

МОДЕЛИРАНЕ И АНАЛИЗ НА СОФТУЕР

Павел Кюркчиев

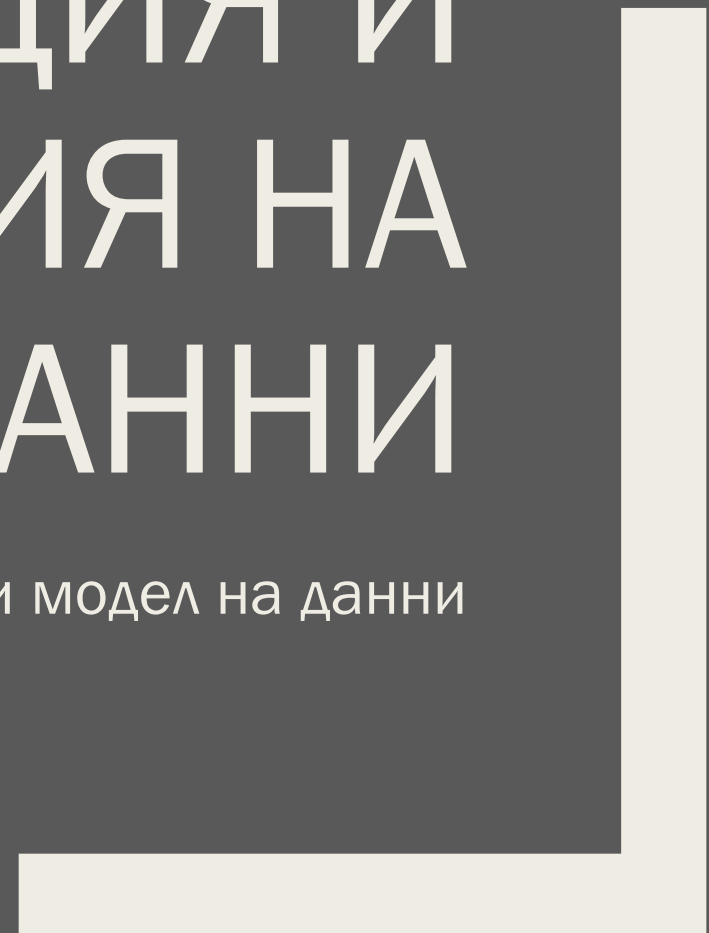
Ас. към ПУ „Паисий Хилендарски“

<https://github.com/pkyurkchiev>

@pkyurkchiev

НОРМАЛИЗАЦИЯ И ДЕНОРМАЛИЗАЦИЯ НА ДАННИ

Логически и Физически модел на данни



Защо нормализираме?

Аномалии в поведението на операциите

- Ако данните представляват различни семантични факти по едно и също време, тогава операции като Insert, Update, Delete могат да имат различно поведение. Изтриването може да доведе до изтриване на факт, който не трябва да бъде изтриван. Изпълнението на промяна би изисквало пълно сканиране, вместо директна единична промяна. Вмъкването на факт не може да бъде направено поради липсата на друг факт.

Съществуване на неконсистентни данни

- Данните са взаимно свързани. Всяка промяна върху обект в базата данни трябва да бъде приложена и върху свързаните с него обекти. Базата данни и изгледите често са базирани на макро данни (получени данни). Ако значението на получените данни не е разбираемо от потребителя, тогава могат да бъдат направени грешни изводи. Ако базовата информация бъде променена, то и получената информация трябва да се промени, за да се избегне неконсистентност.

Съществуване на информацията на повече от едно място (redundancy)

- Данните в базата данни могат да бъдат съхранени в различни асоциации. Ако това не е направено преднамерено и ако не бъде предвиден механизъм за промяната на информацията на различните места, на които се среща, може да възникне проблем. Информацията може да бъде част от други данни.

Пример:

Факултетен номер на студент. Номерът може да съдържа годината на записване. Ако променим самостоятелно годината на записване, то трябва да се погрижим и за номера.

Нестабилност на схема след промяна

- Приложения, базирани на данни, не са стабилни в дългосрочен план. В жизненият им цикъл рано или късно се появяват нови изисквания, които променят схемата за доставка и съхранение на информацията. Единственото възможно решение е създаването на гъвкава схема плюс оставяне на възможности за промяна и надграждане.

