

# МОДЕЛИРАНЕ И АНАЛИЗ НА СОФТУЕР

Павел Кюркчиев

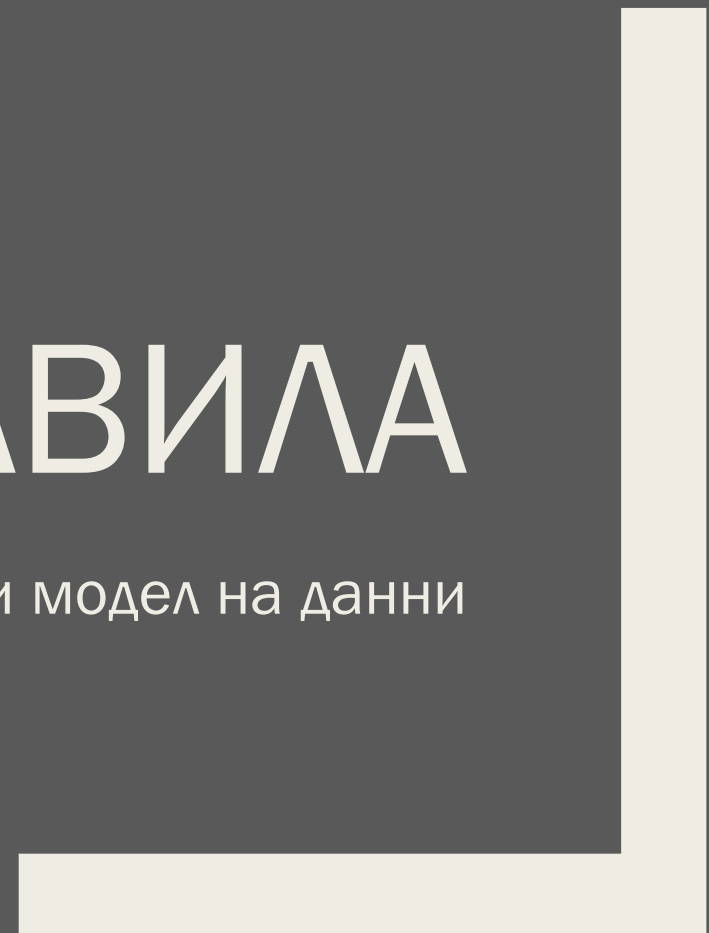
Ас. към ПУ „Паисий Хилендарски“

<https://github.com/pkyurkchiev>

@pkyurkchiev

# БИЗНЕС ПРАВИЛА

Физически модел на данни

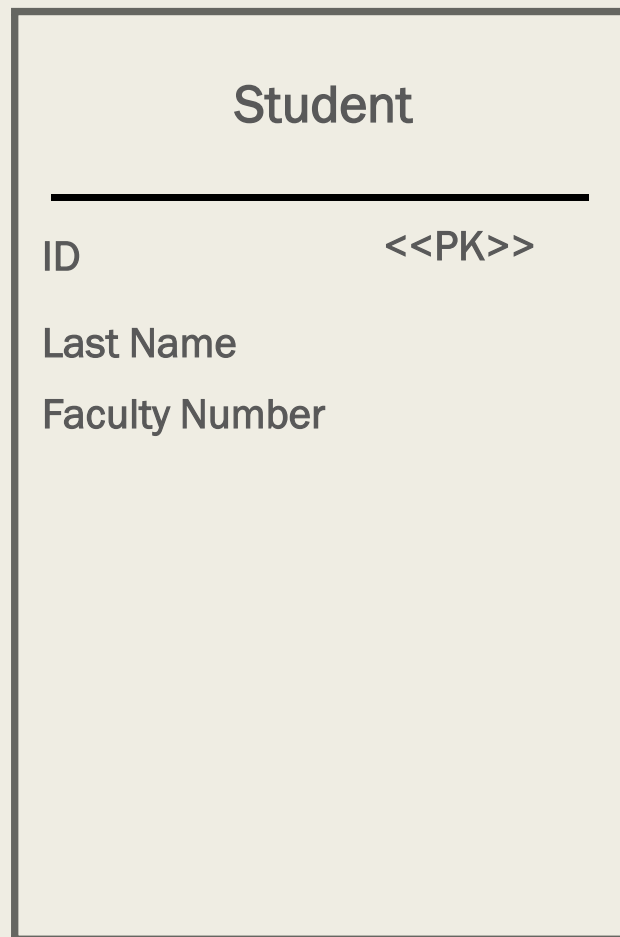


# UML Database Notation – физически модел на данни

# Първичен ключ (Primary key)

- Първичен ключ е поле, което определя еднозначно запис в база от данни.
- Този тип ключове са използвани от OLTP схеми.

