МОДЕЛИРАНЕ И АНАЛИЗ НА СОФТУЕР

Павел Кюркчиев
Ас. към ПУ „Паисий Хилендарски“
https://github.com/pkyurkchiev
@pkyurkchiev

ВРЪЗКИ И БИЗНЕС ПРАВИЛА

Логически и Физически модел на данни

Crow`s Foot Database Notation – логически и физически модел на данни

Връзки (Relationships)

- На ниво логически модел на данни става пълното описание на връзките (определянето на взаимоотношенията между обектите).

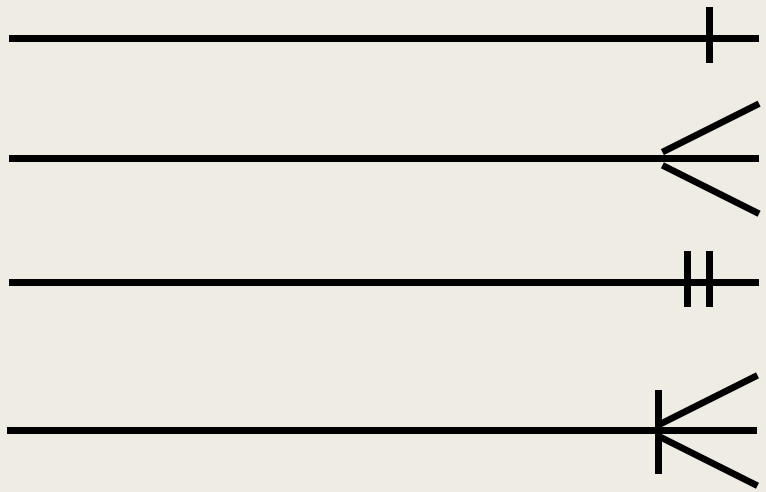
Връзки и конвенции

- Cardinality(кардиналност)
- Optionality(възможност)
- Assertions(твърдения)

Кардиналност (Cardinality)

- „Crown foot“ или три крака връзка може да бъде интерпретирана като връзка към „много“, а нейното отсъствие като връзка към „едно“.

Видове връзки



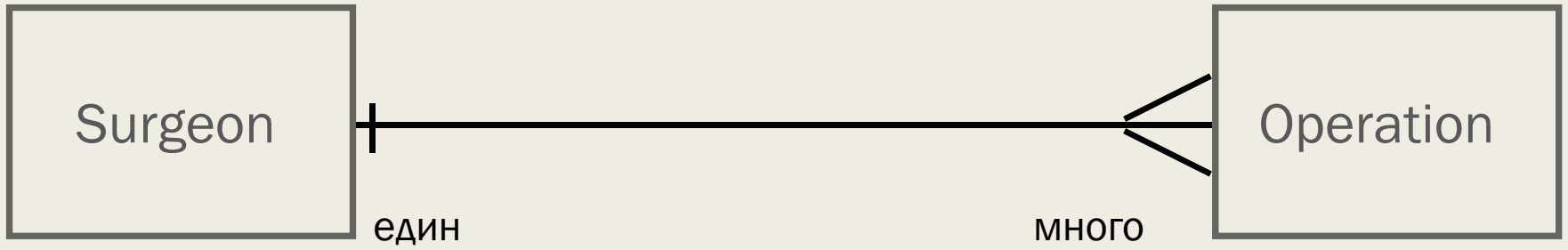
един

много

един(или само един)

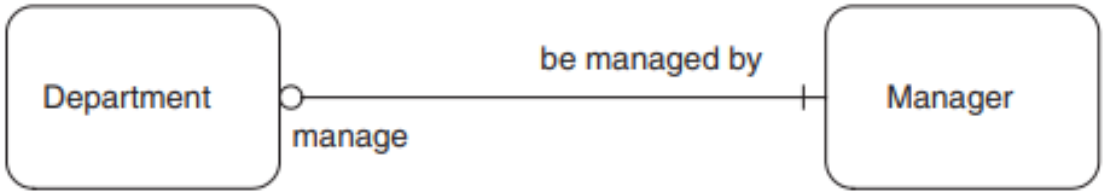
един или много

Модел на данни



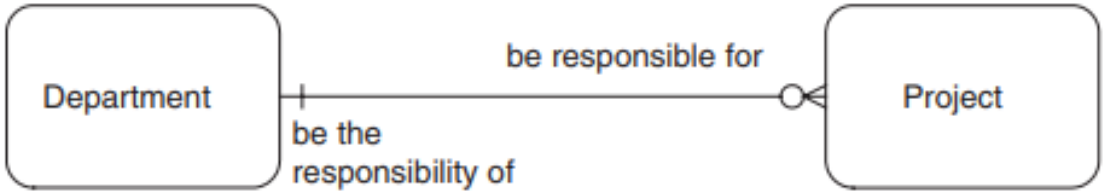
Бележки

- Възможно е в никои от двата края на връзката да няма „crow’s foot“. В този случай имаме 3 алтернативи one-to-one, one-to-many, и many-to-many връзки.
- Възможно е да има повече от една връзка между едни и същи класове обекти.
- Възможно е двата края на една и съща връзка да сочат към един и същи клас обект. Това се нарича „self-referencing“ или „рекурсивна“ връзка.



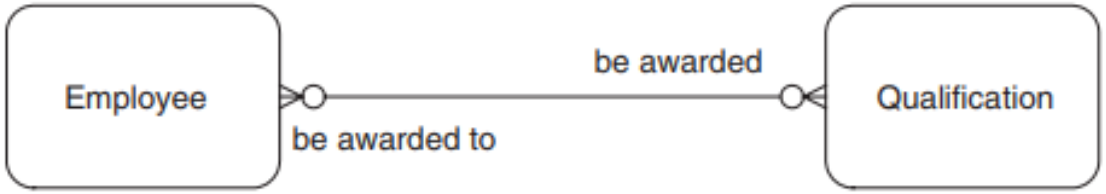
one-to-one

Each Department must be managed by one Manager.
Each Manager may manage one Department.



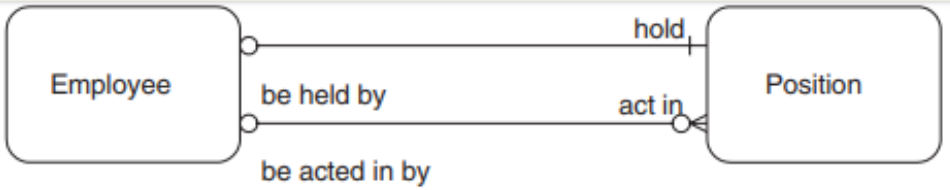
one-to-many

Each Department may be responsible for one or more Projects.
Each Project must be the responsibility of one Department.



many-to-many

Each Employee may be awarded one or more Qualifications.
Each Qualification may be awarded to one or more Employees.

