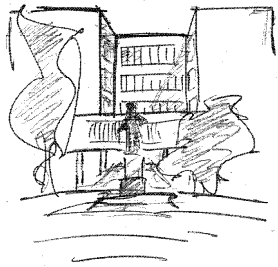


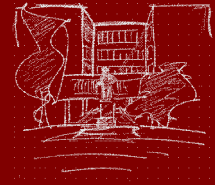
[P273]  
Пројектовање база података

5



Саша Малков  
Универзитет у Београду  
Математички факултет  
2023/2024

[P273]  
Пројектовање база података  
Саша Малков



Тема 8  
Логичко моделирање

[P273] Пројектовање база података - Саша Малков - 2023/24 - час 5

1

Логичко моделирање

## Логичко моделирање



- Концептуални модел је фокусиран на семантику и на односе
  - у великој мери не зависи од врсте база података која ће бити употребљавана у имплементацији
- Логички модел се спушта ближе изабраном имплементационом моделу
  - узима у обзир модел података који ће се употребљавати у имплементацији
- Апстрактни концепти концептуалног модела се преводе у конкретне логичке моделе
  - На пример, у случају релационог модела, хијерархије класа из концептуалног модела се замењују одговарајућим скуповима релација

Универзитет у Београду - Математички факултет

[P273] Пројектовање база података - Саша Малков - 2023/24 - час 5

2

Логичко моделирање

## Логички модел



- Предуслов за започињање логичког моделирања је одабирање врсте база података, тј. модела података
  - Логички модел узима у обзир изабрани модел података али то не значи да је логички модел исто што и физички модел
  - Логичка схема ће претрпети још додатних измена када се изабере конкретан СУБП и када се направи физички модел
- Улаз за поступак логичког моделирања је концептуални модел
- Излаз је детаљан логички (нпр. релациони) модел
  - логичка схема свих трајних објеката (релација)
  - спецификација свих услова и ограничења (*constraints*)
  - познати су сви кључеви и одговарајући сурогат-атрибути
  - као и начини имплементације свих односа

Универзитет у Београду - Математички факултет

[P273] Пројектовање база података - Саша Малков - 2023/24 - час 5

3



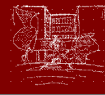
## Улога логичког модела

- Логички модел узима у обзир
  - све специфичности изабраног циљног модела података
  - додатне нефункционалне захтеве, који су у фази концептуалног пројектовања можда могли да остану занемарени
- Води се рачуна о
  - флексибилности
  - проширивости
  - али не на уштрб јасноће и чистоће модела
- Логички модел тачно одређује који се подаци чувају и како ће се њима руковати



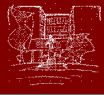
## Прескакање логичког модела

- Пројектанти су често у искушењу да после концептуалног модела одмах праве физички модел
  - Или се прескаче логички модел, или се заправо прескаче концептуални и одмах прави логички
  - Врло често је већ при прављењу концептуалног модела познато која имплементација СУБП ће се користити, па може да изгледа да такво прескакање не представља проблем
- Такав приступ носи ризик да се изгубе неке од пожељних карактеристика базе података:
  - комплетност
  - интегритет
  - флексибилност
  - ефикасност
  - употребљивост



## Прескакање логичког модела (2)

- Дobar концептуални модел је изузетно важан први корак за достизање ових карактеристика
- Али није и довољан
- Ако се концептуални модел преведе непосредно у физички модел, онда могу да се изгубе комплетност и интегритет, али и друге карактеристике



## Прескакање логичког модела (3)

- Мање базе података могу да се добро испројектују и без логичког (и/или концептуалног) модела, али код већих база такав приступ обично води проблемима и при пројектовању и при одржавању
- За одговоре на нека питања је врло често потребно имати сва три модела
- На пример, при променама структуре базе података често се питамо зашто је нешто у физичком моделу направљено баш тако а не другачије? Да ли је то природан модел или оптимизација?
  - Одговор се често може добити само анализом сва три модела.



## Итеративно моделирање

- Логички модел се прави итеративно
  - Прва итерација се прави на основу концептуалног модела
    - “превођење КМ у ЛМ”
  - Свака наредна се прави мењањем претходне (енгл. *flexing*)
    - “пречишћавање схеме”



## Критеријуми за унапређивање итерација

- Критеријуми за мењање су жељене карактеристике:
  - Да ли модел испуњава функционалне захтеве?
  - Да ли модел испуњава нефункционалне захтеве?
  - Да ли је модел комплетан?
  - Да ли модел гарантује интегритет података?
  - Да ли модел пружа потребну флексибилност?
  - Да ли модел омогућава ефикасан рад?
  - Да ли је модел употребљив?
  - Да ли је модел у складу са циљним моделом података?
- Неки од критеријума су “наслеђени” из концептуалног модела, али неки су нови



## Значајне одлуке

- Логичко моделирање може да има за резултат и неке значајне одлуке, нпр:
  - може да се не прави једна него више база података
  - може да се структура прилагоди очекиваним променама (или очекиваној врсти промена)
  - могу да се доносе и неке одлуке у складу са специфичностима изабране имплементације СУБП
  - некада не могу да се остваре све тражене карактеристике, па је неопходно одлучити која је у конкретном случају важнија



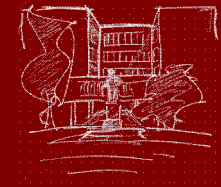
## Релациони модел као основ логичког модела

- У пракси се најчешће пројектују и имплементирају релационе базе података
  - Даљу обраду ове теме настављамо уз претпоставку да је циљни модел управо релациони
  - Логички модел ћемо да представљамо језиком релационог модела
    - Ентитети се представљају као релације
    - Сложени односи се представљају као релације
    - Једноставни односи се представљају страним кључевима
- У случају да имамо неки други циљни модел података (нпр. скуп каталога кључ-вредност), већина наведеног и даље важи али у прилагођеном облику



## Релациони модел као основ логичког модела (2)

- Најпре ћемо се подсетити дијаграма табела
- Затим ћемо детаљно обрадити превођење концептуалног модела на логички модел, на примеру релационог модела
- На крају ћемо детаљно обрадити пречишћавање схеме, на примеру релационог модела



## Пројектовање база података

Саша Малков

### Тема 8.1

## Логичко моделирање - Дијаграми табела



## Дијаграми табела

- Често се погрешно називају „ЕР-дијаграми“
- То *нису* ЕР-дијаграми него дијаграми табела (релација)
  - ознаке и термини су обично табеле/колоне/кључеви
  - нема довољно семантичких информација
  - кардиналности се представљају другачије него на ЕР-дијаграмима
    - по узору на УМЛ
    - обично се користе само 0, 1 и \*
- Прилагођени су логичком и (посебно) физичком нивоу релационог модела
- Семантика односа је представљена у мери у којој то допушта релациони модел



## Дијаграми табела (2)

- Табеле се представљају правоугаоницима
  - пожељно са заобљеним угловима
- Односи међу табелама се представљају линијама

