Данко Милашиновић

УВОД У ИНФОРМАЦИОНЕ ТЕХНОЛОГИЈЕ

Данко Милашиновић

Увод у информационе технологије

Универзитет у Крагујевцу
Факултет за хотелијерство и туризам у Врњачкој Бањи
Врњачка Бања, 2016.

Увод у информационе технологије
- прво издање -

Аутор:
др Данко Милашиновић, ванредни професор
Факултет за хотелијерство и туризам у Врњачкој Бањи,
Универзитет у Крагујевцу
www.linkedin.com/in/dmilashinovic

Рецензенти:
др Владимир Цвјетковић, доцент
Природно-математички факултет,
Универзитет у Крагујевцу
др Александар Пеулић, ванредни професор
Факултет инжењерских наука,
Универзитет у Крагујевцу

Издавач:
Универзитет у Крагујевцу,
Факултет за хотелијерство и туризам у Врњачкој Бањи
www.hit-vb.kg.ac.rs

За издавача:
проф. др Драго Цвијановић, декан

Технички уредник:
др Данко Милашиновић

Уредник MHTSPS издања:
др Марија Мандарић, доцент
Факултет за хотелијерство и туризам у Врњачкој Бањи,
Универзитет у Крагујевцу

Штампа:
Принт Промет д.о.о. Краљево

Тираж:
300 примерака

ISBN 978-86-89949-14-8

Садржај

Предговор

Ова књига је припремљена као уџбеник и резултат је реализације ТЕМПУС пројекта „Modernization and Harmonization of Tourism Study Programmes in Serbia“, No. 544543-TEMPUS-1-2013-1-RS-TEMPUS-JPCR. Партнер на овом ТЕМПУС пројекту је Универзитет у Крагујевцу - Факултет за хотелијерство и туризам у Врњачкој Бањи.

Садржај ове књиге у складу је са наставним планом и програмом наставног предмета Информационе и комуникационе технологије у хотелијерству и туризму који се изучава на основним академским студијама на Факултету за хотелијерство и туризам у Врњачкој Бањи Универзитета у Крагујевцу.

Информациони системи су „кичмени стуб“ савременог пословања и то је основна тема ове књиге. Пошто је текст ове књиге припреман за будуће менаџере, многи технички елементи су поједностављени. Структура књиге осмишљена је тако да буде наставак на наставни материјал изложен у књизи Основе пословне информатике која је у употреби као уџбеник Универзитета у Крагујевцу на Факултету за хотелијерство и туризам у Врњачкој Бањи. Текст је конципиран тако да на смислен и сажет начин приближи тему читаоцима.

Мада структура књиге прати структуру курса наставног предмета Информационе и комуникационе технологије у хотелијерству и туризму, постоје одређени делови курса који су

у књизи изостављени. Ти делови курса тичу се практичне употребе информационих система у хотелијерству и туризму - део наставе који се изводи на вежбама овог наставног предмета. Планирано је да због своје природе поменути наставни елементи буду део практикума овог наставног предмета.

Захваљујем се својим драгим колегама, др Владимиру Цвјетковићу и др Александру Пеулићу, који су се прихватили тога да буду рецензенти ове књиге, на успешној дугогодишњој сарадњи.

Аутор

01 Увод

У савремено доба у пословне и у приватне сврхе свакодневно се генеришу велике количине података. Ти подаци се готово подразумевано чувају у дигиталној форми [Д.Милашиновић, 2014.]. Системи који чувају, преносе, врше обраду ових података и испоручују их корисницима када они то затраже зову се информациони системи. Једноставно речено то су системи који садрже податке, а у складу са захтевима корисника испоручују информације. Због тога је време у коме живимо време информационих система.

Интернет сајтови преко којих купујемо различите врсте производа, софтвер који нам помаже да квалитетније послујемо у различитим привредним гранама, банкарски софтвер, медицински софтвер, друштвене мреже на Интернету... Лако је приметити да су информациони системи присутни у свим сферама савременог друштва.

Примењивост информационих система за различите активности човека (пословање, забава, истраживање...) условило је појаву њихових разноликости. Због тога је и категоризација информационих система подложна константној промени. Готово све нове информатичке технологије брзо налазе примену и у различитим информационим системима, тако да се редовно појављују

информациони системи потпуно нових намена. Рачунарске мреже су се појавиле у употреби у другој половини прошлог века. Значај ових технологија може се по технолошкој аналогији упоредити са најбитнијим проналасцима у историји [В.Цвјетковић, 2008.]. Информациони системи су готово увек конципирани тако да користе и рачунарске мреже. У том контексту ову књигу треба третирати као увод у једну велику, важну и примењиву научну област.

Да би даљи текст био једноставнији за разумевање у делу који следи биће дефинисани основни појмови. Пошто је књига првенствено намењена менаџерима, дефиниције које следе биће мање стриктне, а више описног карактера.

01.01 Основни појмови и термини

Систем се уопштено може дефинисати као скуп објеката (односно ентитета)[1] и веза између њих. Објекти система карактеришу се, и међусобно разликују, својим атрибутима.

Уобичајено систем има неку сврху, те се објекти који га чине заједно са везама постављају ка остваривању циља.

Као пример за систем може послужити пословање туристичке агенције. У таквом систему објекти би могли бити запослени, хотели, транспортне компаније и тако даље. Комплетан систем успостављен је да функционише тако да на што бољи начин пружа туристичке услуге својим клијентима.

1 У наредном тексту читаоцу ће бити предочена разлика између ентитета и објеката.

Ентитети се међусобно разликују по својим атрибутима. Ако је примера ради у систему ентитет хотел, атрибути везани за њега могу бити назив, категорија, број соба и слично. Сваки појединачни хотел система је објекат система и у складу са ентитетом којим су сви хотели дефинисани - има конкретан назив, категорију, број соба и све остале вредности дефинисаних атрибута.

Скуп свих објеката система и тренутних вредности атрибута свих објеката система одређује стање тог система [В.Девеџић, 2000.]. Често је захтев, приликом израде информационих система, да постоји могућност увида у било које од претходних стања система.

Скуп објеката система одређује и његове границе. Сви објекти који не припадају овом скупу део су околине система.

Систем готово увек интерагује са околином. Са једне стране околина система условљава промене стања тог система (улаз), а са друге стране дати систем врши промене стања околине (излаз).

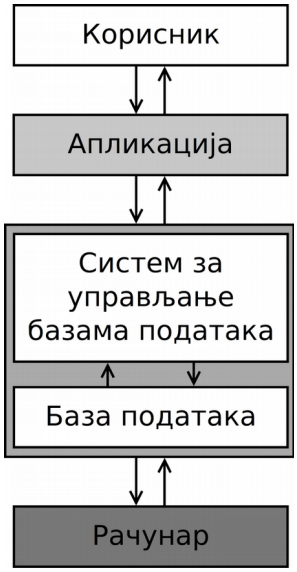
Мењањем стања, мењају се вредности атрибута објеката система. Када се промена врши у систему мењају се вредности атрибута објеката система, а када се врши промена стања околине вредности атрибута објеката околине бивају промењене. У поменутом примеру у околину система туристичке агенције може да спада банка или неки клијент агенције, а трансфер новца може да буде пример за промену стања система пошто мења вредност атрибута објекта система (новац којим располаже агенција). Тако је могуће прецизно предвидети будућност система уколико је познато стање система и улаз.

Информациони систем[2] - IS (eng. information system) је интегрисани скуп компоненти за сакупљање, снимање, чување,

2 Препоручени видео материјал: www.youtube.com/watch?v=Qujsd4vkqFI

обраду и преношење информација.

Мада се у литератури може срести и велики број других дефиниција информационих система, да би разумевање даљег текста било олакшано, горе наведена дефиниција ће бити потребна и довољна.



Сл. 1: Информациони систем

За разлику од општег случаја, информациони систем је систем у коме се везе између његових објеката, као и везе између система и окружења остварују искључиво разменом информација[3] [В.Девеџић, 2000.].

Са становишта информатике на основном (најнижем) нивоу, било какав запис (на било каквом меморијском медијуму) у битовима (eng. bit) је податак, односно информација. Разлог за то је што је бит елементарна јединица података свих дигиталних уређаја. Пошто

3 У општем случају везе између објеката система могу се остварити и разменом материје, односно енергије.

дигитални уређаји оперишу са само два стања (нема напона и има напона[4]) бит може имати само две вредности. Ове две вредности се уобичајено обележавају са 0 и 1 [Д.Милашиновић, 2014.]. Било какви подаци у оквиру базе података информационог система, као и сам систем за управљање базом података (о чему ће бити више речи касније), у складу са наведеном дефиницијом на најнижем нивоу нису ништа друго до велики број нула и јединица које су у одговарајућој физичкој форми записане на неком меморијском медијуму.

Када се говори о подацима у оквиру база података, и информационих система генерално, подразумева се ниво који је више апстрактан. То је зато што бинарно записане информације људима нису од неке претеране користи, јер нису читљиве. Корисник информационог система уобичајено очекује информације о банковним рачунима, именима људи и предмета и тако даље. За потребе информационих система дефиниције података и информација треба да се подигну на виши (апстрактнији) ниво. У тексту који следи биће дефинисани податак и информација у контексту информационих система.

Податак је кодирана чињеница, углавном из реалног света, који је носилац информације.

У складу са наведеном дефиницијом податак је било каква кодирана чињеница. Примера ради податак може бити адреса становања, величина патика, величина кошуље или назив компаније. У случају адресе становања географска (односно физичка) локација кодирана је (рецимо UTF8 скупом карактера) улицом, бројем и

4 Прецизно технички речено напон увек постоји, због тога се дефинише напонски праг на основу кога се одређује дигитална нула и јединица.

називом града у коме неко станује. Величина патике кодира се бројем (у декадном запису) - рецимо 45, а величина кошуље словима - примера ради XL (eng. extra large). Назив компаније кодира се, слично као и адреса, ниском карактера - рецимо Puma. Такав податак - број, односно низ карактера бива унет у информациони систем (односно у базу података информационог система).

Информација се добија анализом података и зависи од контекста. За контекст информације меродаван је корисник.

Да би се направила јасна дистинкција између појмова информација и податак може се замислити да корисник информационог система жели да добије одговор на питање који је број клијената његове компаније који имају адресу становања у Врњачкој Бањи. Потребна информација је број, а од великог броја клијената (из велике количине података) потребно је да се (анализом) дође до информације о броју клијената који имају дату адресу становања. Када корисник од информационог система добије број, то је информација коју је тражио. Уколико неки други корисник информационог система приступи истим подацима, и информациони систем му из неког разлога прикаже број клијената са адресом становања у Врњачкој Бањи (потпуно исти број као и претходном кориснику), лако се може догодити да тај број третира као сасвим небитан. У случају другог корисника тражени број није информација већ податак који му није интересантан. Информација за једног корисника може бити третирана као (сасвим неважан) податак од стране другог корисника, због тога се каже да је за контекст информације меродаван корисник.

01.02
Информациони системи у пословању

У савременом пословном окружењу информациони систем чува све податке о пословању. Без обзира о којој пословној грани је реч, сви аспекти пословања њиме морају бити подржани. То практично значи да сваки запослени своје активности треба да уноси у информациони систем на одговарајући начин, као и да преузима пословне информације. Употребом одговарајућег информационог система, запосленима је олакшано пословање, а и менаџерима је поједностављено управљање. Због овога се често каже да је информациони систем компаније њен „кичмени стуб“ пословања.

У зависности од конфигурације информационог система, уобичајено поједини запослени немају приступ свим подацима (примера ради платама запослених имају приступа само запослени у рачуноводству). Ограничавањем приступа у складу са пословним позицијама омогућава се лакша употреба информационог система, као и већа безбедност података које систем чува.

Углавном се према пословној грани компаније користи информациони систем који је у складу са датом пословном граном прављен и доступан на тржишту (хотелски информациони систем, банкарски информациони систем, информациони систем хипермаркета, информациони систем болнице). Практично за све уобичајене типове пословања на тржишту постоји велики број прилично рафинисаних информационих система. Као и у случају многих других производа, постоји велики број компанија које врше развој информационих система за различите намене.

Чак и уколико се нека компанија бави специфичном пословном граном, њени запослени треба да користе информациони систем. У том случају, пошто вероватно на тржишту нема већ постојећих информационих система који би одговарали датом пословању, компанија треба да изради информациони систем према својим потребама. Тада је обично најбоља опција да менаџери фирме пронађу одговарајућу софтверску компанију која може да врши развој информационог система за њихове потребе. У пракси је чест и сценарио у коме велика компанија после неког времена успешне сарадње на развоју информационог система купује (у целости или делимично) и софтверску компанију која је вршила његов развој, а након тога од ње ствара одељење за ту намену.