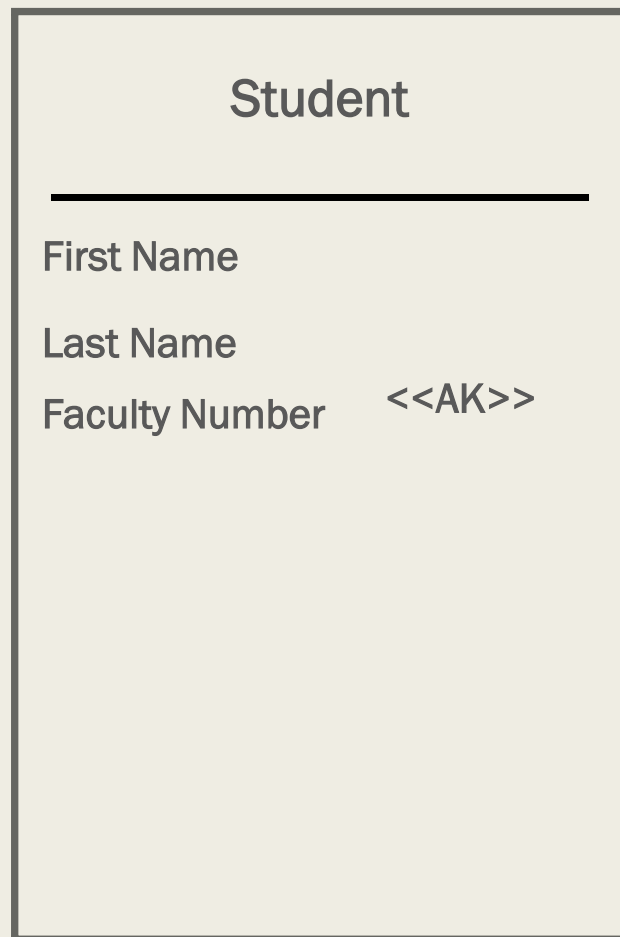
# Представяне на първичен ключ на атрибути



# Алтернативен ключ атрибути (Attributes)

- Показва, че една колона е част от алтернативен ключ, известен също като вторичен ключ, за таблица.