Различно ниво на абстракция в приложенията

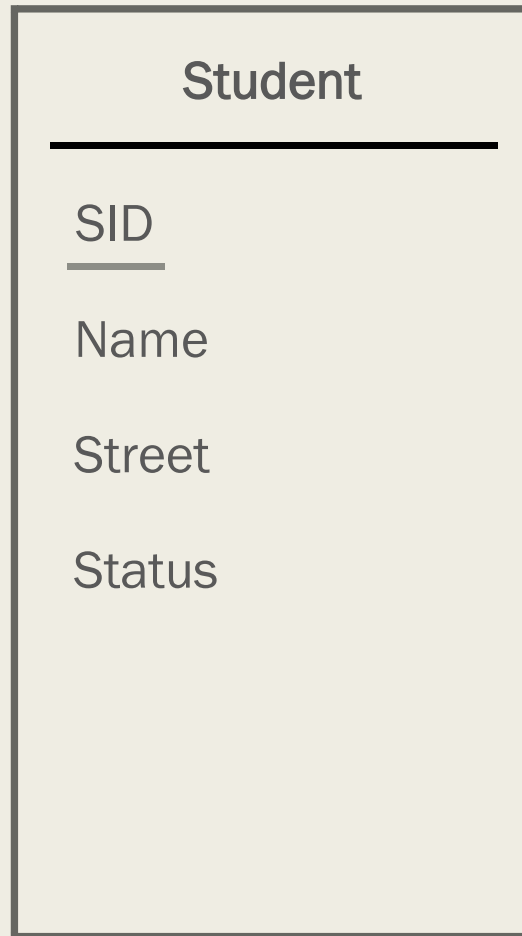
- Потребителските групи имат различни изисквания за абстрактността на данните в сравнение с изискванията на релационния модел. Хората използват информацията по различен начин.

Нормализация – логически модел

Първа нормална форма (First Normal Form)

- Правилото за първа нормална форма гласи, че не трябва да има атрибути, съдържащи повтарящи се групи от информация.
- Клас обект, който съдържа само една стойност за даден атрибут в обект-инстанция, гарантира, че клас обектът е от първа нормална форма.

Клас обект Student



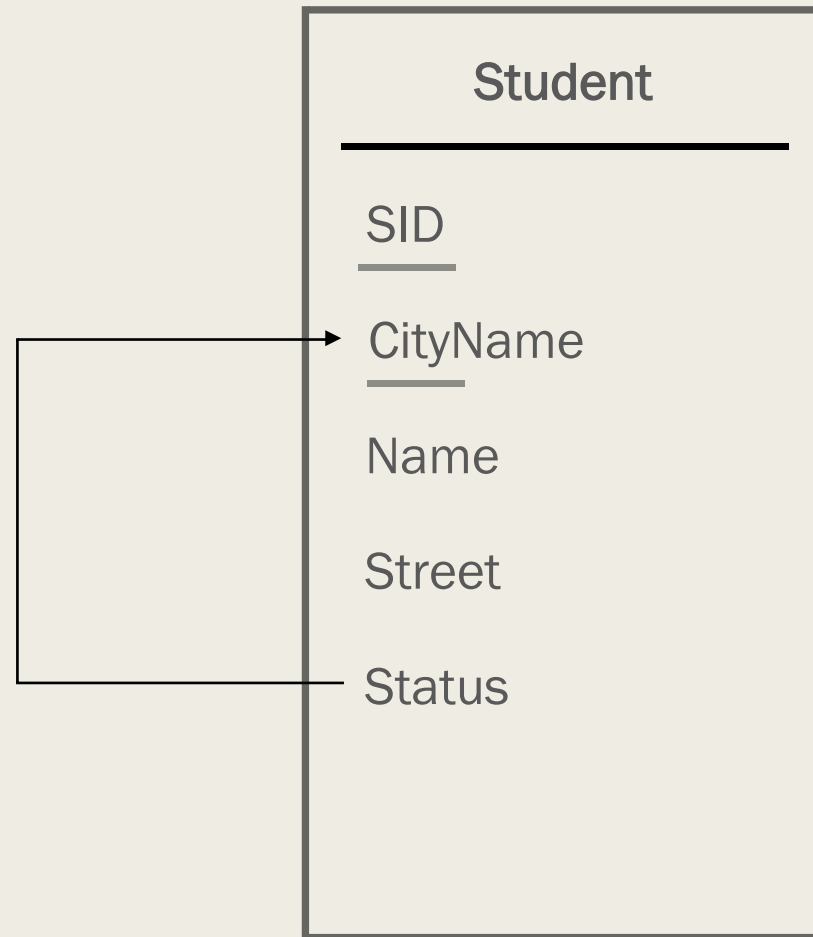
Втора нормална форма (Second Normal Form)

- Втората нормална форма гласи, че всички ключови атрибути трябва да определят неключовите атрибути. Втората нормална форма разглежда ситуациите, когато идентификаторът на обект е изграден от два или повече атрибута, и неключов атрибут зависи само от част от идентификатора.

