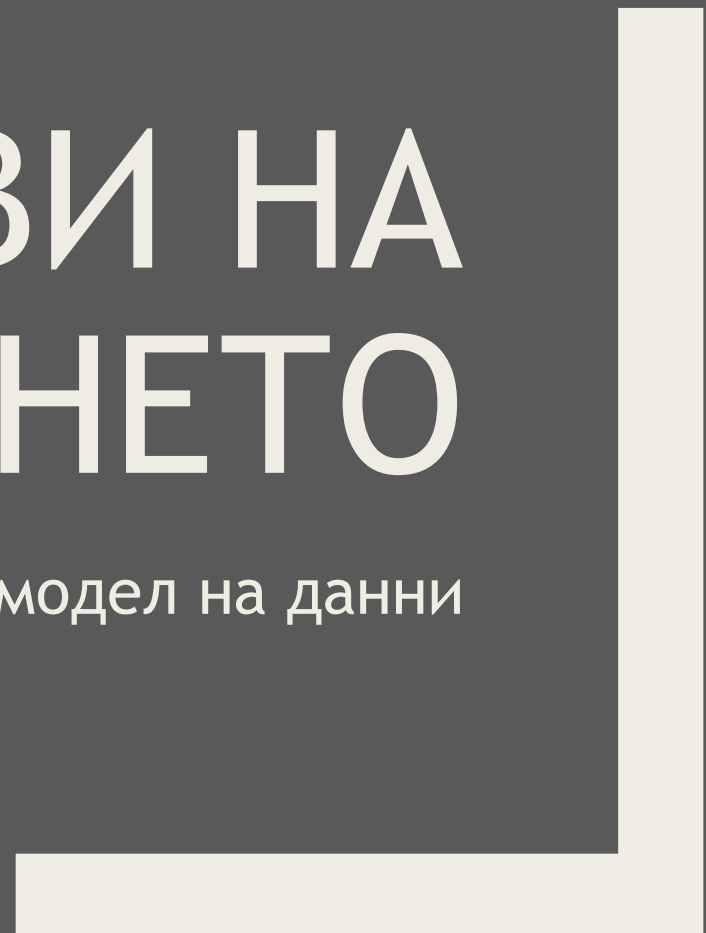


МОДЕЛИРАНЕ И АНАЛИЗ НА СОФТУЕР

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

ОСНОВИ НА МОДЕЛИРАНЕТО

Концептуален модел на данни



The Entity-Relationship Approach

Диаграмно представяне

- Диаграмното представяне е процеса по описание на изискванията(правилата) от бизнеса посредством графични елементи. Процеса по представянето на диаграмата е част концептуалния модел.

