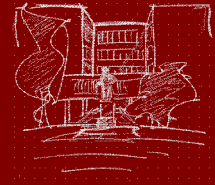


[P273] Пројектовање база података 2



Саша Малков
Универзитет у Београду
Математички факултет
2023/2024

[P273] Пројектовање база података Саша Малков



Тема 3 Процес пројектовања база података

[P273] Пројектовање база података – Саша Малков – 2023/24 – час 2

1

Процес пројектовања база података

Место пројектовања база података



- Пројектовање базе података се одвија у оквиру ширег процеса развоја софтвера или информационог система
- Да би могло да се приступи пројектовању базе података претпоставља се да је већ обављено:
 - истраживање система
 - т.ј. прикупљање информација о домену чију слику ће да представља база података која се пројектује
 - анализа система
 - т.ј. анализа прикупљених информација и формулисање захтева
- Већ постоје потребне информације и дефинисани захтеви
 - али често нешто недостаје, па неки елементи истраживања и анализе морају да се понове или допуне у току пројектовања

Универзитет у Београду – Математички факултет

[P273] Пројектовање база података – Саша Малков – 2023/24 – час 2

2

Процес пројектовања база података

Процес пројектовања база података



- Базе података се пројектују по нивоима
- Нивои пројектовања *делимично* прате нивое архитектуре
- Границе нивоа пројектовања се не поклапају тачно са границама архитектуре
- Такво неподударање је природно:
 - Неки нивои пројектовања су апстрактни, а неки су конкретни
 - Сви нивои архитектуре су конкретни – постоје у имплементацији

Универзитет у Београду – Математички факултет

[P273] Пројектовање база података – Саша Малков – 2023/24 – час 2

3



Основне целине при пројектовању

- **Концептуално пројектовање**
 - одговара апстракцији спољашње схеме и дела концептуалне схеме
 - најпре се прави по апстрактан *појединачни концептуални модел* за сваку од спољашњих схема
 - затим се од њих прави јединствен апстрактан *обједињени концептуални модел*
- **Логичко пројектовање**
 - углавном одговара концептуалној схеми
 - прилагођавање концептуалног модела конкретном моделу података
 - прави се *логички модел* на језику модела изабраног СУБП
- **Физичко пројектовање**
 - прави се *физички (имплементациони) модел*
 - на крају се имплементирају и спољашње схеме
- **Пројектовање безбедности**
 - Углавном (али не сасвим) ортогонално у односу на остале кораке
 - Преплиће се са њима, али се у највећој мери одвија у току и након физичког пројектовања



Основни кораци при пројектовању

- **Концептуално пројектовање**
 - 1 Анализа захтева
 - 2 Концептуално пројектовање појединачних спољашњих схема
 - назива се и *Пројектовање погледа*
 - 3 Прављење обједињеног концептуалног модела
 - назива се и *Обједињавање погледа*
 - 4 Груписање ентитета
 - **Логичко пројектовање**
 - 5 Превођење концептуалног модела у логички модел
 - 6 Пречишћавање схеме
 - **Физичко пројектовање**
 - 7 Физичко пројектовање имплементације
 - 8 Физичко пројектовање спољашњих схема
 - **Пројектовање безбедности**
 - 9 Пројектовање безбедности
- * Првих 8 корака се изводе редом, а 9. се изводи паралелно са њима.



1 - Анализа захтева

- **Циљ:**
 - Разумевање домена
 - елементи домена и односи међу њима
 - њихови атрибути (подаци)
 - оквирна процена обима података и односа међу подацима
 - Разумевање апликација
 - потребе за подацима
 - учесталост упита и трансакција
 - перформансе



2 - Концептуално прој. појединачних схема

- **Циљ:**
 - Прављење модела домена за сваку спољашњу схему посебно
 - назива се и "модел података високог нивоа"
 - Описивање
 - структуре домена
 - односа међу елементима домена
 - услова интегритета
- Све се описује из угла корисника
 - Веома је важно да се опише тачна семантика података и односа
 - Смисао концептуалног модела је у верном представљању семантике



3 – Обједињавање погледа

- **Циљ:**
 - Прављење обједињеног апстрактног модела домена
 - Уједначавање речника и описа домена
 - Додатно апстраховање ради обједињавања
 - Интегрисање погледа у јединствени модел



4 – Груписање ентитета

- **Циљ:**
 - Прављење већег броја мањих и прегледнијих дијаграма модела
 - Груписање јаче повезаних ентитета у целина
 - Уочавање централних ентитета за одговарајуће групе
- Овај корак није суштински (тј. не прави нови модел), али може да буде значајан за лакше разумевање сложених модела



5 – Превођење концептуалног у логички модел

- **Циљ:**
 - Сви елементи модела се преводе на језик логичког модела
 - модел домена се преводи у модел базе података
 - Иницијално прављење логичког модела
 - Основно прилагођавање концептуалног модела конкретном имплементационом моделу података



