

Савети за израду пројекта

1. На почетку спецификације потребно је нагласити за који систем се моделује шема базе података, било помоћу наслова или помоћу прве уводне реченице. Нпр. ставите да је наслов документа *Аутошкола* уколико моделујете шему базе података за аутошколу, или напишите уводну реченицу „У наставку спецификације дат је опис реалних ентитета и њихових односа за систем аутошколе“ или „Моделује се шема базе података за аутошколу“.
2. Доњи кардиналитет са обе стране повезника не сме бити 1. Није могуће да оба повезана типа ентитета буду егзистенцијално зависна један од другог.
 - а. Посматрајмо однос школе и учионице. Делује да би требало измоделовати систем тако да свака школа има барем једну учионицу и да свака учионица припада тачно једној школи. Иако је реалан систем такав, информациони систем који га подржава не сме имати повезане међусобно егзистенцијално зависне типове ентитета. Проблем би настао приликом уноса података о школи, где би се одмах захтевали подаци о некој повезаној учионици, или приликом уноса података о учионици, где би се захтевали подаци о школи којој припада. Генерално говорећи, проблем настаје приликом уноса где се захтева претходно постојање ентитета са којим је везан. У оваквим случајевима је потребно размишљати којим редоследом подаци улазе у информациони систем. Нпр у овом случају би требало омогућити унос података о школама пре него што им се додели нека учионица, односно школа би требала да може да не буде иницијално повезана ни са једном учионицом.
3. Ако имате 2 повезана типа ентитета (у даљем тексту ТЕ), где су кардиналитети са обе стране (1,1), та 2 ТЕ се вероватно требају спојити.
4. Доњи кардиналитети повезника могу бити искључиво 0 или 1, а горњи искључиво 1 или N. Није дозвољен кардиналитет 2, 3, 4...
 - а. Некада се може направити више повезника који ентитете повезују различитом семантиком. Нпр. уместо да се специфицира да фудбалску утакмицу играју 2 тима (односно N тимова), проблем се може решити тако да што утакмица повеже са 1 домаћим и 1 гостујућим клубом, односно коришћењем 2 повезника чији су горњи кардиналитети 1.
 - б. Некада је најприродније користити горњи кардиналитет више и не ограничавати кроз модел број повезаних ентитета једног типа са ентитетом другог. Нпр. 8 спринтера трчи трку, али у моделу ће бити кориштен кардиналитет који исказује да више спринтера трчи трку, (уз евентуални захтев да барем 1 спринтер мора да је трчи).
5. Подтипови ентитета у IS-A хијерархији не би требало да буду „висећи типови“ – типови који немају никакво своје карактеристично обележје нити посебну везу са другим ентитетима. Нпр. радници предузећа могу да имају више занимања, али нека нису од посебног значаја (немају посебних веза нити обележја), те неће бити укључена у модел. У том случају, IS-A хијерархија ће увек бити парцијална, односно неће се свака инстанца надтипа уједно сматрати и инстанцом неког подтипа.
6. Надтип IS-A хијерархије увек треба да има класификационо обележје на основу ког се може ефикасно утврдити ком подтипу припада одређена инстанца надтипа.

