Datenbanken
Wintersemester 2019/20
Prof. Dr. W. May

1. Übungsblatt: ER-Modell und Relationales Modell

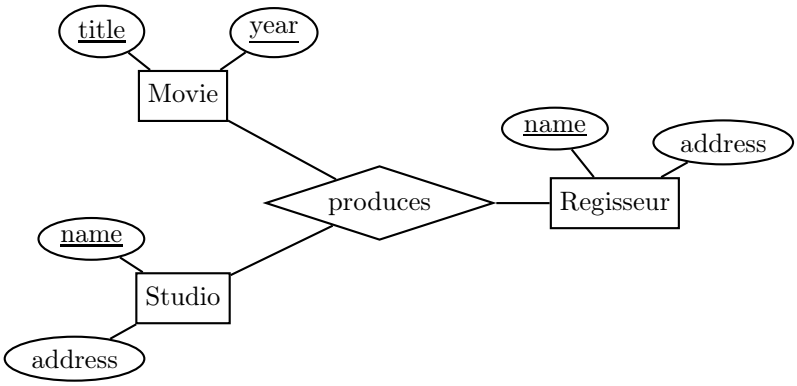
Besprechung voraussichtlich am 13./14/21.11.2019

Aufgabe 1 (ER-Modell: Film) Geben Sie ein ER-Modell für den folgenden Sachverhalt an: Filme werden in Filmstudios von Regisseuren gedreht. Filmstudios gehören einem Besitzer. In Filmen treten Schauspieler auf. Schauspieler erhalten eine Gage für jeden ihrer Verträge.

Entwickeln Sie zuerst ein einfaches Modell, und überlegen Sie dann, wie und ob Sie das Modell ergänzen könnten, um z.B. zu modellieren, dass sowohl Schauspieler als auch Regisseure und Besitzer von Filmstudios Personen sind, und manche Personen auch im selben Film oder in verschiedenen Filmen in mehreren dieser Rollen auftreten.

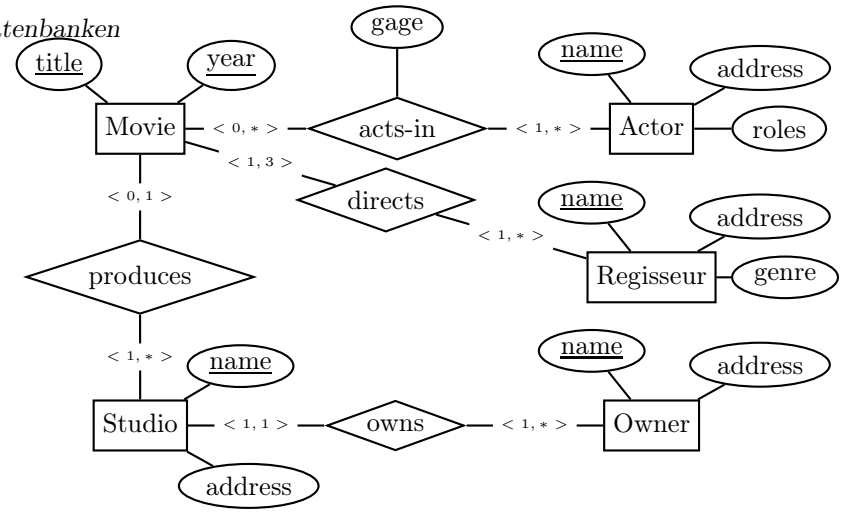
Nur die (bzw. eine) korrekte Lösung zu zeigen, wäre zu einfach und auch didaktisch nicht sinnvoll. Aus diesem Grund werden hier verschiedene Wege und Holzwege diskutiert.

Erster Ansatz: Man betrachtet den Satz "Filme werden in Filmstudios von Regisseuren gedreht." Naheliegend ist hier eine dreistellige Beziehung:



[Präsentation: Umsetzung in Relationen.] Man sieht an der Relation, bzw. an der anzustellenden Kardinalitätsbetrachtung, dass diese Modellierung nicht davor bewahrt, für einen Film, der von mehreren Regisseuren gemeinsam gedreht wird, auch unterschiedliche Studios zu speichern. Also sollte man diese Modellierung so nicht wählen.