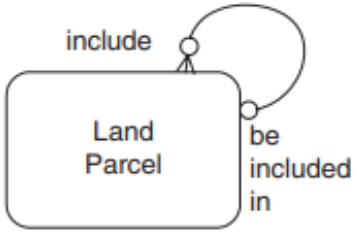


two relationships

Each Employee must hold one Position.
Each Position may be held by one Employee.

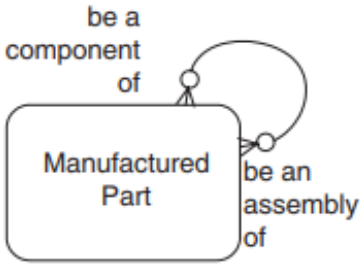
and

Each Employee may act in one or more Positions.
Each Position may be acted in by one Employee.



self-referencing one-to-many

Each Land Parcel may include one or more Land Parcels.
Each Land Parcel may be included in one Land Parcel.



self-referencing many-to-many

Each Manufactured Part may be an assembly of one or more Manufactured Parts.

Each Manufactured Part may be a component of one or more Manufactured Parts.

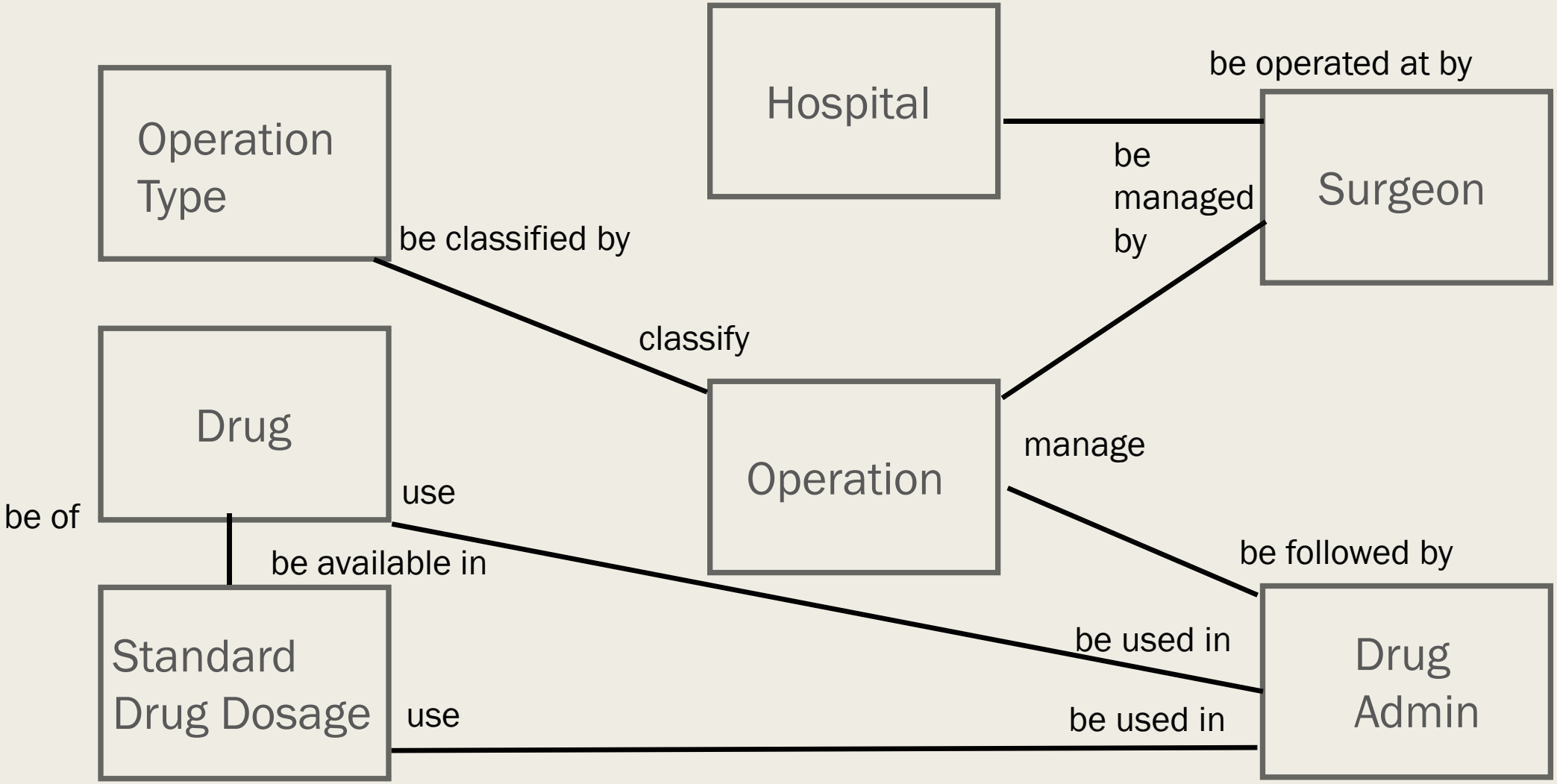
Many-to-Many връзки

- Many-to-many е връзка, която не може да бъде имплементирана във физическия модел. Най – често тази връзка бива свеждана до one-to-many.

Hospital диаграма

- Ако прегледаме диаграмата Hospital ще установим, че в нея има само връзки one-to-many. Това е резултата от нормализацията на диаграмата. Не трябва да забравяме, че всяка стойност на връзката сочи към един ред (представляваща една обект инстанция), и че всяка стойност може да се покаже многократно;

Модел на данни Hospital



Как ще имплементираме many-to-many връзка с Foreign Keys?

- Отговора е лесен, в стандартна релационна DBMS е невъзможно.
- Ние не можем да държим PK от Qualification в Employee таблицата, защото работника може да има повече от една квалификация. Същото се отнася и за противоположния случай.

- В нормализиран логически модел само с Employee и Qualification не може да бъде представена many-to-many връзка, но е възможно с повтарящи се групи и де-нормализация да представим Employee.

- Не нормализирана форма

EMPLOYEE (Employee Number, Employee Name, {Qualification ID, Qualification Name, Qualification Date})

Ако изходим от тази форма, чрез нормализации бихме могли да постигнем по елегантно решение.