# Представяне на алтернативни ключови атрибути

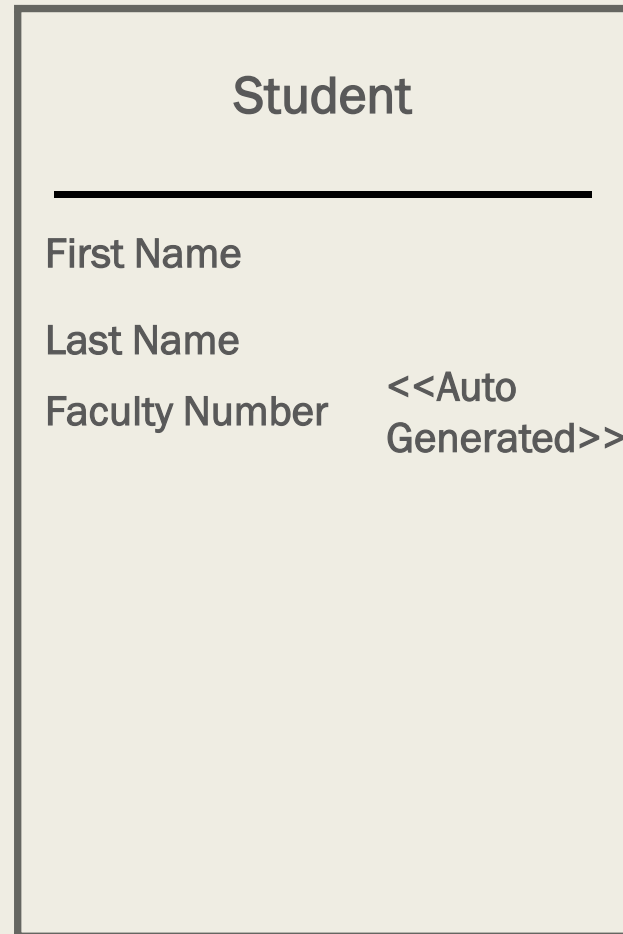


# Автоматично генерирани атрибути (Auto Generated Attributes)

- Показва, че стойността на колоната се генерира автоматично от базата данни.



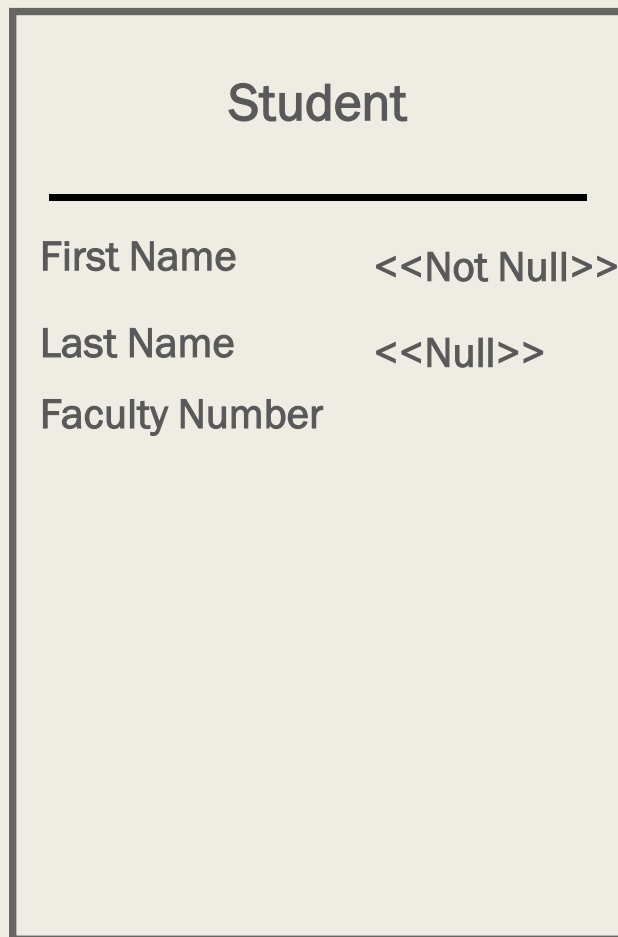
# Представяне на автоматично генерирани атрибути



# Задължителни/Незадължителни атрибути

- Задължителни - Показва, че в колоната не може да има нулеви стойности.
- Незадължителни - Показва, че колоната може да има нулеви стойности.

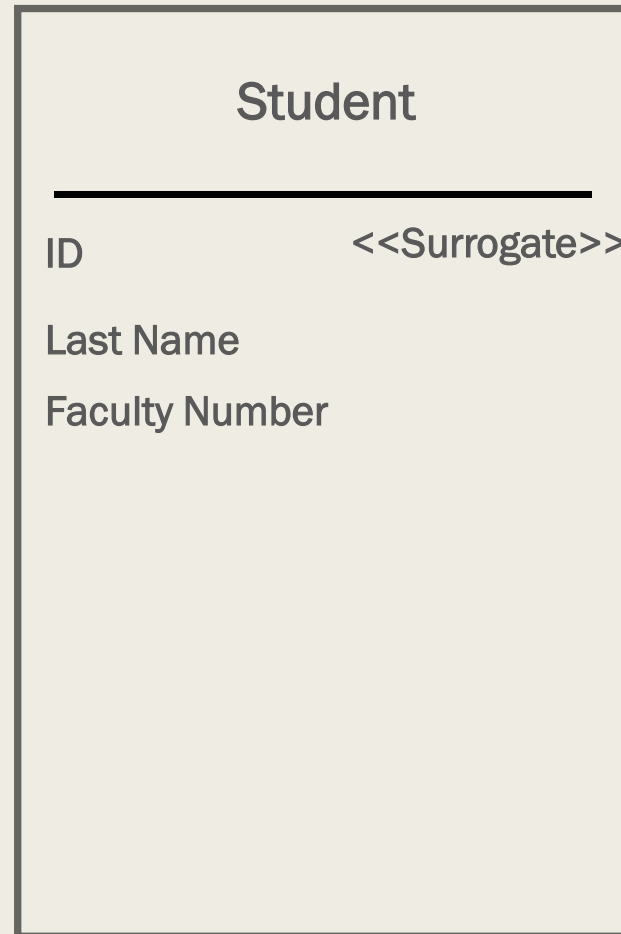
# Представяне на задължителни/незадължителни атрибути