Основни елементи: Box и Arrow

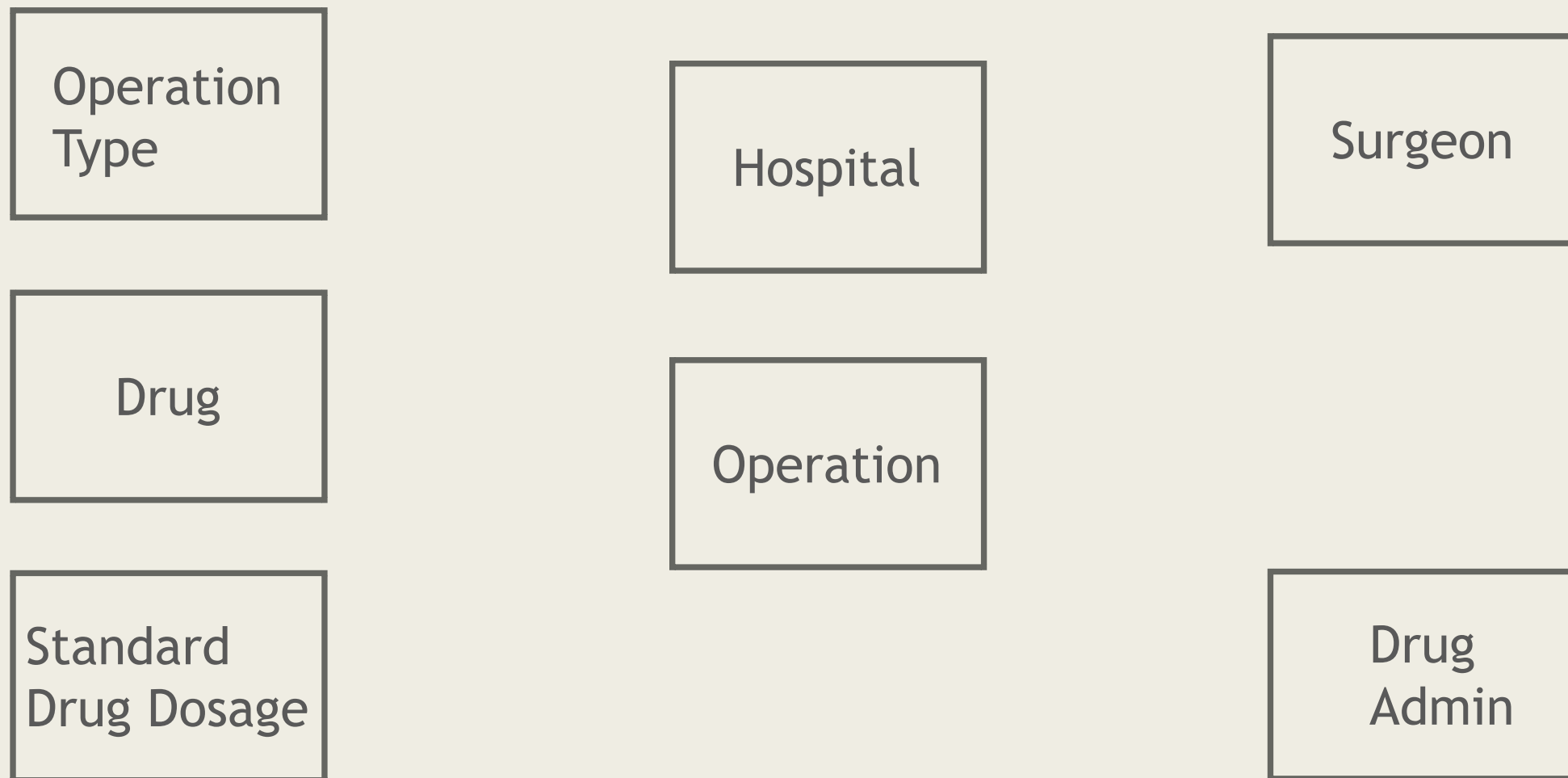
- “Box” (строго казано, правоъгълник) представлява таблицата в логическия модел.
- “Arrow” поставена между два “Box”-са, представяща връзките между тях.