- Първа нормална форма

EMPLOYEE (Employee Number, Employee Name)

EMPLOYEE QUALIFICATION (Employee Number*, Qualification ID, Qualification Name, Qualification Date)

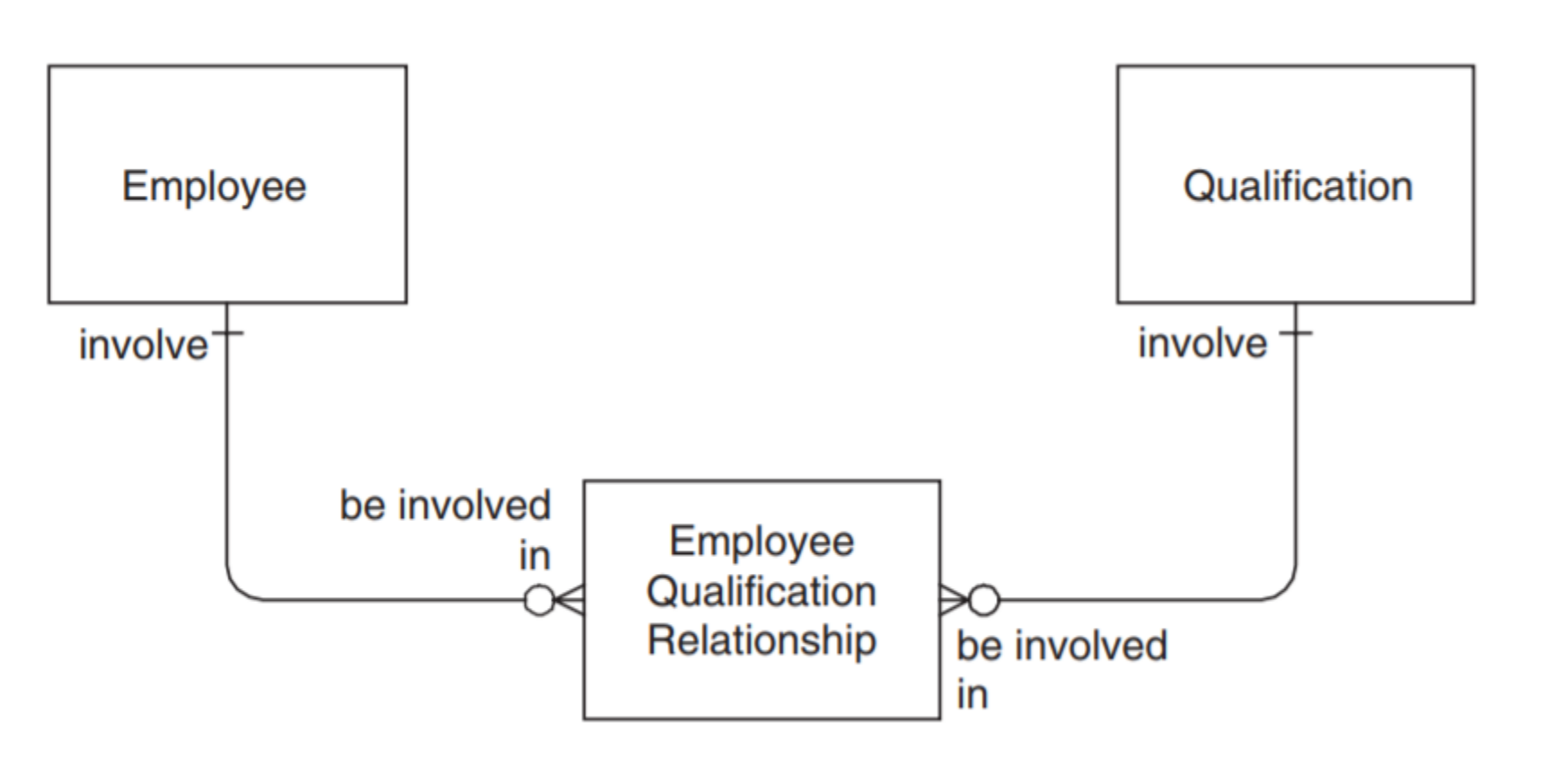
- Втора нормална форма

EMPLOYEE (Employee Number, Employee Name)

EMPLOYEE QUALIFICATION RELATIONSHIP (Employee Number*, Qualification ID*, Qualification Date)

QUALIFICATION (Qualification ID, Qualification Name)

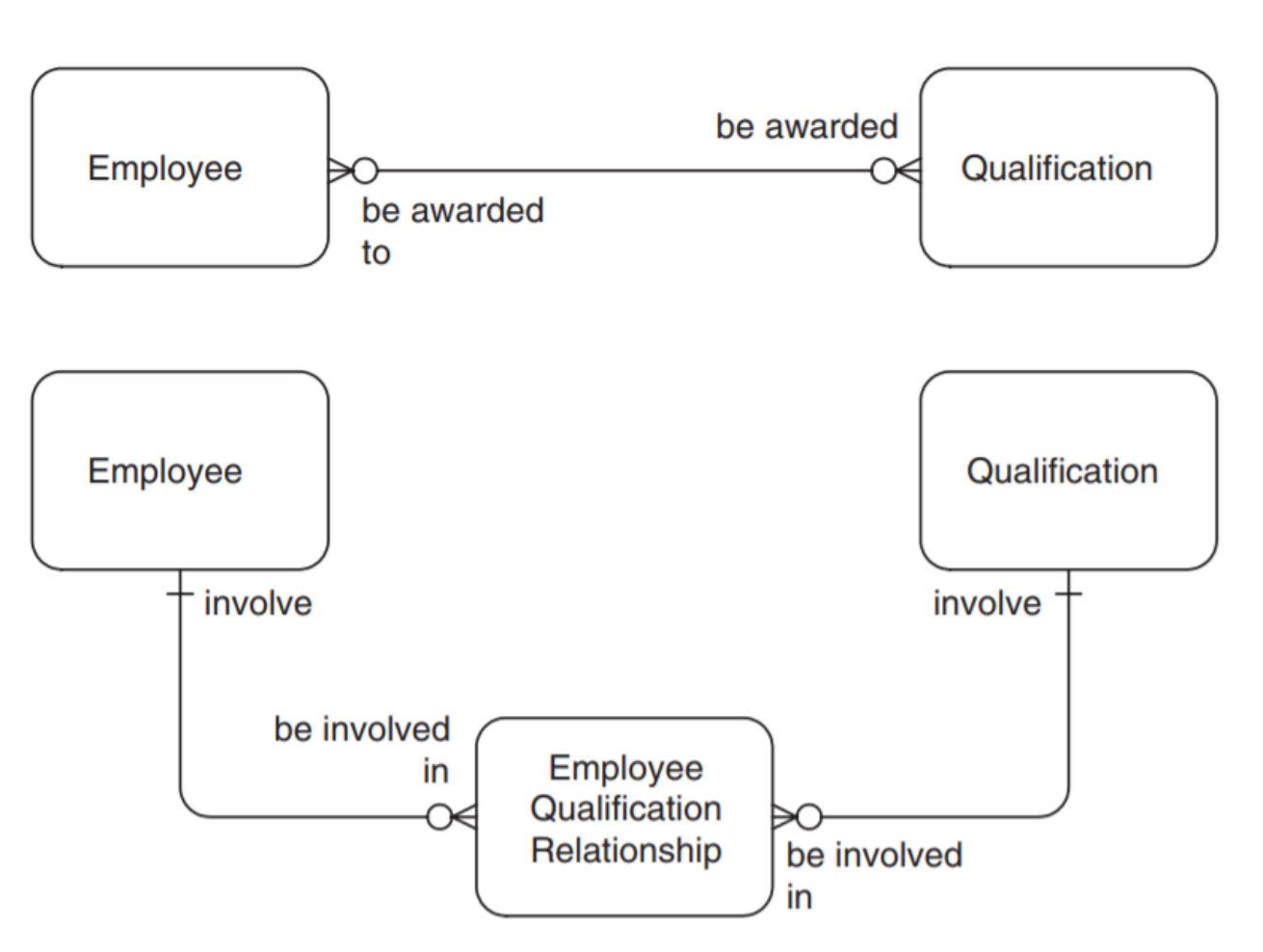
Може да се каже, че този проблем може да бъде решен с една допълнителна таблица.