01.03 Информационе и комуникационе технологије

Термин информационе технологије - IT (eng. information technology) може се дефинисати као „изучавање, дизајн, развој, имплементација и подршка или управљање рачунарским информационим системима, софтверским апликацијама и хардвером” [K.S.Proctor, 2011]. У оквиру информационих технологија користе се рачунари, рачунарске мреже и рачунарски програми да би се конвертовале, ускладиштиле, штитиле, обрадиле, безбедно послале и примиле информације (односно подаци).

Информационе и комуникационе технологије - ICT (eng. information and communication technology) је проширени термин за информационе технологије. Термином информационе и

комуникационе технологије, међутим, акцентује се унификована комуникација. У том смислу подразумева се интеграција телекомуникација, рачунара, неопходног софтвера, складишта информација, мултимедијалних система. Ове технологије омогућавају корисницима да приступе, складиште, пошаљу, приме и манипулишу подацима. Иако је проширени термин дескриптивнији, због контекста ове књиге, у даљем тексту оба термина биће третирана истоветно.

На основу наведених дефиниција треба да буде јасно да информационе и комуникационе технологије обједињују:

- хардвер;
- софтвер;
- телекомуникације;
- особље.

У пословном свету данашњице практично сви људи употребљавају информационе системе, а некада није било тако. Многе особе данас нису ни свесне тога да их директно или индиректно користе. Примера ради особа која није заинтересована за информационе системе одлази до велике продавнице у којој купује неке прехрамбене намирнице. Након плаћања на каси подаци о датој куповини остају у информационом систему продавнице, и вероватно ће бити употребљени у анализи (значајној за побољшање продаје, маркетинг и тако даље).

Може се рећи да је увођење информационих система у традиционалне сфере пословања заправо увођење иновација. Тиме се постиже већа конкурентност на тржишту.

Поред увођења информационих система у пословно окружење, они су данас присутни и ван пословног окружења (социјалне мреже, web базиране галерије слика и слично). Мада понекад није једноставно разграничити ова окружења, обзиром на тему ове књиге фокус ће бити на пословној примени информационих система.

01.04 Информациони системи и туризам

Туризам је од фундаменталног значаја за привреду и економију. У одређеним подручјима у свету туризам је практично најбитнија компонента економије.

Савремене информационе и комуникационе технологије су у великој мери трансформисале туризам. Са једне стране, појавом ових технологија туризам је битнији него икада до сада, а са друге стране организације и појединци који се баве туризмом, а који их нису благовремено прихватили трпе велике губитке [K.Al-Busaidi - H.Al-Shihi, 2010.]. Много је примера којима се ово може илустровати. Употребом информационих технологија људи широм света постају свесни великог броја нових туристичких понуда, туристи обично прве информације о некој туристичкој дестинацији проналазе самостално на Интернету [Организациони одбор научне конференције TISC 2016]. Употребом савремених информационих система на Интернету се лако самостално могу извршити резервације смештаја и превоза. Могуће је користити се искуствима других људи приликом одлучивања...

У зависности од конкретне примене постоји више стандардних

типова информационих система који се примењују у туризму. Туристичке агенције углавном користе системе за резервације, као и информационе системе које се тичу њиховог интерног пословања. Хотелски објекти уобичајено користе хотелске информационе системе, односно системе за управљање објектима - PMS (eng. property management systems). Туристичке организације користе информационе системе у које похрањују податке о свим туристичким понудама (углавном из одређене географске регије) - DMS (eng. destination management systems). Авио компаније користе системе за избор и резервисање карата [N.Minić.., 2014]. Велики број компанија за приказ географске дистрибуције различитих ресурса користи географске информационе системе - GIS (eng. geographic information systems). Поред података које чувају поменути типови информационих система, велика количина података доступна је јавно на Интернет сајтовима. То пружа могућност за употребу различитих типова интелигентних информационих система у ефикасној обради тих података у циљу доношења правилних пословних одлука [U.Gretzel, 2011.].

Пошто у светској економији туризам заузима значајно место, и пошто је туризам тесно повезан са информационим и комуникационим технологијама, потребе развоја туризма у значајној мери диктирају и развој ових технологија. То је због тога што се технологија углавном увек дизајнира на начин на који би корисници желели да је употребљавају.

02 Развој информационих система

Без обзира на то што већина људи не познаје много детаља о томе како се праве ципеле, готово сви имају прилично добре идеје о томе какве би ципеле желели да имају. Пошто скоро сви људи носе ципеле, за квалитетнији живот згодно је да имају знања о томе какве су им ципеле потребне. За сваког човека појединачно постоји низ особина које би ципеле требале имати (у одговарајућој боји или комбинацији боја, са пертлама, без пертли и тако даље). На сличан начин менаџери компанија углавном немају детаљног техничког знања о томе како се врши израда информационих система. Они међутим треба да у тај процес буду упућени до одговарајуће мере да би на адекватан начин могли да квалитетније изаберу информациони систем за своју компанију или да утичу на корекције већ постојећег.

Када се говори о одабиру информационог система за компанију то може бити готови производ (неки од већ направљених информационих система) или нови производ који се прави према спецификацији. Налик на поменуту куповину или израду ципела, у оба случаја битно је да особе које врше одабир у што већој мери „знају шта желе“.

Само по себи јасно је да ће особа (без обзира на евентуални природни таленат) бити далеко више оспособљена да врши правилан избор ципела уколико има што више знања и искуства са ципелама. Исто тако менаџер компаније може бити компетентан, када је реч о информационим системима, само уколико има знања и искуства са таквим системима. Такав менаџер може адекватно изабрати или драстично унапредити развој и употребу информационог система за своју компанију тиме што утиче на израду спецификације потреба. Пошто информациони систем спада у основни алат за функционисање савремене компаније, способност менаџера у овој области је веома битна.

02.01 Архитектура информационог система

Архитектура информационог система је оквир по коме особе различитих стручних профила и погледа на информациони систем могу контролисати основне блокове његовог развоја. Најједноставније је те блокове развоја замислити као блокове грађевинског материјала на неком објекту. Да би објекат био употребљив сви блокови морају да постоје (у супротном би било шупљина). Исто тако сви блокови развоја информационог система морају бити комплетирани да би он био адекватно урађен.

Различите стране у току развоја информационог система називамо заинтересованим странама (eng. stakeholder). Заинтересоване стране у основи и у општем случају се могу поделити на [А.Његуш, 2010.]:

- Власнике (eng. system owners)
- Кориснике (eng. system users)
- Пројектанте (eng. system developers)
- Градитеље (eng. system builders)

Већ на први поглед може бити јасно да приказана подела илуструје општи случај пошто може бити (било каквих) преклапања заинтересованих страна. Примера ради корисници могу једновремено бити и власници система, пројектанти једновремено могу бити и власници и тако даље.

Развојем информационог система управља менаџер. Он је особа која је еквивалент диригента, дистрибуира задатке свима и одговоран је за правовремену и квалитетну реализацију.

У развоју информационог система у општем случају учествује велики број људи различитих стручних профила. Особа која је задужена да буде медијатор у њиховој комуникацији је систем аналитичар (eng. system analytic). Систем аналитичар се довољно технички разуме у све делове посла, тако да олакшава међусобну интеракцију заинтересованих страна, као и приказивање резултата менаџменту и крајњим корисницима. Он се мора постарати да све заинтересоване стране до одговарајуће мере разумеју све делове информационог система [А.Његуш, 2010.].

На слици у наставку приказани су блокови развоја информационог система. У оквиру приказане схеме могу се сагледати перспективе на развој информационог система сваке од заинтересованих страна.

	подаци	процеси	комуникација
власници	оквирни садржај IS	оквирне функције IS	домен комуникација у IS
корисници	захтев за специфичне пословне податке	захтев за одређеним пословним процесима	захтев за одговарајући кориснички интерфејс
пројектанти	пројектовање базе података	дизајн софтвера	дизајн интерфејса
градитељи	израда базе података уз употребу одговарајућег DBMS	израда или куповина одговарајућег софтвера	израда или куповина решења за интерфејс

Сл. 2: Блокови развоја информационог система
[J.Whitten – L.Bentley, 2005.]

Свака заинтересована страна треба да допринесе развоју информационог система у оквиру блокова који су за њу намењени. Само у том случају може се направити одговарајући информациони систем.

Мишљење и жеље власника се у сваком послу морају уважити. Они су ти који обезбеђују финансије. Да се не би догодило да жеље власника ремете правилан развој информационог система, неопходно је да систем аналитичар у координацији са менаџером и осталим заинтересованим странама на пројекту предочи зашто је нешто добро или неопходно урадити на дати начин. Наравно уколико је власник упућен у информационе системе, у значајној мери може унапредити развој. Генерално нема потребе да власници залазе у детаље, већ да дају оквирне смернице за развој информационог система.

Корисници су ти који су најмеродавнији за то како информациони

систем треба да функционише и изгледа. Разлог за то прилично је јасан, пошто су корисници ти који употребљавају систем. У току употребе информационог система корисници стичу искуство, тако да они временом коригују своје захтеве. Са већим искуством они су више меродавни да кажу шта је тачно потребно да систем ради или памти и на који начин треба да то приказује. Примера ради постају свесни тога да одређене функционалности нису толико корисне, а да би неке које иницијално нису предвиђене могле да буду.

Пројектанти према правилу имају велико техничко разумевање. Они усаглашавају захтеве (корисника и власника) са техничким могућностима. Њихова улога је да у складу са захтевима оптимално испројектују систем у свим његовим аспектима, довољно детаљно да градитељи могу да врше имплементацију. Пројектанти треба да изврше и спецификацију свих технологија које ће у изради информационог система бити коришћене.

Након што су планови пројектаната припремљени, градитељи врше имплементацију. Улога градитеља у развоју информационог система је да га направе, тестирају, поставе у рад, као и да врше његово одржавање и развој у складу са пројектом.

02.02
Блокови података

Основна сврха информационог система јесте да корисницима пружа и чува жељене информације, односно податке. У зависности од његове намене, подаци које систем чува могу бити веома различити. Да би систем био у стању да ефикасно ради са потребним подацима

(чува, претражује, обрађује) потребно је да се испројектује и имплементира на одговарајући начин. У зависности од података које треба да чува, као и самог система (хардвера и софтвера система) оптимално решење може бити веома различито од случаја до случаја. Свака заинтересована страна треба да пружи допринос да би било јасно које је најбоље решење када је реч о подацима са којима систем треба да ради.

Власници треба да оквирно опишу са каквим подацима ће радити информациони систем. За објекте система (купце, производе, добављаче...) и њихове везе (добављачи достављају робу, производи се чувају у складишту...) власници треба да уопштено опишу циљеве, проблеме, ограничења, могућности и тако даље.

Корисници оперативно употребљавају информациони систем. Због тога су њихови захтеви од фундаменталног значаја за развој. Када је реч о подацима, потребно је да корисници детаљно дефинишу које све податке систем мора чувати. Према неком правилу ово је увек итеративан процес, пошто у току употребе система корисници често мењају захтеве (примера ради потребно је да систем чува и обим струка купаца јер на основу тога радње компаније ефикасније могу требовати нову робу), те систем временом почиње да бива све употребљивији. Подаци могу бити везани за било који од објеката система (купац има 30 година, прегледао је само спортске производе, носи број ципела 45 и тако даље).

У складу са захтевима власника и корисника, пројектанти у складу са описом праве моделе за базу података. Осим тога пројектанти треба да дефинишу који систем за управљање базама података - DBMS[5] (eng. database management system) треба да се

5 Појам DBMS ће у даљем тексту бити објашњен

користи, као и серверски рачунар и одговарајућу мрежну опрему (оперативни систем тог рачунара и тако даље).

На основу пројекта градитељи врше имплементацију. Поред прављења информационог система употребом специфицираних технологија на основу пројекта, градитељи треба да изврше и додатне послове. Генерално када је реч о подацима то подразумева постављање и конфигурацију серверских компјутера (на којима се налазе подаци и софтвер који укључује и DBMS), конфигурацију и постављање DBMS у рад, и осталих серверских апликација које су неопходне за рад информационог система (примера ради web базирана апликација која омогућава спољну контролу над DBMS).

02.03

Блокови процеса

Осим тога што информациони систем треба да чува и испоручује жељене податке, односно информације, неопходно је и да врши неке функције, односно процесе. Да би систем био сврсисходан неопходно је да се жељене функционалности иницијално спецификују као и да се изврши њихова правилна имплементација.

Да би правилно одговорили на питања која се тичу процеса, власници треба да пруже уопштену слику група пословних процеса у компанији. Уобичајено то обухвата рачуноводство, маркетинг, људске ресурсе и тако даље. Пожељно је да се обезбеди организациона структура компаније. Тада је могуће дискутовати и о примећеним проблемима, ограничењима и могућностима који су везани за процесе у оквиру система.