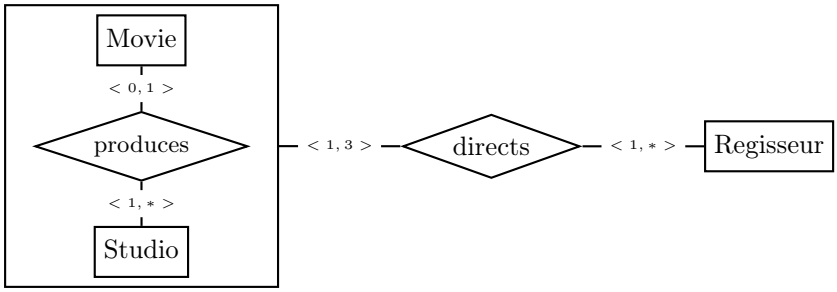
Naheliegend ist, die Beziehung in zwei zweispaltige Beziehungen (*Movie-Studio*) und (*Movie-Regisseur*) aufzulösen, womit man insgesamt die folgende "Musterlösung" erreicht:



Roles ist ein spezifisches Attribut von Schauspielern, das angibt, welche Arten von Rolle ein Schauspieler spielen kann (Erweiterung: die Beziehung *acts-in* ebenfalls *roles* zu erweitern, das angibt, welche Art von Rolle ein Schauspieler in diesem Film spielt). *Genre* ist ein spezifisches Attribute von Regisseuren, das angibt, welche Art von Filmen ein Regisseur dreht (auch hier kann man überlegen, *genre* als Attribut des Films zu nehmen).

Alternative Modellierung der linken Seite.

Eine mögliche Alternative ist, die dreistellige Beziehung *directs* aufzuspalten, indem man die Beziehung *produces* zwischen einem Studio und einem Film betrachtet, diese aggregiert (Produktion), und das Aggregat in eine *directs*-Beziehung mit Regisseuren stellt. *Owner* stehen weiterhin in Beziehung mit ihrem Studio, Schauspieler kann man wahlweise in Beziehung mit dem Film oder der Produktion stellen.



Hinweis: man erhält dasselbe relationale Modell, wenn man an der richtigen Stelle einbezieht, dass die 1:n-Beziehung zwischen Film und Studio dazu führt, dass der Schlüssel von *Production* nur (*title,year*) ist. (Es ist somit auch eigentlich keine echte Aggregation, da jede Instanz der Aggregation auch genau einer Instanz des Entitätstyps "Film" entspricht.)

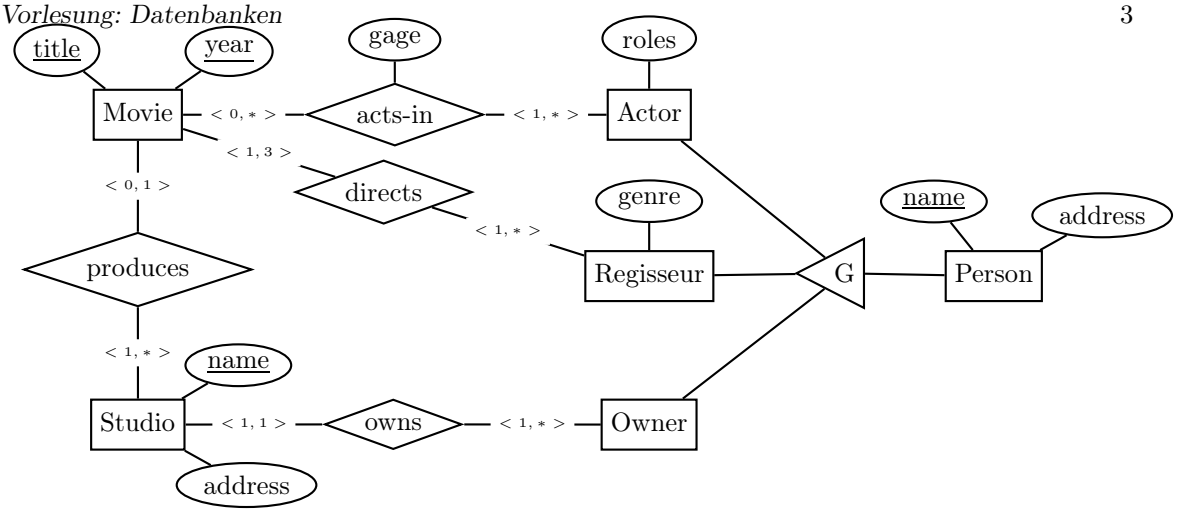
Hinweis: eine Aggregation (*Film, Regisseur*) und eine Beziehung des Aggregates zu *Studio* ist keine geeignete Modellierung, da dann für Filme, die von mehreren Regisseuren zusammen gedreht würden, auch verschiedene Studios angegeben werden könnten.

