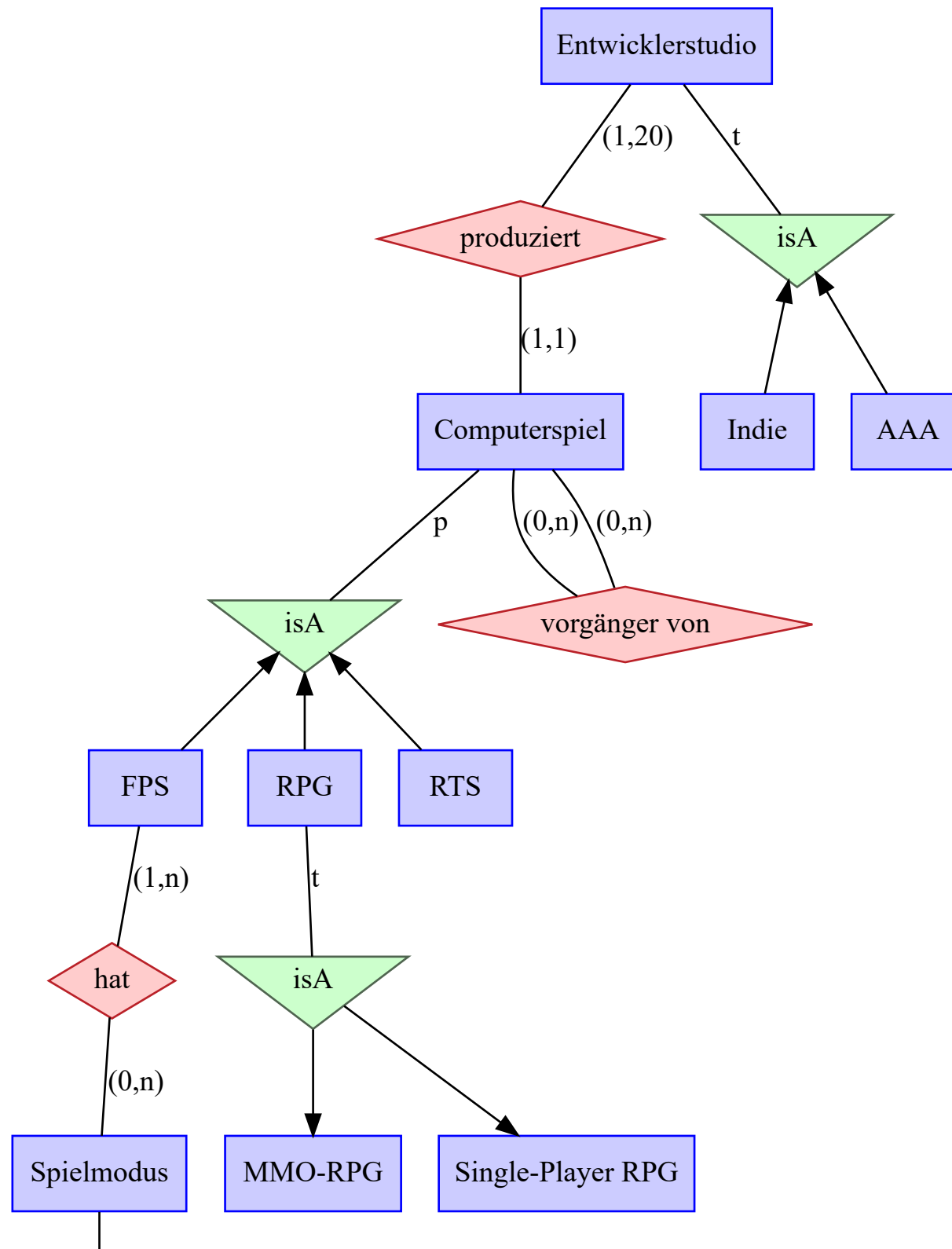
g.add_is_a('Entwicklerstudio', ['Indie', 'AAA'], 't', is_disjunct=False)

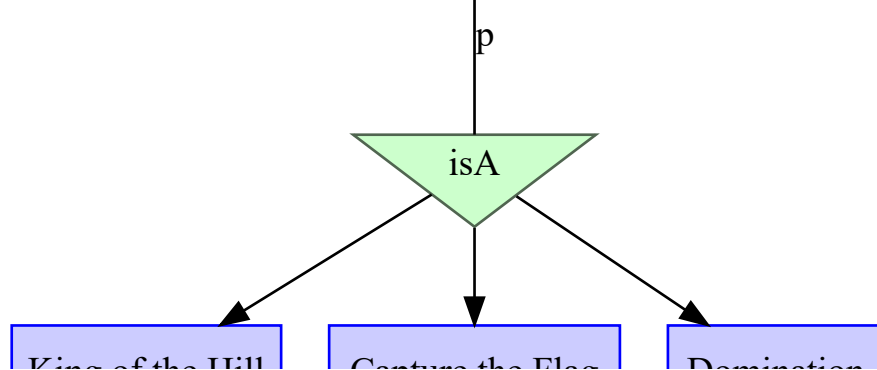
g.add_relation('Computerspiel', 'vorgänger von', 'Computerspiel', '(0,n)', '(0,n)')