Сваки појединачни пословни процес је потпуно јасан за кориснике система. Због тога корисници имају кључну улогу у детаљној спецификацији свих функционалности информационог система. Сваки процес појединачно треба да буде што је боље могуће дефинисан. То уобичајено подразумева да се дефинишу улазни подаци, време трајања, излазни подаци (eng. workflow). Сваки процес дефинисан је прецизно корацима који су неопходни да би се реализовао. У току дефинисања процеса у изградњи информационог система, готово увек дешава се да се открију редундансе и сувишни кораци у пословању. Тако и сама документација у току рада на развоју информационог система може утицати на побољшање пословања.

Када процеси буду идентификовани и дефинисани, задатак пројектаната информационог система јесте да направе пројекат на основу кога градитељи могу да изврше њихову имплементацију. Пројектанти прво треба да дефинишу које технологије ће да буду коришћене (на каквом хардверу и оперативном систему ће софтвер радити, којим програмским језиком писати софтвер и тако даље). Након тога да у складу са специфицираним процесима направе технички пројекат са адекватним дијаграмима на основу кога градитељи могу да направе софтвер са жељеним функционалностима.

На основу докумената које пројектанти предају градитељима врши се писање софтверског кода у дефинисаном програмском језику (прављење софтвера). Када наследе пројекат градитељи на тражени начин стварају софтвер који ће имати све функционалности које су пројектанти специфицирали.

02.04

Блокови комуникације

Да би информациони систем био сврсисходан неопходно је да омогући корисницима (што укључује особе или друге информационе системе) адекватан интерфејс (eng. interface). У општем смислу те речи интерфејс је место на коме се сусрећу два независна система која интерагују међусобно. Када је реч о информационим системима том речју описује се интеракција корисника и информационог система. Корисник система може бити човек (примера ради рецепционар интерагује са информационим системом хотела) или други информациони систем (информациони систем хотела интерагује са информационим системом банке). За све предвиђене интеракције неког информационог система мора се омогућити адекватан интерфејс за комуникацију. Често је потребно обезбедити и да корисници употребом информационог система могу да комуницирају међусобно, ако је такав случај у изградњи система и за такву врсту комуникације мора се обезбедити одговарајући интерфејс.

И када је реч о комуникацији власници треба да направе оквир по коме систем треба да се развија. У случају блокова комуникације власници треба да дефинишу који ће подаци и функционалности система бити доступне којим групама корисника (примера ради подаци о платама запослених могу бити доступни само рачуноводству, стање у магацину могу мењати само магационери и слично). Битно је да власници дефинишу које групе корисника система имају међусобну интеракцију и са којим другим системима информациони систем који се гради може интераговати. Тиме што се

дефинишу могућности и ограничења у комуникацији у великој мери се олакшава пословање и употреба система.

За кориснике информационог система блокови комуникације углавном се своде на размишљање о томе како треба да изгледа кориснички интерфејс. Већ дуги низ година уназад подразумевани кориснички интерфејс је графички кориснички интерфејс – GUI (eng. graphical user interface). То практично значи да корисник углавном у виду прозора и дијалога (налик на оне који се појављују у оперативном систему MS Windows) има интеракцију са информационим системом. Корисници могу да сугеришу уколико сматрају да треба да постоје и неке комуникације у оквиру система које нису предвиђене по плану. Сугестије корисника о томе где би се у оквиру графичког интерфејса требале да налазе одређене функционалности или информације такође треба следити, јер су они ти који ће употребљавати информациони систем.

Осим тога што размишљају о истоветном приказу – графичком интерфејсу као и корисници, пројектанти система размишљају и о томе шта се налази „иза тих прозора“. Када је реч о графичком интерфејсу пројектанти имају на уму како треба да изгледа софтвер графички, али и како елементе графичког интерфејса треба увезати са остатком софтвера информационог система. Пројектанти треба да осмисле и интерфејс за комуникацију информационог система са другим системима. То значи да пројектантима мора да буде познато којим протоколима и на који начин треба да се врши комуникација. На основу тога они треба да предвиде како да се дати протокол за комуникацију имплементира у информациони систем чији развој се врши. За сву евентуалну међусобну комуникацију корисника у оквиру информационог система пројектанти такође треба да осмисле

адекватан начин и кориснички интерфејс.

Као и за друге блокове у оквиру информационог система, градитељи, што се комуникације тиче, треба да у складу са документацијом коју им предају пројектанти изврше имплементацију, инсталацију и тестирање. У складу са дефинисаним технологијама градитељи креирају информациони систем, затим врше његово правилно постављање на жељене рачунаре и на крају тестирају да ли све ради на жељени начин.

02.05 Континуални развој информационог система

Промена потреба успешне компаније која се временом дешава условљава и промене у њеном информационом систему. Због тога је информациони систем компаније вечито жив.

Било какве промене које се праве на информационом систему морају бити тестиране пре употребе. Чак и решења која раде тачно у складу са спецификацијама (захтевима) не морају бити финална, јер се сами захтеви са временом мењају.

Због овога се увек подразумева итеративни развој информационог система, у коме заинтересоване стране стално интерагују.

03 Основе база података

У савременом животу (дигиталној ери) термин база података се често употребљава. У досадашњем тексту такође је тај термин више пута је био употребљен. Базе података настале су из потребе да се ефикасно чувају подаци. Многи системи који раде на рачунарима имају потребе за чувањем велике количине података (банке, осигуравајућа друштва, компјутерске игрице, социјалне мреже и тако даље). Због тога су базе података присутне на великом броју различитих уређаја (класични рачунари, мобилни телефони, таблет рачунари и тако даље) и користе се за рад са различитим врстама података.

База података - DB (eng. database) је добро структурирана колекција података на рачунару. Такву колекцију података уобичајено употребљава више клијената.

Пошто је „добро структурирана“ може се ефикасно прегледати, сортирати, поредити, мењати... За некога ко није упућен у концепт базе података најједноставније је да је замисли као уређену библиотеку. У таквој библиотеци далеко је ефикасније наћи жељену књигу него је пронаћи у неуређеној гомили књига. Због тога се у библиотеци књиге слажу у складу са правилима, а у свакој књизи жељене информације могу се пронаћи употребом садржаја и индекса.

Слично се поступа и са базом података са разликом што је проналажење информација далеко брже него у библиотеци. Због тога је намена база података да поједноставе рад са подацима приликом продаје, у менаџменту било које врсте, у банкама, компјутерским игрицама, истраживањима...

Основу сваког информационог система чини база података. Сви подаци информационог система снимљени су у бази података. Ако замислимо да је информациони систем модел реалног система, примера ради пословања неке компаније, онда база података одговара стању тог система.

База података обезбеђује физичку и логичку независност података, чува интегритет, омогућава флексибилан приступ и ефикасан рад.

Уколико је потребно анкетирати велики број особа, уобичајено се прво креира упитник (формулар који је штампан на папиру). Након тога упитник попуњава свака особа у току анкете. Разлог што је ефикасније користити упитник него оставити празан папир, на коме би особа уносила податке према свом нахођењу, исти је као и разлог због кога је неопходно да база података чува интегритет (или да се књиге у библиотеци морају слагати према правилима). Када униформно форматиран упитник правилно попуне све анкетиране особе, далеко је лакше вршити претрагу, сортирање и проналажење жељених информација у односу на то када би сваки упитник био другачијег формата (када би се формулари разликовали). Разлог је што особа која врши претрагу увек зна где се, примера ради, налази број телефона у сваком од упитника, јер су сви формулари правилно и на исти начин структурирани. Када се уносе подаци у базу података, систем за управљање базом података такође чува

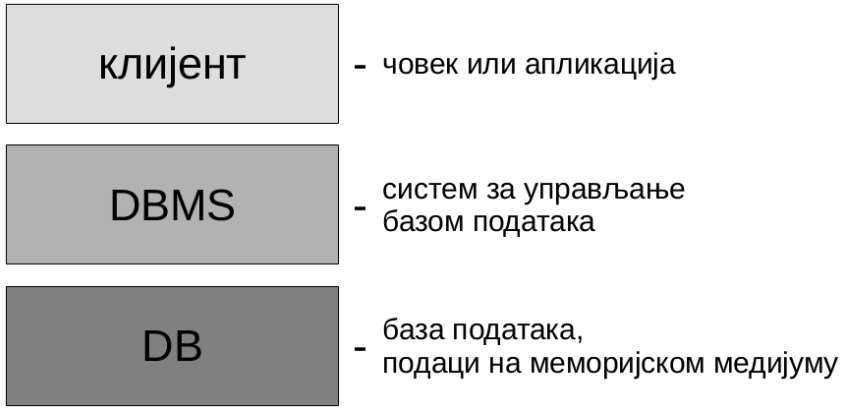
интегритет. Уколико клијент базе података покуша да, примера ради, унесе текстуални податак на место на коме систем очекује број телефона, унос неће бити дозвољен. Након тога компјутеру је далеко лакше да ради било какве операције са логично уређеним подацима.

03.01 Систем за управљање базама података

Интеракцију са књигама у библиотеци има библиотекар. Он је тај који смешта и узима књиге са полица библиотеке (према некој логици и у складу са захтевима корисника). Корисник библиотеке нема потребе да зна према каквим правилима поступа библиотекар са књигама. На сличан начин са подацима који се налазе у бази података оперише систем за управљање базама података - DBMS. То је специјализовани софтвер који служи да интерагује са базом података са једне стране, и клијентом базе података са друге стране. Све податке у базу смешта тај софтвер према неким правилима која су непозната за клијенте. Такође у зависности од требовања клијената DBMS проналази тражене податке и према потреби врши операције над њима. Резултате тих операција DBMS може али и не мора да похрањује натраг у базу.

Клијент не треба да размишља о томе на који начин DBMS завршава то што клијент од њега очекује. Дужност клијента је само да на адекватан начин затражи да DBMS уради неку операцију са подацима односно базом података. Клијент базе података је углавном друга апликација (други софтвер). Такве апликације употребљавају особе које су обучене за њихово коришћење - те особе су корисници

система. Осим тога, наравно, и човек може бити директно (мануелно) клијент који интерагује са DBMS, али се то углавном практикује само у току развоја апликација које треба да користе базу података (односно апликација које ће да интерагују са DBMS).



Сл. 3: Универзални начин рада са базом података

Пример за употребу базе података може да буде хотелски софтвер и рецепционар. Рецепционар нема потребе да интерагује са базом података директно. Он једноставно попуњава форме у оквиру хотелског софтвера када похрањује неке податке (примера ради податке о новом госту хотела), а са друге стране употребом софтвера може и да прегледа податке који већ постоје. Ти подаци не налазе се у оквиру хотелског софтвера, већ у бази података. Хотелски софтвер заправо интерагује са DBMS који интерагује са базом података. DBMS тада снима, претражује, преузима (и тако даље) податке из базе података и испоручује их хотелском софтверу, који их у адекватној форми (прозорима програма или у штампаној форми на папиру) испоручује рецепционару.

Изучавање база података може се генерално поделити у две категорије:

- изучавање DBMS;
- изучавање модела података.

Изучавање DBMS је нешто што превазилази оквире ове књиге. Због тога неће бити даље коментарисано.

Изучавање модела података на вишем нивоу такође превазилази оквире ове књиге. Елементарно познавање детаља око креирања модела података у оквиру базе података приликом развоја информационог система међутим јесте у домену ове књиге. То је оно што би на елементарном нивоу менаџер данашњице требао да зна, јер се тиче једног од основних алата у савременом пословању. Због овога ће о моделима података бити више речи у даљем тексту.

Уколико DBMS ради са релационим базама података, назива се и RDBMS (eng. relational database management system). Релациона база података је база података код које се организација података заснива на релационом моделу. Често се у литератури подразумева да се употребљава релациони DBMS, те се префикс или слово које означава релациони модел не назначава. Осим релационог модела, постоје и други. Примера ради данас је све чешће у употреби објектни (односно објектно-оријентисани) модел базе података. Објектни модел базе података има занимљиве и практичне предности у односу на релациони. Без обзира на то у овој књизи акценат ће бити на релационом моделу базе података пошто је он најзаступљенији.

Принципи по којима раде релационе базе података изведени су из математике (теорије скупова и предикатне логике). Др Едгар Ф. Код (1923. – 2003.) (eng. Edgar F. Codd) је први применио ове принципе на моделовање података 1960-их година. Овај истраживач

је тада радио за чувену компанију IBM. У оваквим базама података подаци се организују према релационом моделу у табеле (које се праве према релацијама модела), између којих се према потреби дефинишу везе. Др Е.Ф.Код је употребио назив релација, а не табела у моделу, јер у то време реч релација у информатици није имала друге конотације [R.Riordan, 2005]. Због овога овај модел и носи назив релациони.

03.02 Управљање базом података

На основу досадашњег текста јасно је да базом података управља DBMS. На који се начин врши интеракција између DBMS и базе података, као што је већ било напоменуто, превазилази оквире ове књиге. Оно што остаје да се разјасни је како интераговати са DBMS.