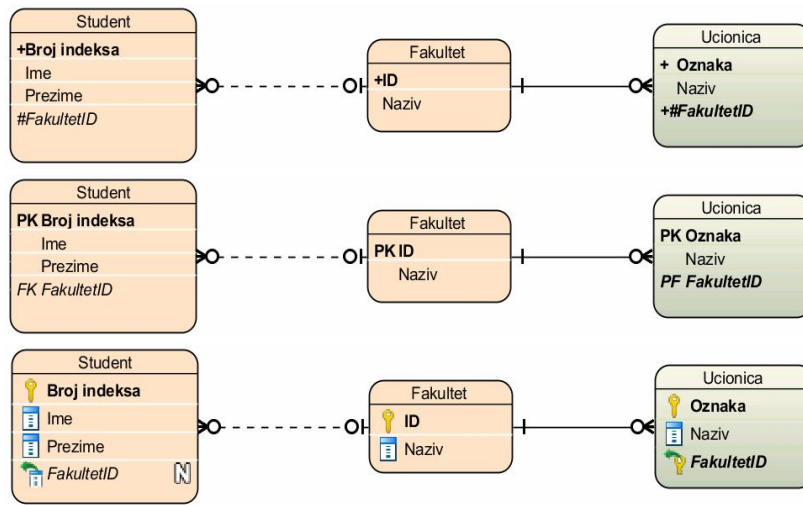
## Табеле

- Табеле се представљају правоугаонцима
  - обично заобљеним
- Табела је хоризонталном линијом подељена на два дела
  - горњи део садржи назив табеле
  - доњи део садржи описе колона
  - у неким варијантама поред описа колона могу да се наведу и друге информације

## Табеле (2)

- Колоне су посебно означене симболима, словним ознакама или цртежима
  - ако колона припада примарном кључу
    - симбол +
    - ознака PK
    - цртеж кључа
  - ако колона припада страном кључу
    - симбол #
    - ознака FK (или PF, ако је и примарни и страни, тзв. слаби ентитет)
    - цртеж кључа са стрелицом
    - обично и искошен назив
  - ако колона може да садржи недефинисану вредност
    - симбол, цртеж или ознака који асоцирају на слово N

## Табеле (примери)



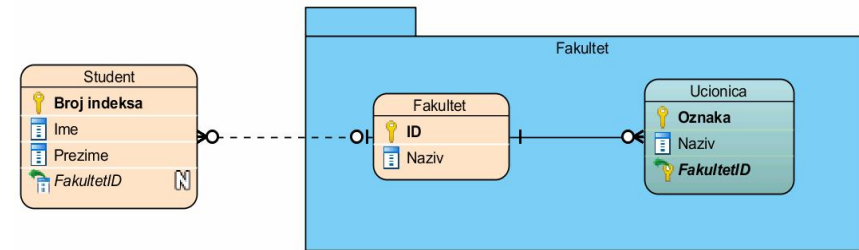
## Врсте табела

- Неки алати и аутори разликују табеле према боји
  - једном бојом јаки ентитети
  - другом бојом слаби ентитети
  - трећом бојом односи
  - може да се уведе и више различитих боја, према контексту
- Неки алати и аутори означавају табеле различитим симболима или цртежима у десном горњем углу
  - слично као стереотипови у УМЛ-у
  - обично се користи посебна ознака за погледе
- Обе праксе су добре и олакшавају разумевање модела

## Груписање

- За груписање табела се користе правоугаоници који личе на фасцикле
  - као пакети у УМЛ-у
- Могу да се користе и боје, али то може да прави проблем у односу на означавање врста табела

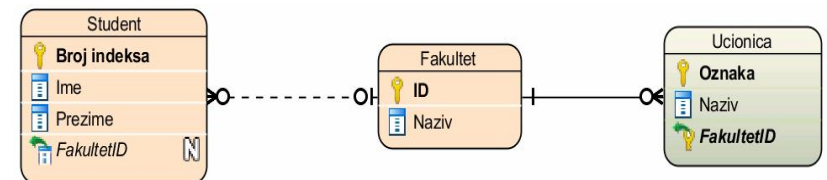
## Груписање (пример)



## Односи

- Основна врста “односа” су страни кључеви и односи више-више
- Обично се представљају испрекиданим линијама
  - пуна линија представља однос слабог ентитета са матичним
- На крају линије се представља кардиналност
  - у духу УМЛ-а
    - број ентитета уз који стоји ознака, који могу да буду у односу са једним ентитетом који је на другој страни
  - графичке ознаке:
    - 0 – кружић
    - 1 – управна линија
    - \* - линија се грана у три кратке линије према табели
    - 0-1, 0-\*, 1-\* – комбинације претходних случајева

## Односи (примери)



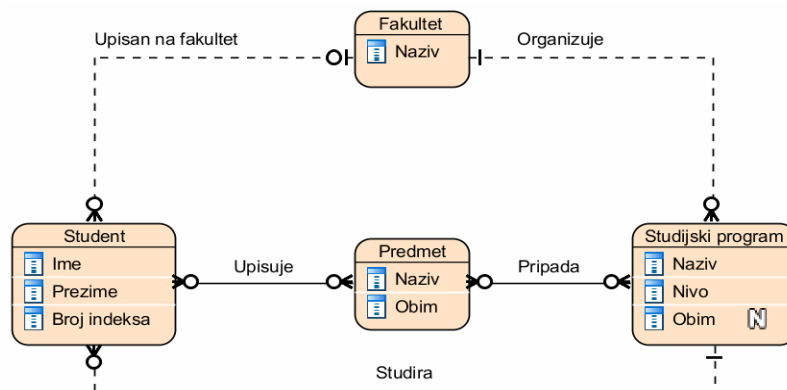
## Ниво модела

- Основни облик дијаграма највише одговара логичком нивоу
  - На концептуалном нивоу се обично прилагођава
    - не препоручује се употреба, осим за једноставне базе података
  - На логичком нивоу се обично користи у основном облику али може и да се прилагођава
  - На физичком нивоу се користи у основном облику али могу да се додају и нови елементи за описивање елемената физичке имплементације

## Концептуални ниво

- Дијаграм је поједностављен
  - садржи само основну структуру података
- Дијаграм уобичајено не садржи:
  - додатне колоне сурогат кључева
  - додатне колоне страних кључева
  - ознаке кључева (осим можда једног кандидат-кључа)
  - типове колона
- Овакав поједностављен дијаграм се понекад користи и на логичком нивоу
  - мада би требало да буде потпунији

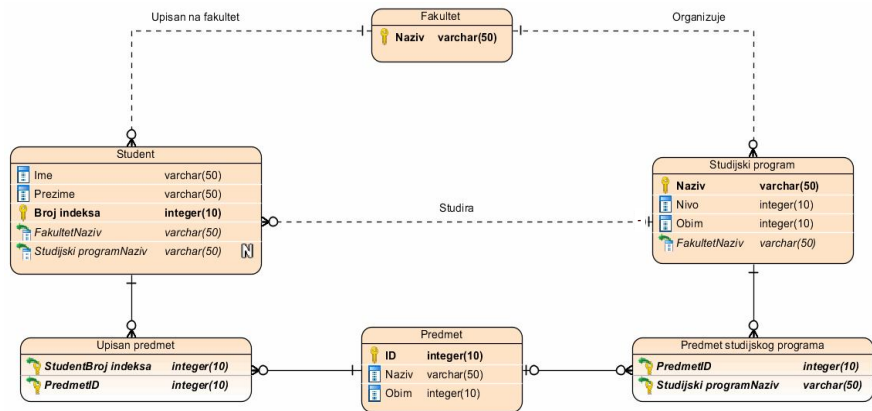
## Концептуални ниво (пример)



## Логички ниво

- Додају се сви остали подаци неопходни за пресликавање у релациони модел:
  - колоне које су сурогат кључеви (ако су потребне)
  - колоне везаних табела који чине стране кључеве
  - ознаке кључева и врста кључева
  - типови колона
  - односи више-више се замењују табелама које их моделирају
  - правила интегритета

## Логички ниво (пример)

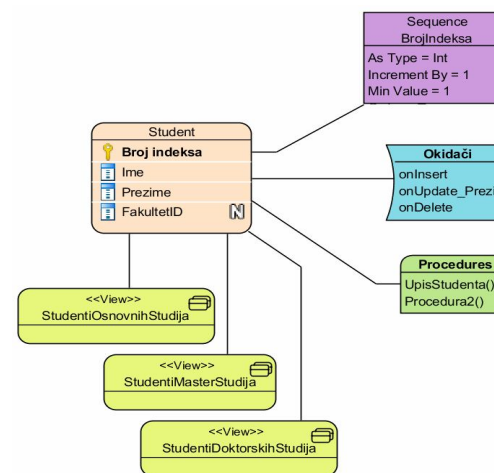


## Физички ниво

- Додају се и елементи физичке структуре:
  - погледи
  - индекси
  - окидачи
  - простори за табеле
  - ...