g.add_node('Spielmodus')
g.add_is_a('Spielmodus', ['King of the Hill', 'Capture the Flag', 'Domination'], 'p', is_disjunct=True)
g.add_relation('FPS', 'hat', 'Spielmodus', '(1,n)', '(0,n)')

g.add_is_a('RPG', ['MMO-RPG', 'Single-Player RPG'], 't', is_disjunct=True)

g.display()
```





Aufgabe 2.3 (Relationales Datenmodell)

(10 Punkte)

Geben Sie eine Umsetzung des folgenden ER-Diagramms in ein relationales Datenbankschema an. Bitte gruppieren Sie Ihre Lösung in Relationen und interrelationale Abhängigkeiten. Verwenden Sie die in der Vorlesung vorgestellte Notation.

Kurzreferenz LaTeX Notation

- $E1(\underline{PK}, A1, \dots)$
- $R1 \subseteq R2$
- $R1[ID] \cap R2[ID] = \emptyset$

In [5]:

```
from assets.ER import ERDiagram
g = ERDiagram()

g.add_node('Brauvorgang', is_weak = True)
g.add_attribute('Brauvorgang', 'Name', is_pk = True, is_weak = True)

g.add_node('Bier')

g.add_relation('Brauvorgang', 'benötigt', 'Zutat', '1', 'n')
g.add_relation('Brauvorgang', 'ergibt', 'Bier', 'n', '1')

g.neue_relation("Bier", "trinkt ", "Person", "1", "n")
g.neue_relation("", "trinkt ", "Ort", "1", "m")

g.add_attribute('Bier', 'BatchNr', is_pk = True)
#g.add_attribute('Bier', 'Menge')
g.add_attribute('Bier', 'Name')
g.add_attribute('ergibt', 'Menge')

g.add_node('Zutat')
g.add_attribute('Zutat', 'ID', is_pk = True)
g.add_attribute('Zutat', 'Einheit')
g.add_attribute('Zutat', 'MHD')

g.add_node('Brauerei')
g.add_relation('Brauerei', 'kooperiert', 'Brauerei', 'm', 'n')
g.add_attribute('kooperiert', 'Projekt')
g.add_attribute('kooperiert', 'Laufzeit')
g.add_relation('Brauerei', 'führt durch', 'Brauvorgang', '1', 'n', is_weak = True)

g.add_attribute('Brauerei', 'HRN', is_pk = True)
g.add_attribute('Brauerei', 'Name')
g.add_attribute('Brauerei', 'Rezeptur', is_multiple = True)