6 – Пречишћавање схеме

- **Циљ:**
 - Доследно прилагођавање модела базе података језику и правилима конкретног модела података
 - Препознају се и решавају се потенцијални проблеми
 - У случају релационог модела основни циљ је елиминација редундантности



7 – Физичко пројектовање имплементације

- **Циљ:**
 - Моделирање начина имплементирања базе података
 - назива се и “модел података”
 - Оптимизовање логичког модела према специфичностима примењеног СУБП и према очекиваном моделу употребе
 - Одређивање тачне интерне схеме базе података
 - параметри имплементације табела базе података
 - помоћне структуре (индекси, бафери,...)
 - параметри функционисања СУБП
 - пројектовање дистрибуирања
 - ...
 - Одређивање механизма пресликавања физичке у концептуалну схему



8 – Физ. пројектовање спољашњих схема

- **Циљ:**
 - Пројектовање имплементације спољашњих схема
 - Пројектовање апликативних интерфејса за приступање подацима
 - Прилагођавање интерне схеме специфичностима примењених спољашњих алата



9 – Пројектовање безбедности

- **Циљ:**
 - Препознавање врсте и улоге сваког корисника
 - Препознавање врсте апликације
 - Дефинисање корисничких група и одговарајућих минималних скупова привилегија за све различите улоге и врсте корисника и апликација
 - Препознавање делова базе података који су посебно осетљиви и којима је потребно додатно редуковати приступ
 - Дефинисање и имплементирање одговарајућих безбедносних механизма



Однос пројектовања и архитектуре

- При пројектовању се два пута пролази кроз нивое архитектуре
 - прво наниже
 - концептуално па логичко пројектовање прво разматрају најапстрактније елементе архитектуре
 - затим навише
 - физичко пројектовање дефинише имплементације свих схема, одоздо навише – најпре физичке, па концептуалне па спољашњих схема

Однос пројектовања и архитектуре

Концептуално пројектовање

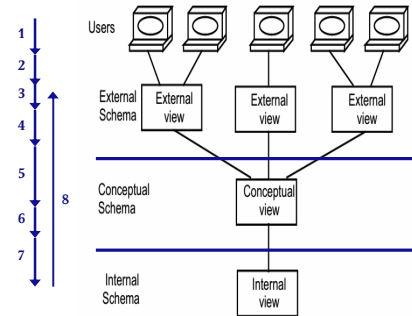
1. Анализа захтева
2. Концептуално пројектовање погледа
3. Обједињавање концептуалног модела
4. Груписање ентитета

Логичко пројектовање

5. Превођење концептуалног модела у логички
6. Пречишћавање схеме

Физичко пројектовање

7. Оптимизовање логичког модела и пројектовање имплементације према моделу употребе
8. Имплементација спољашње схеме



Универзитет у Београду - Математички факултет

Општи и специфични елементи пројектовања

- Концептуално пројектовање
 - односи се на домен и на апликације
 - одвија се углавном уопштено, без много обраћања пажње на врсту и имплементацију СУБП
- Логичко пројектовање
 - односи се на конкретан модел података изабраног СУБП
 - узимају се у обзир и неке специфичности конкретне имплементације СУБП
- Физичко пројектовање
 - односи се на конкретан СУБП
 - неки елементи су уопштени за врсту (модел) база података
 - већина елемената се више или мање прилагођава конкретној имплементацији СУБП

Универзитет у Београду - Математички факултет

Пројектовање релационих база података

- У складу са претходно представљеним концептима
- Специфичности су последица особина релационог модела
- Логичко пројектовање има за циљ релациони модел
 - концептуални модел се преводи на језик релационог модела
 - пречишћавање схеме се односи првенствено на елиминацију редундантности
- Физичко пројектовање
 - интерна схема зависи од импл. РСУБП, али има неких дељених концепата
 - дефинисање концептуалне схеме помоћу погледа (и окидача)
 - дефинисање спољашњих схема се остварује на исти начин

Универзитет у Београду - Математички факултет

[P273]

Пројектовање база података

Саша Малков

Тема 4

Концептуално моделирање



Концептуални модел

- Описује препознату семантику посматраног домена
- Посебно важан
 - за разумевање целине система
 - за размену информација са имплементаторима апликација
 - за размену информација са клијентима
- Модел се обично обликује и записује применом
 - ЕР моделирања
 - УМЛ моделирања



Основне карактеристике

- Кључне особине концептуалног модела су
 - **Исцрпно описана семантика**
 - од суштинске важности
 - мора да буде јасно представљена улога сваког елемента у домену
 - ентитети, односи, атрибути
 - **Апстрактност**
 - циљ је лакше и боље разумевање семантике целине и делова модела
 - смањује се ниво детаљности
 - изостављају се сувишне информације
 - без небитних или мање битних атрибута
 - без помоћних табела неопходних за физичку имплементацију
 - уопштава се представљање односа