## Додатни елементи

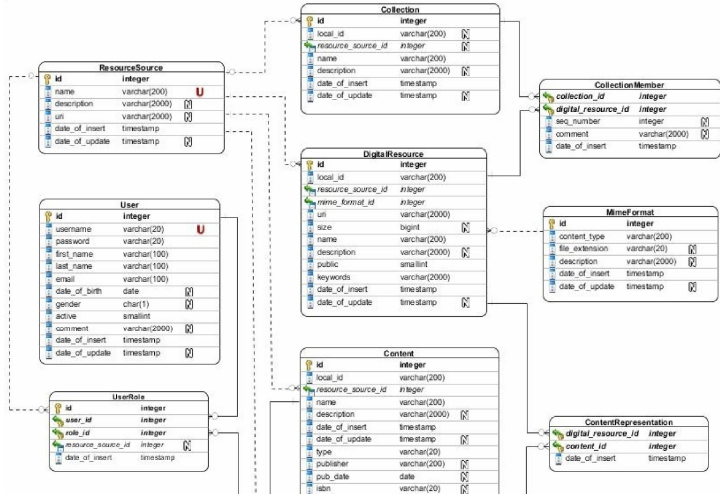
- Додатни елементи омогућавају прецизније описивање
  - Користе се примарно на физичком нивоу, али неки могу и на логичком
- погледи
  - обично правоугаоници означени додатним симболом и везани линијом са табелама на основу којих се дефинишу
  - ако допуштају мењање података може и додатно означавање
- секвенце
- окидачи
- услови интегритета
  - додају се на различите начине, обично у виду коментара
- простори табела и други елементи физичког модела
- индекси
- уграђене процедуре
- и друго



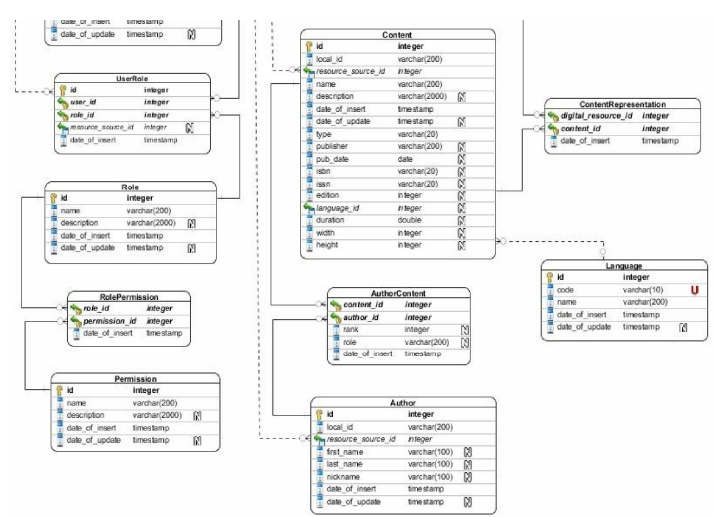




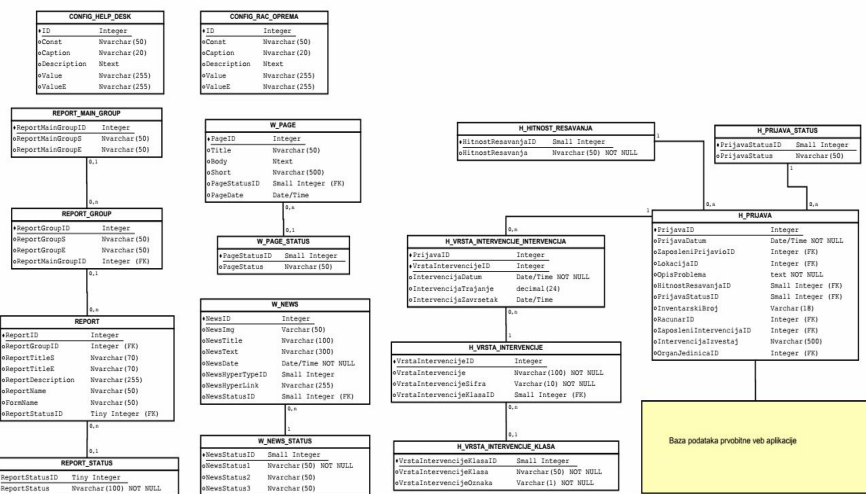
Пример 2а – Регистар дигиталних ресурса



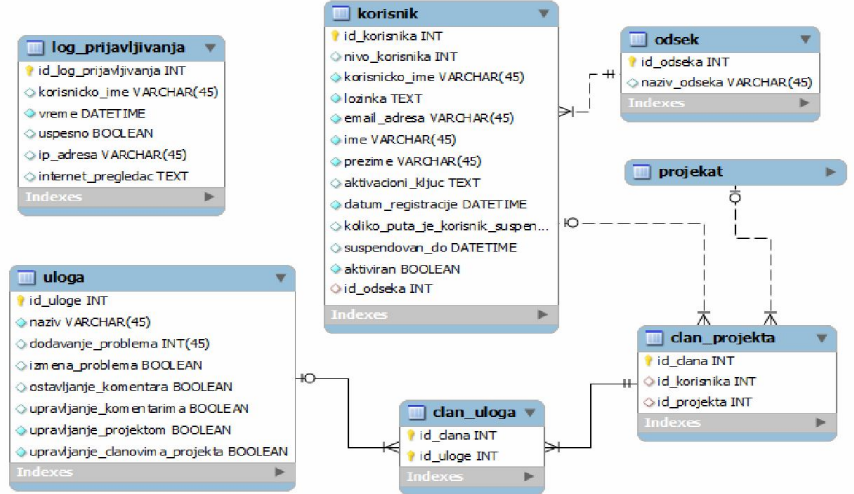
Пример 2б – Регистар дигиталних ресурса



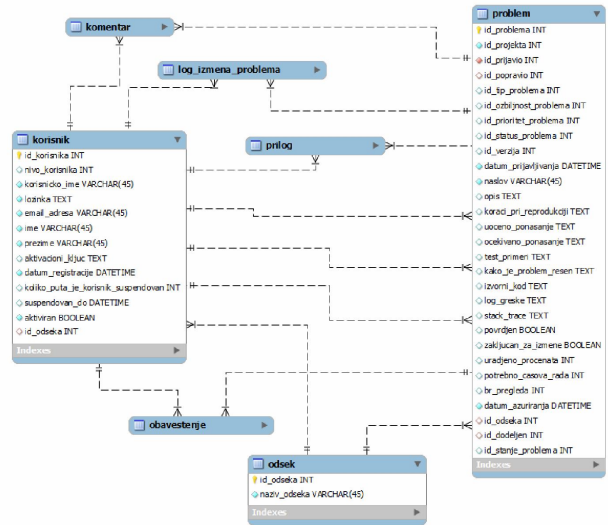
Пример 3 – Решавање хардверских проблема