Описание на модел Hospital, чрез таблицы

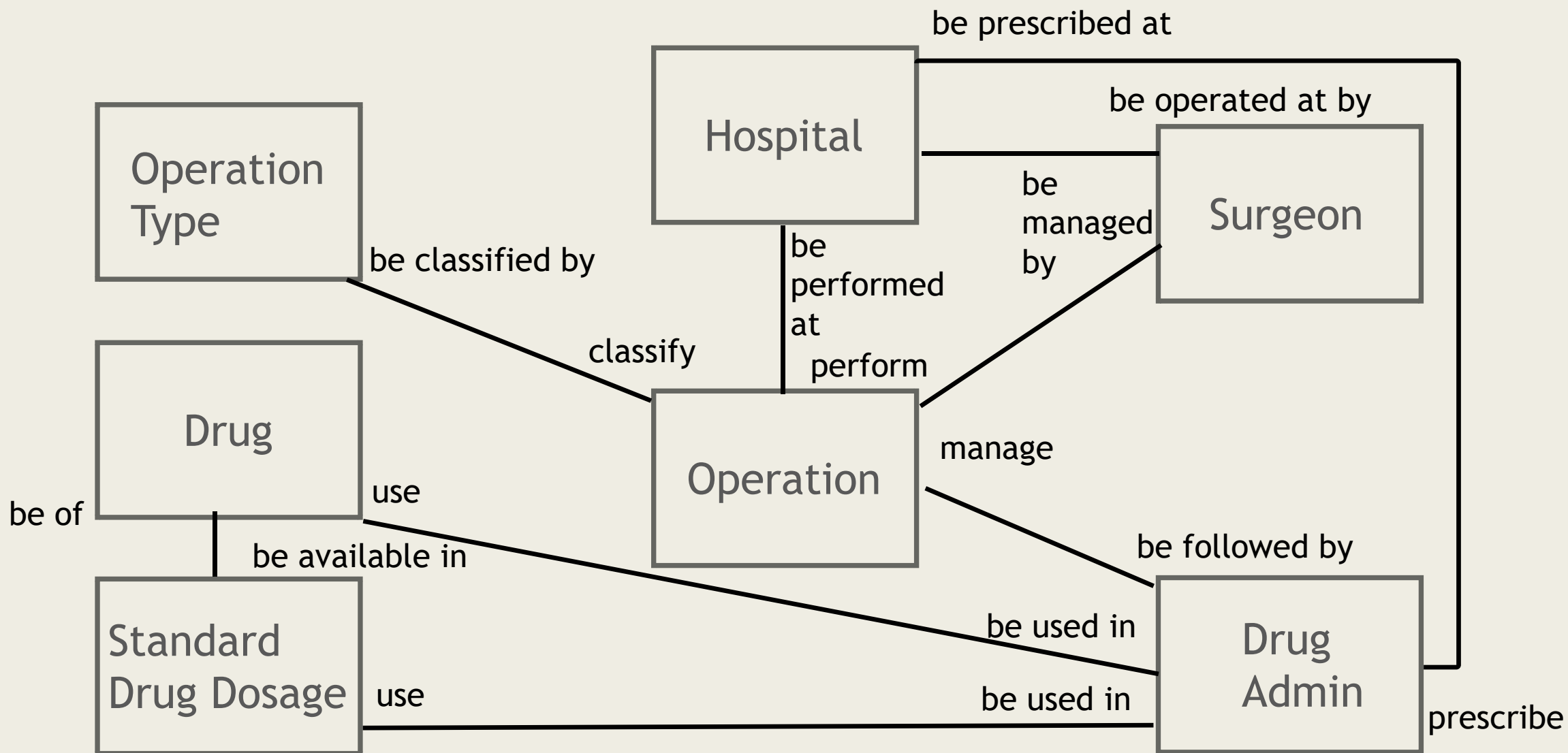
- **OPERATION** (Hospital Number*, Operation Number, Operation Code*, Surgeon Number*)
- **SURGEON** (Hospital Number*, Surgeon Number, Surgeon Specialty)
- **OPERATION TYPE** (Operation Code, Operation Name, Procedure Group)
- **HOSPITAL** (Hospital Number, Hospital Name, Hospital Category, Contact Person)

- **DRUG** (Drug Short Name, Drug Name, Manufacturer)
- **STANDARD DRUG DOSAGE** (Drug Short Name*, Size of Dose, Unit of Measure, Method of Administration, Standard Dose Cost)
- **DRUG ADMINISTRATION** (Hospital Number*, Operation Number*, Drug Short Name*, Size of Dose*, Unit of Measure*, Method of Administration*, Number of Doses)

Представяне на модел Hospital



Модел на данни Hospital + връзки



The Top-Down Approach: Entity-Relationship Modeling

Top-Down Approach

- В метод Top-Down разработката на концептуалния модел започва, чрез запознаване с проблема, задаване на въпроси свързани със средата и условията и чак след това стартира работата по модела на данни. По този начин може да бъде избегнато допълнителното нормализиране.

Entity-Relationship model(E-R model)

- Процеса по създаването на правилни класове от класове обекти, връзки и атрибути, които да удовлетворяват бизнес проблем се нарича entity-relationship modeling(E-R modeling за кратко), и по общо казано концептуален модел.

Бизнес ориентирана терминология

- Entity classes - са категории от бизнес логиката; представени с кутии(box) по диаграмата; обикновено се имплементират като таблици.
- Attributes - Представят знанието за класа обект; обикновено не са показани на диаграмата и се представят като колони в таблиците.
- Relationship - са линии между два клас обекта, представляващи чужди ключове.

Chen`s Database Notation - концептуален модел

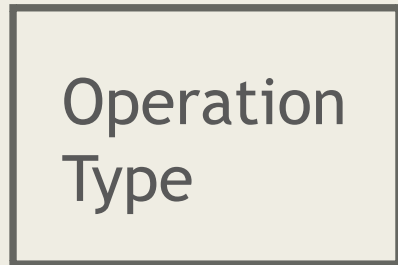
Класове обекти (Entity Classes)

- Клас обект в реалния свят представляват класове от обекти (Пример: болница).
- Трябва да се прави разлика между обекти като „УМБАЛ СВ. Георги“, и класове обекти като „болница“. В практиката E-R modelers използват думата обект за клас обект и инстанция на обект в случаите, когато искат да реферират само единствена инстанция.