# Ключ заместител (Surrogate key)

- Ключ заместител е поле, което определя еднозначно запис в база от данни.
- Този тип ключове са използвани от OLAP схеми.
- Този тип ключове са прости числа. Не представляват сложни изрази докато първичните ключове могат да бъдат сложни изрази.

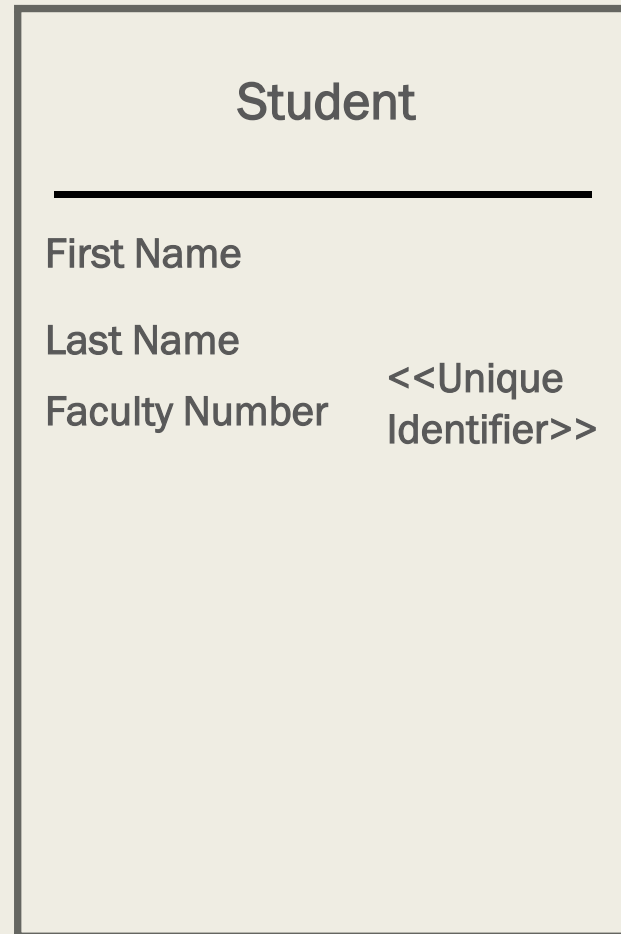
# Представяне на ключ заместител на атрибути



# Уникално ограничение на атрибути (Unique Constraint)

- Уникалните ограничения спомагат за утвърждаването на уникалността на конкретни колони. Уникални ограничения могат да бъдат добавени към даден обект, за да се осигури невъзможност за въвеждане на дублирани стойности в конкретни колони. Уникалното ограничение може да се състои от една колона или комбинация от колони.