Нарушение на втора нормална форма

- Нарушение на втората нормална форма възниква, когато имаме комбиниран ключ и част от ключа определя някои неключови атрибути.

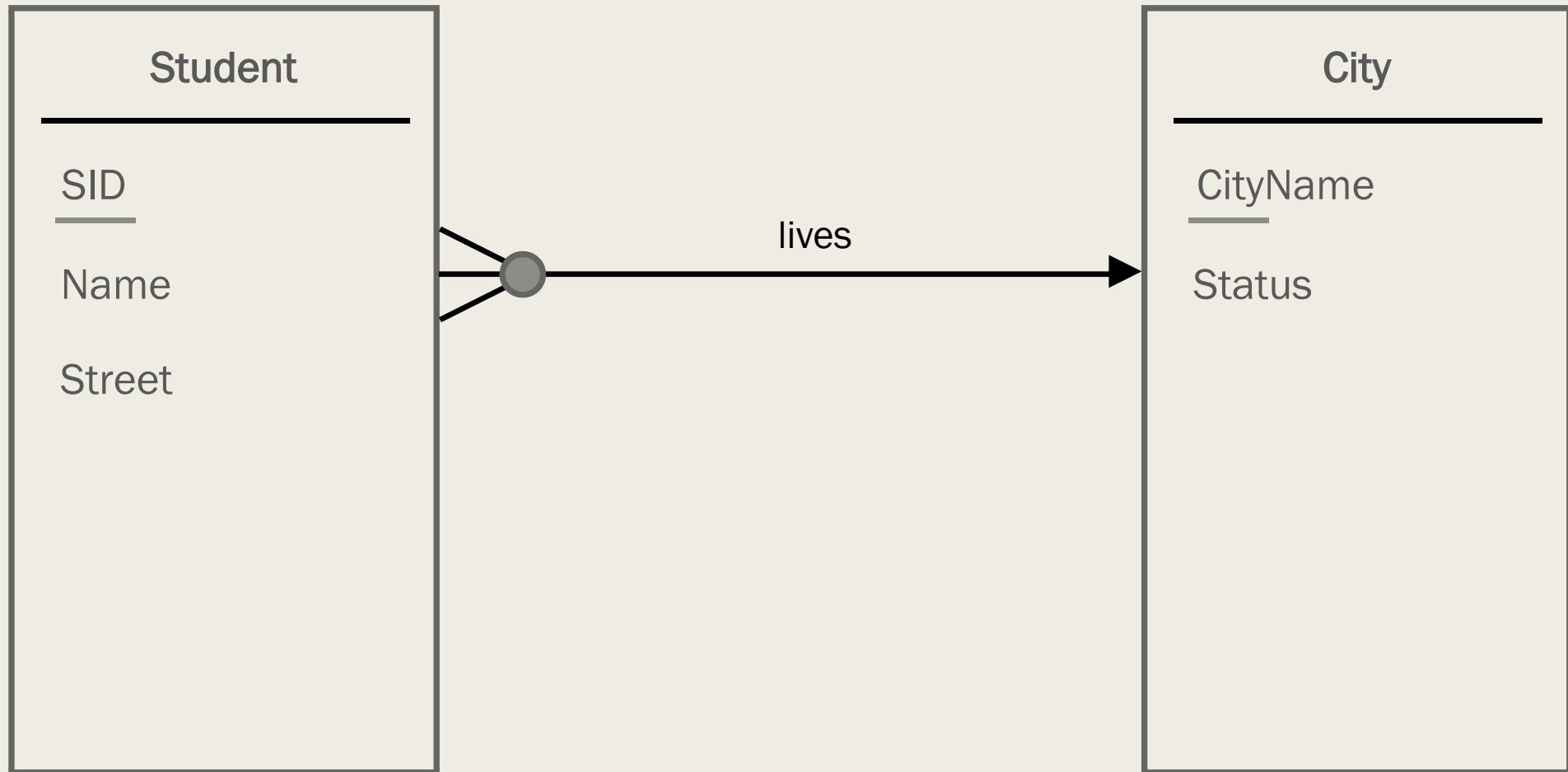
Проблемен клас обект



Объекти инстанции

Student	Student	Student
<u>100</u> St. Louis Jack Taylor 212 S. Page Good	<u>200</u> New York Tom Larson 444 N. Monroe Excellent	<u>100</u> Detroit Jack Taylor 212 S. Page Good