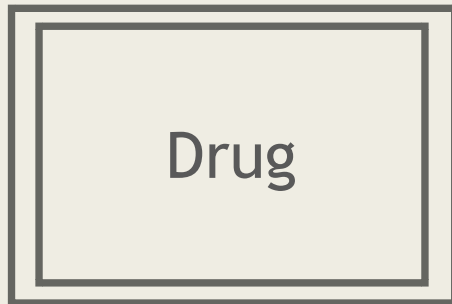
Типове класове обекти - концептуален модел

- Силени класове обекти - тези класове обекти за независими от останалите класове обекти, доста често са наричани родители. Доста често са във връзка със слаби класове обекти.
- Слаби класове обекти - тези класове обекти зависят от други класове, които допълват техния смисъл. Те нямат идентификатор.
- Асоциирани класове обекти - тези класове обекти се отнасят към няколко други класове обекти. Те съдържат и специфични атрибути носещи информация за връзките им.

Представяне на класове обекти



Силен клас обект



Слаб клас обект



Асоцииран клас обект

Клас обект именуване

- Името на класа обект трябва да е в единствено число и да се отнася до една единствена инстанция (в релационни отношения, ред) - не към цялата информация. По този начин колективните термини като файл, таблица, каталог, история и график са неподходящи.

Примери

Account вместо **Accounts**

Customer вместо **Customer File** и **Customer Table**,
или **Customer Record**

Product вместо **Product Catalog**

Historical Transaction вместо **Transaction History**

Причини за прилагане на правилата за именуване

- Consistency(съгласуваност) - това е основата на стандарта за именуване на класовете обекти.
- Communication(свързване) - класа обект е „нещо, за което искаме да пазим информация“, като например клиент а не клиентки файл.
- Generating business assertions(създаване на бизнес твърдения) - ако следваме някои прости правила за именуване на компонентите на E-R model, можем автоматично да генерираме граматически стабилни твърдения.

Причини за прилагане на правилата за именуване

- Consistency(съгласуваност) - това е основата на стандарта за именуване на класовете обекти.
- Communication(свързване) - класа обект е „нещо, за което искаме да пазим информация“, като например клиент а не клиентки файл.
- Generating business assertions(създаване на бизнес твърдения) - ако следваме някои прости правила за именуване на компонентите на E-R model, можем автоматично да генерираме граматически стабилни твърдения.

Връзки (Relationships)

- Представят знанието за класа обект. В модела на Hospital, линиите между различните кутии(box) могат да бъдат интерпретирани в реалния свят като връзки между класове обекти. Като пример имаме връзки между hospitals и surgeons, и между operations и drug administrations.

Представяне на връзки



Връзка

Оптимизация

- Модела трябва да бъде винаги подържан и да се избягва излишно дублиране на връзки между отделните елементи.