Пример 4 – Систем за праћење активности

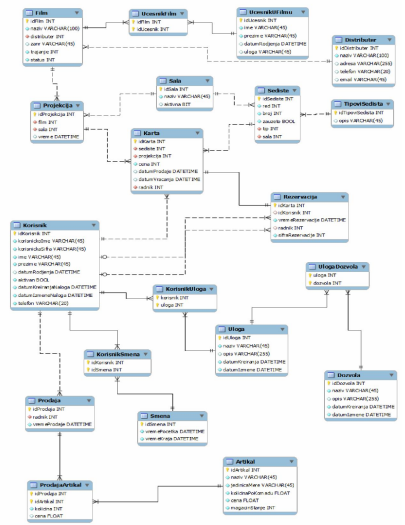


Пример 5 – Систем за праћење активности



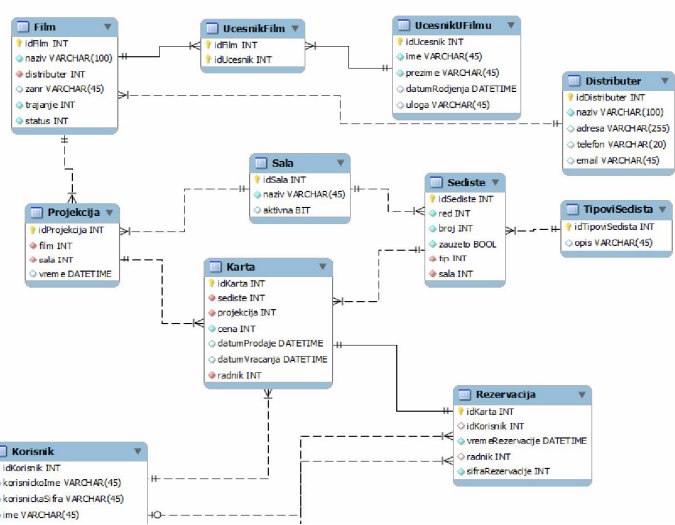
Универзитет у Београду – Математички факултет

Пример 6 – ИС биоскопа



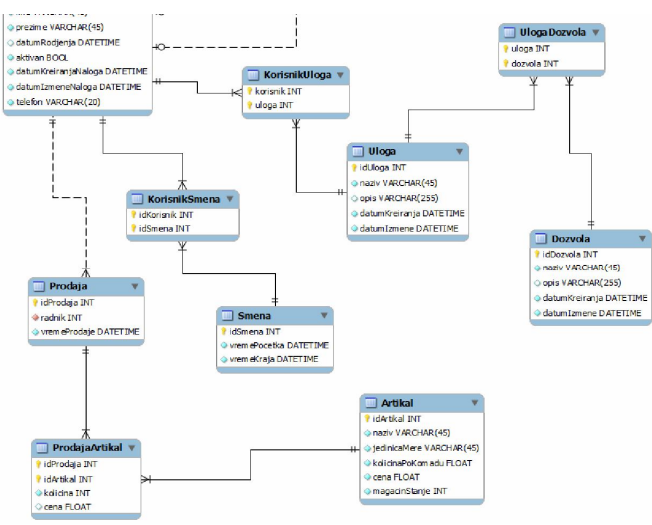
Универзитет у Београду – Математички факултет

Пример 6а – ИС биоскопа



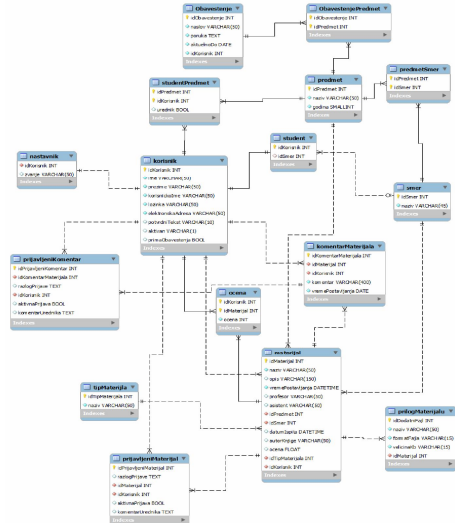
Универзитет у Београду – Математички факултет

Пример 6б – ИС биоскопа



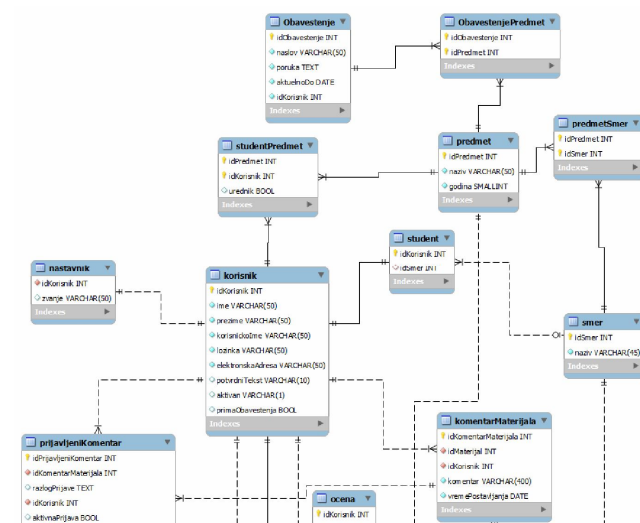
Универзитет у Београду – Математички факултет

Пример 7 – Збирка помоћних наставних материјала



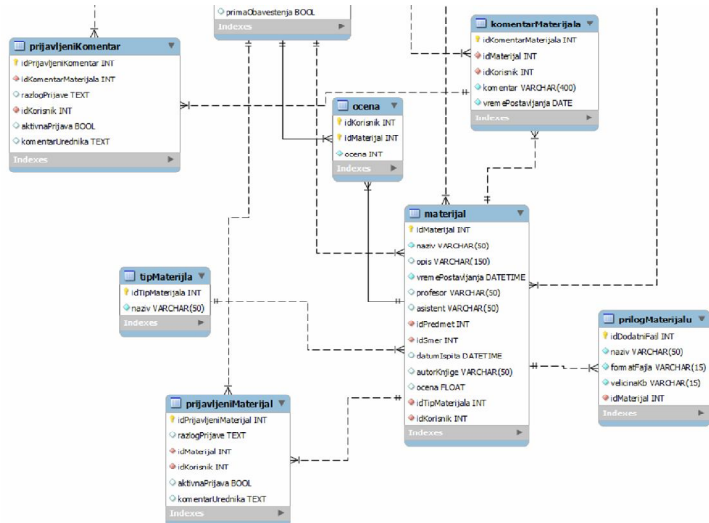
Универзитет у Београду – Математички факултет

Пример 7а – Збирка помоћних наставних материјала



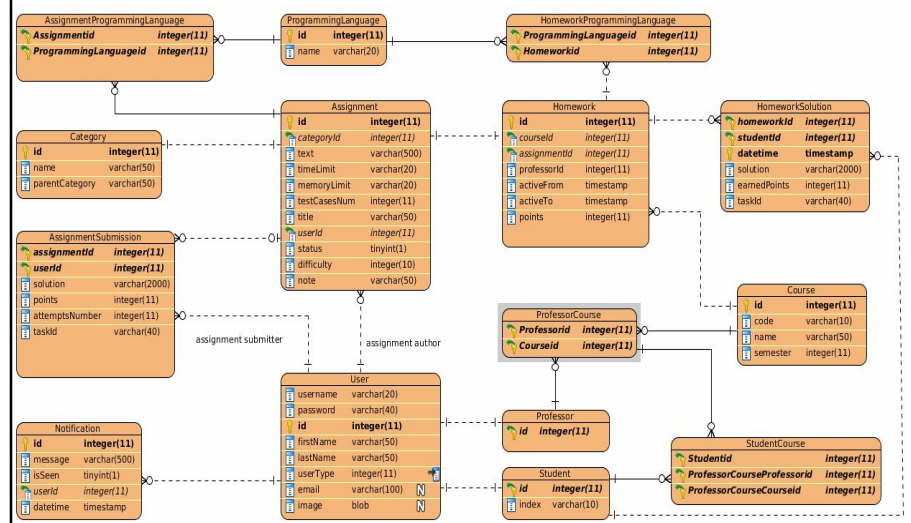
Универзитет у Београду – Математички факултет