Решение



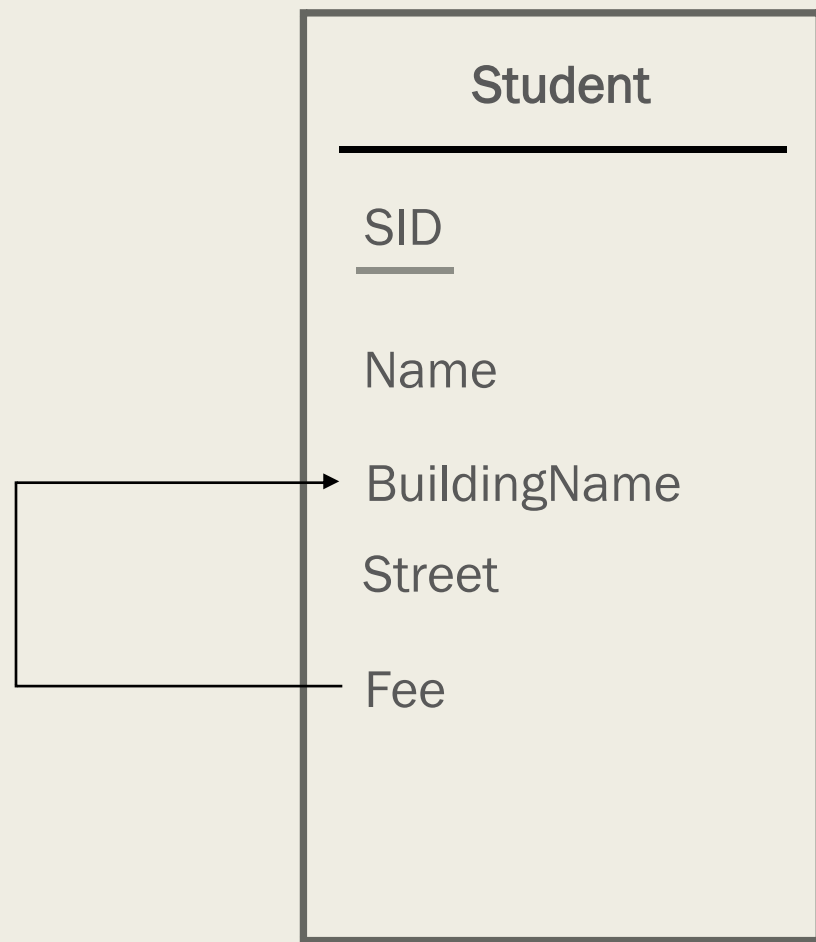
Трета нормална форма (Third Normal Form)

- Правилото на третата нормална форма е, че неключовите атрибути трябва да бъдат независими.

Нарушение на трета нормална форма

- Трета нормална форма се нарушава, когато съществува зависимост от неключови атрибути под формата на транзитивна зависимост.