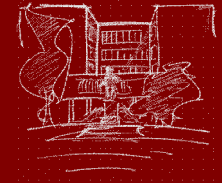
Кораци

- Прављење концептуалног модела почива на следећим корацима:
 - Анализа захтева
 - Концептуално моделирање података и односа
 - Интегрисање погледа (локалних схема)
 - Груписање ентитета

[P273]

Пројектовање база података

Саша Малков



Тема 4.1

Концептуално моделирање

-

Анализа захтева



Анализа захтева

- Анализа захтева
 - Веома важна!
 - Веома захтевна!
- Тесно повезана за процесом анализе захтева у контексту општег пројектовања софтвера или информационог система
- Да би се разумеле потребе за подацима често морају да се детаљно разумеју сви процеси који се одвијају у систему



Циљеви анализе захтева

- Основни циљеви:
 - Превођење функционалних захтева у контекст трајних података
 - Разумевање и описивање података и њихове улоге
 - елемената домена и њихових односа
 - тј. структура одговарајућих потребних података и њихових односа
 - уочава се који атрибути елемената домена имају значаја за моделирање
 - свих појединачних трансакција које се извршавају у домену
 - над елементима домена, односно над подацима о њима
 - Дефинисање нефункционалних захтева
 - интегритет
 - перформансе
 - безбедност
 - и разни административни аспекти
 - ...



Циљеви анализе захтева (2)

- Основни циљеви:
 - ...
 - Одређивање и обликовање додатних услова имплементације
 - технологије
 - хардверске и софтверске претпоставке
 - апликативни и кориснички интерфејси
 - и друго
 - Израда исцрпне документације за све наведено



Циљеви анализе захтева (3)

- Један од најважнијих циљева анализе захтева, у оквиру концептуалног пројектовања, је **декомпозиција система на скуп слабије повезаних делова / сегмената**
- Идеја је да се пројектовање сложеног система подели на мање целине, које се лакше изучавају и моделирају
 - т.ј. на већи број мањих концептуалних модела
- Касније, у фази груписања, ти делови ће се спојити у целину



Циљеви анализе захтева (4)

- Декомпозиција обично почиње од природних разлика – од различитих уочених спољашњих схема, тј. врста корисника
 - на пример, у случају БП факултета, можемо да уочимо *студенте, наставнике, студентску службу, финансијску службу,...*
- Наставља се, ако је потребно, уочавањем мањих функционалних целина и даљим декомпоновањем
 - уочавамо послове који су издвојени од остатка система
 - на пример, организација уписних рокова
 - у оквиру једне спољашње схеме уочавамо различите послове
 - нпр. послови студента као што су *пријављивање испита, упис године* и сл.



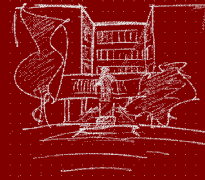
Послови у оквиру анализе захтева

- Морају да се обухвате сви битни аспекти анализе система
 - Анализа захтева обухвата детаљно изучавање и моделирање система
 - То може да обухвати (и често је неопходно) и моделирање пословних и имплементационих аспеката система
 - Ова фаза се преплиће са пројектовањем ИС и не може да се из њега издвоји, а да при томе остане потпуна
- Самом анализом захтева се нећемо бавити посебно детаљно
 - (види предмет Информациони системи)
 - У случају ИС, анализа захтева за ПБП се обично не издваја из анализе захтева за читав ИС
 - мада могу да се уоче неки захтеви који се односе само на БП
 - У случају изоловане апликативне базе података, анализа се ради одвојено, али на сличан начин као при пројектовању ИС, с тим да се изучавање и моделирање ограничавају на домен апликације

[P273]

Пројектовање база података

Саша Малков



Тема 4.2

Концептуално моделирање - Моделирање података и односа



Концептуално моделирање домена / података

- Циљ концептуалног моделирања је прављење верне слике домена
- Зато се назива и *моделирање домена*
- Моделирају се само они елементи домена који имају значаја за циљну базу података
- Зато се конкретан корак описивања елемената домена назива и *концептуално моделирање података*



Концептуално моделирање података

- У овом кораку се бавимо концептуалним моделирањем једног дела система – тј. једне спољашње схеме
 - углавном на сличан начин као што би се моделирао цео систем, када не би био декомпонован
- Главни кораци
 - Класификација скупова података
 - препознавање ентитета / класа и атрибута
 - Препознавање односа
 - генерализација / специјализација, хијерархије
 - асоцијација и сложенији односи



Класификација скупова података

- Сваки објекат/предмет у домену је кандидат за ентитет или атрибут
 - т.ј. свака именица у текстуалном опису домена или прикупљеним захтевима је кандидат за ентитет или атрибут
 - није увек очигледно шта је ентитет (или класа) а шта атрибут
- Неке смернице
 - Ентитети би требало да садрже описне информације
 - Ако нешто има вишеструку или сложу вредност (а не скаларну) онда је вероватно ентитет
 - Атрибути би требало да припадају оним ентитетима које најнепосредније описују
 - Ако је нешто описано једном скаларном вредношћу, онда је вероватно у питању атрибут