Пример 7а – Збирка помоћних наставних материјала



Универзитет у Београду – Математички факултет

Пример 8 – Систем за аутоматско тестирање решења програмских задатака

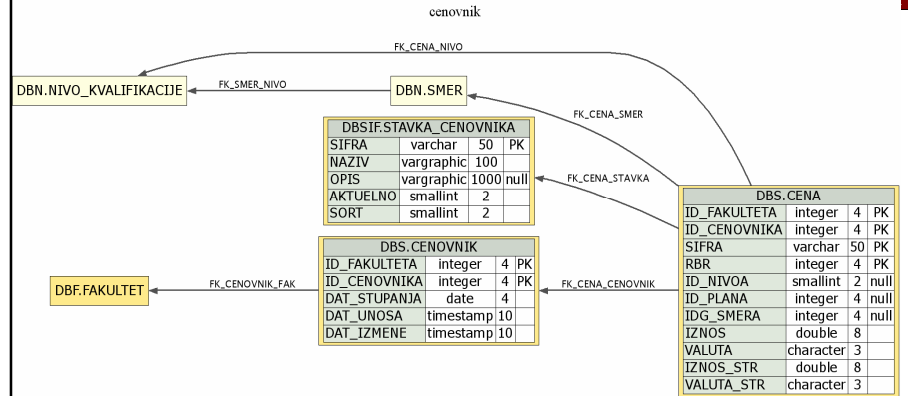


Универзитет у Београду – Математички факултет

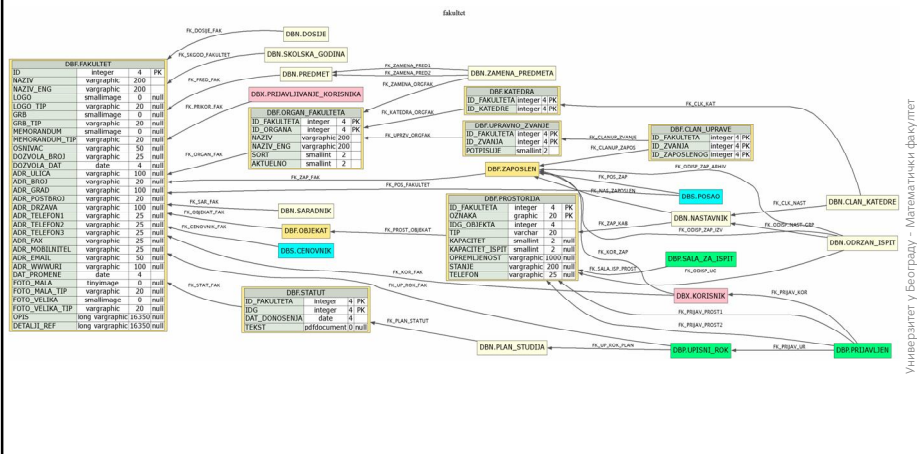
## Модификовани дијаграми

- Неки алати доносе модификације
  - модификације често имају за циљ приближавање ОО-моделу
  - на пример, страни кључеви као стрелице у смеру реферисања
- модификације су начелно допуштене
  - ако су промене интуитивно јасне
  - док постоји доследност у оквиру пројекта

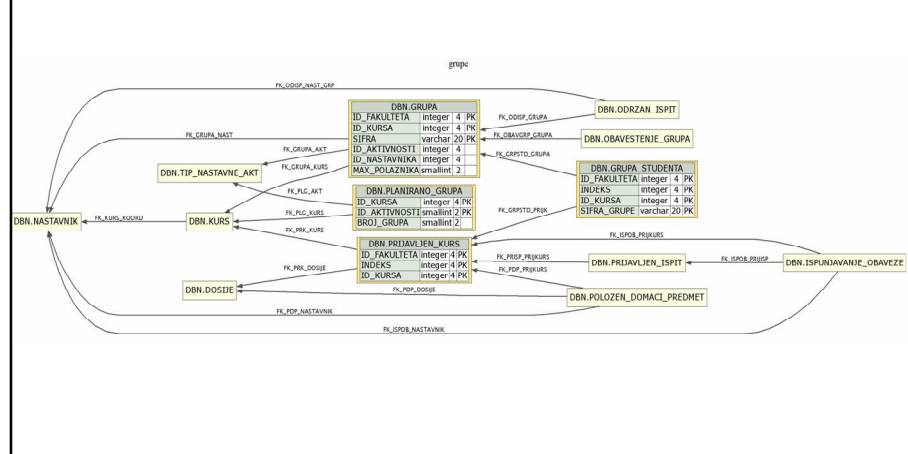
## Пример 9а - Аутоматски генерисан дијаграм дела модела базе података система Студинфо



## Пример 9б - Аутоматски генерисан дијаграм дела модела базе система Студинфо



## Пример 9ц - Аутоматски генерисан дијаграм дела модела базе система Студинфо



## Литература за тему

- Molina, **Database Systems**, 2.ed, *Pearson Prentice Hall* (2009)
- Teorey et al., **Database Design**, *Morgan Kaufmann* (2009)
- Muller, **Database Design for Smarties – Using UML for Data modeling**, *Academinc Press* (1999)
- Alvaro Monge, **Database design with UML and SQL**, 4.ed.
  - <https://web.csulb.edu/colleges/coe/cecs/dbdesign/dbdesign.php>
- Алати за моделирање, препоручује се:
  - Visual Paradigm Suite
    - <http://www.visual-paradigm.com/>
    - бесплатна верзија *Community Edition*

## Пројектовање база података

[P273]

Саша Малков

Тема 8.2

### Логичко моделирање

-

### Превођење концептуалног модела у логички

Логичко моделирање – Превођење концептуалног модела у логички

## Превођење конц. у логички модел

- Први корак (прва итерација) при прављењу логичког модела је превођење концептуалног модела у логички модел
  - Ако се оба модела изражавају на ЕР-у или УМЛ-у, онда “превођење” може да постане “трансформисање” или “доуњавање и надограђивање”
  - У савременој пракси је уобичајено да се логички модели описују дијаграмима табела или евентуално дијаграмима класа

Логичко моделирање – Превођење концептуалног модела у логички

## Превођење конц. у логички модел (2)

- Као што је већ наглашено, у овој теми ћемо се посветити релационим базама података
- Покушаћемо да превођење опишемо у форми скупа правила
- Постоје мање разлике у поступку у зависности од тога да ли је КМ прављен коришћењем модела ЕР или ОО



## Превођење ЕРМ на РМ

- Превођење ЕР-модела на релациони модел
  - неки аспекти превођења су “правoliniјски”
  - неки други нису
  - може да зависи и од семантике и од контекста
- Неки случајеви су једноставни:
  - Сваки ентитет се преводи у релацију
  - Атрибути који представљају скупове података превод се у релације
  - ...
  - ...једноставних случајева је највише...
- Али неки случајеви су сложенији:
  - ...сложенијих случајева је мање, али су од пресудне важности...