Employee
Qualification
involve
involve
be involved in
Employee Qualification Relationship
be involved in

Заключение

- Края на връзката „one“ е винаги задължителна;
- А края на новата връзката „many“ ще бъде незадължителна или задължителна в зависимост от съответните краища на първоначалната връзка.

Възможности за представя на many-to-many връзка



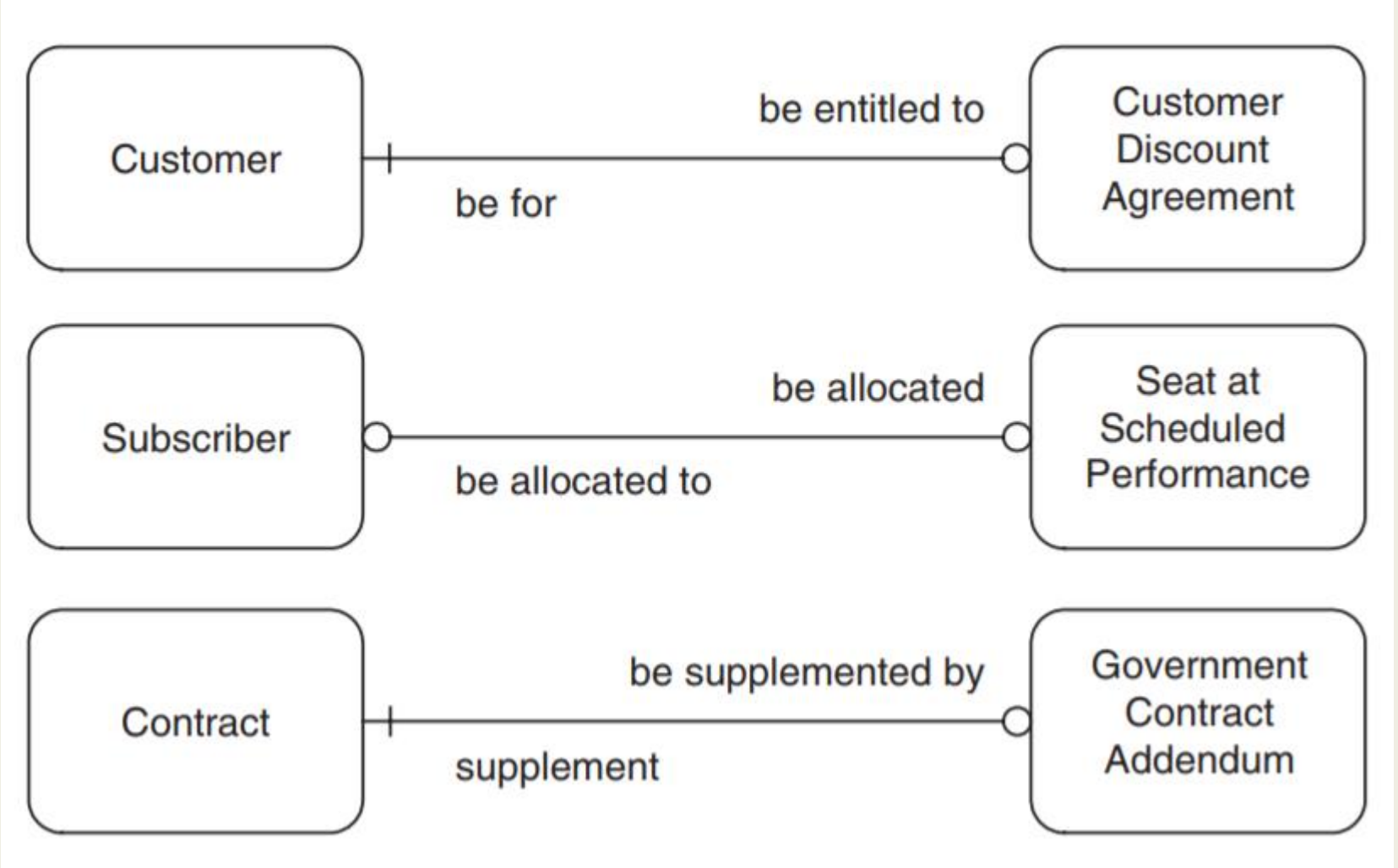
One-to-One връзки

- Връзките one-to-one са най – рядко срещаните. При определянето на подобен вид връзка винаги трябва да се питаме дали това е най – добрия вариант. Препоръчително е да не се превръща веднага връзката в обект или таблица (в логическия модел).

Проблем при създаването на подобна връзка

- Основната последица от разделянето на клас обекти в концептуалния модел по този начин, е че вмъкването и изтриването на пълни редове в базата данни на ниво логически модел, става по-сложно. На кратко това води до актуализирането на две или повече таблици вместо една.