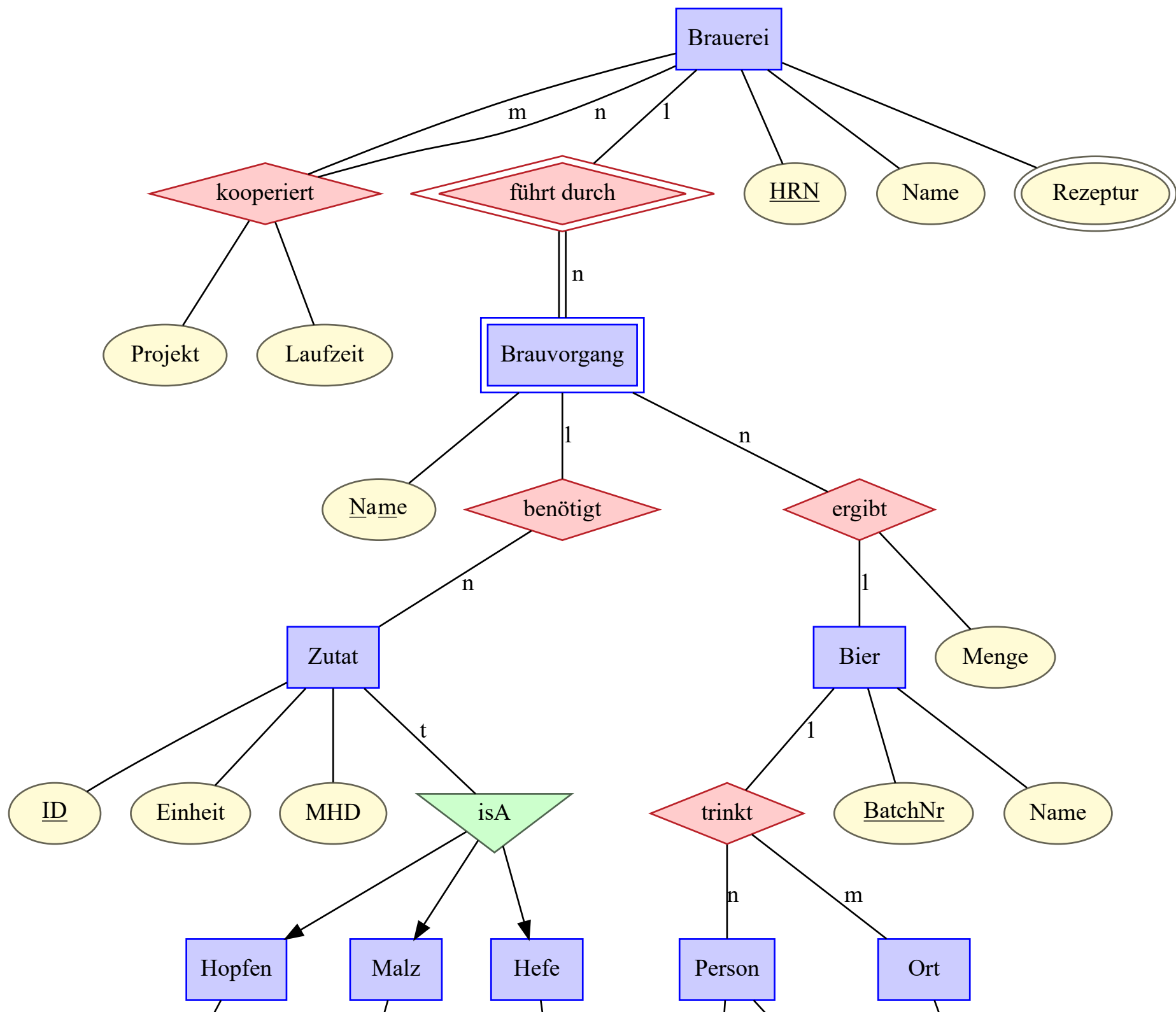
g.add_attribute('Ort', 'PLZ', is_pk = True)
g.add_attribute('Person', 'ID', is_pk = True)
g.add_attribute('Person', 'Alter')

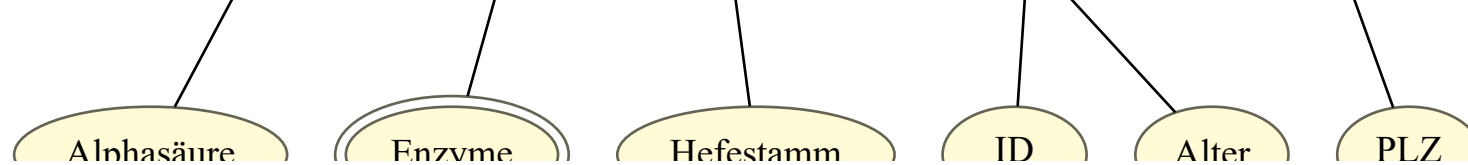
g.add_is_a('Zutat', ['Hopfen', 'Malz', 'Hefe'], super_label = 't', is_disjunct = True)
g.add_attribute('Hopfen', 'Alphasäure')

g.add_attribute('Malz', 'Enzyme', is_multiple = True)

g.add_attribute('Hefe', 'Hefestamm')

g.display()
```





1. Brauerei(HRN, Name)
2. Rezeptur(BrauereiHRN)
3. Rezeptur[BrauereiHRN] \subseteq Brauerei[HRN]
4. Kopiert(Vorgänger, Nachfolger, Projekt, Laufzeit)
5. Kopiert[Vorgänger] \subseteq Brauerei[HRN]
6. Kopiert[Nachfolger] \subseteq Brauerei[HRN]
7. Brauvorgang(Name, BierBatchNrErgibt, Menge)
8. Brauvorgang[HRN] \subseteq Brauerei[HRN]
9. Hopfen(Alphasäure, ID, Einheit, MHD, BrauvorgangNameBenötigt)
10. Malz(ID, Einheit, MHD, BrauvorgangNameBenötigt)
11. Enzyme(MalzID)
12. Enzyme[MalzID] \subseteq Malz[ID]
13. Hefe(Hefestamm, ID, Einheit, MHD, BrauvorgangNameBenötigt)
14. Hopfen[ID] \cap \ Malz[ID] \cap Hefe[ID] = \emptyset
15. Hopfen[BrauvorgangNameBenötigt] \cup Malz[BrauvorgangNameBenötigt] \cup Hefe[BrauvorgangNameBenötigt] \subseteq Brauvorgang[Name]
16. Bier(BatchNr, Name)
17. Person(ID, Alter)
18. Ort(PLZ)
19. Trinkt(ID, PLZ, BatchNr)
20. Trinkt[ID] \subseteq Person[ID]
21. Trinkt[PLZ] \subseteq Ort[PLZ]
22. Trinkt[BatchNr] \subseteq Bier[BatchNr]