7. IS-A хијерархија није једини начин да се назначи да неки тип ентитета припада одређеној подврсти спрема неког критеријума. Нпр. филм може бити комедија, драма, хорор и слично. Да бисмо исказали такво својство филма не цртамо IS-A хијерархију са подкласама комедија, драма и хорор (које би свакако биле висеће), већ уводимо нови тип ентитета жанр, који повезујемо са филмом помоћу типа повезника припада. IS-A хијерархију уводимо када нам је потребно да моделујемо специфичности подкласа.
8. Могућ је сценарио где је потребно да се класификују инстанце одређеног типа ентитета спрема одређених категорија, при чему инстанца може припадати више од једне категорије, а постоји фиксан, предефинисан и непромењив број категорија и једино што карактерише категорију је њен назив. Пример тога су позиције које играч може да игра у одређеном спорту – постоји фиксан и непромењив сет позиција које зависе искључиво од спорта. Ова ситуација се може решити и тако што се ентитету који је потребно класификовати (овде, играчу) придодат онолико обележја колико постоји категорија (овде, позиција), при чему свако од тих обележја садржи логичку вредност која указује на то да ли ентитет припада одређеној категорији. На тај начин је могуће ентите повезати са произвољним подскупом фиксног сета категорија.
9. Идентификациона зависност никада није једино решење. Сваки идентификационо зависни тип ентитета се може заменити са регуларним типом ентитета који је повезан са неким другим типом ентитета кардиналитетом (1,1).
 - а. Нпр. студенти могу да се идентификују на основу своје ознаке, али у оквиру смера који похађају (у том случају је ТЕ *Студент* идентификационо завијестан од ТЕ *Смер*). Алтернативно решење је да се све студенти, независно од смера, идентификују искључиво на основу неког свог обележја (у том случају је ТЕ *Студент* само повезан са ТЕ *Смер* кардиналитетом (1,1)).
10. Не треба претеривати са увођењем идентификационих зависности. Ако се пречесто користе, могу да се добију „ланци“ типова ентитета повезаних идентификационим зависностима, шта доводи до ентитета (и последично повезника) чији кључеви се састоје од великог броја елемената, што није генерално пожељно.
11. На идентификационо зависни тип повезника не стављајте никаква обележја нити повезнике (она никада неће бити герунд). Све што би ставили на тај повезник, преместите на идентификационо зависни тип ентитета.
 - а. Нпр. ситуација у којој студент који припада одређеном смеру слуша одређени предмет би се могла моделовати тако да *студент који припада одређеном смеру* буде герунд. Са друге стране, ако је студент идентификационо завијестан од свог смера, онда не би постојао герунд него би предмет био повезан са студентом уместо са герундом који обједињава студента и смер.
12. Сваки тип ентитета мора имати барем једно примарно и барем једно непримарно обележје.
13. Моделујете систем за неку организацију, нпр. за кошаркашки клуб. Ако моделујете софтверско решење за које сте замислили да би било кориштено у више кошаркашких клубова, онда има смисла имати клуб као ентитет. Ако је систем кориштен унутар једног клуба, нема га потребе укључивати у модел. Информациони систем се у већини случајева користи на нивоу једне организације, те обично нема смисла моделовати ТЕ чије би инстанце биле саме организације.
 - а. Додатно, генерално треба бити опрезан када долазите на идеју да моделујете ТЕ који би имао само једну инстанцу.
14. Ако хоћете да радите праћење нечега кроз време (нпр. како се кретала цена неког артикла, функције које је особа имала у неком предузећу, како се кретала пошлица приликом

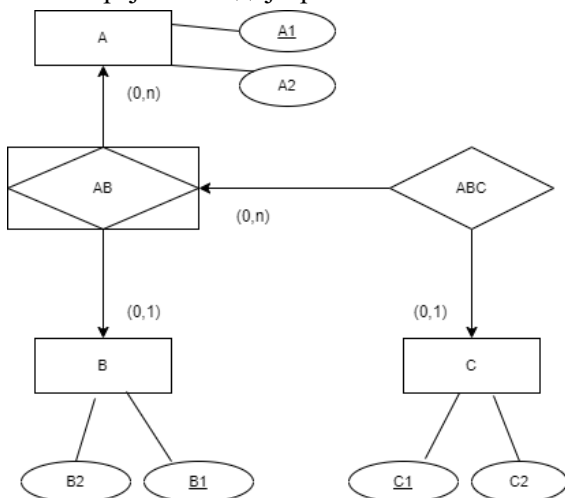
транспорта), то решавате тако што имате посебан ентитет који представља записе о стању посматраног објекта у одређеном моменту. Такав ентитете увек има у себи обележје које је индикатор времена када је одређена вредност постављена/забележена. Додатно, показатељи морају референцирати ентитете на које се односе и често их је погодно моделовати као идентификационо зависне типове ентитета.