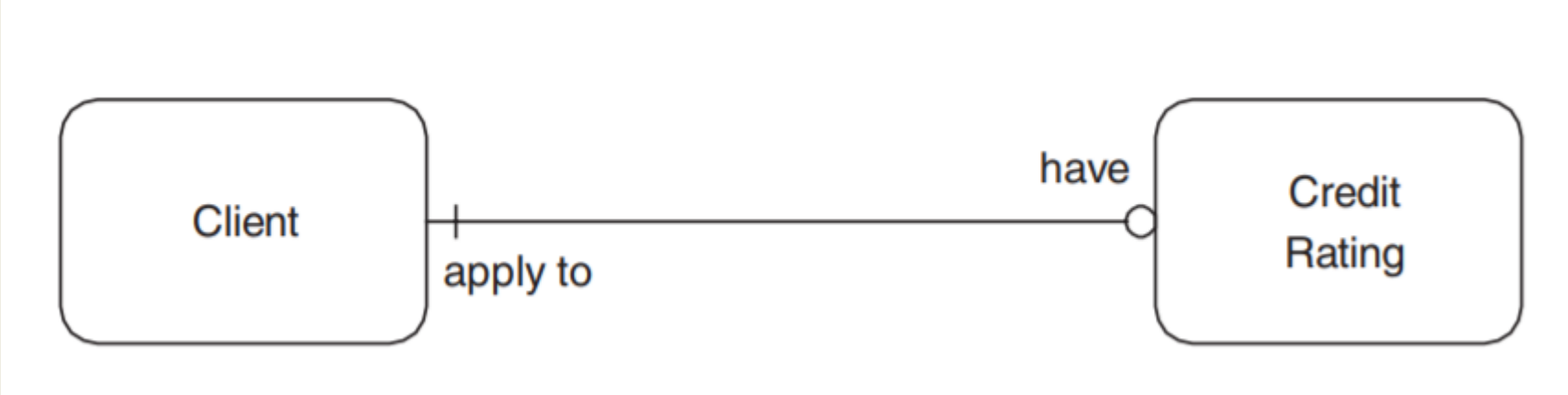
One-to-one връзки



Отделяне на групите атрибути

- Пример за това могат да бъдат Client и Credit Rating. Ако е необходимо да пазим клиент и неговия кредитен рейтинг в един клас обект, това би създало неудобства (не всеки клиент може да има кредитен рейтинг). Възможен и варианта да съхраняваме клиентите с рейтинг в един клас обект а тези без в друг, но това също би създало доста неудобства (като по-често прехвърляне на информация в таблиците на ниво логически модел).

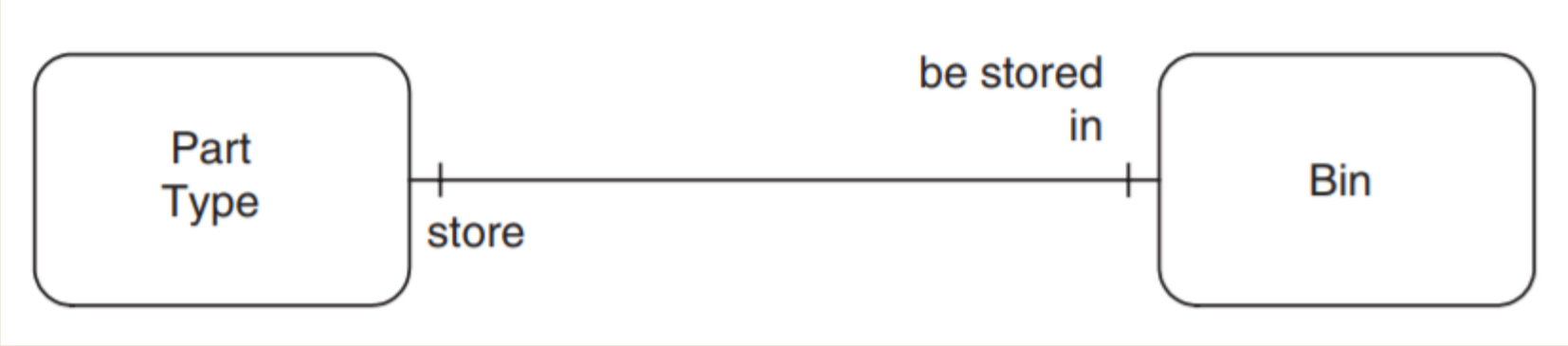
Client to Credit Rating



Преносими взаимоотношения

- Пример за това могат да бъдат Part Type и Bin. Да приемем, че двата клас обекта могат да бъдат обединени, тогава прехвърлянето на части от едно кошче в друго би изисквало ъпдейт на всички атрибути на описаната част.