Садржај ентитета

- Ентитети садрже описне информације
 - ако имамо описне информације о неком податку, онда би то требало да буде ентитет
 - “предмет” има “назив” и “број ЕСПБ”, па би требало да буде ентитет
 - ако неки податак има само идентификатор (који може да буде описног карактера али нема додатних описа) онда би то требало да буде атрибут
 - “оцена” се идентификује својом вредношћу и нису потребни додатни описи, па је то атрибут



Садржај ентитета (2)

- Пример: “град”
 - ако нам је довољан назив града, онда може да буде атрибут
 - али ако су нам важне и неке друге информације о граду (на пример локација града или број становника), онда је то ентитет



Вишеструке вредности

- Ако нешто може да има више вредности, сложену вредност, или вредност која је подложна променама, онда је то ентитет
 - то некада не мора да буде неопходно, али је једноставније за анализу и разматрање, а касније може да се оптимизује ако је потребно
- Пример:
 - ако “удружење студената” има само назив, онда је то атрибут
 - али ако један студент може да буде члан више удружења, онда је можда боље да се удружење прогласи за ентитет
 - такође, пошто више студената може да буде у истом удружењу, евентуална промена назива удружења може да буде проблематична ако је у питању само атрибут...



Додељивање атрибута ентитетима

- Атрибути се додељују оним ентитетима које најнепосредније описују
 - То некада није сасвим очигледно
 - Обично може да се установи из функционалних зависности
 - ако се погрешно, обично може да се поправи у фази логичког пројектовања
 - Некада одговарајућа зависност постоји само у домену али не и у подацима
 - тј. вредност атрибута се у начелу одређује на основу ентитета, али не може да се установи функционална зависност
 - у таквим случајевима се обично ради о (потенцијално прикривеним) транзитивним зависностима



Додељивање атрибута ентитетима (2)

- На пример:
 - ако за неки атрибут А проценимо да припада ентитету Е1
 - а ентитет Е1 је у односу са ентитетима Е2 и Е3
 - можда је атрибуту А место у неком од ентитета Е2 и Е3?



Додељивање атрибута ентитетима (3)

- Конкретан пример:
 - претпоставимо да атрибут *Име* припада ентитету *Студент*
 - ако студент упише два ст.програма (било у исто време или један после другог), онда ће се његови лични подаци понављати, укључујући и *Име*
 - можемо да препознамо скривену зависност *Студент* -> *Особа* -> *Име*
 - може да има смисла издвајање новог ентитета *Особа*, тако да више *Студент*а може да се односи на једну исту *Особу*
 - *** у пракси се ово често не ради, зато што име (и други лични подаци) могу да се промене између уписивања различитих програма, па би могло да буде проблема у вези са званичним документима
 - различите референце на физички исту особу се повезују на неки други начин



Препознавање односа

- За сваки однос је потребно препознати
 - учеснике
 - а са њима и ред односа (тј. број учесника)
 - кардиналност
 - број таквих односа једног ентитета
 - опционост / обавезност
 - додатне атрибуте који описују однос
 - јасан назив и семантику односа у домену
 - као и одговарајуће функционалне зависности



Препознавање генерализације

- Генерализација = Сваки “посебан” случај “је” истовремено и “општи” случај
 - нпр. сваки квадрат је и правоугаоник
- Установљавање смера генерализација није увек очигледно
 - У случају ОО дизајна основни критеријум је понашање:
 - квадрат је специјалан случај правоугаоника
 - Код база података смо често у искушењу да игноришемо понашање и посматрамо само структуру:
 - структура правоугаоника је допуна (тј. специјалан случај) структуре квадрата – имамо један податак више



Препознавање генерализације (2)

- При грађењу хијерархија ипак не сме да се потпуно запостави понашање
 - Фокусирање само на структуру “може да прође” само у случају плитких хијерархија
 - У случају иоле дубљих хијерархија евентуално донесене погрешне одлуке могу да имају неугодне последице



Препознавање генерализације (3)

- Што апстрактније – то боље:
 - Атрибуте је потребно “попети” што више уз хијерархију
 - Кључне атрибуте је потребно везати за базу класу (ентитет)
 - И односе је потребно “попети” што више уз хијерархију