## Поступак превођења КМ у РМ

- Поступак превођења почиње превођењем ентитета
  - успут могу да се преведу и неки односи, обично 1-\*, који се непосредно превод у стране кључеве
- Затим се наставља превођењем односа
  - најпре се превод односи који производе релације
    - сложени односи
    - бинарни односи \*-\*
  - затим се превод преостали односи
  - на крају се дотерује превод односа
- Најпре ћемо описати основну идеју превођења, а затим размотрити различите појединачне случајеве



## Превођење ентитета у релације

- Сваки ентитет се преводи у релацију
  - препознаје се примарни кључ
  - превод се кључни атрибути или се уводе нови сурогат кључеви
  - превод се сви некључни атрибути
- Ентитети који чине хијерархије се обично најпре превод у релације на исти начин као остали ентитети
  - касније ћемо видети шта се и како ради са њима
- Слаб ентитет и одговарајући однос се обично моделирају једном релацијом
  - релацији се додаје страни кључн према везаном јаком ентитету
  - више о томе мало касније



## Превођење атрибута

- Скаларни атрибути ентитета се превод у атрибуте релација
- Сложени атрибути се превод у више скаларних атрибута
  - Евентуално у релације, али се то обично оставља за другу фазу
- Атрибути са више вредности се превод у релације



## Превођење односа

- Односи могу де се преведу на два начина:
  - у страни кључ или
  - у нову релацију
- Код превођења односа мора да се узме у обзир природа релационог модела
- Односи се преводе у зависности од кардиналности и врсте односа:
  - Бинарни односи 1-\*
  - Бинарни односи \*-\*
  - Сложени односи
  - Генерализација



## Представљање односа страним кључем

- У релационом моделу, основно средство за моделирање односа су страни кључеви
  - Страни кључ из “везане” релације Б према “базној” релацији А има врло ограничену семантику кардиналности:
    - кард. А је уобичајено 0..1, ако кључ може да садржи *NULL*
    - кард. А може да буде 1, ако кључ не сме да садржи *NULL*
    - кард. Б је уобичајено 0..\*
    - кард. Б може да буде 0..1, ако се дода услов јединствености кључа уз допуштено понављање вредности *NULL*
- Све остале варијанте захтевају постављање додатних услова и ограничења
  - а често и додавање нових релација



## Превођење бинарних односа 1-\*, 0..1-\*

- Овакве односе практично непосредно моделира страни кључ:
  - 0..1-\*, страни кључ може да садржи *NULL*
  - 1-\*, страни кључ не сме да садржи *NULL*
- Такви односи се често преводе већ у фази превођења ентитета
  - Ако ентитет учествује у односима 1-\* као слабији учесник (у ЕР-моделу као учесник на страни 1, или у моделу класа као учесник на страни \*, или чак као експлицитно означен слаб ентитет), у преведену релацију се додају атрибути кључа друге релације као страни кључ
  - Ако однос има додатне атрибуте, они се додају релацији заједно са атрибутима страног кључа
    - ако страни кључ може да садржи *NULL*, онда то морају да могу и ови атрибути
    - штавише, ако страни кључ садржи *NULL*, онда и сви додатни атрибути морају да буду *NULL*
- (Ово ћемо прецизније разградити у наставку)



## Превођење бинарних односа \*-\*

- Сваки бинарни однос \*-\* се преводи у релацију која као стране кључеве садржи кључеве ентитета који учествују у односу
  - Ако однос има додатне атрибуте, ти атрибути се преводе у атрибуте нове релације
- Мора да се води рачуна о усклађивању ограничења за примарне и стране кључеве са ф.зависностима
- (Ово ћемо прецизније разградити у наставку)





## Превођење односа са 3 или више учесника

- Сложени односи се преводе слично бинарним односима \*-\*
  - Сваки однос се преводи у релацију која као стране кључеве садржи кључеве релација које су преводи ентитета који учествују у односу
- Додају се јединствени кључеви у складу са ф.зависностима
- Води се рачуна о усклађивању ограничења за примарне и стране кључеве са ф.зависностима



## Превођење атрибута односа

- Ако се однос преводи у један страни кључ
  - атрибути односа се додају релацији у којој се прави страни кључ
  - ако атрибути страног кључа могу да буду *NULL* онда и остали атрибути пресликани из односа морају да могу да буду *NULL*
- Ако се однос преводи у више страних кључева
  - (т.ј. у стране кључеве у обе релације, биће више речи о томе)
  - атрибути се додају у једну од њих
- Ако се однос преводи у нову релацију
  - атрибути односа се додају тој новој релацији
  - опциони атрибути из КМ морају да могу да буду *NULL*



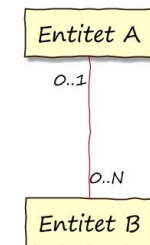
## Разматрање различитих случајева

- У наставку ћемо размотрити различите врсте односа и њихово превођење
- Примере концептуалног модела ћемо представљати (углавном) коришћењем дијаграма класа домена, али поступак је практично исти и у случају дијаграма ЕР



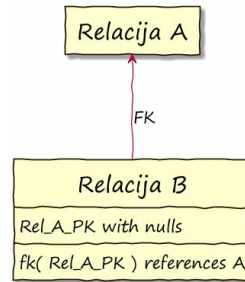
## Бинарни односи 0..1 – 0..N

- Уобичајено за односе
  - *родитељ – њошмак*
  - *целина – гео*
  - ако *целина* **може** да буде без *делова*
  - ако *гео* **може** да буде без *целине*
- Чест случај код агрегација



## Бинарни односи 0..1 – 0..N (2)

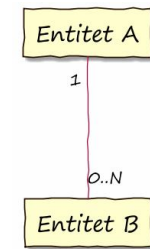
- Релацији која је *geo*
  - додаје се страни кључ у односу на *целину*
  - страни кључ **може** да буде недефинисан



Универзитет у Београду - Математички факултет

## Бинарни односи 1 – 0..N

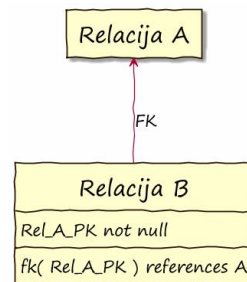
- Уобичајено за односе
  - *родитељ – њошмак*
  - *целина – geo*
  - ако *целина може* да буде без *делова*
  - ако *geo не може* да буде без *целине*
- Чест случај код композиција



Универзитет у Београду - Математички факултет

## Бинарни односи 1 – 0..N (2)

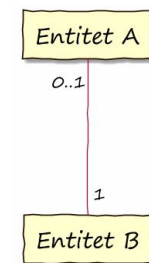
- Релацији која је *geo*
  - додаје се страни кључ у односу на *целину*
  - страни кључ **не сме** да буде недефинисан



Универзитет у Београду - Математички факултет

## Бинарни односи 0..1 – 1

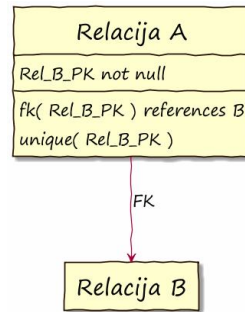
- Уобичајено за односе
  - *нагређени – њогређени*
  - *целина – ојциони geo*
- Пример:
  - Руководилац одељења А је службеник Б



Универзитет у Београду - Математички факултет

## Бинарни односи 0..1 – 1 (2)

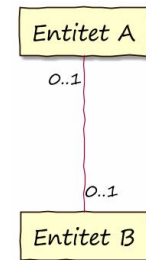
- Релација која је *опционо дефинисана*
  - додаје се страни кључ у односу на *целину*
  - страни кључ **не сме** да буде недефинисан
  - вредност кључа мора да буде јединствена