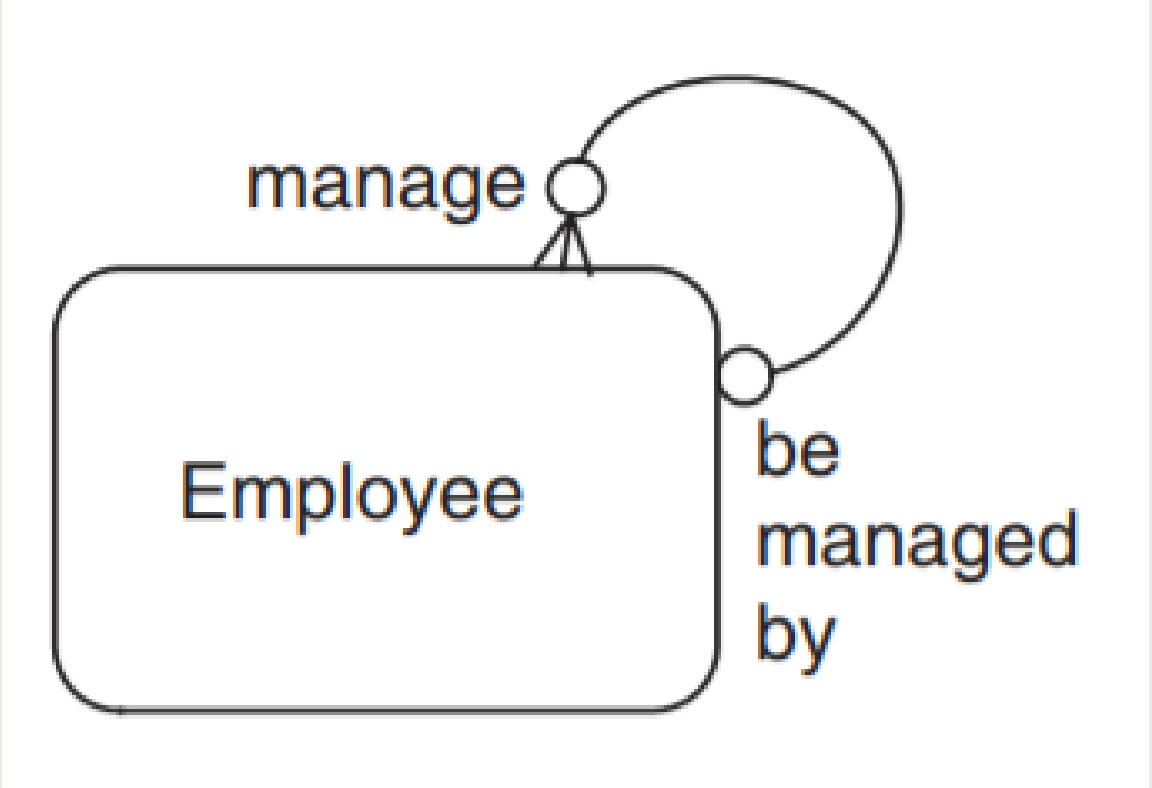
Part Type to Bin



Self-Referencing или Recursive връзки

- Self-referencing(recursive) е връзка, на която и началото и краят сочат към един и същи клас обекти.

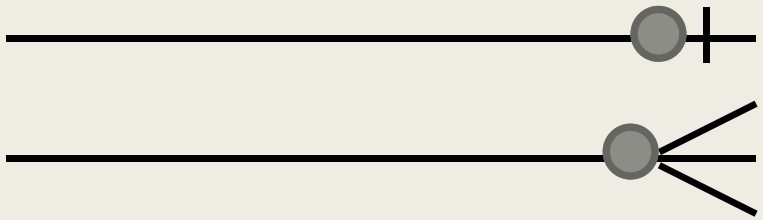
Self-referencing



Възможност за избор (Optionality)

- Определят се връзките между отделните елементи и тяхното отношение един към друг със задължително или незадължително. Отбелязването в диаграмата става с помощта на кръг.

Видове връзки

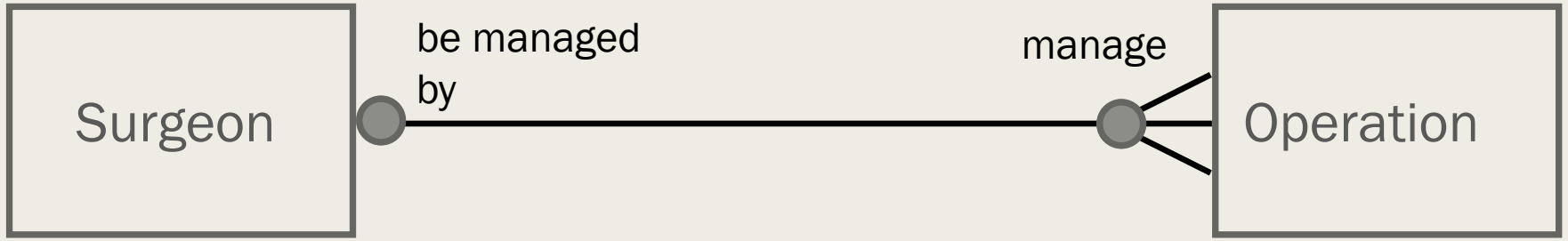


нула или един

нула или много

Всяка операция трябва да бъде проведена от хирург

Всеки хирург може да проведе операция



Всяка операция може да бъде проведена от хирург

Всеки хирург може да проведе операция

Правила: връзки до твърдения за бизнес данни

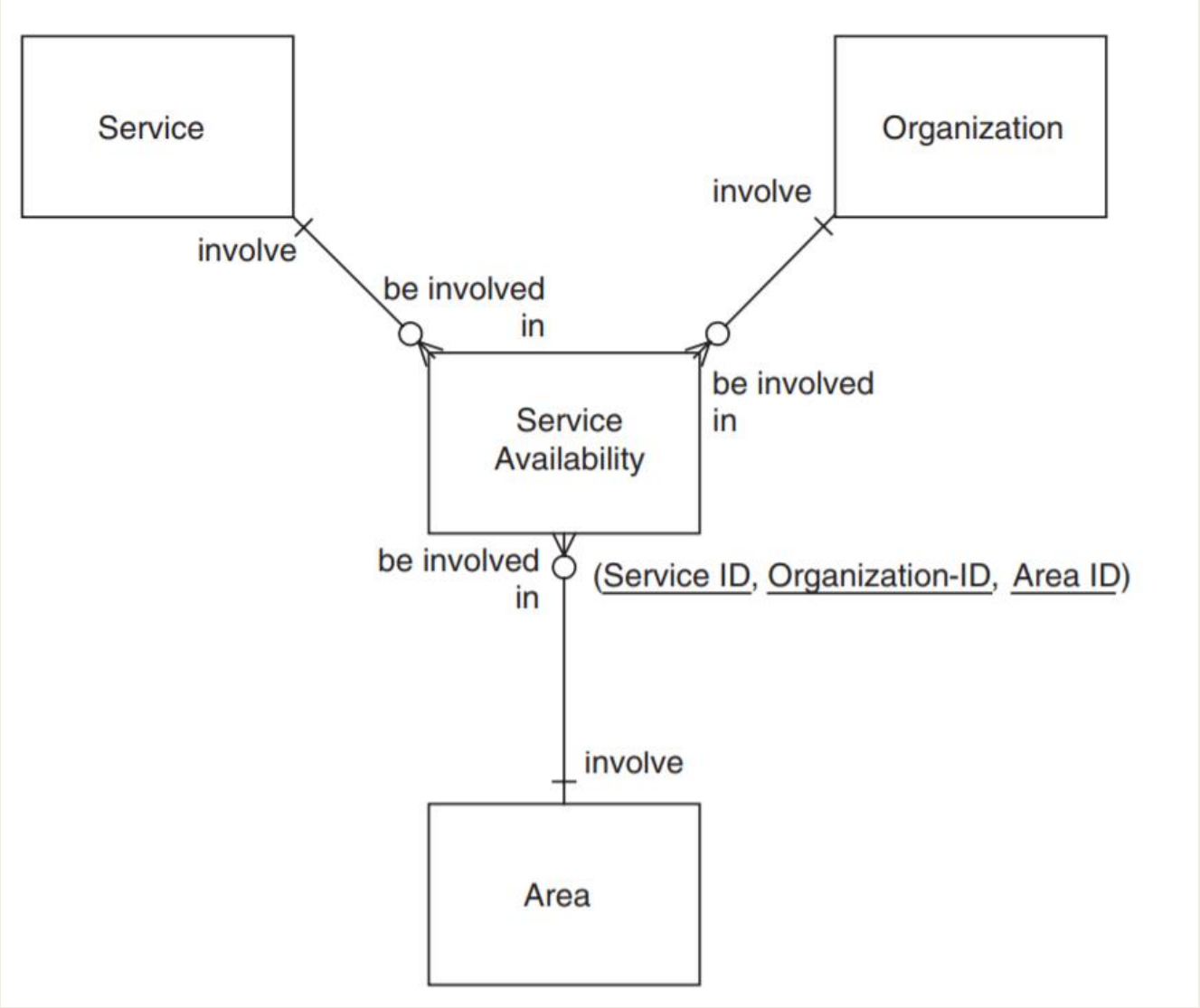
- Трябва да изберем имена на връзките, които съответстват на структурата на изреченията използвани от бизнеса. Използваните глаголи трябва да са еднакви в двете посоки (“hold” и “be held by,” или “be responsible for” и “be the responsibility of”) за да сме сигурни, че връзките ще бъдат разчетени правилно.
- Винаги трябва да се добавят имена на връзките и в двете посоки.