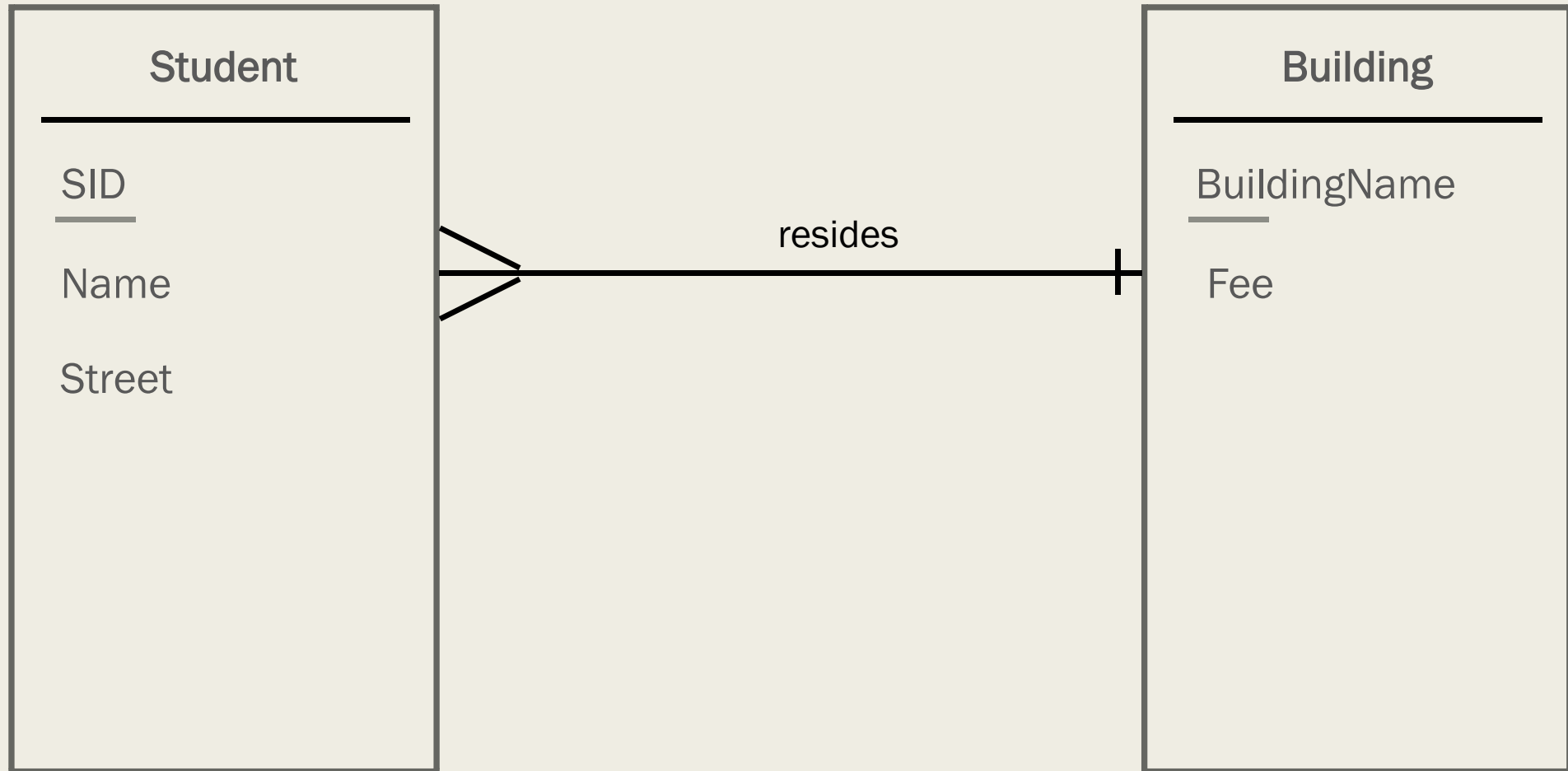
Проблемен клас обект



Объекти инстанции

Student	Student	Student
<u>100</u> Jack Taylor 212 S. Page Keller 250	<u>200</u> Tom Larson 444 N. Monroe Carrington 300	<u>300</u> Kay Beth 4212 S. Normal Keller 250

Решение



Бойс-Код нормална форма (Boyce-Codd Normal Form)

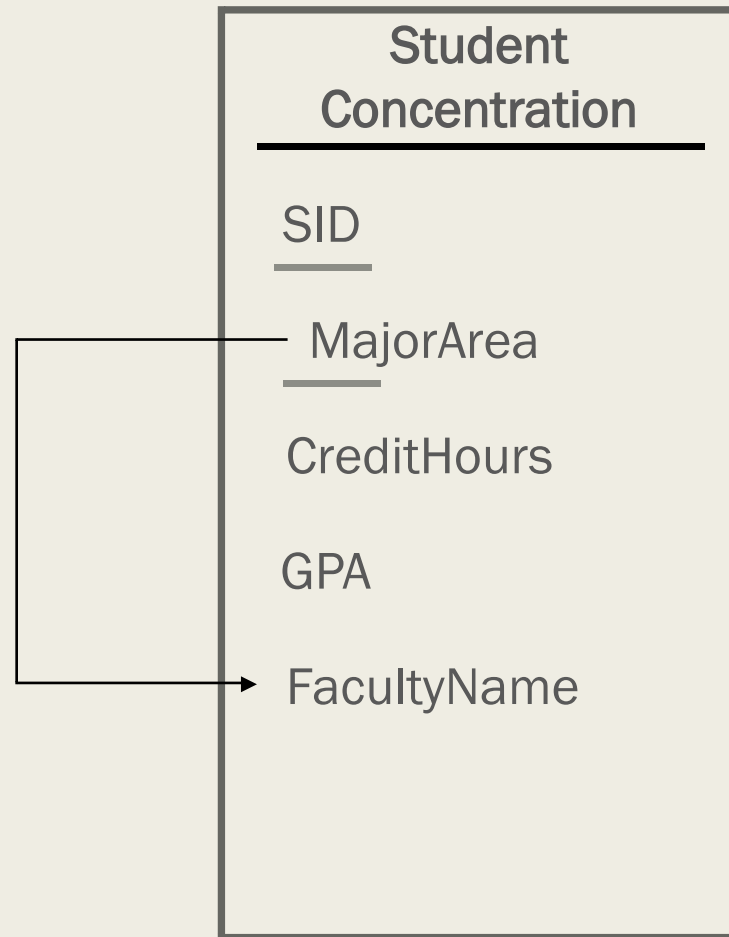
- Бойс-Код разширява трета нормална форма. Правилото на Бойс-Код за нормална форма е, че всеки определящ фактор е кандидат-ключ.
- Въпреки че нормалната форма на Бойс-Код и третата нормална форма като цяло генерират един и същ резултат, нормалната форма Бойс-Код е по-силна дефиниция от третата нормална форма.