Начин на који клијент (човек или апликација) интерагује са базом података јесте посредно употребом DBMS. Да би интераговао са DBMS клијент употребљава упитни језик - QL (eng. query language). Овај назив користи се универзално за све језике којим се постављају упити над DBMS.

Упитних језика има више, али је најпознатији SQL (eng. Structured query language). То је стандардизовани језик који се користи за управљање савременим релационим DBMS. Пандан језику SQL који се користи код објектних база података је OQL (eng. Object query language).

Без обзира на то што је SQL стандардизован, у зависности од

конкретног DBMS (MySQL, MS SQL Server, SQLite и тако даље) постоје његове варијанте. SQL обједињује DDL, DML, TCL и DCL где су:

- DDL (eng. data description language)
  SQL пример: CREATE, ALTER, DROP;
- DML (eng. data manipulation language)
  SQL пример: SELECT, INSERT, UPDATE, DELETE;
- TCL (eng. transaction control language)
  SQL пример: COMMIT, ROLLBACK;
- DCL (eng. data control language)
  SQL пример: GRANT, REVOKE.

Због тога SQL пружа свеобухватну контролу над релационим DBMS.

ACID је скуп правила који гарантује да су трансакције у бази података поуздане. Овај скуп инструкција имплементиран је у све савремене релационе DBMS. Трансакција у бази података је једна операција. У оквиру листе која следи укратко су објашњена ACID правила:

- Acid - Atomicity - карактеристика DBMS да ће сви задаци једне трансакције бити извршени или ниједан;
- aCid - Consistency - трансакција не може нарушити правила, односно интегритет базе;
- acId - Isolation - сви задаци у једној трансакцији обављају се независно (изоловано) од осталих;
- aciD - Durability - након извршене трансакције, она остаје трајна. Често се све трансакције и пописују у дневник (eng.

log).

На следећем примеру биће приказано неколико једноставних SQL упита у оквиру MySQL DBMS. Претпоставља се да је администратор DBMS већ улогован са својим налогом - именом и шифром.

Упитом који следи креира се нова база података која се зове PRODAVNICA:

```
CREATE DATABASE PRODAVNICA;
```

Наредним упитом се додељују све привилегије над том базом (свим њеним табелама, односно релацијама) клијенту са именом PRODAVNICAklijent који може употребљавати те привилегије са локалног компјутера (компјутера на коме се налази DBMS). Начин да добије своје привилегије јесте да се пријави употребом свог имена PRODAVNICAklijent и шифре shifraklijenta:

```
GRANT ALL PRIVILEGES ON PRODAVNICA.* TO
'PRODAVNICAklijent'@'LOCALHOST' IDENTIFIED BY
'shifraklijenta';
```

Следећим упитом се мења скуп карактера који користи база у UTF-8:

```
ALTER DATABASE PRODAVNICA CHARACTER SET UTF8;
```

Затим се ресетују привилегије у оквиру DBMS:

```
FLUSH PRIVILEGES;
```

Након тога, следећим упитом администратор почиње да користи нову базу података PRODAVNICA:

```
USE PRODAVNICA;
```

Уколико је то потребно, могуће је следећим упитом креирати нову релацију, односно табелу у оквиру базе података. У упиту који следи биће дефинисана нова табела Sluzbenik која ће имати 5 атрибута (JMBG, ime, prezime, zanimanje, plata):

```
CREATE TABLE Sluzbenik ( JMBG INTEGER NOT NULL, ime
VARCHAR(255) NOT NULL, prezime VARCHAR(255) NOT NULL,
zanimanje VARCHAR(255) NOT NULL, plata DOUBLE NOT NULL,
PRIMARY KEY(JMBG) );
```

У овом примеру, ради једноставности, неће бити других табела (релација). Уколико администратор жели може да почне са употребом своје нове базе података која има само једну табелу. Пошто је сада табела без унесених података (нема ни једне унете n-торке), иницијално ће бити направљено неколико уноса. Нека први унос буде уписивање једног службеника који ће бити рачуновођа:

```
INSERT INTO Sluzbenik (JMBG, ime, prezime, zanimanje, plata)
VALUES ('1234578910', 'Петар', 'Петровић', 'рачуновођа',
'37000.34');
```

На истоветан начин у наредном упиту биће унет нови службеник који ће бити директор - биће унет нови ред у табелу (нова n-торка у релацију Sluzbenik). Назначеним редом ће сви атрибути добити и вредности:

```
INSERT INTO Sluzbenik (JMBG, ime, prezime, zanimanje, plata)
VALUES ('23457891011', 'Лазар', 'Лазаревић', 'директор',
'62000.14');
```

За сада постоји база података, која садржи само једну релацију, односно табелу. Пошто у тој табели за сваког службеника постоји и атрибут (колона табеле) у оквиру кога се уносе плате, рецимо да клијент базе жели да види само имена запослених који имају плату

већу од 50000. Ову операцију DBMS извршиће на основу следећег упита:

```
SELECT ime FROM Sluzbenik WHERE plata > 50000;
```

Након извршења упита DBMS ће приказати следећи резултат - релацију, односно табелу (која је виртуелна пошто као таква не постоји у бази података, већ је само резултат упита):

```
+----------+
| ime      |
+----------+
| Лазар    |
+----------+
1 row in set (0.00 sec)
```

У претходним примерима планирано су употребљени и термини који ће бити разјашњени у тексту који следи.

03.03

Одлике релационих база података

Основне одлике релационих база података су:

- Према овом концепту сви подаци представљени су, у складу са релационим моделом, као релације (табеле), које су организоване у редове и колоне. У пресеку редова и колона су поља која се називају и ћелије.
- Све вредности су скалари. То практично значи да се на сваком месту које је одређено редом и колоном налази само једна вредност.

- Све операције врше се над целом релацијом. Резултат сваке операције над релацијама је такође релација. Овај концепт назива се целовитост (eng. closure).

У оквиру сваке базе података која функционише према релационом моделу, сви подаци налазе се у релацијама. Те релације записане су у бази података као табеле (имају редове и колоне). Подаци се налазе у пресецима редова и колона релација (ћелијама табела). Разлог због којег су подаци смештени тако да се на месту које је пресек реда и колоне налази само једна вредност (да су вредности скаларне) је поједностављење рада са релацијама. Тиме релације нису ограничене, мада на први поглед може тако изгледати. Ово ће бити јасније након текста који следи.

Све операције које се могу вршити у оквиру релационе базе података, могуће је вршити само над целим релацијама, односно табелама. Резултат сваке операције такође је релација (табела), ово је особина која се зове целовитост. Због тога осим релација (табела) које се заиста налазе у бази података, које се још називају и реалне, релациони DBMS омогућава и формирање виртуелних релација (табела). Такве релације могу послужити и као улазни подаци за нове операције. У складу са наведеном методологијом пројектанти база података лако могу да компликоване операције изделе у више мањих засебних целина.

Да би досадашњи текст био јаснији, биће илустрован једноставним примером базе података библиотеке. Рецимо да у таквој бази података постоји релација у коју су унети сви чланови библиотеке. За сваког члана постоји низ података (име, презиме, број телефона, адреса становања и тако даље). Сваки од тих података уноси се у независну ћелију. Уколико клијент базе направи

одговарајући упит, као резултат од DBMS може добити табелу која садржи само имена чланова и њихове бројеве телефона. Таква релација реално не постоји у бази података, али се може добити као резултат операције над постојећом релацијом (табелом). Тада се добијена релација назива и виртуелна, односно поглед. Из те релације даље је, примера ради, могуће издвојити само бројеве телефона свих чланова чије име почиње словом А. У том случају улазни податак за упит је виртуелна релација (која је добијена претходним упитом), а резултат ће такође бити виртуелна релација (која није реална).

Мада ће о овоме бити више речи у даљем тексту, развој базе података за потребе неког система грубо се може поделити у следеће фазе:

- анализа потреба;
- креирање модела података;
- имплементација;
- тестирање;
- одржавање и развој.

На првом месту неопходно је извршити анализу потреба неког система за базом података. Уобичајено се таква анализа документује у виду кратког текста.

Формирање модела података врши се на основу текста добијеног у првој фази. Постоје правила која се у тој операцији поштују. Након тога, на основу искуства и правила добијени модели се коригују док се не постигне задовољавајуће решење. Овакви модели уобичајено се документују у виду дијаграма.

У зависности од конкретног DBMS, на основу дијаграма врши се имплементација употребом SQL (код релационих база података). Претварање модела из дијаграма у SQL такође је стандардизовано.

Након што се имплементирају модели у оквиру базе података врши се тестирање. Ова процедура ради се са тест подацима (измишљеним подацима) или правим подацима. Уколико се покаже да све ради на планирани начин, база се ресетује у почетно стање. Након тога спремна је за употребу.

У одржавање базе података, поред повремених чувања, односно прављења резервних копија (eng. backup), спада и њена модификација. База се модификује у складу са евентуалним новим потребама - врши се даљи развој. То је није компликовано када је она првобитно ваљано испројектована. У петој глави ове књиге биће више речи о поменутом развоју на конкретном примеру.

03.04 Модел података у релационом моделу база података

Модел података је специјализовани теоријски оквир, помоћу кога се спецификује, пројектује и имплементира нека база података. Важно је направити разлику између података (кодиране чињенице која је заправо носилац информације) и модела података (оквира који спецификује базу података, а самим тим и начин уношења и потраживања података у тој бази). Модел података, ради илустрације, може се упоредити са формуларом који попуњавају особе у некој анкети. Формулар дефинише оквир по коме се подаци на правилан начин уносе, а сами подаци (који су носиоци

информације) су на различит начин кодирани (бројеви, слова, слике, цртежи). Када се подаци уносе у складу са правилно структурираним оквиром, касније је далеко лакше користити унете податке.

Модел података, у релационом моделу база података, састоји се од следећих елемената:

- ентитет (класа);
- атрибут (поље);
- домен атрибута;
- веза.

Веома је важно у потпуности разумети сврху сваког од наведених елемената да би сам концепт релационог модела био јасан.

03.04.1 Ентитети

Објекат у релационим базама података треба да буде све оно о чему систем треба да чува податке (примера ради купац). За сваки тип објеката (службеници, производи, купци...), према основном правилу треба формирати ентитет. Када се прича о томе чему база података треба да служи, већина именица (и глагола) су могући ентитети. Примери за ово могу бити „купац купује робу“ и „продавац продаје робу“. На основу правила лако је закључити да купац, продавац (односно службеник), роба и продаје (наруџбеница) треба да добију ентитете (постану ентитети) у моделу података. Сваки појединачни купац, продавац, одређени производ или наруџбеница

након формирања ентитета постају објекти одговарајућих ентитета (инстанце одговарајућих класа). За сваки од објеката у оквиру базе података чувају се одговарајући подаци, примера ради име купца, презиме купца, његова адреса и тако даље.

Већ је у претходном тексту више пута поменуто да се у релационом моделу база података подаци чувају у релацијама. Важно је приметити да се код ових база података ентитет (односно класа) пресликава у релацију, која је у бази записана као табела. Свака релација у некој релационој бази података заправо је ентитет, који може бити независтан или зависан. О зависности ентитета, односно објеката тих ентитета, ће бити речи у даљем тексту. Релација у релационом моделу база података практично дефинише тип објекта.

Један ред релације назива се торка, или n-торка (где је n број колона, односно поља релације). Једна торка заправо је један примерак, односно један објекат (једна инстанца) те релације (ентитета, односно класе). Уколико се за пример замисли релација Kupac, сваки појединачни купац (особа) биће упамћен у бази података као један ред релације Kupac. Тада се још каже и да је сваки купац заправо један примерак - објекат, односно n-торка (инстанца) тог ентитета (или инстанца те класе).

Број n-торки назива се кардиналност релације. Ако у поменутом примеру у бази података, примера ради, има сачуваних 35 купаца (35 редова табеле Kupac), кардиналност релације Kupac је 35.

03.04.2
Атрибути

Конкретни подаци о објектима записани су у оквиру вредности атрибута датих објеката. Атрибути објеката дефинисани су у оквиру њихових ентитета. Уколико се вратимо на претходни пример и замислимо ентитет Kupac (који је формиран на основу купаца из реалног света), кандидати за његове атрибуте, у зависности од потреба система, могу бити ime (у оквиру кога се чува име сваког појединачног купца), prezime (у оквиру кога се чува презиме сваког појединачног купца), br_tel (у оквиру кога се чува број телефона сваког појединачног купца), adresa (у оквиру кога се чува адреса сваког појединачног купца), br_cipela (у оквиру кога се чува број ципела који носи дати купац)... У том случају за сваког купца, односно сваку n-торку или објекат (инстанцу) релације Kupac постоје вредности атрибута (ime: Марко, prezime: Марковић...). Атрибути дефинишу који ће подаци бити чувани за сваки од објеката. Они атрибути који нису дефинисани у оквиру ентитета (односно релације), изостављају се намерно из модела.