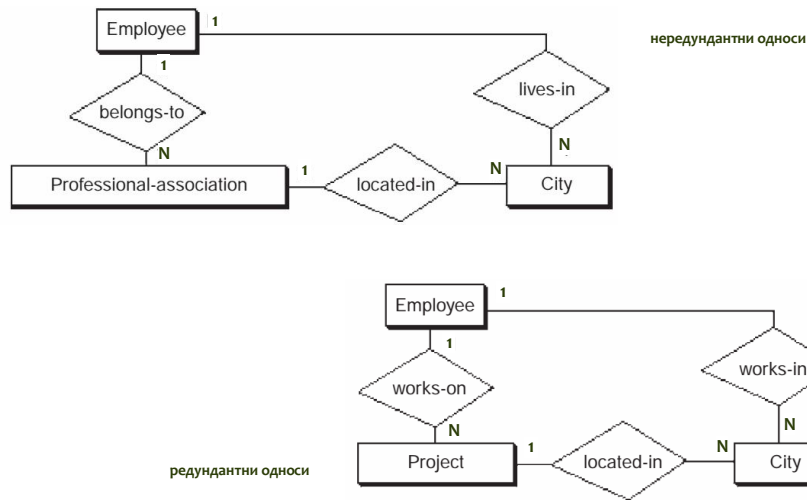
Редундантност односа

- Редундантни односи
 - имају за резултат тешко отклоњиву редундантност у моделу и имплементацији
 - ометају конзистентност и нормализацију
- Потребно (пожељно) је елиминисати редундантне односе

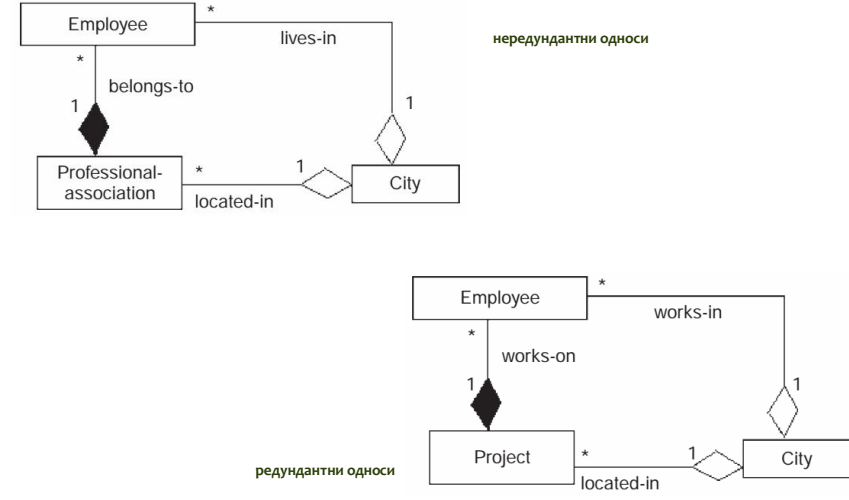
Елиминисање редундантних односа

- На нивоу концептуалног модела редундантност не мора увек да се отклања, али мора да буде документована
- Ако редундантност олакшава разумевање концептуалног модела, онда може да нам буде од користи да оставимо редундантан однос, ако га при томе јасно документујемо
- Ако није документована, може да прође непримећено при прављењу логичког модела, што не би смело да се деси
- **ОПРЕЗ!**
 - “исти” односи могу бити редундантни и нередундантни
 - редундантност зависи од семантике и домена

Редундантни односи, пример 1, ЕР



Редундантни односи, пример 1, УМЛ



Односи вишег реда



- Однос је “вишег реда” ако обухвата више од два учесника
- ОПРЕЗНО са односима вишег реда
 - увек је потребно да се додатно размисли да ли су неопходни
 - често је боље (и семантички и технички) да се однос вишег реда подели на више једноставнијих односа
- Упутство:
 - Увек покушати са више бинарних односа
 - Ако се испостави да је немогуће, тек онда користити односе вишег реда

Односи вишег реда, пример



- На пример:
 - У распореду часова, сложен однос је “наставник А држи час из предмета Б у учионици В у термину Г”
 - Међутим, однос наставника и предмета је важан и независно од распореда и може да се издвоји у концепт групе
 - “група Д” = “наставник А држи предмет Б”
 - може да има и додатне атрибуте, нпр. ознаку
 - може више наставника да држи различите групе за исти предмет
 - може да буде више група истог наставника и предмета
 - може да има и однос са студентима који чине групу
 - почетни однос се онда поједностављује (иако и даље сложен)
 - “група Д има час у учионици В у термину Г”

[P273]

Пројектовање база података

Саша Малков



Тема 4.3

Концептуално моделирање

-

Интегрисање погледа

Раздвојени погледи



- Концептуално моделирање се често одвија по деловима
 - одвојено за различите апликације
 - или делове апликације
- Основна мотивација:
 - анализа и моделирање мањих делова домена се изводе једноставније и ефикасније
 - мањи модели се лакше обликују и разумеју
- Сваки појединачан модел се назива “погледом” или “локалном схемом”
 - нема непосредног додира са “погледима” у језику SQL
 - мада они касније могу да се користе за имплементацију