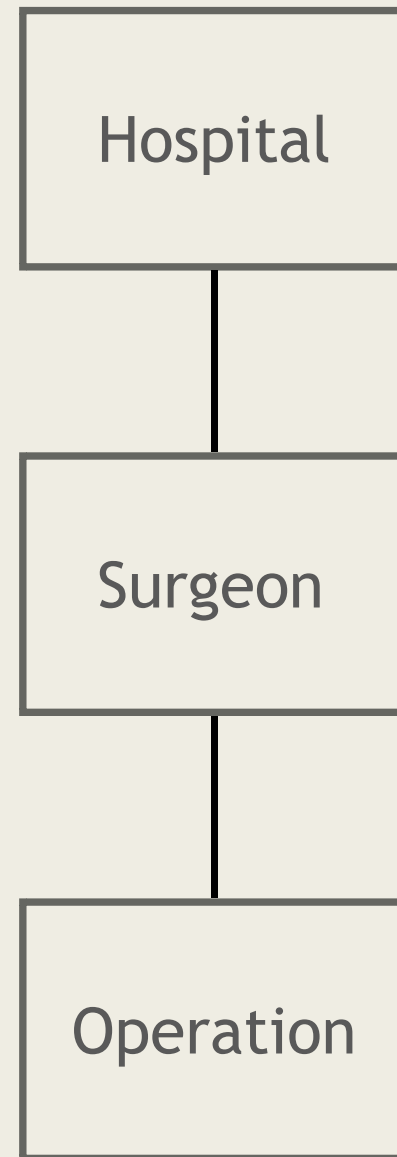
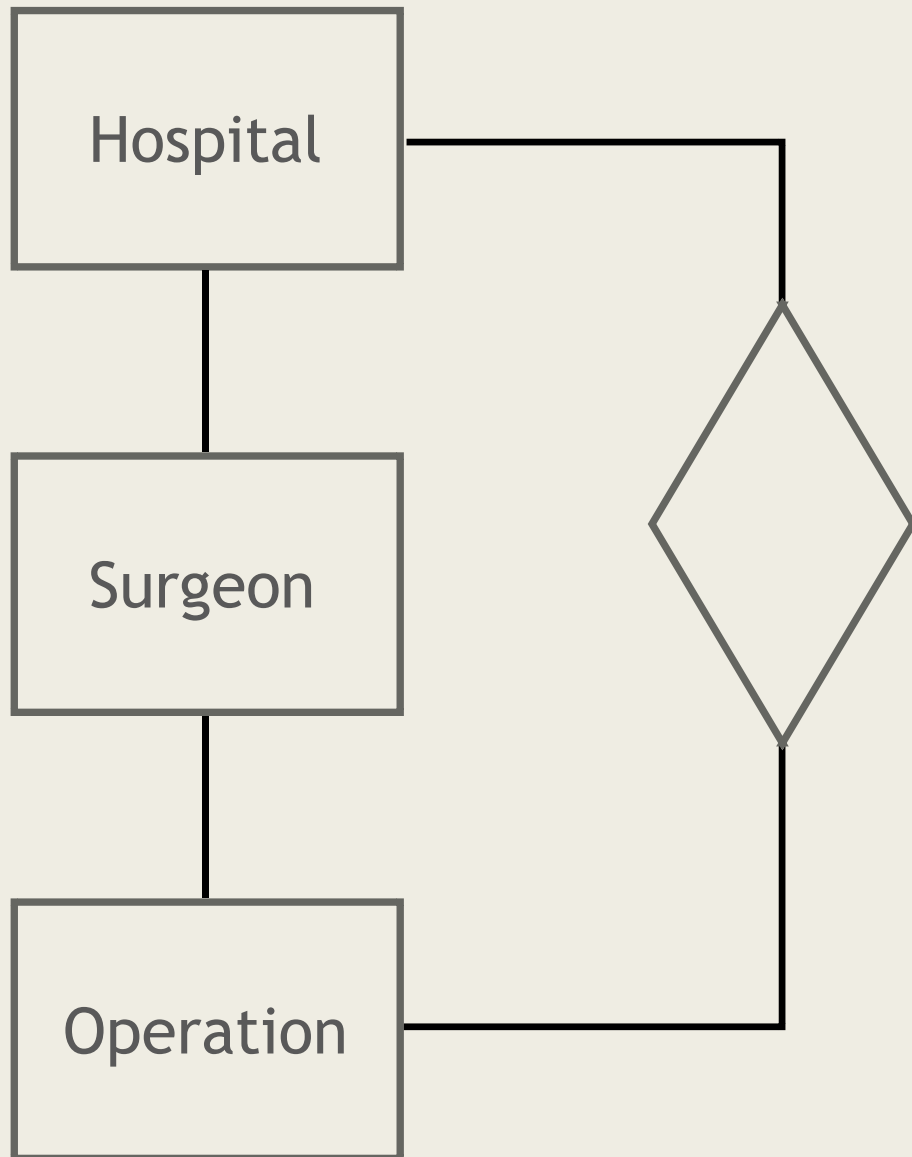
- Ако разгледаме таблиците Hospital, Operation и Surgeon. Имаме стрелки между Hospital, Surgeon и от Surgeon към Operation.

Въпроса е тази трета стрелка, давали някакво допълнително познание към бизнес модела?