Када се каже „потреба система“ мисли се на његову сврху. Уколико се прави систем за потребе продавнице спортске опреме, број ципела може бити важан и треба да фигурише као атрибут за сваког купца, а уколико се прави систем за потребе књижаре онда је овај атрибут сасвим небитан тако да не треба да постоји (треба га изоставити из модела). Контекст тога шта је важно да се чува од података одређује циљ система (његова сврха).

Сваки атрибут има (за ту релацију) јединствено име и домен (скуп свих дозвољених вредности). Вредност атрибута може бити и ништа

односно NULL уколико је и то дозвољено. Треба да буде јасно да атрибут који има назив ime може бити присутан у више релација у једној бази података. Примера ради атрибут ime може фигурисати у релацијама Kupac и Sluzbenik (имена купаца, и имена службеника).

Број атрибута одређује степен релације. Уколико имамо случај да за релацију Kupac постоје 4 атрибута, степен релације Kupac је 4.

03.04.3 Домени

Домен је скуп свих валидних вредности које вредност атрибута може имати.

Уколико се за пример атрибута употреби ime (име купца), домен би могао бити низ карактера. Примера ради низ од 2 до 255 карактера, који за скуп карактера (eng. character set) користе све словне UTF-8 карактере. Уколико се за пример атрибута употреби br_tel (број телефона купца), домен може бити низ цифара.

Домен заправо служи да се дефинише ограничење. Уколико неко покуша да унесе слова уместо бројева за број телефона, DBMS неће дозволити такав унос уколико је претходно правилно дефинисан домен датог атрибута. Са друге стране приликом дефинисања домена важно је бити пажљив јер се може догодити да нешто што треба да буде могуће за унос у базу не буде подржано. Из горе наведеног примера уколико се у току имплементације модела података у базу података за домен, примера ради, не наведе неки шири скуп карактера (у горњем примеру UTF-8) од подразумеваног (који зависи

од конкретног DBMS, али је често ASCII) може се догодити да се имена која користе слова која нису у оквиру енглеског алфабета не могу снимити.

03.04.4 Везе

У оквиру модела података, у релационим базама података, везама се дефинишу односи између ентитета. Након што везе буду дефинисане, приликом уношења n-торки, односно објеката (инстанци) датих ентитета везе морају поштовати, пошто у супротном DBMS неће дозволити унос.

Ентитети између којих постоји нека веза су учесници везе. Број учесника везе одређује степен везе. Углавном се све везе праве таквим да буду бинарне (везе између два ентитета). Због тога везе које нису бинарне неће бити коментарисане (унарне, тернарне).

Кардиналност везе је однос броја објеката ентитета који се повезују. У случају бинарних веза кардиналност везе може бити:

- један према један (1:1)
  (пример: брачни пар);
- нула према више (0:N)
  (пример: особа може да нема телефон, или да има један или више њих, а телефон може бити у власништву особе или бити без власника);
- један према више (1:N)
  (пример: продавац издаје више наруџбеница, а свака наруџбеница издата је од стране једног продавца);

- више према више (M:N)
  (пример: студенти похађају наставу више наставника, а наставници држе наставу већем броју студената).

Теоријски, кардиналност бинарне везе може бити и:

- 0:0 означава да оба ентитета у вези могу да постоје невезано један за други;
- 1:0 означава да ентитет условљава постојање другог (пример: особа може, а не мора, бити програмер, али програмер свакако мора бити особа).

Мада на први поглед везе у релационом моделу могу изгледати као нешто непотребно и компликовано, заправо су корисне у раду са релационим базама података јер се њиховом употребом клијенти базе ограничавају на правилне уносе. Примера ради адекватном везом може се лако постићи да на свакој наруџбеници мора да буде уписан и службеник који је извршио продају (што је корисно).

Ентитети могу учествовати у везама делимично или потпуно. Уколико објекти ентитета не могу да постоје без учествовања у вези (зависган, односно слаб ентитет) реч је о потпуном учествовању, у супротном (независган, односно јак ентитет) оно је делимично.

Једноставности ради, још једном може бити употребљен претходни пример у коме купац купује робу. Веза која може послужити као пример за потпуно учествовање може бити веза између ентитета Sluzbenik и Narudzbenica. Ни једна појединачна наруџбеница (n-торка у оквиру релације Narudzbenica) не може да постоји у бази података уколико не постоји службеник (n-торка у оквиру релације Sluzbenik) који је унео дату поруџбину. У овом примеру учествовање релације Sluzbenik у вези са релацијом

Narudzbenica је делимично, пошто n-торка релације Sluzbenik може да постоји и без постојања било какве n-торке у релацији Narudzbenica (примера ради тек се запослио службеник и није унео ни једну наруџбеницу), али је учествовање релације Narudzbenica у овој вези потпуно (јер на свакој наруџбеници мора да фигурише и особа која ју је издала - један службеник). Narudzbenica је слаб ентитет, јер зависи од ентитета Sluzbenik. Уобичајено је да се Narudzbenica направи тако да зависи и од релације Kupac (пошто за сваку наруџбину мора да се зна и особа која је извршила куповину). Kupac и Sluzbenik у конкретном примеру су јаки ентитети, јер не зависе од других. Ако се овако дефинишу везе приликом прављења модела података у некој бази података, DBMS неће дозволити да у тој бази података постоји ни једна наруџбеница на којој је непознато ко је купац, односно продавац (очигледно је да је то корисно). У овом примеру (као и у било којем другом) тврдња за тип везе мора бити тачна за све објекте (инстанце), односно важи за све n-торке повезаних релација.

03.05

Неке особености релација

Слике и текст који следи су резиме до сада реченог. За ваљано разумевање наредног текста неопходно је усвојити релациону терминологију.

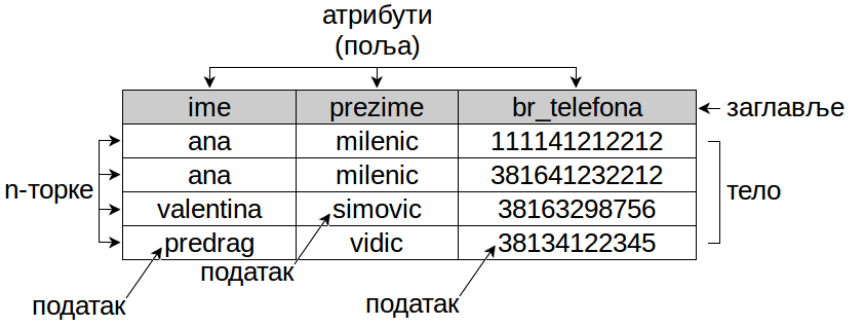
реални свет	релациони модел	релациона база података
ентитет (класа)	релација	табела
објекат (инстанца)	n-торка	ред табеле
атрибут	атрибут (поље)	колона табеле
вредност атрибута	податак	вредност записана у ћелији табеле

Сл. 4: Релациона терминологија

Да би претходна слика била јаснија, може се искористити већ поменути пример. У реалном свету, односно систему из реалног света, за ентитет (класу) може послужити Купац. Тај ентитет формира се на основу купаца из реалног света. Ентитет се у релационом моделу представља релацијом Купац, која се у релационој бази података представља једном табелом. У оквиру релације дефинишу се све заједничке карактеристике (атрибути) свих купаца (име купца, презиме купца...). Објекат (инстанца класе) у овом примеру је један конкретан купац. У релацији се један објекат записује као једна n-торка (где је n број атрибута у релацији), односно један ред табеле у бази података. Атрибути карактеришу ентитете. Све релевантне карактеристике ентитета издвајају се као њихови атрибути. Сваки атрибут се у релационом моделу приказује као једно поље релације, а у бази података то је једна колона табеле. За сваки појединачни објекат (n-торку релације), односно ред табеле, у оквиру базе података вредности атрибута уносе се у пресецима редова и колона, односно ћелијама табеле. Вредности атрибута су подаци у бази података.

У складу са релационом терминологијом у наредном примеру приказана је једна релација (слика 5). Комплетна структура приказана на слици која следи јесте релација. На слици су означене

и компоненте релације. Свака n-торка (у овом случају је n = 3, пошто је степен релације 3) често се назива торка. На основу релација модела формирају се табеле у релационој бази података.



Сл. 5: Структура једне релације

Ни једна релација, односно табела, у оквиру било које релационе базе података није уређена, односно редослед торки није уређен. Празна релација је такође релација. Број торки је кардиналност (eng. cardinality) релације (у приказаном примеру кардиналност релације је 4, пошто има укупно 4 реда табеле). Број атрибута (поља) је степен релације. Вредност сваког атрибута (податак) може бити унета само у складу са његовим доменом. Релацију је најједноставније разумети као скуп.

03.06

Модел објекти-везе

У ранијем тексту више пута помињани су дијаграми којима се илуструју модели података за потребе база података. Једна од најпознатијих варијанти дијаграма тог типа позната је под називом

дијаграми модела објекти-везе, односно E/R дијаграми.

Питер Чен (ch. 陳品山, eng. Peter Pin-Shan Chen) је 1976. године предложио ER модел (eng. entity-relationship model), односно МОВ - модел објекти-везе. Модел објекти-везе приказује ентитете, њихове атрибуте и везе између ентитета. Везе између ентитета практично дефинишу везе између свих објеката тих ентитета. Овај научник једновремено је осмислио концепт графичког приказивања (нотације) оваквих модела путем МОВ дијаграма, односно E/R дијаграма.

У складу са концептом E/R дијаграма:

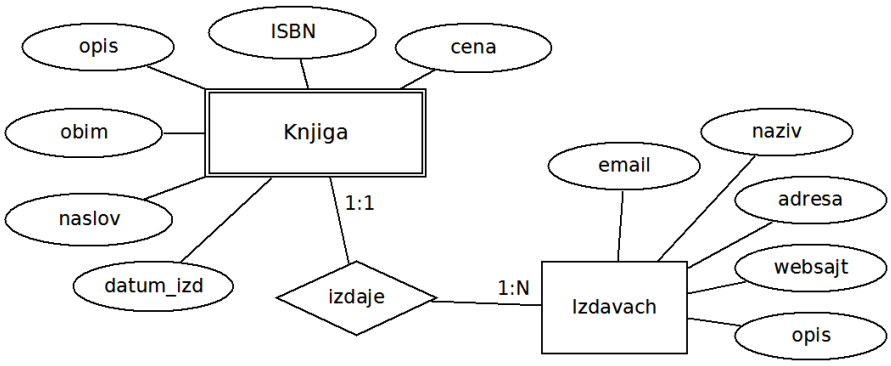
- правоугаоник репрезентује ентитет;
- елипса репрезентује атрибут;
- ромб репрезентује везу.

Осим тога поред линија које спајају ентитете, односно веза, наглашава се (уобичајено бројевима) њихова кардиналност.

Слаби ентитети означавају се правоугаоницима са дуплом линијом. Објекти слабих ентитета зависни су од постојања објеката других ентитета у моделу. Дуплом линијом често се означавају и ромбови - везе које ће у релационом моделу произвести нове релације, односно нове слабе ентитете. Ово ће бити детаљније објашњено на примеру у петом поглављу ове књиге.

На поједностављеном примеру који следи приказан је пример E/R дијаграма (слика 6). На слици је лако уочити да ентитет Knjiga има 6 атрибута datum_izd, naslov, obim, opis, ISBN и cena, као да и ентитет Izdavach има 5 атрибута email, naziv, adresa, websajt и opis. Ентитети Knjiga и Izdavach су у вези према којој једна књига има једног издавача, а један издавач може издати више књига. Пошто је ентитет

Knjiga слаб, означен је дуплом линијом (јер за сваку књигу мора да постоји један издавач). Ентитет Izdavach није слаб, што је уочљиво одмах, пошто нема дупле линије на правоугаонику.



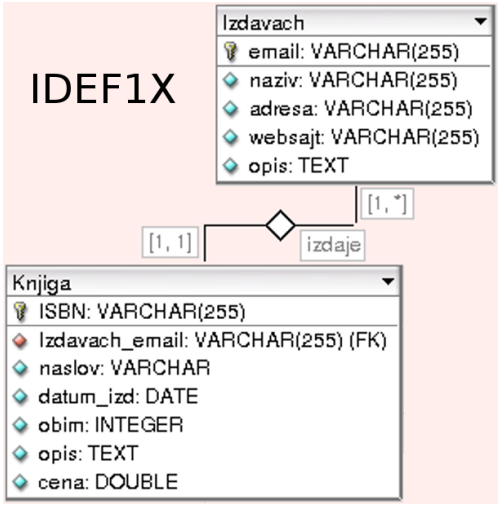
Сл. 6: MOB дијаграм (E/R дијаграм)

У складу са дијаграмом приказаним на претходној слици један издавач (објекат ентитета Izdavach) може да постоји у моделу без обзира што није издао ни једну књигу (објекат ентитета Knjiga). За разлику од тога свака књига (објекат ентитета Knjiga) мора да буде повезана са својим издавачем (објекат ентитета Izdavach) јер у супротном не може да постоји у тако направљеној бази података.