Бойс-Код нормална форма

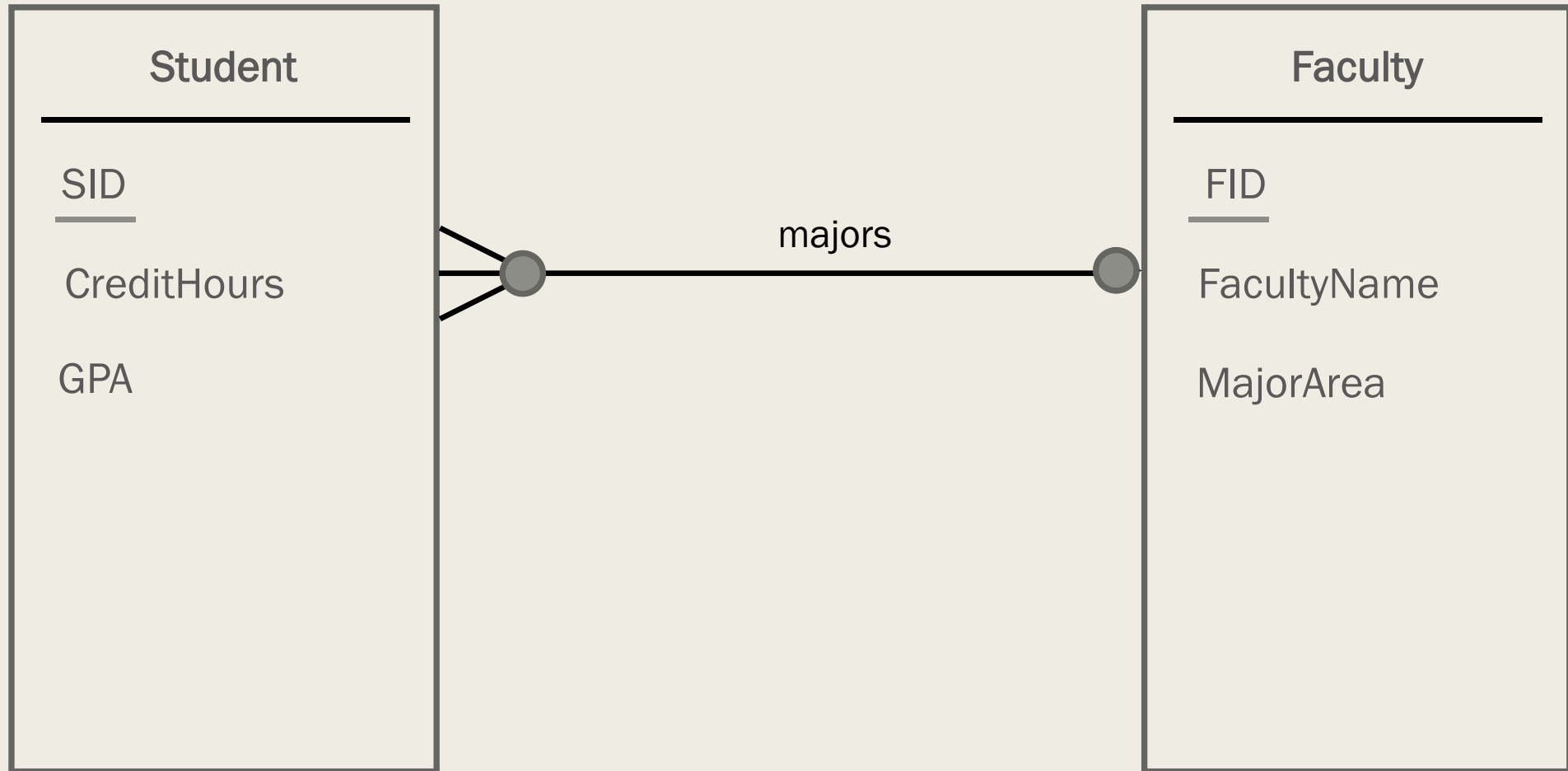
Бойс-Код разглежда два специални случая, които не са обхванати от трета нормална форма:

- Част от клас обект-идентификатора определя друг атрибут, и
- Неключов атрибут определя част от клас обект-идентификатора.