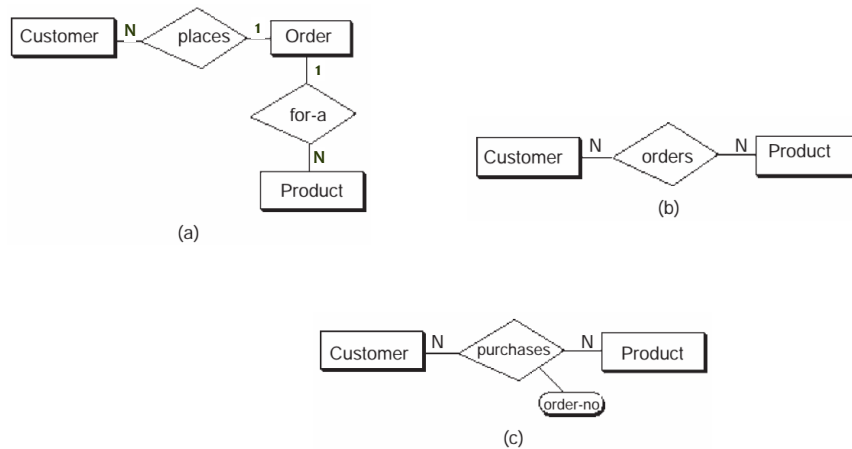
Раздвојени погледи, пример

- ИС факултета
 - из угла студента
 - потребно је да се зна да је неки наставник задужен за неки предмет
 - није битно када је ко изабран у које звање, па ни у ком је звању
 - из угла кадровске службе
 - није битно ко је задужен за који предмет
 - важно је када је ко изабран у које звање, када је био на одсуствима, који је период избора и друго

Интегрисање погледа (локалних схема)

- Општа (глобална) концептуална схема се добија интегрисањем локалних схема (погледа)
- То је некада једноставно али може да буде и веома сложено
- Зато се *интегрисање погледа* обично описује као посебан корак
 - саставни део концептуалног моделирања
- ВАЖНО: Термини “локална схема” и “глобална схема” у контексту концептуалног пројектовања имају другачији смисао него у контексту архитектуре дистрибуираних база података

Пример 2, различити модели за исти домен



Кораци при интегрисању погледа

- Основни кораци:
 - Поређење схема и препознавање конфликта
 - Преуређивање схема и разрешавање конфликта
 - Спајање и реструктурирање схема



Препознавање конфликта

- Врсте конфликта
 - конфликти имена
 - структурни конфликти
 - конфликти кључева
 - конфликти зависности



Конфликти имена

- Неуједначено именовање ентитета и односа
 - речници ентитета и односа морају бити уједначени
- Синоними
 - различити називи за исте концепте
 - први и најлакше решив проблем
- Хомоними
 - исти назив се користи за различите концепте
 - може да представља озбиљан проблем ако је различито тумачена спецификација захтева
 - теже решиво од синонима и зато има приоритет при решавању
 - често је повезан са структурним конфликтима



Конфликти имена (2)

- Пример синонима
 - Ентитет који представља пријаву и полагање испита се у једном погледу зове *Полагање*, а у другом *Испит*
- Пример хомонима
 - Именом *Испит* се назива
 - у једном погледу ентитет који представља положен испит
 - у другом погледу ентитет који представља пријаву и полагање испита



Структурни конфликти

- Различити структурни елементи се употребљавају за моделирање истих концепата
 - (пример 2 са наруџбеницама)
- Да би погледи могли да се споје, одговарајући концепти морају да имају идентичне структуре

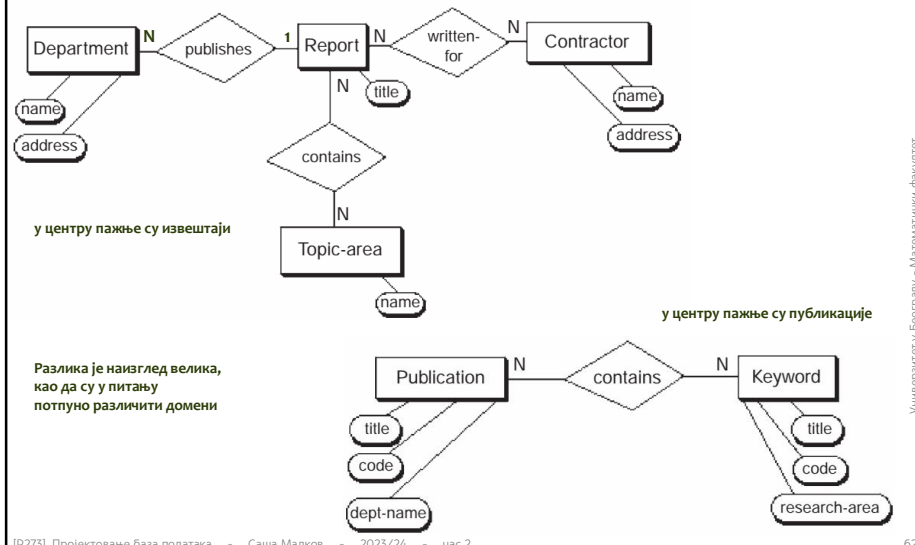
Конфликти кључева

- Истом ентитету су у различитим погледима додељени различити кључеви
- Неопходно је да се кључеви уједначе
- У супротном прети опасност од редундантних кључева