Modellierung mit Generalisierung "Person".

Betrachtet man bei der obigen Modellierung die Umsetzung ins relationale Modell, so stellt man fest, Information zu Personen, die z.B. sowohl als Regisseur als auch als Schauspieler gespeichert sind, *redundant* gehalten wird. Dies kostet nicht nur Platz, sondern kann auch zu inkonsistenten Datenbankzuständen führen:

Actor(AS, Hollywood Drive 42 - LosAngeles) und Regisseur(AS, Graben 1 - Wien).

Es ist daher sinnvoll, einen Entitätstyp *Person* als Generalisierung dieser Typen einzuführen:



Generalisierung bedeutet hier, dass

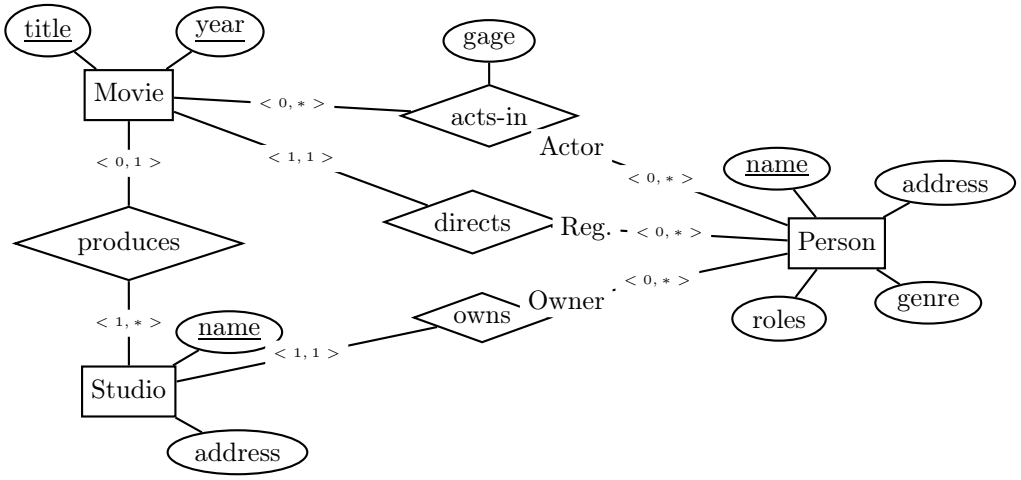
$$Persons = Actors \cup Regisseurs \cup Owners$$

ist. Würde man stattdessen eine Spezialisierung wählen, könnte man auch Personen haben, die nicht in eine dieser spezielleren Klassen fallen.

Bei einer Umsetzung ins relationale Modell erhält man je eine Tabelle für *Person(Name,Address)*, *Actor(Name,role)* (ggf. mehrwertig), *Regisseur(Name,genre)*, die jeweils den Primärschlüssel *Name* einer Person referenzieren.

Modellierung nur mit "Person" und Rollen.

Als dritte Möglichkeit könnte man nur Personen modellieren und ihre Rollen (in der Anwendungswelt) durch Rollen(bezeichner im ER-Modell) darstellen:



Die Attribute *roles* und *genre* müssen damit alle bei Person angesiedelt sein. Damit ist nicht klar, dass z.B. nur Schauspieler das Attribut *roles* besitzen. Außerdem sind potentiell viele Nullwerte in der Tabelle enthalten.

Aufgabe 2 (Umsetzung in das relationale Modell: Film) In Aufgabe 1 haben Sie ein ER-Modell für eine kleine Filmdatenbank erstellt. Transformieren Sie dieses in ein relationales Modell.