03.07 Модел IDEF1X

Поред класичне нотације у дизајну модела података за потребе база података у употреби су и разне друге. У овој књизи ће бити приказан још један познати стандард за дијаграме - IDEF1X (eng. Integration definition for information modeling). Стандард IDEF1X веома је ефикасан у приказивању модела релационих база података.

У оквиру овог стандарда ентитети су приказани правоугаоницима, у оквиру којих су листе њихових атрибута. Мада приказ варира у зависности од програма који цртају дијаграме, уобичајено поред назива атрибута стоји спецификација њихових типова. Ромбовима су приказане везе, а кардиналности најчешће бројевима (слика 7).



Сл. 7: IDEF1X дијаграм

На претходном примеру уочљиво је да релација Knjiga има 7 атрибута, а релација Izdavach има 5. Поред сваког атрибута стоји и његов тип. Први атрибут са листе релације Knjiga (ISBN), као и први атрибут са листе релације Izdavach (email) користе се као примарни кључеви. Због тога су означени сличицама кључева. Пошто је атрибут Izdavach_email релације Knjiga преузет из релације Izdavach, тај атрибут је и у функцији страног кључа. Због тога стоји поред типа овог атрибута ознака (FK), која се уобичајено поставља.

О кључевима релација биће више речи у наредном поглављу ове књиге.

03.08

Основне одлике објектних база података

Објектни, односно објектно-оријентисани, модел база података потпуно је различит од релационог. Основна идеја код ових база података је да се подаци у оквиру базе података чувају према моделу који је дефинисан у програмском језику.

Из перспективе програмера објектни модел база података може се тумачити као далеко природнији. Моделе података које програмер специфицира у оквиру класа свог програмског кода (који је написан у неком објектном, односно објектно-оријентисаном високом програмском језику) практично наслеђује и објектна база података. Тако се избегава превођење из једног модела који се користи у програмском коду (објектног, односно објектно-оријентисаног модела) у други који се користи у бази (углавном релациони модел).

Модели података које објектна база треба да користи дефинисани су у оквиру класа програмског кода (примера ради C++, Java...). Инстанце класа, односно објекти (еквивалент n-торкама релационог модела), чувају се (налик на низове) у објектним базама података. Овакав концепт у потпуно је природан из перспективе програмера јер се на практично идентичан начин поступа са објектима у програмским језицима и у базама података.

Стање објеката које објектна база чува, дефинисано је вредностима њихових атрибута. Атрибути објеката су специфицирани у оквиру дефиниција њихових класа у програмском коду. Уобичајено је да се поред података, у оквиру дефиниције класа, специфицирају и методе. Те методе се углавном користе за промену стања објеката

који су инстанце тих класа.

Сви објекти у оквиру објектне базе података изгледају исто као и објекти у коду програмског језика. У зависности од класа које их дефинишу ти објекти могу бити различитих сложености. Због тога се може рећи да објектни модел база података интегрише карактеристике објектних програмских језика и база података у јединствену целину [В.Девеџић, 2000].

У новије време постоји и техника која се у програмирању понекад користи, а позната је као ORM (eng. Object-relational mapping). Може се рећи да је тај процес креирање „виртуелне објектне базе података“ од уобичајене релационе базе података. Софтверски алати који врше овај процес заправо праве класе у оквиру кода датог објектног (објектно-оријентисаног) програмског језика на основу модела података дате релационе базе података. Овим процесом програмеру постају доступни, у програмском коду објектног (објектно-оријентисаног) високог програмског језика који употребљава, типови података из релационе базе података. Разлог због којег се ова техника користи и јесте тај што се на поменути начин програмерима омогућава природнији приступ моделима података који се користе у релационој бази података. Може се рећи да је идеја за настанак ове технике инверзна идеји за настанак објектне базе података.

04 Основе пројектовања релационих база података

У развоју информационих система ваљано пројектовање базе података је од круцијалног значаја. У овом делу књиге на основном нивоу биће више речи о овој процедури, када је реч о пројектовању релационе базе података.

Приликом пројектовања релационе базе података основна ствар је дефинисање модела података. Модели података који се у току пројектовања дефинишу, биће употребљавани од стране клијента базе података. Када је реч о развоју информационих система, клијент базе је апликација (компјутерски програм).

Редунданса коју било какав модел података повлачи са собом треба да буде минимална. То практично значи да се пројектант труди да редундантно чување података сведе на прихватљиви минимум. Суштински квалитетан модел треба да буде у стању да одговори на свако „разумно“ питање које може да буде постављено, односно мора да буде флексибилан.

Да би досадашњи текст био јаснији, флексибилност о којој је реч може се илустровати примерима. Замислимо информациони систем неке компаније. Питање колико запослених старости између 40 и 45

година има у компанији потпуно је „разумно“. Због тога модел мора да буде конципиран тако да се једноставним упитом може добити одговор. За разлику од претходног, питање које су друге речи у називима улица адреса становања запослених у компанији не би могло да буде третирано као „разумно“, те квалитетан модел не мора да буде конципиран тако да се једноставним упитом може добити одговор.

Принципијелно, у било како направљеном моделу података у оквиру неке базе података клијент може са одговарајућим упитом да пронађе то што му је потребно. Уколико је, међутим, модел лоше конципиран клијент ће морати да постави компликован упит да би добио то што му је потребно. Компликован упит захтева да компјутер више времена проведе док га не изврши, што успорава његов рад. То је слично као што се на великој гомили насумично набацаних књига библиотекар далеко лошије сналази него у лепо сортираној библиотеци. У случају те велике гомиле библиотекар мора да изврши далеко компликованију радњу док не пронађе жељену књигу, а то захтева више његовог времена. Због тога се полице библиотеке, као и модел података у базама података, конципирају тако да се „разумни упити“ ефикасно (са мало утрошеног времена и рада) могу извршити.

И у добро осмишљеним пројектима информационих система компанија се, како време пролази, појављује потреба за надоградњама. Када су модели података у оквиру базе података квалитетно решени, било какве њихове надоградње су далеко једноставније. То је још један, важан разлог због којег је креирање ваљаних модела у бази података битно.

narudzbenica_br	kupac	telefon_kupca	cena
1	Иван	123223	100
2	Маја	113333	130
3	Ана	554333	100
4	Лаза	135677	200
5	Ана	554333	420

Сл. 8: Редундантно чувани подаци

Када је реч о редундантно чуваним подацима може се, ради илустрације, употребити следећи поједностављени пример (слика 8). Нека се приликом сваке продаје у некој радњи формира наруџбеница и нека се у оквиру сваке уноси и број телефона купца. Јасно је да се у том случају сваки пут када се иста особа појави као купац наново мора унети број телефона те особе. Мада на први поглед не изгледа проблематично, поред проблема са компликованијим уносом овакав модел повлачи и друге проблеме. Шта би се догодило уколико би особа која је редовни купац променила број телефона? На приказаном примеру видимо да Ана као купац фигурише на две наруџбенице. Уколико она промени број телефона да ли то значи да ће на свим претходним наруџбеницама да остане стари? Ако остане стари, који је стварни телефон, пошто сада у бази фигуришу два? Ако се, ипак, мења на свим наруџбеницама, колико пута је неопходно извршити ту измену и на који начин?

И у једноставном примеру, какав је био претходни, лако је уочити да редундантно чување података у употреби неке базе података повлачи са собом много проблема. У претходном примеру приказано је редундантно чување података у оквиру једне релације. Проблем генерално може бити и са редундантним чувањем података у више релација.

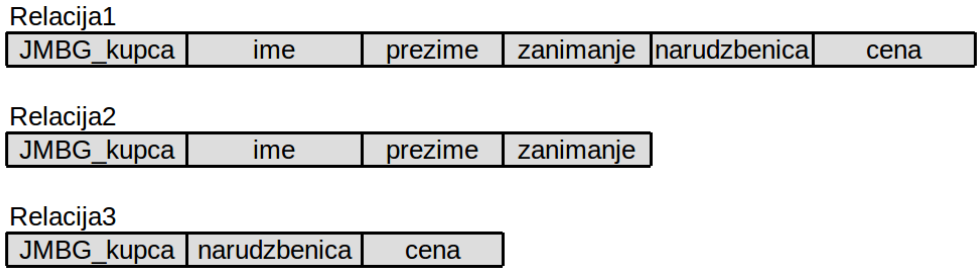
04.01
Атомске вредности атрибута

Приликом формирања релација у неком моделу података у релационој бази података атрибути се дефинишу тако да њихове вредности буду атомске. Атомска вредност атрибута, у релационом моделу база података, је вредност која не може да се дели на мање смислене делове. Примера ради, атомске вредности могу бити Adidas (назив произвођача), Врњачка Бања (назив места), 45 (величина патика), Марка Краљевића 38 (улица и број)... Јасно је да назив места Врњачка Бања може бити подељен на две речи које чине назив, а исто тако свака од речи може бити издељена на независна слова, али би се тиме добиле вредности које саме за себе не значе ништа (немају смисла).

Атомска вредност атрибута често може зависити и од контекста у оквиру базе података. Уколико се, примера ради, прави информациони систем за продају спортске опреме на нивоу Европе, онда се вредност атрибута Марка Краљевића 38 (улица и број) може третирати као атомска. Уколико се међутим, прави база података за прехрамбену продавницу у којој је већина купаца из неколико суседних улица, онда има смисла да се улица и број раздвоје као две независне атомске вредности, јер се, рецимо, може вршити нека статистичка анализа посета купаца продавнице на нивоу бројева у одређеној улици.

04.02
Декомпоновање релација

Релације које су сложене, односно релације са великим бројем атрибута, могуће је раздвајати у више једноставнијих. Тада се добија више једноставних релација, односно релација са мање атрибута. Овај процес назива се декомпоновање без губитка (eng. lossless decomposition). Декомпоновање без губитка често се назива и декомпоновање. Разлог за други део назива лежи у томе што се њиме акцентује особина да се употребом декомпоновања без губитка не губе атрибути (нема губитка података). Декомпоновање без губитка је процес који се примењује да се на тај начин елиминише редундантност. Следећи пример илустроваће овај процес.



Сл. 9: Декомпоновање без губитка

На приказаном примеру (слика 9) може се видети декомпоновање релације Relacija1 на две нове релације Relacija2 и Relacija3. Лако је уочити да су сви атрибути који су постојали у иницијалној релацији присутни и у нове две. Овим процесом добијене су релације са мањим бројем атрибута. Додатне предности које такав концепт повлачи биће јасније након текста који следи.

04.03
Зависности атрибута

У великом броју случајева приликом одабира атрибута, у току стварања модела података за потребе базе података, постоје њихове међусобне зависности. Ради илустрације може се замислити било каква релација у оквиру базе података неке продавнице. Рецимо да у оквиру релације Proizvod постоје само 3 атрибута shifra_proizvoda, cena и pdv (уобичајено их наравно има више). Већ на први поглед лако је закључити да ће од самог производа (његове шифре која га идентификује) зависити цена производа, а од цене да зависи порез.

Ако вредност атрибута A једнозначно одређује вредност атрибута B каже се да атрибут B функционално зависи од атрибута A. Оваква зависност се записује A → B.

Детерминанта је атрибут (или скуп атрибута) о којем је неки други атрибут функционално зависан.

Уколико се вратимо на илустрацију која је горе наведена може се записати shifra_proizvoda → cena, као и cena → pdv. У датом примеру може да се каже да је cena детерминанта за pdv, као и да је shifra_proizvoda детерминанта за cena.

Каже се да C транзитивно зависи од A ако B функционално зависи од A, а C функционално зависи од B. Јасно је да се за наведену илустрацију може рећи да pdv транзитивно зависи од shifra_proizvoda.

Ако вредност атрибута B функционално зависи од скупа атрибута A, а функционално не зависи ни од једног подскупа A каже се да B

потпуно функционално зависи од A.

Да би била илустрована потпуна функционална зависност, у случају када постоји више атрибута (у случају једног је тривијалан доказ) од којих функционално зависи неки атрибут релације, биће употребљена измењена варијанта релације Proizvod. Рецимо да у релацији Proizvod постоје атрибути naziv (у који се записују називи производа - рецимо модела спортских патика), boja (у који се записује боја производа) и p_n_stanju (у који се записује број производа у магацину). Рецимо да патике које се зову Предатор постоје у две различите боје. Нека у магацину црвених патика Предатор има 15, а наранџастих 17. У зависности од назива производа (вредности атрибута naziv) и боје производа (вредности атрибута boja) зависи број датих производа у магацину (вредност атрибута p_n_stanju). На основу тога може се записати (naziv, boja) → p_n_stanju. Пошто атрибут p_n_stanju не зависи ни од атрибута naziv, ни од атрибута boja (појединачно), каже се да p_n_stanju потпуно функционално зависи од (naziv, boja).