Универзитет у Београду - Математички факултет

## Бинарни односи 0..1 – 0..1

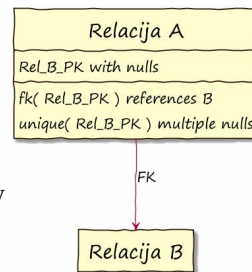
- Уобичајено за двосмерне опционе односе
- Пример:
  - Пројекат А може да има највише једног главног програмера Б
  - Програмер Б може да буде главни програмер на највише једном пројекту



Универзитет у Београду - Математички факултет

## Бинарни односи 0..1 – 0..1 (2)

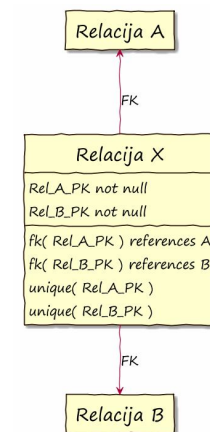
- Једној од релација
  - додаје се страни кључ у односу на другу
  - страни кључ **може** да буде недефинисан
  - вредност кључа мора да буде јединствена
    - недефинисане вредности смеју да се понављају (то је проблем код неких система)



Универзитет у Београду - Математички факултет

## Бинарни односи 0..1 – 0..1 (3)

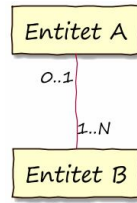
- ...алтернатива...
- Прави се нова *везна* релација
  - садржи само стране кључеве у односу на обе везане релације
  - кључеве **не смеју** да буду недефинисани
  - сви атрибути чине примарни кључ
  - вредност сваког од страних кључева мора да буде јединствена
- Решење са везном релацијом има предност
  - ако је овај однос релативно редак
  - ако је важно поштовање јединствености а СУБП не омогућава јединствене кључеве са вишеструким недефинисаним вредностима



Универзитет у Београду - Математички факултет

## Бинарни односи 0..1 – 1..N

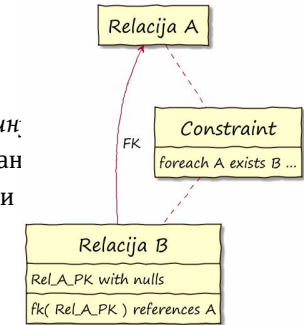
- Уобичајено за односе
  - агрегације са обавезним деловима
- Пример:
  - Свако одељење А има најмање једно додељено паркинг-место Б
  - Свако паркинг-место Б може да се додели највише једном одељењу А



Универзитет у Београду - Математички факултет

## Бинарни односи 0..1 – 1..N (2)

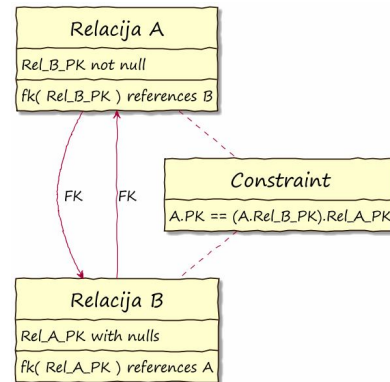
- Релацији која је *geo*
  - додаје се страни кључ у односу на *целин*
  - страни кључ **може** да буде недефинисан
  - додаје се услов базе података: “за сваки ентитет А мора да постоји бар један одговарајући ентитет Б”



Универзитет у Београду - Математички факултет

## Бинарни односи 0..1 – 1..N (3)

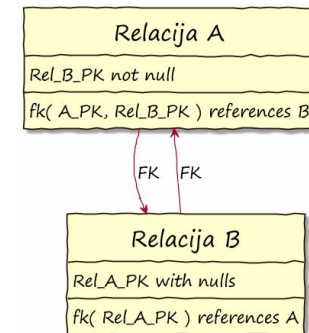
- ...алтернатива 1...
- Међу свим деловима једне целине се препознаје (често вештачки) један изабран
  - у релацију А се додаје страни кључ у односу на Б
    - целина се повезује са изабраним делом
  - олакшано је проверавање постојања “бар једног”
  - али је уведена редундантност
    - мора да се дода правило које проверава узајамност
    - изабрани део мора да припада баш тој целини



Универзитет у Београду - Математички факултет

## Бинарни односи 0..1 – 1..N (4)

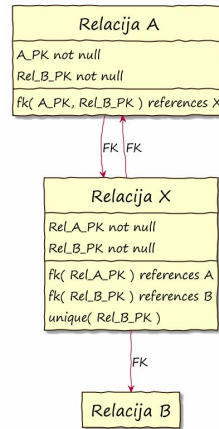
- ...алтернатива 1a...
- Веома слично као претходни случај
  - експлицитан услов се замењује проширивањем страног кључа из А према Б, тако да укључује атрибуте примарних кључева и А и Б
  - не омогућавају сви РСУБП да се реферисање по страном кључу врши према скупу атрибута који није примарни кључ



Универзитет у Београду - Математички факултет

## Бинарни односи 0..1 – 1..N (5)

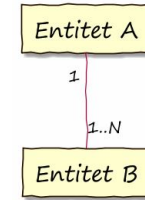
- ...алтернатива 2...
- Уводи се везна релација X
  - она имплементира 0..\* – 0..\*
  - додаје се јединствени кључ према Б
    - тиме се решава однос 0..1 – 0..\*
  - остаје да се провери да за свако А постоји бар једно Б
- У релацију А се додаје страни кључ у односу на X
  - целина се повезује са изабраним делом
  - а истовремено се проверава узајамност



Универзитету Београду - Математички факултет

## Бинарни односи 1 – 1..N

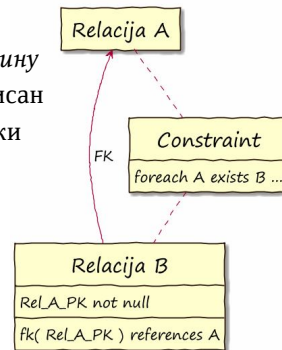
- Уобичајено за односе
  - композиције са обавезним деловима
- Пример:
  - Свако одељење А има најмање једно додељено паркинг-место Б
  - Свако паркинг-место Б припада тачно једном одељењу А



Универзитету Београду - Математички факултет

## Бинарни односи 1 – 1..N (2)

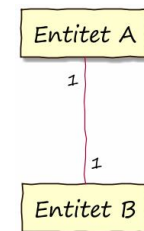
- Релацији која је *geo*
  - додаје се страни кључ у односу на *целину*
  - страни кључ **не сме** да буде недефинисан
  - додаје се услов базе података: “за сваки ентитет А мора да постоји бар један одговарајући ентитет Б”
- У свему слично као 0..1-1..N
  - па и у алтернативним решењима...