Проблемен клас обект



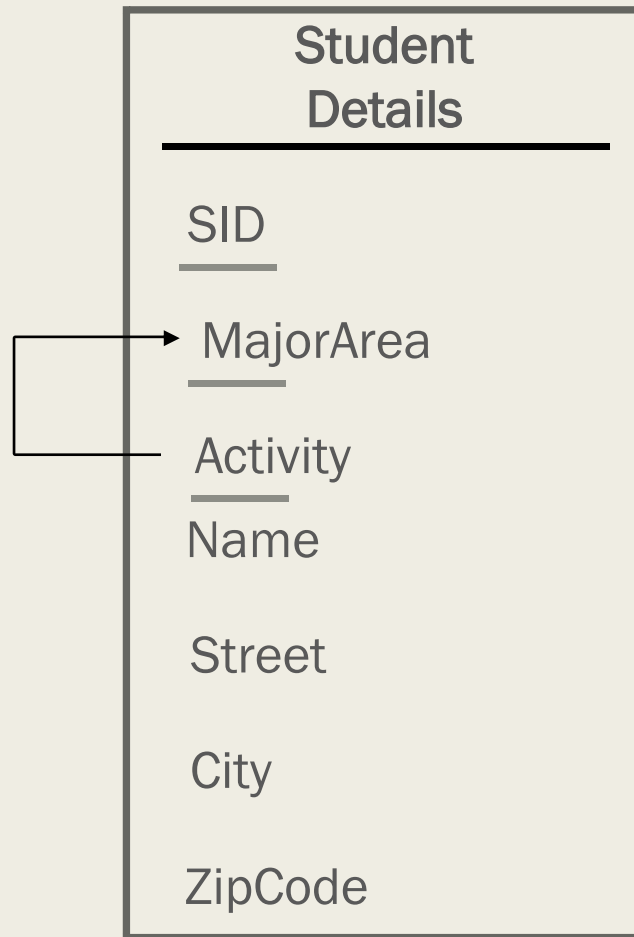
Решение



Четвърта нормална форма (Fourth Normal Form)

- Четвърта нормална форма гласи, че в клас обект не трябва да има повече от една многозначна зависимост.