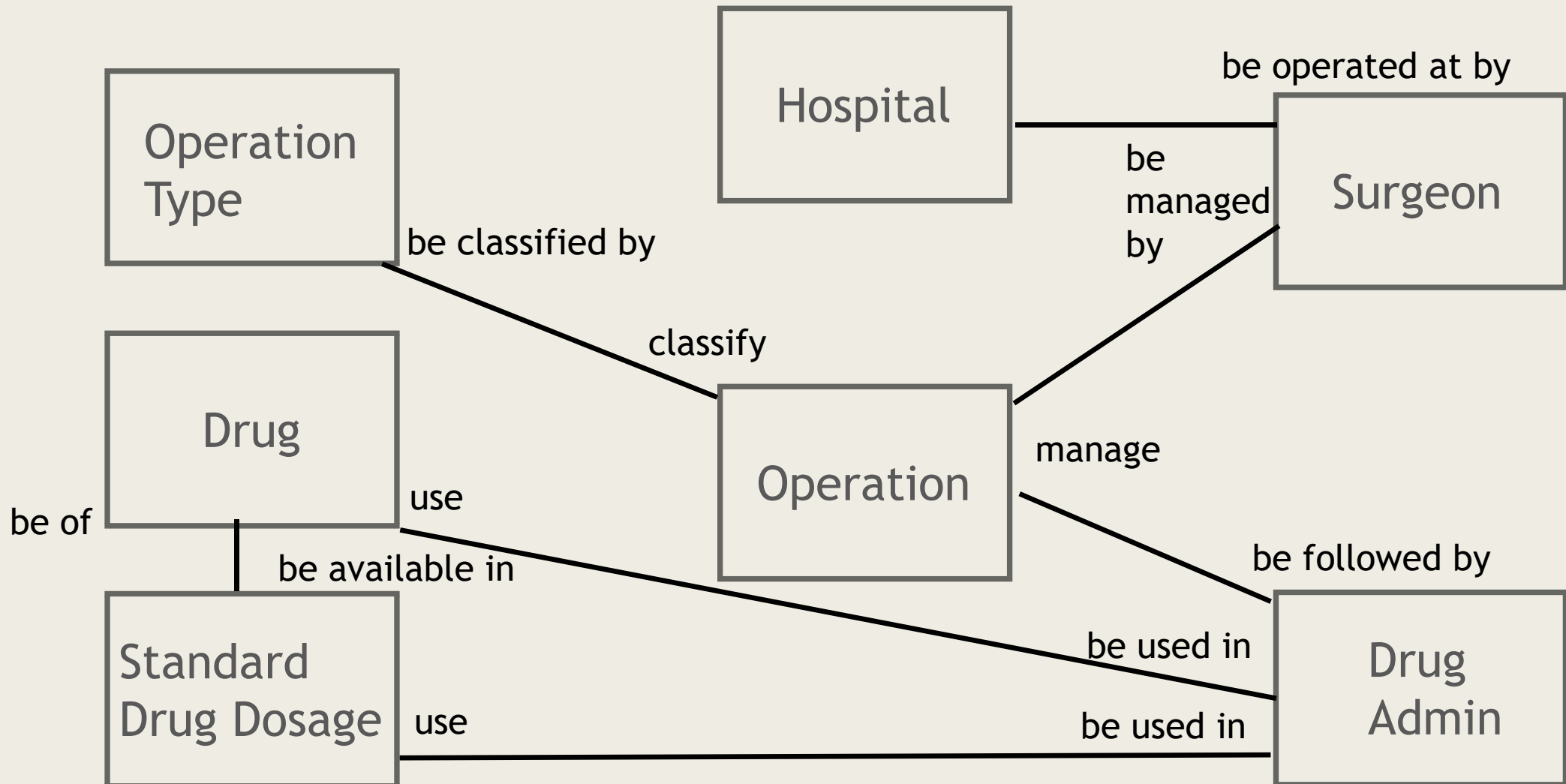
Отговора е не. Връзката може да бъде остановена при анализ на модела(всяка операция трябва да бъде направена в една болница).

Анализ: Всяка операция трябва да бъде направена от хирург а всеки хирург работи в болница.

Оптимизация



Модел на данни - ОПТИМИЗАЦИЯ



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.

- Ако погледнем модела още веднъж 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“.

Атрибути (Attributes)

- Атрибутите в концептуалния модел съответстват на колоните в физическия модел. Показваме само няколко атрибута на диаграмата за изясняване на смисъла на определен клас обект. Никога не се описват всички атрибути. Атрибутите представляват отговор на въпроса „Какви данни искаме да запазим за този клас обект?“.

Типове атрибути - концептуален модел

- Обикновени атрибути (Simple attributes)
- Комбинираните атрибути (Composite attributes)
- Многостойностни атрибути (Multivalued attributes)
- Производни атрибути (Derived attributes)

Обикновени атрибути (Simple attributes)

- Обикновените атрибути са стойностите, извлечени от домейн областта на анализ; също са наричани и атрибути с единична стойност.