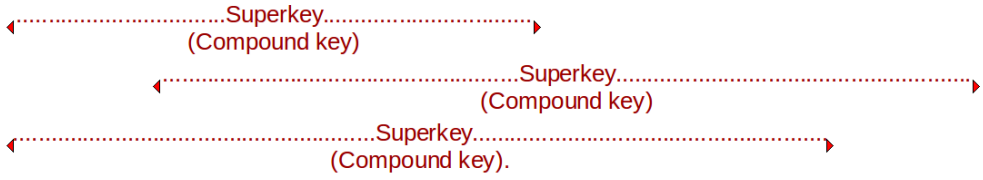
04.04

Кључеви релација

У тексту који следи биће описани кључеви релација. Сваки опис, односно дефиниција, ради бољег разумевања, биће илустрована и примером. Кључеви могу бити различитих типова и у складу са типом различитих намена. У развоју релационих база података кључеви се увек користе.

Атрибут или скуп атрибута релације може се употребити као

кључ, или део кључа. Кључ који се састоји од само једног атрибута назива се још и једноставни кључ (eng. simple key). Кључ који се састоји од више атрибута назива се композитни кључ или сложени кључ (eng. compound key, composite key).



Studenti

JMBG	email	br_indeksa	god_upisa	ime	prezime
3112345565	petar@hehe.com	34	2012	Петар	Лазић
1254512535	marko@yahoo.com	2	2012	Марко	Марић
1243614366	ana@hotmail.com	15	2011	Ана	Аничић
6634653465	maja@asdf.com	17	2012	Марија	Вишњић
3465346566	nina@nat.org	11	2012	Наталија	Николић

Сл. 10: Суперкључ - SK

Било који скуп атрибута који једнозначно одређују један ред релације (табеле у бази података) (слика 10), односно једну торку у релацији, назива се суперкључ - SK (eng. superkey).

Studenti

JMBG	email	br_indeksa	god_upisa	ime	prezime
3112345565	petar@hehe.com	34	2012	Петар	Лазић
1254512535	marko@yahoo.com	2	2012	Марко	Марић
1243614366	ana@hotmail.com	15	2011	Ана	Аничић
6634653465	maja@asdf.com	17	2012	Марија	Вишњић
3465346566	nina@nat.org	11	2012	Наталија	Николић

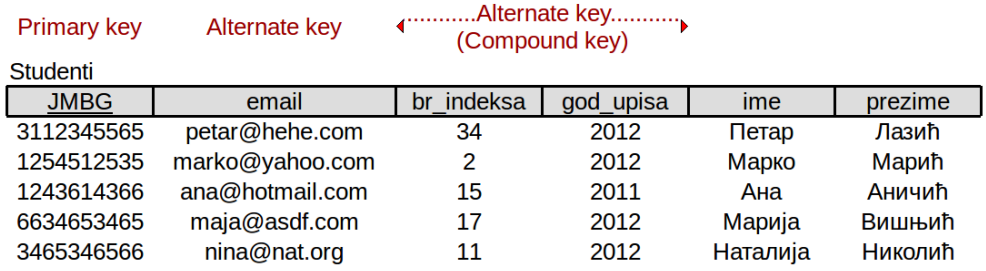
Сл. 11: Кључ кандидат - CK

Кључ кандидат - CK (eng. candidate key) је минимални суперкључ (слика 11), јер се не може раставити на једноставнији (уколико би се додатно раставио више не би једнозначно одређивао једну торку

релације). Овај кључ назива се кандидат кључ пошто било који овакав кључ представља кандидата за примарни кључ - може бити употребљен као примарни кључ.

Кључни атрибут (eng. key attribute) је сваки атрибут који улази у састав било ког кључа кандидата.

Алтернативни кључ - AK (eng. alternate key) је било који кључ кандидат који није подешен као примарни кључ. Назива се алтернативни пошто је алтернатива за примарни кључ.



Сл. 12: Примарни кључ - PK и алтернативни кључ - AK

Примарни кључ - PK (eng. primary key) је скуп атрибута, односно атрибут, који једнозначно одређује једну n-торку у релацији (један ред табеле у бази података). Било који кандидат кључ може бити подешен (одабран) као примарни кључ.

Страни кључ - FK (eng. foreign key) је атрибут (или скуп атрибута) једне релације који (је повезан са n-торком друге релације) једнозначно одређује торку друге релације. Овај кључ се користи за повезивање релација пошто у једну релацију уноси кључ кандидат (уобичајено примарни кључ) из друге релације, тако да једнозначно одређује (повезује) једну торку у другој релацији. Употреба страног кључа биће илустрована примерима.

На слици 7 приказана је употреба страног кључа за реализацију бинарне везе кардиналности један према више. Примарни кључ релације Izdavach (email) у датом случају је употребљен као страни кључ у релацији Knjiga (Izdavach_email), да би приликом уноса свака појединачна књига (објекат ентитета Knjiga) била увезана са одговарајућим издавачем (објектом ентитета Izdavach).

Studenti

JMBG	email	br_indeksa	god_upisa	ime	prezime
3112345565	petar@hehe.com	34	2012	Петар	Лазић
1254512535	marko@yahoo.com	2	2012	Марко	Марић
1243614366	ana@hotmail.com	15	2011	Ана	Аничић
6634653465	maja@asdf.com	17	2012	Марија	Вишњић
3465346566	nina@nat.org	11	2012	Наталија	Николић

Primary Key
(Compound key)
Foreign Key 1 + Foreign Key 2

Uchesnici

JMBG	sport	semestar
3112345565	фудбал	летњи
1243614366	фудбал	летњи
3112345565	кошарка	летњи
6634653465	кошарка	зимски

Sportovi

sport	tip	lokacija
фудбал	такмичарски	сц Владе Дивац
кошарка	рекреативни	кк Младост

Сл. 13: Страни кључ - FK

Атрибут JMBG једнозначно дефинише торку релације Studenti, а атрибут sport једнозначно дефинише торку релације Sportovi (слика 13). Ови атрибути „позајмљени“ су релацији Uchesnici. Пошто су преузети из других релација у којима фигуришу као кључеви кандидати (који су подешени као примарни кључеви у датом примеру) у релацији Uchesnici ови атрибути фигуришу као страни кључеви, а заједно формирају примарни кључ ове релације (релације Uchesnici). На тај начин у овом случају реализована је бинарна веза кардиналности више према више између релација Studenti и Sportovi (уз употребу референтне табеле, односно релације Uchesnici).

04.05

Нормалне форме

Да би се смањила редундантност и обезбедила флексибилност, приликом стварања модела података у релационим базама података, примењује се поступак који се назива нормализовање (eng. normalization). У поступку нормализовања примењују се нормалне форме.

Нормалне форме - NF (eng. normal forms) су правила по којима се врши нормализовање структуре података. Оне су конципиране са растућим степеном строгости. Због тога се свака надовезује на претходну тако што спречава одређене врсте аномалија. Нормалне форме су примењиве на релације у релационом моделу података. За саму базу података се може рећи да је у нормалној форми n уколико све њене табеле задовољавају нормалну форму n-тог реда.

Концепт нормалних форми први је предложио др Е.Ф.Код, у време када је радио за компанију IBM. Дефиниције нормалних форми поставио је први овај научник 1971. године.

Обзиром на то да се једна за другом нижу са растућим степеном строгости нормалне форме практично одређују степен рањивости. Што је на релацију примењивија нормална форма вишег реда, то је релација отпорнија на аномалије (уноса, брисања, промене). HNF (eng. Highest normal form) је нормална форма највишег реда која је примењива на релацију. Релација задовољава критеријуме свих нормалних форми које су нижег и једнаког реда HNF. Треба да је очигледно да је ово тачно због растућег степена строгости нормалних форми у низу.

У тексту који следи биће објашњене основне нормалне форме. Да би њихов смисао био јаснији поред објашњења и дефиниција биће упоредно приказан и илустративни пример. Нормалне форме које су изложене у следећем тексту, најчешће су у употреби приликом пројектовања релационих база података.

За пример ће послужити замишљена база података радње са спортском опремом. У оквиру те базе за сваки производ постоји фирма, односно произвођач (firma), назив производа (proizvod), боја (boja), гаранција (garancija), цена (cena), број примерака у магацину (p_n_stanju)[6]. Нека за дати пример важе функционалне зависности које су приказане у табели која следи.

shifra	→	firma
shifra	→	proizvod
proizvod	→	cena
shifra	→	cena
firma	→	garancija
(proizvod, boja)	→	p_n_stanju
(shifra, boja)	→	p_n_stanju

Табела 1: Функционалне зависности за пример базе података радње са спортском опремом.

04.05.1 Прва нормална форма

Релација је у Првој нормалној форми - 1NF (eng. First normal form) уколико је вредност сваког њеног атрибута атомска и уколико је свака њена торка јединствена (одређена употребом примарног

6 У заградама су приказана имена одговарајућих атрибута у оквиру релационе базе података.

кључа).

1NF је основна нормална форма. Њоме се гарантују атомске вредности атрибута и јединственост сваке торке. Уколико вредности атрибута нису атомске, табела у оквиру базе података престаје да буде флексибилна (њена употребна вредност се драстично смањује). Разлог за то је што би тада упити који би морали да буду постављани DBMS, да би корисник могао да добија жељене информације, морали да буду компликованији.

Relacija1

proizvodi	boje	cena	garancija	p_n_stanju
ni4gw1af, Nike, Tmax	црвена, зелена	10	3	25
pu3q2we5, Puma, shirtOne	плава	14	5	7
puiiys44, Puma, ClassyT	зелена	12	5	11

Сл. 14: Релација није у Првој нормалној форми

На претходном примеру (слика Error: Reference source not found) уочљиво је да вредност атрибута proizvodi није атомска. Ова вредност садржи шифру, име произвођача и назив производа. Уколико би рецимо клијент базе података, из табеле направљене према приказаној релацији, желео да прочита само назив жељеног производа, морао би да вредност коју добије из атрибута proizvodi накнадно дели на сегменте док не добије жељену информацију. Тада би се морао задати компликованији упит, а компјутер би утрошио више времена за проналажење тражене информације.

Уколико у претходној релацији (слика Error: Reference source not found) није дефинисан примарни кључ, могуће је имати два потпуно иста производа у виду две различите торке. То би учинило релацију прилично неупотребљивом, примера ради, не би било јасно колико је продатих примерака тог производа (ако је претходно тај број промењен у само једној од њих).

Relacija1

shifra	firma	proizvod	boja	cena	garancija	p_n_stanju
ni4gw1af	Nike	Tmax	црвена	10	3	15
ni4gw1af	Nike	Tmax	зелена	10	3	10
pu3q2we5	Puma	shirtOne	плава	14	5	7
puiiys44	Puma	ClassyT	зелена	12	5	11

PK (shifra, boja) AK (proizvod, boja)

Сл. 15: Релација испуњава услов Прве нормалне форме

На слици 15 приказана је релација која задовољава 1NF. Већ на први поглед требало би да је јасно да она није осетљива на проблеме који су поменути у претходном тексту. Ова релација добијена је нормализацијом у Прву нормалну форму релације из претходног примера.

У приказаној релацији (слика 15) уочљиво је да су вредности атрибута атомске. Њихово даље дељење не би било смислено у датом контексту. Осим тога битно је приметити да су сада атрибути shifra и boja фиксирани као композитни примарни кључ (називи ових атрибута су подвучени). Тиме је гарантовано да производ који је одређен шифром и бојом може бити приказан само једним редом у оквиру дате табеле у бази података (једном торком релације). Након тога јасно је да потпуно дефинисано колико на стању има производа (дефинисаних шифром и бојом).

Као примарни кључ претходне релације (слика 15) могао би да послужи и скуп атрибута proizvod и boja. Пошто би и ова два атрибута могла да буду употребљена као примарни кључ, они чине композитни алтернативни кључ дате релације. У складу са тим кључни атрибути приказане релације су shifra, proizvod и boja.

04.05.2
Друга нормална форма

Релација је у Другој нормалној форми - 2NF (eng. Second normal form) уколико је у Првој нормалној форми и сваки њен атрибут који није кључни атрибут потпуно функционално зависи од сваког кључа кандидата.

Битно је приметити да у Другој нормалној форми атрибути који нису кључни атрибути могу зависити и транзитивно од кључа кандидата, а да ова нормална форма не буде нарушена. Осим тога из дефиниције је јасно да уколико релација која је у Првој нормалној форми има само један кључ кандидат који се састоји од једног атрибута, по аутоматизму задовољава и Другу нормалну форму.

На основу дефиниције Друге нормалне форме јасно је да је она строжија у односу на Прву нормалну форму. Разлог за то је што, да би испунила Другу нормалну форму, релација мора задовољити услове Прве нормалне форме, као и додатни услов. Овим додатним условом захтева се да сви атрибути који нису кључни атрибути морају да потпуно зависе од сваког кандидата за кључ.