# Представяне на уникално ограничение на атрибути

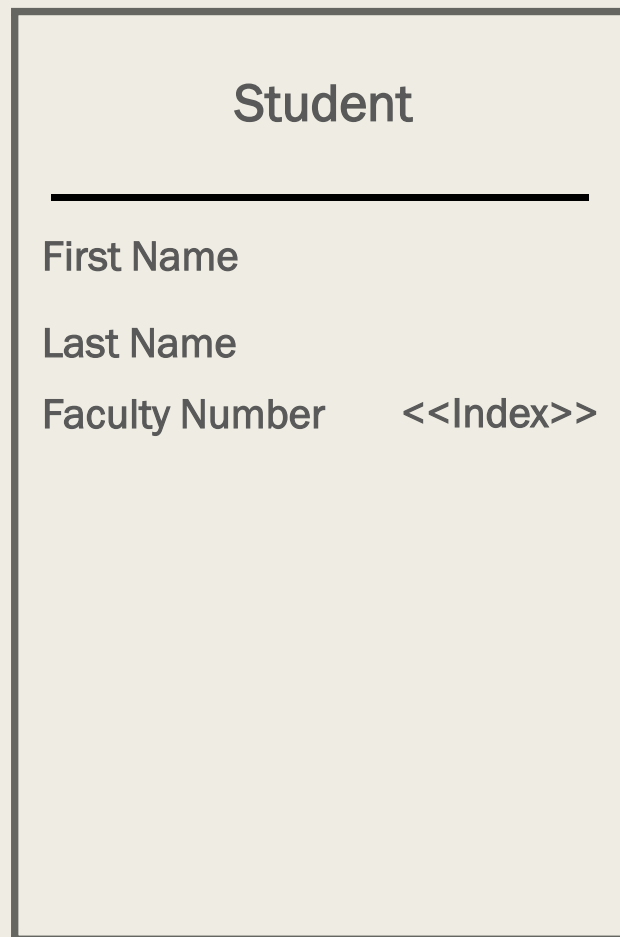


# Индексирани Атрибути (Index Attributes)

- Индексът на базата данни е структура от данни, която подобрява скоростта на операциите за извличане на данни в таблица на база данни, за сметка на допълнителното пространство за запис и съхранение, за да се запази структурата на индексните данни.



# Представяне на индексирани атрибути



# Изчислими колони (Calculated column)

- Това са колони, които се изчисляват или обработват от бизнес логика, която е заложена в базата данни.

# Представяне на изчислими КОЛОНИ

## Student

---

First Name

Last Name

/ Faculty Number

# Съхранени процедури (Stored Procedures)

- Съхранена процедура е предварително написан код на процедура, който позволява да бъде изпълняван отново и отново за валидиране или бързо извличане на данни. Използването на съхранена процедура помага да се поддържа последователно внедряване на логиката в програмните модули и приложения. Той също така прави дизайна, кодирането и тестването по-лесни, защото логиката се поставя на едно място - съхранената процедура.

# Описание на съхранени процедури

- Съхранените процедури трябва да бъдат моделирани като част от един клас. Този клас е отбелязан със стереотип <<Запазени процедури>>.

# Представяне на съхранена процедура

<<Stored Procedure>>  
StudentDB

---

getAllStudents(...)  
getStudentTotal(...)

# Тригери (Triggers)

- Тригерът в база данни е процедура, която автоматично се изпълнява в отговор на определени събития възникнали в таблица на базата данни. Обичайната употреба на тригери е за одитиране на база данни. Тригерът, който регистрира вмъкването, модифицирането и изтриването на важни данни, ще ви информира кога и защо е направена промяна в базата данни.

## Описание на тригер

- При описанието на тригера освен задаване на име на тригера трябва да бъде моделирано и събитието, което задейства метода. Например {събитие = преди вмъкване | след актуализация, цел на следене = име на колона}



# Представяне на тригер

Student

---

---

InsertStudent(...)  
<<Trigger>>  
{event = after insert}

UpdateStudent(...)  
<<Trigger>>  
{event = after update}

# Изгледи (Views)

- В база данни изгледът представлява набор от резултати от запаметена заявка върху данните, които потребителите на базата данни могат да заявят, както биха направили заявка за извличане на информация от таблица.

# Представяне на изгледи

<<View>>

StudentTaxes

---

Tax

Tax Unit

Tax Total

# Асоциативни таблици (Associative Table)

- Асоциативните таблици са не основни таблици, на които първичния ключ е съвкупност от чужди ключове.

# Представяне на Асоциативна таблица

<<Associative Table>>  
StudentToMajor

---

StudentID

MajorID

# Таблицы списъци (Lookup Table)

- Този вид таблици съдържа в себе си номенклатури.