- a. Избори у академска звања особља на факултету се могу моделовати као засебан тип ентитета *Избор* повезан са типом *Особље* и типом *Звање*. Осим тога, имали би обележје које представља датум избора одређене особе у одређено звање.
 - b. Промена цене неког артикла би се могла мерити коришћењем типа ентитета *Ценовник* који би уз то могао евентуално бити и идентификационо зависан од типа ентитета *Артикал*. Артикал може да има своје обележје, актуелну цену, а инстанце ценовника представљају све забележене цене артикала у назначеним моментима у времену.
15. Када моделујете цену неке робе, увек је згодно креирати типове ентитета као *Ценовник*.
16. Ако моделујете неку купопродају (најчешће малопродају), рачуне никада не повезујете директно са артиклима. Рачуни имају своје ставке које имају своје редне бројеве, везу са артиклима и количину артикла на одређеној ставци рачуна (ту се додатно могу наћи и друге ствари, нпр. попуст који се односи само на одређени артикал конкретнoг рачуна).
17. Уколико желимо да одређене ТЕ повезујемо тако да одређена конкретна инстанца неког типа може више пута да буде у вези са одређеном конкретном инстанцом другог типа, потребно је њихове везе моделовати посебним типом ентитета који опционо може бити идентификационо зависан од типова ентитета које повезује.
 - a. Нпр. ако имате ТЕ *Студент* и ТЕ *Предмет*, и желите да повежете студента са предметима које је положио, при чему сваки студент са сваким предметом може бити повезан само једном (када студент положи предмет) ту можете користити повезник који повезује та два ТЕ.
 - b. Али, ако желите да исказете да студент може да полаже предмет (излази на испит) више пута, то није могуће урадити користећи повезник (јер је кључ повезника у овом случају комбинација кључева повезаних ентитета, па свака два ентитета могу бити упарена највише једном, што не задовољава наше захтеве). Тада је потребно креирати нови ТЕ *Полаже*. Свако полагање се везује за тачно једног студента и један предмет, при чему полагања могу, али не морају нужно, да буду идентификационо зависна од студената и предмета који се полажу.
18. Уколико желите да моделујете наруџбенице или моделујете магацине и магацинско пословање погледајте пример 1.10. из збирке задатака. У оквиру овог примера моделован је тип ентитета складишни документ и тип ентитета тип документа, као и још неколико додатних типова ентитета. На такав начин, осим наруџбенице, покривени су и остали важни типови складишних докумената попут отпремнице, отписне листе и пријемне листе.
19. У одређени системима се ствара конфузија где студенти моделују и особе и документе путем којих особе бивају препознате у систему (нпр. студент и индекс на факултету, студент и студентска картиза за мензу или студентски дом, особа и њега возачка дозвола).
 - a. Нпр. нема превише смисла моделовати ТЕ *Студент* и ТЕ *Индекс*. Та два типа ентитета се свде на исто. Индекс је документ у који су унети подаци о студенту као реалној особи који се евидентирају у ИС факултета.
20. Догађаји које је потребно логовати се најчешће моделују као посебни типови ентитета. Нпр. ако клијент плаћа одређену врсту чланарине, потребно је имати ТЕ Уплата која ће бити повезана са клијентом и одређеном категоријом чланарине, а не само повезник који ће

исказати коју врсту чланарине клијент иначе плаћа. (ово важи уколико се желе евидентирати неке уплате, уколико само желите да исказете ка клијент припада одређеној категорији спрам које би иначе требало да плаћа неку чланарину, проблем се може решити и повезником).

21. Када желите да моделујете ентитете који представљају регистар/каталог нечега (нпр. ред вожње који обухвата више полазака неких аутобуса, ценовник који обухвата више цена одређених артикала, радни календар који обухвата више ангажмана особа на одређеним позицијама) никада нећете имати ТЕ који представља читав каталог него ћете имати ТЕ чије инстанце представљају један запис тог каталога (уместо да имате ТЕ *Ред вожње*, имаћете ТЕ *Полазак*; уместо да моделујете *Ценовник*, моделоваћете *Цену*; уместо да моделујете *Радни календар*, моделоваћете *Ангажман*). Свака торка тако добијене релације након превођења ће представљати један запис регистра/каталога, а када се скуп инстанце посматра заједно, оне представљају читав регистар/каталог.
22. EER модел служи томе да се дефинише шема базе чије стање треба да одговара стању реалног система који она подржава. Разна ограничења која су могућа у реалном систему не могу бити специфицирана EER моделом, јер он ни не служи томе. Таква ограничења би била задовољена у слоју бизнис логике посредством кога би садржај базе био ажуриран (што није тема овог пројекта).
 - а. Такође, EER модел не служи томе да се опише одређен процес нити ток одређених докумената и информација кроз систем. База би својим дизајном требала да омогући евиденцију докумената креираних унутар система, без претераног оптерећивања на хронолошке везе у експлоатацији докумената. Ако имате дуже ланце герунда, вероватно имате овај проблем.