Конфликти зависности

- Исти односи се у различитим погледима моделирају на различите начине
 - различите врсте односа (слаби и јаки ентитетски односи, агрегација, асоцијација и слично)
 - различите кардиналности
 - могу да буду и суштински различити односи
- Решавају се уједначавањем карактеристика односа
 - у неким случајевима и додавањем нових односа или чак мењањем структуре

Пример 3.1, већи концептуални конфликт



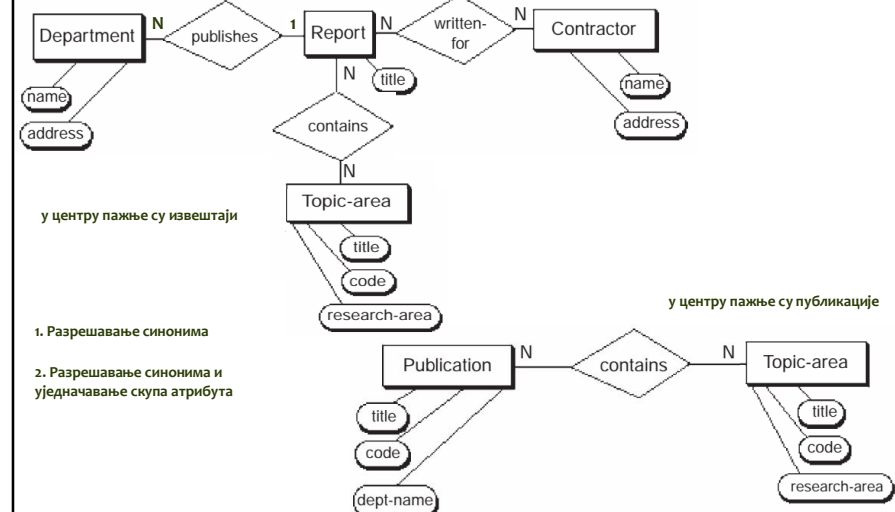
Разрешавање конфликта

- Може да захтева додатну или поновљену анализу захтева
- Може да захтева озбиљно модификовање погледа
- Добро је да учествују пројектанти конфликтних погледа
- Технике разрешавања конфликта су исте као и за прављење концептуалног модела

Разрешавање конфликта (2)

- На пример, у моделу 3.1 се у два погледа користе различити термини за исту ствар (конфликт синонима):
 - *Topic-area* и *Keyword*
 - штавише, имају и различите атрибуте (конфликт структуре)
- У моделу 3.2 (на наредној страни) је овај конфликт решен:
 - у оба модела се користи назив *Topic-area*
 - задржани су називи атрибута из модела публикација

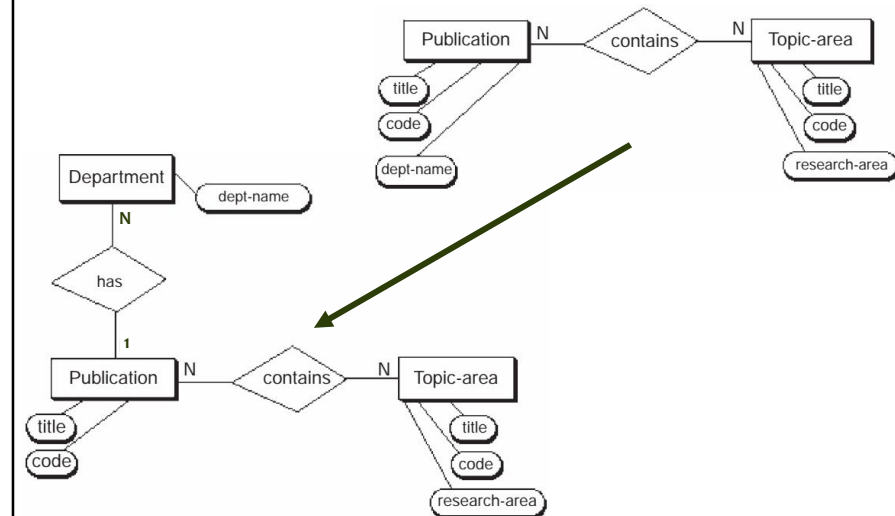
Пример 3.2, већи концептуални конфликт



Разрешавање конфликта (3)

- Даље, видимо да се у публикацији департман помиње само кроз атрибут *dept-name* а у извештајима представља ентитет
- То је структурни конфликт – потребно је да се уведе одговарајући ентитет који је у односу са публикацијом