Einfachste Modellierung

Annahme: roles wird als String als Kommaliste abgelegt

Movie: (<u>title</u>, <u>year</u>)
Studio: (<u>name</u>, address)
Actor: (<u>name</u>, address, roles)
Regisseur: (<u>name</u>, address, genre)
Owner: (<u>name</u>, address)

produces: (studio, <u>title</u>, <u>year</u>)
acts-in: (<u>actor</u>, <u>title</u>, <u>year</u>, gage)
directs: (<u>regisseur</u>, <u>title</u>, <u>year</u>)
owns: (<u>owner</u>, <u>studio</u>)

Alternative, um zu einem Schauspieler mehrere Rollencharakterisierungen ablegen zu können:

Actor: (<u>name</u>, address)
ActorRoles: (<u>name</u>, <u>role</u>)

Alternativen

Da ein Film jeweils nur in einem Studio gedreht werden kann (siehe Beziehungskomplexitäten im ER-Diagramm), kann man die *produces*-Beziehung in *Movie* mit hineinnehmen:

Movie: (<u>title</u>, <u>year</u>, studio)

Modelliert man Filmproduktion als Aggregation aus *Movie, Studio*, erhält man dafür genau die Relation

production: (<u>title</u>, <u>year</u>, studio)

Die Produktion steht nun in Beziehung zu Regisseuren, womit man hierfür die Relation

directs: (regisseur, title, year)

erhält.

Generalisierung als Personen

... verschiedene Möglichkeiten.

- Personen mit Name und Adresse, roles und genre jeweils in Relationen Actor bzw. Regisseur die als Fremdschlüssel den Namen einer Person haben. Die Fremdschlüsselbeziehungen der Relationen zu den Beziehungstypen referenzieren die Relationen Actor, Regisseur, Owner:
  Movie: (<u>title</u>, <u>year</u>)
  Studio: (<u>name</u>, address)
  Person: (<u>name</u>, address)
  Actor: (<u>name</u>, roles)
  Regisseur: (<u>name</u>, genre)
  Owner: (<u>name</u>)

  produces: (studio, <u>title</u>, <u>year</u>)
  acts-in: (<u>actor</u>, <u>title</u>, <u>year</u>, gage)
  directs: (<u>regisseur</u>, <u>title</u>, <u>year</u>)
  owns: (<u>owner</u>, <u>studio</u>)

  $Actor.name \rightarrow Person.name$ (und analog)
  $acts\text{-}in.actor \rightarrow Actor.name$
  Aufgrund der detaillierten Modellierung sind die Konsistenzbedingungen an die Datenbank sehr scharf.

- Die vier Relationen *Person*, *Actor*, *Regisseur*, *Owner* kosten relativ viel Platz, weil die Namen mehrfach abgelegt sind (als *Person(AS, LosAngeles)*, *Actor(AS, hero)* und *Regisseur(AS, action)*. Man kann sie einsparen, wenn man stattdessen *roles* und *genre* als Attribute zu *Person* nimmt und mit Nullwerten auffüllt (z.B. *Person(AS, LosAngeles, hero, action, NULL)*). Dies entspricht auch dem ER-Diagramm, wo nur der Entitätstyp *Person* verwendet wird, und mit Rollenbezeichnungen gearbeitet wird:

  Movie: (<u>title</u>, <u>year</u>)
  Studio: (<u>name</u>, address)
  Person: (<u>name</u>, address, roles, genre)

  produces: (studio, title)
  acts-in: (person as actor, title, year, gage)
  directs: (person as regisseur, title, year)
  owns: (person as owner, studio)

  $acts.name \rightarrow Person.name$ etc.

  Man verliert die Integritätsbedingungen, dass nur Schauspieler das Attribut *roles* haben sowie dass die *acts*-Beziehung nur mit Schauspielern besteht (und analoges).
- Nun hat man ziemlich viele Nullwerte in *roles*, *genre*. Die meisten Personen werden Schauspieler sein. Man kann also z.B. das Attribut *roles* bei Person lassen, und *genre* wieder in separate Relationen für Regisseure (mit nur relativ wenigen Personen) legen:

  Movie: (<u>title</u>, <u>year</u>)
  Studio: (<u>name</u>, address)
  Person: (<u>name</u>, address, roles)
  Regisseur: (<u>name</u>, genre)
  Owner: (<u>name</u>)

  produces: (studio, title, year)
  acts-in: (person, title, year, gage)
  directs: (regisseur, title, year)
  owns: (owner, studio)