23. Посматрајте EER дијаграм са слике:



Будући да су горњи кардиналитети повезника АВ и АВС 1-N, ови повезници се неће превести као засебне шеме релација, него ће настати страни кључ у шеми релације на страни максималног кардиналитета 1.

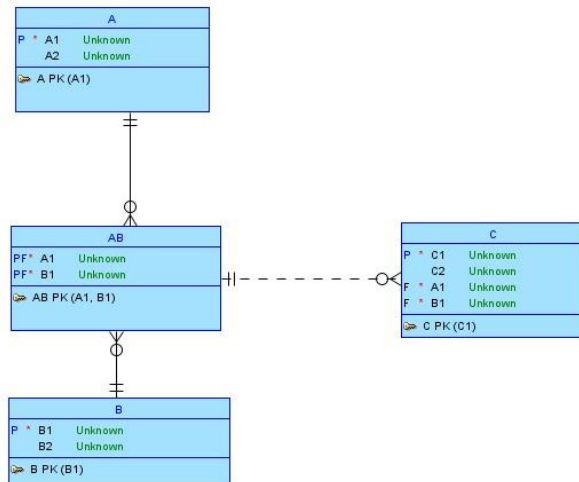
Релациони модел преведеног система на крају би био:

```

A({A1,A2}{A1})
B({B1,B2,A1}{B1})
  B[A1] ≤ A[A1]
C({C1,C2,B1}{C1})
  C[B1] ≤ B[B1]
  
```

Због тога нема ни потребе направити герунд АВ, него је довољно да тип повезника АВС повезује тип ентитета В и тип ентитета С.

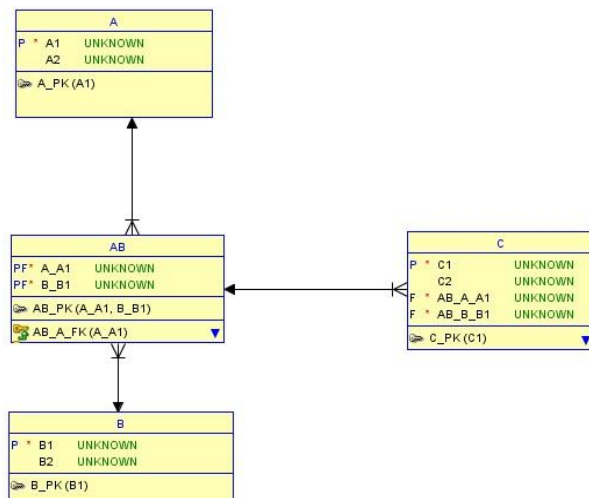
Обзиром да се у data modeler-у герунди не могу направити другачије него коришћењем идентификационо зависних типова повезника, овај систем би у data modeler-у био измоделован на следећи начин:



Пошто је овде АВ измоделовано као ентитет, а не као герунд, што заправо јесте, data modeler ће приликом превођења АВ тако и третирати. Биће креирана посебна релација која је резултат превођења оваквог ентитета и она ће имати стране кључеве ка ентитетима А и В.

Резултат превођења је дат на наредној слици. Добијени превод у алату и „на папиру“ се не поклапају те их је потребно ускладити. Крајњи релациони модел у data modeler-у треба да буде

идентичан оном на папиру, односно њега треба преуредити. Потребно је обрисати релацију АВ, што ће за собом повући брисање страних кључева од АВ ка А и В, као и страног кључа од С ка АВ. Након тога је потребно ручно у релационом моделу креирати страни кључ од В ка А, као и од С ка В.



Поново можемо доћи до закључка да није ни било потребе направити герунд АВ (због максималних кардиналитета 1-N), већ направити везу између типова ентитета А и В. Да су максимални кардиналитети били N-N, онда морамо направити герунд АВ и превођење би свакако било другачије (настала би посебна шема релације АВ).