Пример

- Разглеждана област компания (COMPANY):

Name = {John} ; Age = {23}

Комбинираните атрибути (Composite attributes)

- Комбинираните атрибути са тези, които се състоят от йерархия на атрибути.

Пример

- Представянето на адрес (Address), където имаме Number, Street и Suburb:

Address = {59 + 'Meek Street' + 'Kingsford'}

Многостойностни атрибути (Multivalued attributes)

- Атрибутите с много стойности са атрибути, които имат набор от стойности за всяки клас обект.

Пример

- Представянето на научна степен на работник
(Degrees of an employee) :

BSc, MIT, PhD

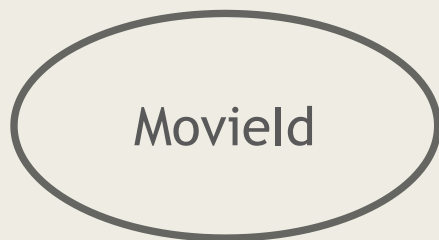
Производни атрибути (Derived attributes)

- Атрибут, чиято стойност се изчислява (получена) от други атрибути. Полученият атрибут може или не може да бъде физически съхраняван в базата данни.

Пример

- Представянето на дата на раждане(Birthdate) :
Date, Month, Year

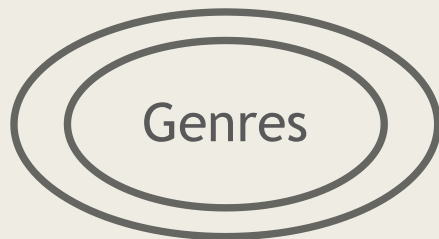
Представяне на атрибути - концептуален модел



Атрибут и комбиниран

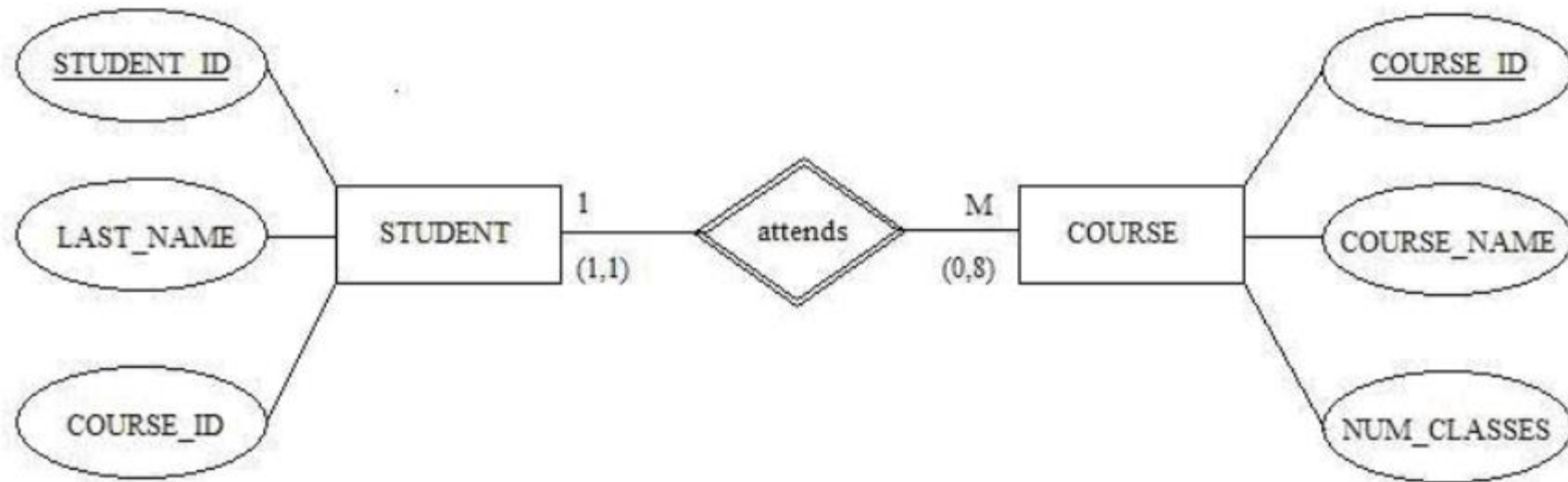


Производен атрибут



Многостойностен атрибут

Диаграма студент курс



ВЪПРОСИ ?