# Представяне на таблица списък

<<Lookup Table>>  
LevelOfEducation

---

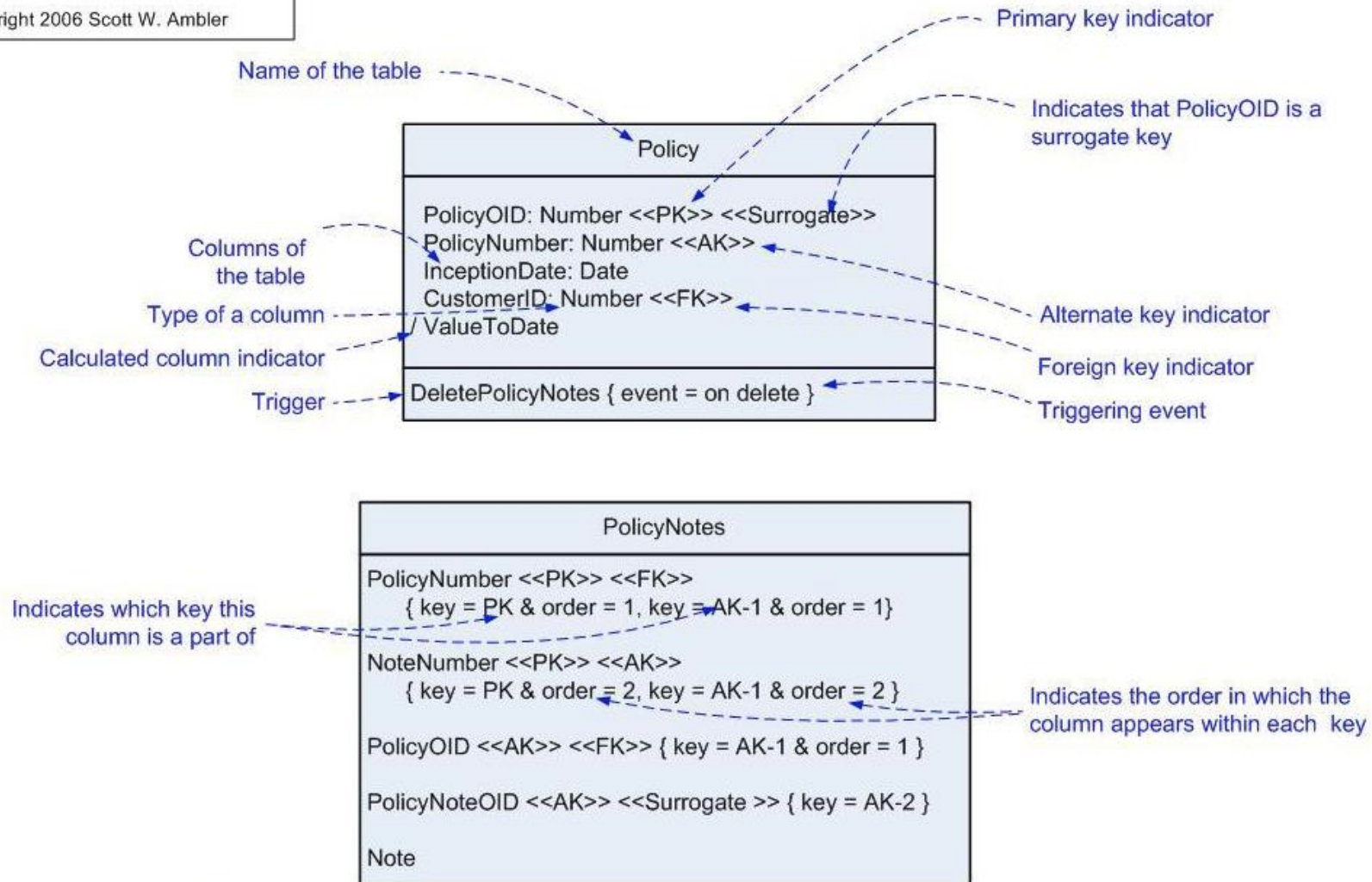
LevelOfEducationID

Value

# Диаграма 1

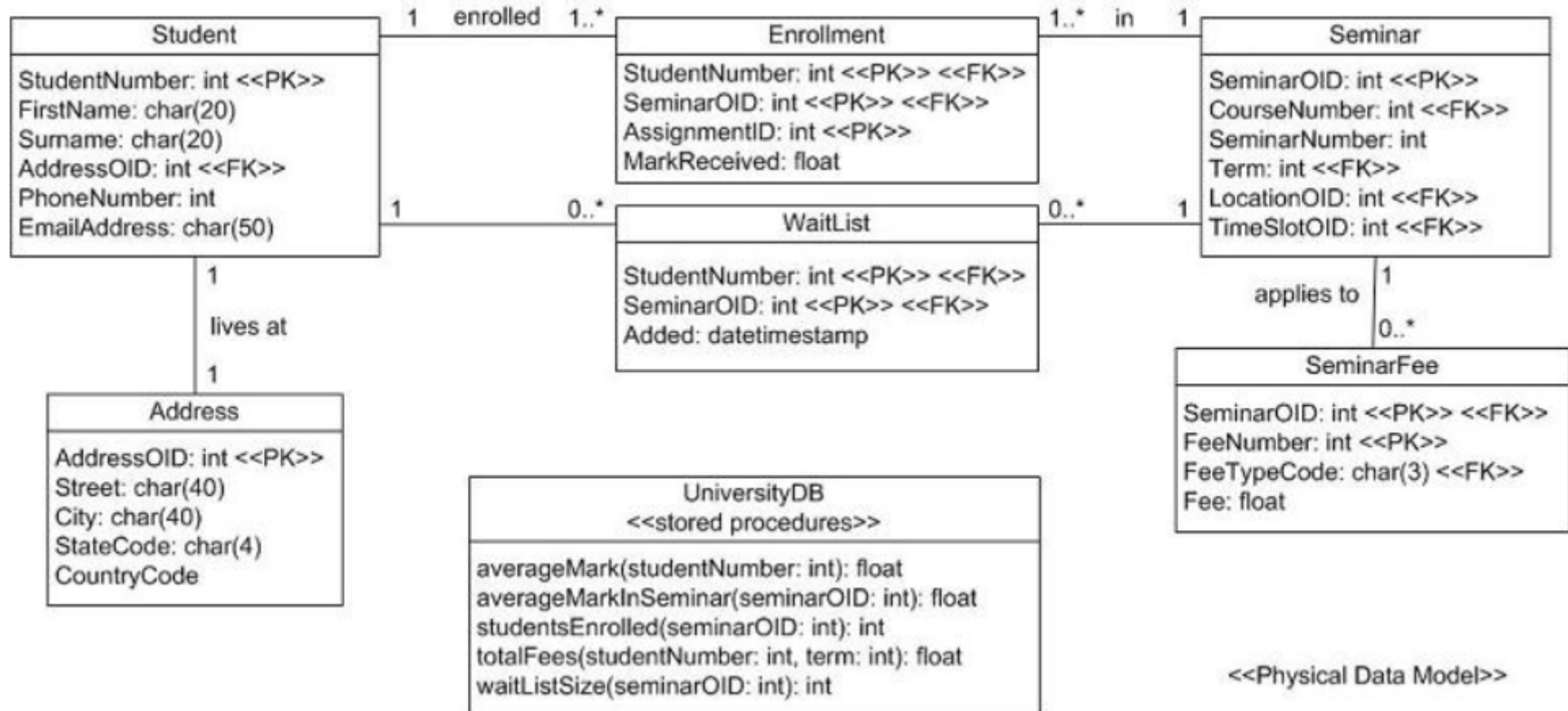
## Physical Data Model

pyright 2006 Scott W. Ambler