Пример 3.3, већи концептуални конфликт





Спајање и реструктурирање схема

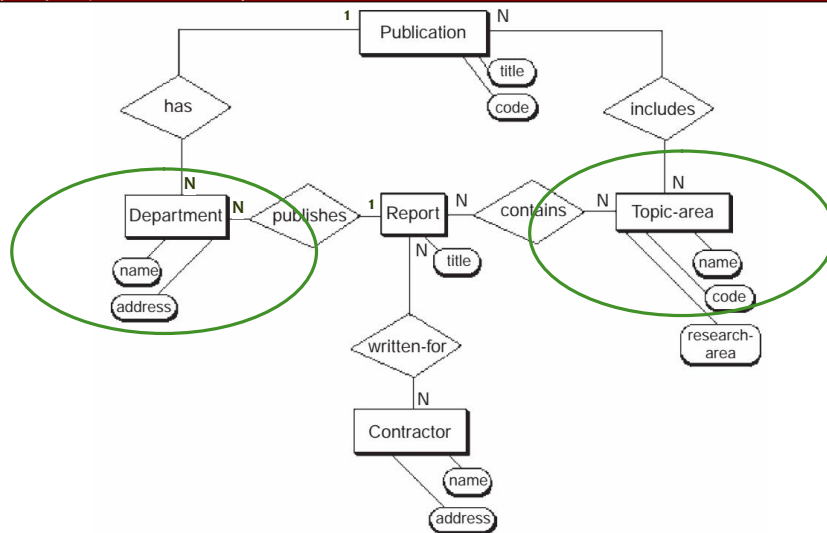
- Наступа након усаглашавања конфликта
- Руководи се принципима
 - **потпуности** – сви концепти из локалних схема морају да се очувани и у потпуности пренесу у глобалну схему
 - **минималности** – сви редундантни концепти (и ентитети и односи) морају да се уклоне из глобалне схеме
 - **разумљивости** – глобална схема мора да буде разумљива свим корисницима



Принцип потпуности

- Након разрешавања конфликта најпре спајамо све одговарајуће концепте и остварујемо потпуност

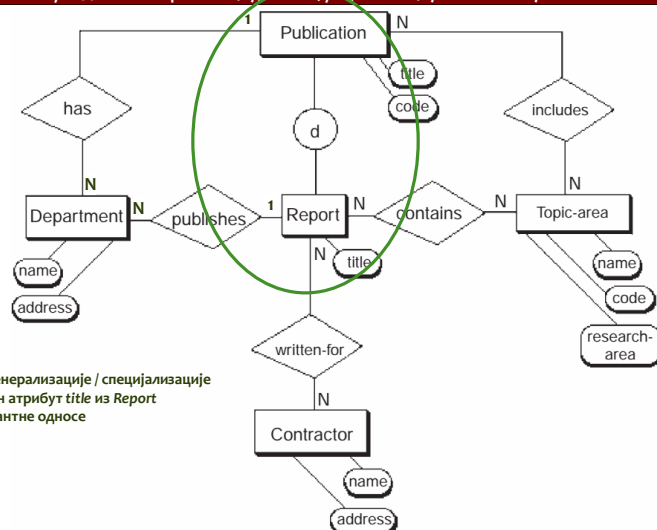
Пример спајања схема – потпуно, али не и минимално



Минималност

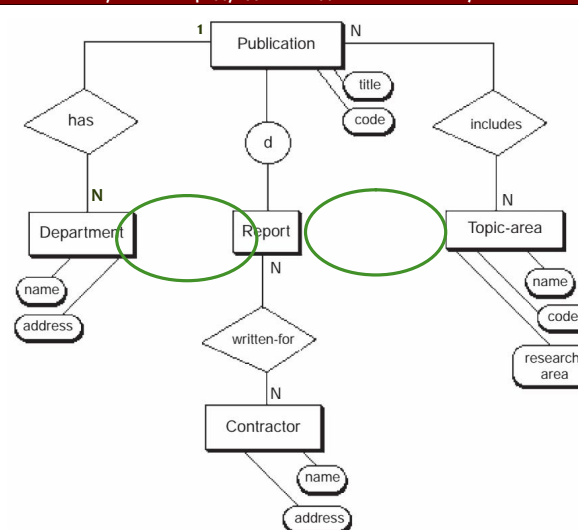
- Препознајемо редундантне концепте
- Покушавамо да их решимо
 - увођењем генерализације
 - уклањањем редундантних односа

Пример спајања схема – уводимо генерализацију између публикације и извештаја



1. Уводимо однос генерализације / специјализације
2. Бришемо сувишан атрибут title из Report
3. Бришемо редундантне односе

Пример спајања схема – уклањамо редундантне односе са извештајем

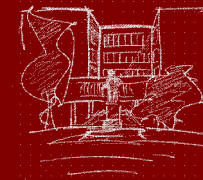


Концептуално моделирање – Интегрисање погледа

Разумљивост

- По потреби додатно уређујемо модел да би био разумљивији

[P273]
Пројектовање база података
 Саша Малков



Тема 4.4
Концептуално моделирање
 -
Груписање ентитета

- На наредном часу...
- Пре тога ћемо прво да детаљније упознамо модел ентитета и односа



- Teorey, Lightstone, Nadeau, Jagadish, **Database Modeling and Design, 5.ed, Elsevier, 2011.**