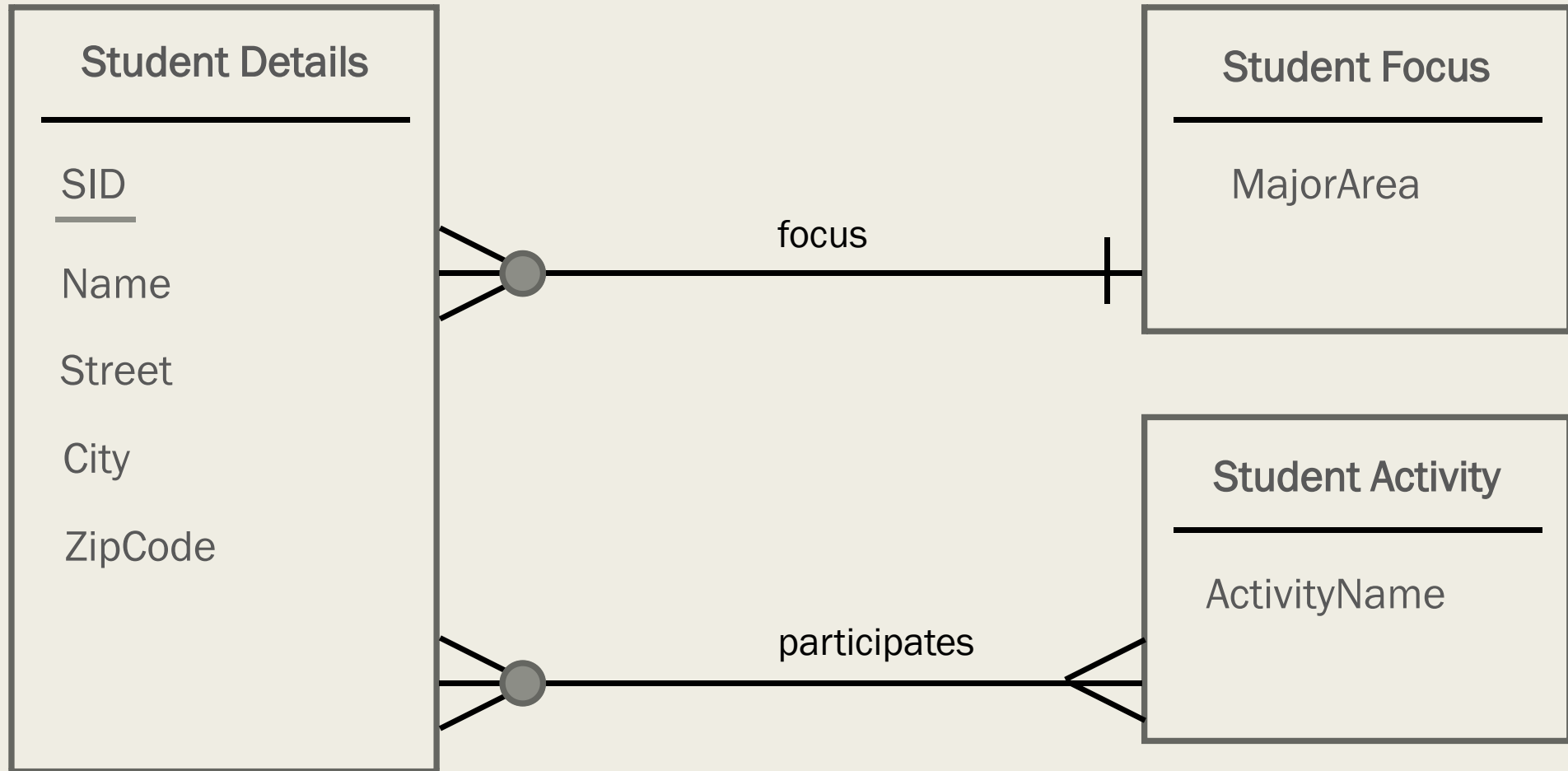
Проблемен клас обект



Решение

За да се отстрани нарушената четвърта нормална форма, трябва да бъдат отделени слабите обектни типове с идентифициращи връзки.

Решение



Модел на данни „Продажба на автомобили“:

Производител на автомобил

Цена на автомобил

Цвят на автомобил

Модел автомобил

Година на производство

Адрес на централен офис (Дилър)

Дилър

Мощност

Създаване и нормализация на следния модел (логически)

Денормализация – физически модел

Денормализация (Denormalization)

- Денормализация е стратегия, използвана в предварително нормализиран модел (база данни), за да се увеличи производителността. Идеята зад това е да добавим излишни данни, за които смятаме, че ще ни помогнат най-много.

Цел

Целта на процеса по денормализация е да намалим времето за изтегляне на информацията, като направим информацията по-достъпна чрез заявка или да създаде обобщена извадка в отделен клас обект.

Създаване на денормализирана форма

За денормализирането на физически модел, можем да използваме допълнителни атрибути във вече съществуващия клас обект, може да добавим и нов клас обект, възможно е и създаването на инстанция от съществуващия клас обект.

Кога трябва да използваме денормализация

- Поддържане на история – Имаме нужда от съхраняване на информация, която може да бъде променяна във времето;
- Подобряване на извличането на информация – Някои заявки може да зависят от информация, която е разпръсната между повече от 10 таблици. Достъпът до информация по този начин изисква много ресурси и най-вече време за доставяне;

- Подобряване извличането на справките – Нуждаем се от конкретна статистика доста често. Възможно е тази информация да пристига всяка секунда;
- Изчисляеми стойности – Нуждаем се от стойности, които се получават при изчислението на различни атрибути.

Кога не трябва да използваме денормализация

- Недостатъчно дисково пространство – Денормализация на данните в повечето случаи води и до заемането на по-голямо дисково пространство;
- Аномалии в данните – Трябва да се внимава при прилагането на каквато и да е операция върху данни. Дублираната информация трябва да бъде поддържана еднаква;

- Документация – При денормализацията всички допълнителни правила трябва да бъдат документиран;
- Забавяне на други операции – При денормализация може да се получи забавяне при следните операции: добавяне, промяна или изтриване;
- Повече програмиране – Правила 2 и 3 биха изисквали допълнително писане на код.

ВЪПРОСИ ?