# Диаграма 2



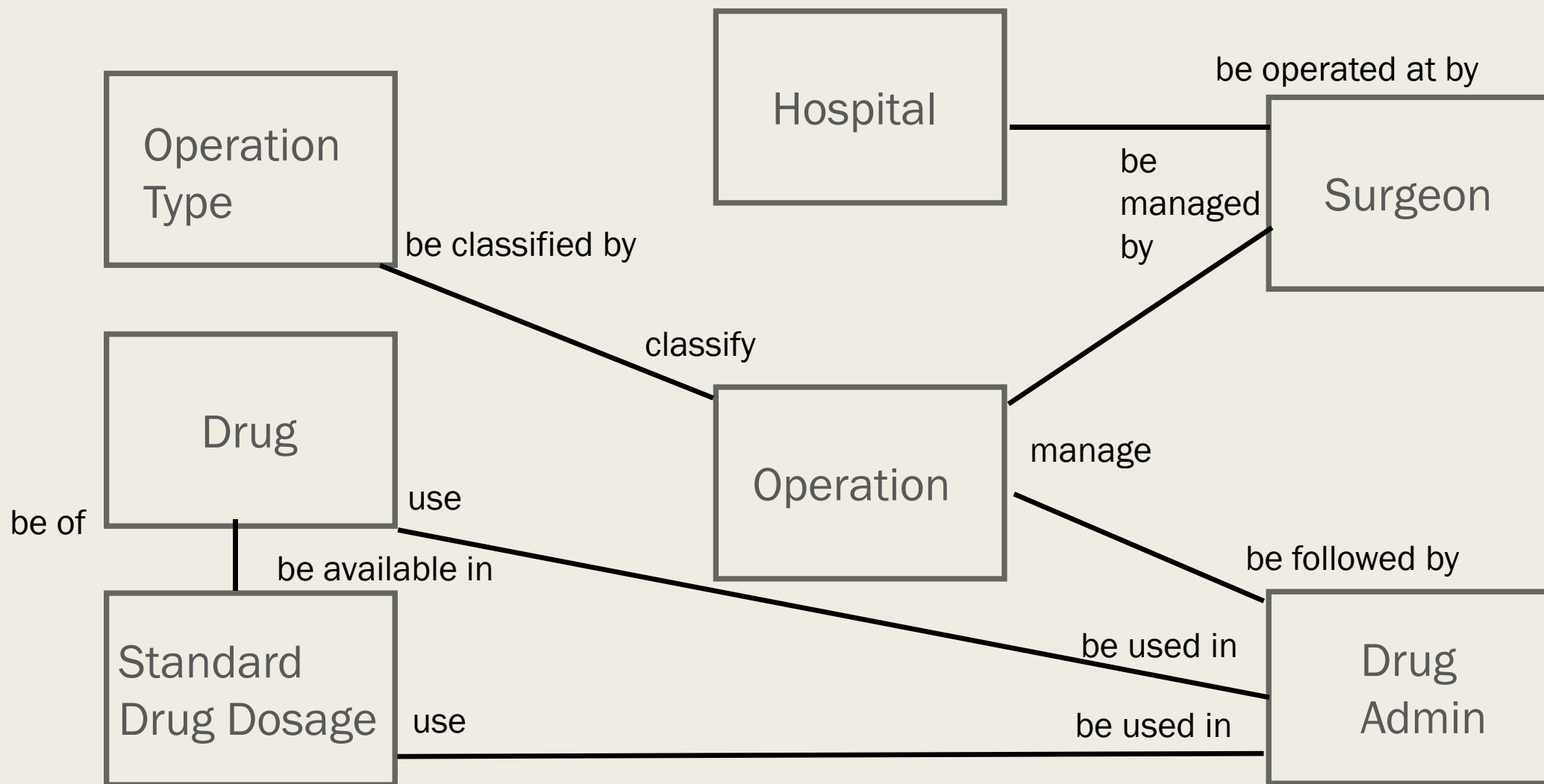
## Стъпки

- Определяне на таблици
- Нормализиране на таблици
- Определяне на колони
- Определяне на съхранени процедури
- Прилагане конвенциите за именуване
- Определяне на връзките
- Прилагане на шаблони за моделиране на данни
- Определяне на ключове

Да се дефинират тригери  
Да се дефинират съхранени процедури  
Да се генерират изгледи

Допълване на Hospital модел

# Модел на данни Hospital



ВЪПРОСИ ?