Универзитету Београду - Математички факултет

## Бинарни односи 1 – 1

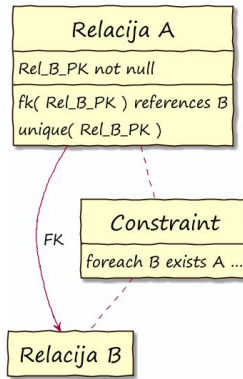
- Уобичајено за односе
  - са обостраним ексклузивним придруживањем
  - (релативно ретко, питање је да ли су то различити ентитети)



Универзитету Београду - Математички факултет

## Бинарни односи 1 – 1 (2)

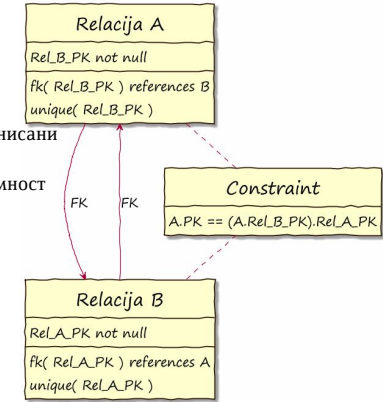
- Једној од релација
  - додаје се страни кључ у односу на другу
  - страни кључ **не сме** да буде недефинисан
  - вредност кључа мора да буде јединствена
  - додаје се услов базе података: “за сваки ентитет Б мора да постоји бар један одговарајући ентитет А”



Универзитет у Београду - Математички факултет

## Бинарни односи 1 – 1 (3)

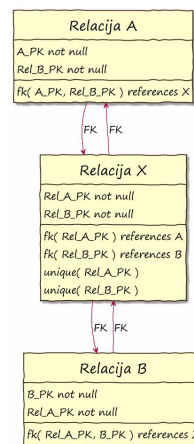
- ...алтернатива 1...
- У свакој од релација
  - додаје се страни кључ у односу на другу
  - атрибути кључа не смеју да буду недефинисани и морају да буду јединствени
  - додаје се услов којим се проверава узајамност везе
- ...алтернатива 1a...
- У релацијама се дефинише исти примарни кључ
  - онда није потребан додатни услов за проверу узајамности везе



Универзитет у Београду - Математички факултет

## Бинарни односи 1 – 1 (4)

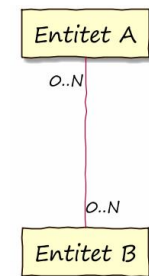
- ...алтернатива 2...
- Додаје се везна релација X
  - јединствен кључ према А
  - јединствен кључ према Б
- У свакој од релација А и Б
  - додаје се страни кључ у односу на везну релацију X
  - тиме се проверава узајамност везе
- ...и овде може алтернатива 1a, као у случају односа 0..1-1 ..N



Универзитет у Београду - Математички факултет

## Бинарни односи 0..N – 0..N

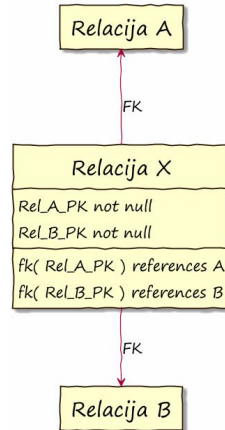
- Уобичајено за односе
  - асоцијације без посебних ограничења



Универзитет у Београду - Математички факултет

## Бинарни односи 0..N – 0..N (2)

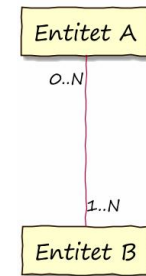
- Прави се нова *везна* релација
  - садржи само стране кључеве у односу на обе везане релације
  - кључеви **не смеју** да буду недефинисани
  - сви атрибути чине примарни кључ



Универзитет у Београду - Математички факултет

## Бинарни односи 0..N – 1..N

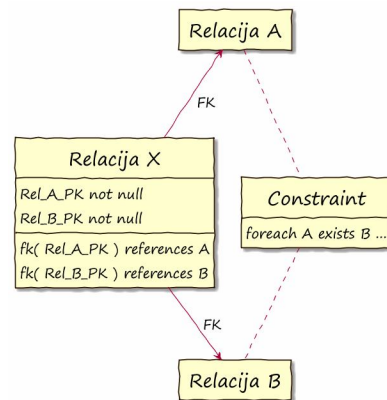
- Уобичајено за односе
  - асоцијације слабих и јаких ентитета
  - агрегације код којих *geo* може да чини више целина, али не може да постоји самостално



Универзитет у Београду - Математички факултет

## Бинарни односи 0..N – 1..N (2)

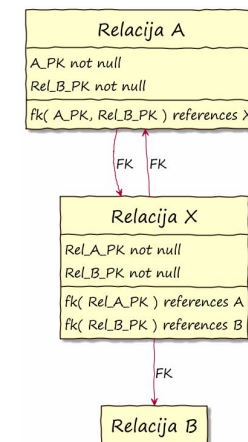
- Прави се нова *везна* релација
  - садржи само стране кључеве у односу на обе везане релације
  - кључеви **не смеју** да буду недефинисани
  - сви атрибути чине примарни кључ
  - додаје се услов базе података: “за сваки ентитет А мора да постоји бар један одговарајући ентитет Б”
- ...алтернатива...
  - у А се додаје страни кључ према изабраном Б односно X...



Универзитет у Београду - Математички факултет

## Бинарни односи 0..N – 1..N (3)

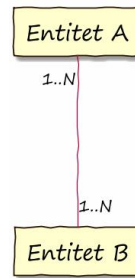
- ...алтернатива...
- У релацију А се додаје страни кључ према X
  - тако се повезује А са изабраним Б (тј. паром А,Б)
  - нису потребни додатни услови
- Нешто је сложеније ажурирање у случају промене изабраног пара...



Универзитет у Београду - Математички факултет

## Бинарни односи 1..N – 1..N

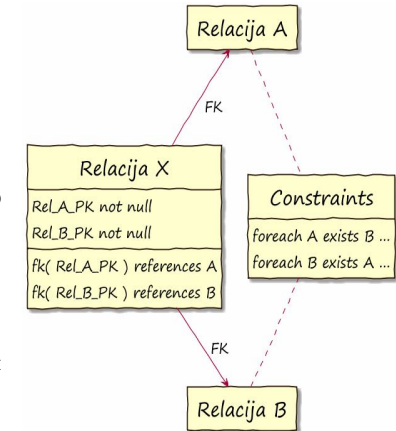
- Уобичајено за односе
- агрегације код којих
  - geo* може да чини више целина, али не може да постоји самостално
  - целина не може да постоји без *делова*
- (релативно ретко се користи)



Универзитет у Београду - Математички факултет

## Бинарни односи 1..N – 1..N (2)

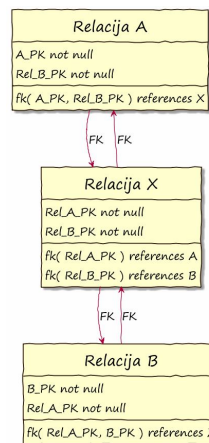
- Прави се нова *везна* релација
  - садржи само стране *кључеве* у односу на обе *везане* релације
  - кључеви **не смеју** да буду недефинисани
  - сви атрибути чине примарни *кључ*
  - додаје се услов базе података: “за сваки ентитет А мора да постоји бар један одговарајући ентитет Б”
  - додаје се услов базе података: “за сваки ентитет Б мора да постоји бар један одговарајући ентитет А”
- ...алтернатива...
  - и у А и у Б се додају страни *кључеви* према изабраним деловима/целинама...



Универзитет у Београду - Математички факултет

## Бинарни односи 1..N – 1..N (3)

- ...алтернатива...
  - слично као и за 0..N – 1..N
  - и у А и у Б се додају страни *кључеви* према релацији X, којим се реферишу изабрани парови
  - нису потребни додатни услови



Универзитет у Београду - Математички факултет

## ...наставак на следећем часу...

- Примери...
- Сложени односи...
- Хијерархије ентитета...

Универзитет у Београду - Математички факултет



## Литература за тему



- Teorey, Lightstone, Nadeau, Jagadish, **Database Modeling and Design**, 5.ed, Elsevier, 2011.
- Watt, Eng, **Database Design**, 2.ed, Open Edition, 2014.