- Имената на класовете обекти трябва да са в единствено число и да са съобразени със средата, за която се разработва модела.

Връзки включващи три или повече класове обекти

- Ако си представим че имаме три класове обекти с две many-to-many връзки, и искаме да нормализираме концептуалния модел. Решението би било да премахнем двете връзки и да добавим клас обект, който да държи всичките връзки между трите класове обекти. Този подход може да бъде прилаган и при по - голям брой класове обекти.

Три обекта от класове

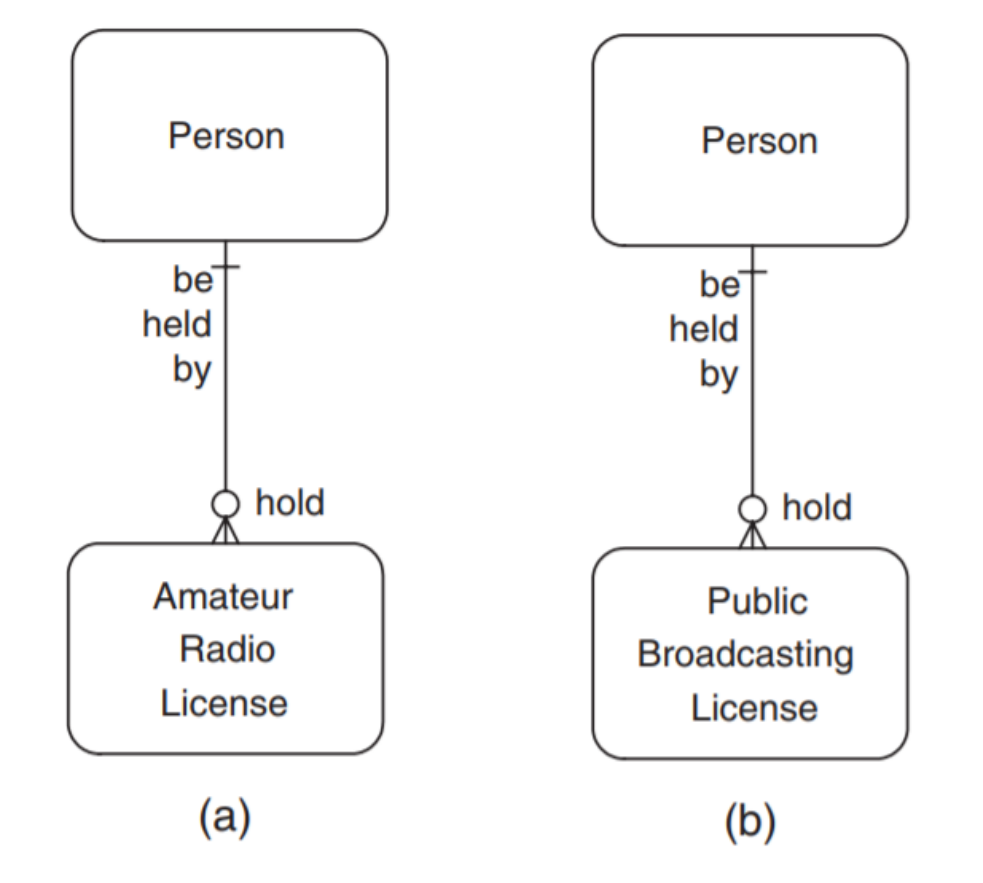


Transferability (Условия за прехвърляне)

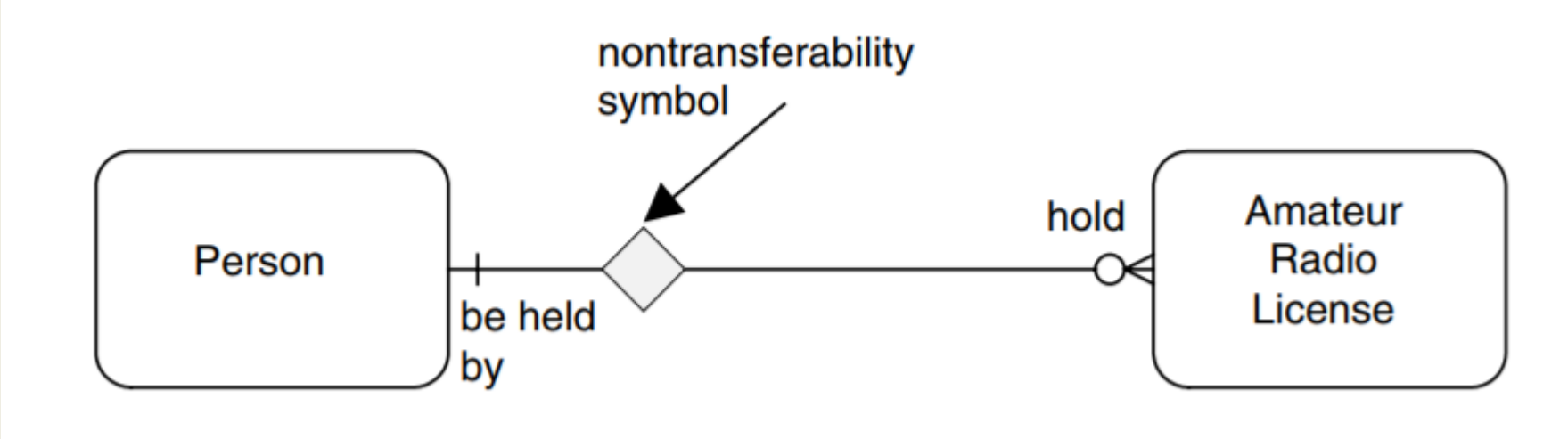
- Условието за прехвърляне дава възможност за размяна на връзката между различни обекти инстанции. По подразбиране не се отбелязва върху диаграмата. Ако имам връзка с NonTransferability отбелязваме с ромб.

Сравнение

- Двете диаграми са напълно еднакви.
- Ако погледнем внимателно ще установим, че при едната Public Broadcasting License може да бъде прехвърлен от един човек към друг (ако първия загуби лиценза си), докато при Amateur Radio License това е невъзможно.



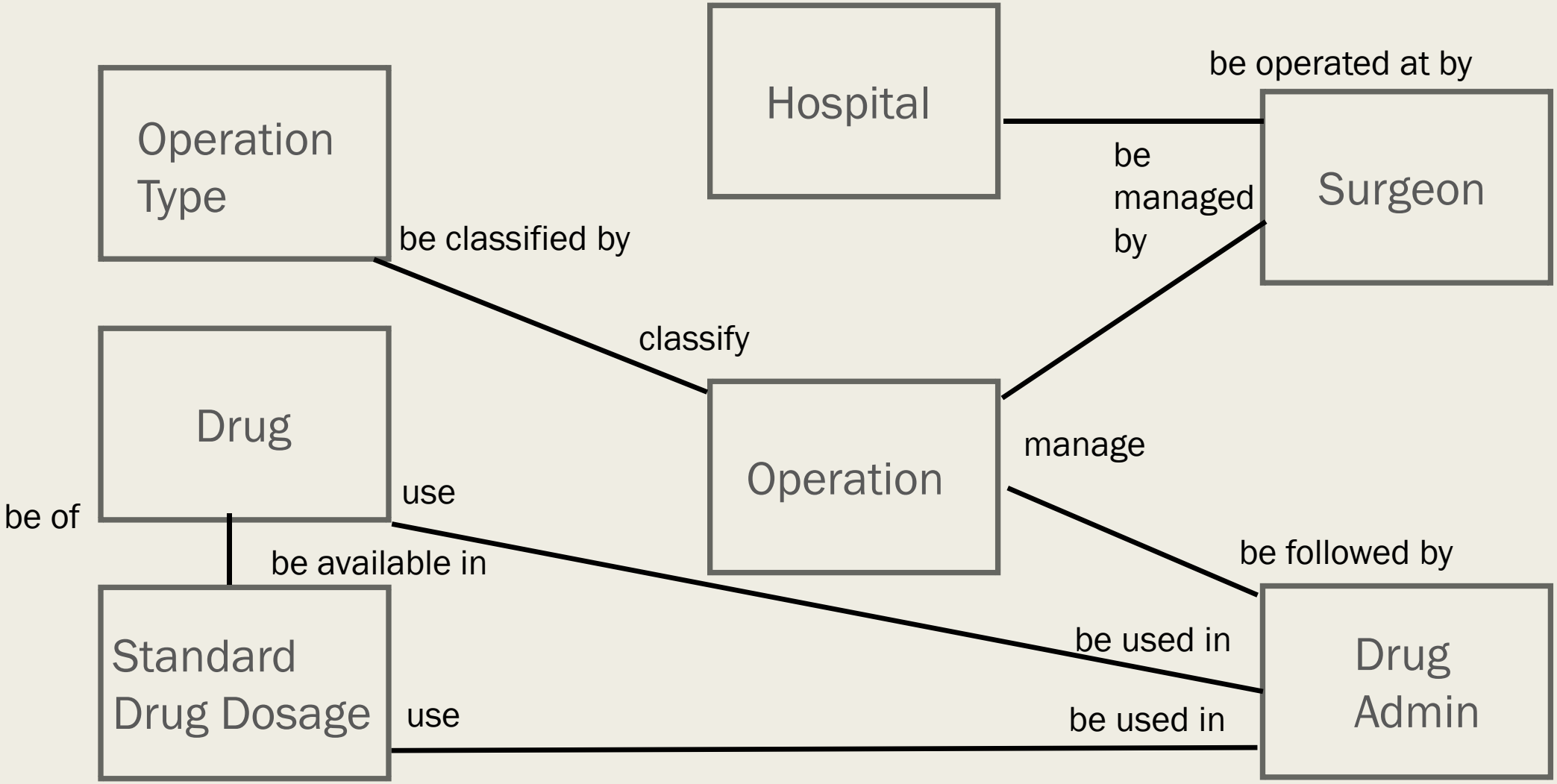
Представяне на NonTransferability



Проверка на модела данни

- Диаграмата осигурява отлична отправна точка за проверка на модела, в който участват потребители и бизнес специалисти. Интелигентната и задълбочена проверка на всяка стрелка на диаграмата често разкрива неблагоприятни предположения и недоразумения. Решението на тези проблеми увеличава доверието на заинтересованите страни в приложимостта на модела.

Модел на данни Hospital



- Ако разгледаме модела на Hospital и по специално връзката между Operation и Operation Type бихме се запитали следното:

Сигурни ли сме, че всяка операция може да бъде само от един тип? Решения:

1. Разрешени са само "прости" видове операции, като: „Отстраняване на жлъчка“ и „Отстраняване на апендикс“. Ако този курс е бил избран, моделът ще трябва да бъде преработен въз основа на информацията за типа операция, която е повтаряща се група в операцията; или

2. Позволяваме комплексни видове операции като „ Премахване на жлъчка и апендикс“. Типове операции(„Апендикс“, „ Отстраняване на жлъчка“ и „ Отстраняване на жлъчка и апендикс“)

- Ако имплементацията на базата данни и потребителския интерфейс е направена, бихме били поставени пред варианта да изберем Решение 2, освен ако не сме подготвени да направим много промени по интерфейса. Вариант 1 изглежда много по елегантно и би опростило много последвали справки, като „Покажи всички операции, които са от тип апендикс“.

ВЪПРОСИ ?