Релација која је приказана на претходном примеру (слика 15) не задовољава Другу нормалну форму. Разлог за то је што атрибут firma, према задатим функционалним зависностима (табела 1) не зависи потпуно од кандидата за кључ (пошто важи shifra → firma), јер су кандидати за кључ композитни и састоје се од атрибута shifra и boja, или proizvod и boja.

Relacija1

shifra	boja	proizvod	p_n_stanju
ni4gw1af	црвена	Tmax	15
ni4gw1af	зелена	Tmax	10
pu3q2we5	плава	shirtOne	7
puiiys44	зелена	ClassyT	11

Relacija2

shifra	firma	garancija	cena
ni4gw1af	Nike	3	10
pu3q2we5	Puma	5	14
puiiys44	Puma	5	12

Сл. 16: Релације испуњавају услов Друге нормалне форме

Да би се избегао поменути проблем могу се, употребом декомпозиције без губитка, формирати две релације (слика 16) које задовољавају све тражене услове за Другу нормалну форму. На примеру који је приказан на претходној слици лако је увидети да у складу са задатим функционалним зависностима Relacija1 има два композитна кандидата за кључ (shifra, boja), као и (proizvod, boja). У истом примеру Relacija2 има само један једноставни кандидат за кључ који чини атрибут shifra. Пошто у обе релације сви атрибути који нису кључни атрибути потпуно зависе од свих кандидата за кључеве, обе релације у овом примеру задовољавају Другу нормалну форму.

04.05.3 Трећа нормална форма

Релација је у Трећој нормалној форми - 3NF (eng. Third normal form), уколико је у Другој нормалној форми и ако су сви њени атрибути који нису кључни атрибути међусобно независни.

Пошто се Трећом нормалном формом захтева да атрибути који нису кључни буду међусобно независни, транзитивне зависности се елиминишу из релација.

Једну од најпознатијих алтернативних дефиниција Треће нормалне форме поставио је 1981. године Карло Заниоло (it. Carlo Zaniolo). Ради бољег разумевања ове нормалне форме у тексту који следи ће бити дата и дефиниција коју је поставио овај научник.

Релација је у Трећој нормалној форми ако је за свако X → A испуњен бар један од услова [C.Zaniolo, 1982]:

- A је садржано у X
- X је суперкључ
- разлика A - X састоји се из једног или више кључних атрибута

Да би разумевање концепта Треће нормалне форме било једноставније биће употребљен претходни пример. На примеру (слика 16) се може уочити да Relacija2 не задовољава Трећу нормалну форму. Према првој дефиницији Треће нормалне форме која је дата наглашено је да атрибути који нису кључни морају бити међусобно независни, а у релацији постоји функционална зависност firma → garancija. Осим тога ни један од услова који су набројани у другој дефиницији Треће нормалне форме за поменуте атрибуте није испуњен, те релација не испуњава ову нормалну форму.

Relacija2

shifra	firma	cena
ni4gw1af	Nike	10
pu3q2we5	Puma	14
puiiys44	Puma	12

Relacija3

firma	garancija
Nike	3
Puma	5

Сл. 17: Релације испуњавају услов Треће нормалне форме

Уколико би се релација из примера декомпоновала на две приказане релације (слика 17), Трећа нормална форма била би

испуњена пошто је испуњава и Relacija1 (слика 16). За сваку од приказаних релација (Relacija1, Relacija2 и Relacija3) лако је уочити да су испуњени услови за Трећу нормалну форму јер нема међусобно зависних атрибута који нису кључни атрибути.

05 Пример пројектовања релационе базе података

Ради бољег разумевања концепта базе података и њене употребљивости, у тексту који следи биће дат пример пројектовања и имплементације једноставне релационе базе података.

У општем случају моделовање (дизајнирање) - пројектовање релационе базе података могуће је извести у четири корака:

- опис система на основу кога се формира модел (уобичајено је то систем из реалног света);
- модел објекти-везе;
- релациони модел;
- имплементација у систему за управљање базама података.

Након што се реализују сва четири поменута корака, нова база података спремна је за употребу. Пре него што корисници почну да употребљавају нову базу података, у склопу информационог система, она се тестира.

Ваљано пројектована и имплементирана база података касније уз много мање рада и на логичан начин може надоградити у складу са евентуалним новим потребама њених корисника.

05.01

Опис система на основу кога се формира модел

Систем из реалног света је уобичајено (али не и нужно) систем на основу кога се врши пројектовање базе података. Углавном се такав систем базира на пословању неке компаније. У зависности од намене базе података, такав систем може имати и потпуно другачију конотацију (класификација биљака или неких других биолошких података, технички подаци о различитим врстама грађевинских материјала, категоризација небеских тела и тако даље). Такође је могуће моделовати базу података на основу система који није реалан (примера ради база података возила у свемирској опери Ратови Звезда).

У примеру који ће у овом делу књиге бити разматран потребно је пројектовати и имплементирати базу података за књижару. База података мора да подржи следеће:

(А) Унос нових књига - за сваку књигу мора да се формирају одговарајући подаци (ISBN, назив, област[и], аутор[и], биографија аутора, формат, обим, издавач, адреса издавача, цена, опис књиге..)

(Б) Наручивање књиге - приликом наручивања књиге формира се одговарајући формулар за наручивање у коме се евидентирају подаци о купцу који купује једну или више књига, адреси на коју треба да буду испоручене књиге, цени коју треба да плати, продавцу који реализује продају и датуму.

Уобичајено се у првом кораку у пројектовању базе података у разговору са пословодством компаније (у случају када се база података користи за неки пословни информациони систем) формира

опис налик на претходни текст. Када је реч о уобичајеним пословним процесима искусни пројектанти већ у току писања таквог текста могу да формирају прилично рафинисане моделе података. Да би овај процес био у потпуности разјашњен, у даљем тексту ће на основу датог описа, бити извршено детаљно разматрање.

05.02 Модел објекти-везе

На основу описа који је дат, односно описа који је направљен на основу система из реалног света, могуће је изградити модел објекти-везе (E/R модел). Дијаграм овог модела (E/R дијаграм) је доста разумљив и за особе које нису имале искуства са базама података. Готово је подразумевано потребно у току пројектовања нове базе података предочити њен дизајн и особама које нису упућене у техничке елементе овог процеса. За ту намену E/R дијаграм се често употребљава.

Модел објекти-везе је значајан јер се на основу њега може релативно лако извршити трансформација у релациони модел. Начин на који се ова трансформација врши у приличној мери је стандардизован, што повећава употребну вредност овог модела. Релациони модел се даље може једноставно имплементирати у сваки од савремених система за управљање релационим базама података.

Модел објекти-везе описује систем као скуп ентитета са својим атрибутима и њихових веза. Ентитет у моделу може бити формиран на основу неког ентитета из реалног система, али може бити и концепт реалног система. У оквиру овог модела везама су

дефинисани односи између ентитета.

05.02.1 Формирање ентитета модела објекти-везе

На основу описа система врши се дизајн модела. Први корак у изградњи модела објекти-везе је да се на адекватан начин формирају ентитети (који дефинишу објекте).

За исти опис система, могуће је направити више различитих квалитетних решења, односно модела објекти-везе (слично као што се једна фудбалска утакмица може квалитетно одиграти на много различитих начина). Због тога и ентитети у оквиру модела могу бити реализовани на више различитих начина. На пројектанту је да се определи за једну варијанту по којој ће реализовати модел. У тексту који следи биће издвојено више илустративних детаља о формирању ентитета модела у овом примеру.

На основу правила из 03.04.1 ове књиге, требало би да буде очигледно да ће у моделу бити потребан ентитет Knjiga (који се формира на основу мноштва књига из реалног система). За ентитет Knjiga у моделу објекти-везе јасно је да ће један од атрибута свакако бити ISBN (који одговара истоименом називу броја из реалног система којим је једнозначно одређена свака књига) као и naslov (који одговара називу књиге из реалног система).

Оно што се такође може лако уочити као кандидат за ентитет у моделу је Autor - јер једна особа може бити аутор више књига, те би креирање (једног или више) атрибута ентитета Knjiga за аутора било

погрешно пошто би повлачило редундансу. Рецимо да треба да буде подржано и то да једна књига може имати и више аутора. То је битно приметити иницијално, јер тај однос повлачи одређену кардиналност везе између новоформираних ентитета модела Knjiga и Autor.

У опису који је дат појављује се и издавач. Издавач може бити везан за више књига (може издати више књига), те би формирање новог атрибута објеката Knjiga за издавача било погрешно јер би то довело до редундансе. Тада би на много места, за сваку књигу датог издавача, морали у базу уносити телефон издавача, адресу издавача и тако даље (све податке које треба база да чува за сваког издавача). Уколико би издавач добио атрибут или више атрибута везаних за ентитет Knjiga ситуација би била иста као и када би било стављено да аутор добије атрибут или атрибуте везане за ентитет Knjiga. За издавача значи треба формирати нови ентитет - Izdavach. Нека ради једноставности, за овај пример важи да једна књига може имати само једног издавача. То значи да ће веза између ентитета Knjiga и Izdavach да буде различита (бити различите кардиналности) у односу на везу између ентитета Knjiga и Autor.

Да ли и опис треба да постане нови ентитет у моделу? - Не, јер опис директно зависи од конкретне књиге (постоји функционална зависност између примарног кључа, који може бити ISBN и описа) и не би требало да исти опис буде употребљен за две различите књиге. Због тога по аутоматизму за опис књиге треба направити атрибут opis ентитета Knjiga.

Други део задатка говори о наручивању књига. Већ се из самог исказа може уочити формулар о наручивању (наруџбеница) као ентитет Narudzbenica. Јасно је међутим да тај ентитет нема никаквог смисла без ентитета Kupac, те ће свакако морати да зависи бар од

овог ентитета. За купца се свакако мора формирати ентитет Kupac, јер би формирање неког атрибута ентитета Narudzbenica за купца повлачило редундансу налик на ону поменуту за формирање атрибута за аутора ентитета Knjiga (пошто исти купац може више пута куповати, исто као што један аутор може написати више књига). Narudzbenica ће морати да има и атрибут adresa_dostavljanja који говори о адреси преузимања (односно достављања), јер та адреса није нужно иста као и адреса купца (те је карактеристична за сваку појединачну наруџбину). Поред тога јасно је да мора да постоји и датум (односно време) наручивања као атрибут datvreme ентитета Narudzbenica. Време и датум куповине карактеристике су само једне конкретне куповине (једног објекта ентитета Narudzbenica, односно једне наруџбенице) и треба да фигуришу као атрибут (нема редундансе).

Захтев је да се евидентира у свакој наруџбеници и службеник који је попунио формулар и јасно је да ће један службеник више наруџбеница попунити. Због тога ће модел добити независтан ентитет Sluzbenik, од кога ће ентитет Narudzbenica морати да зависи.

05.02.2 Формирање веза између ентитета модела објекти-везе

Након што пројектант модела објекти-везе креира ентитете, неопходно је да правилно постави везе између њих.

Један аутор може написати више књига, а књига може имати више аутора. Због оваквог односа потребно је да веза између ентитета Knjiga и Autor буде кардиналности више према више.

Издавач може издати једну или више књига, а књига може имати једног издавача. На основу тога између ентитета Izdavach и Knjiga у моделу треба формирати везу кардиналности један према више. Уколико би база података морала да подржи и могућност да једну књигу може издати више издавачких кућа, онда би кардиналност везе морала бити више према више.

Уколико би требало ставити ограничење у бази података да свака књига мора имати издавача, онда би ентитет Knjiga морао бити слаб ентитет (који зависи од ентитета Izdavach). Тада би сваки објекат ентитета Knjiga морао бити повезан са једним објектом ентитета Izdavach.

Ако је подразумевано да се у бази података за сваку књигу мора унети и област књиге и формат у којој је књига штампана, треба ентитете Oblast и Format повезати са ентитетом Knjiga на истоветан начин као и ентитет Izdavach. Разлог што је добро за област и формат формирати ентитете (Oblast и Format) у оквиру модела је да би се избегла редунданса (као и у случају издавача).

На једној наруџбеници може фигурисати једна или више књига, а једна књига може бити више пута наручена. Због тога је кардиналност везе између објеката Knjiga и Narudzbenica више према више.

Исти купац може више пута куповати, тако да може фигурисати на више наруџбеница, а свака наруџбеница мора имати једног купца. Из овога је јасно да кардиналност везе између ентитета Kupac и Narudzbenica један према више.

Службеник може попунити једну или више наруџбеница, а свака наруџбеница мора имати службеника који је реализовао наручивање.

На основу овога у моделу треба формирати везу између ентитета Sluzbenik и Narudzbenica кардиналности један према више.

05.02.3 Формирање атрибута ентитета модела објекти-везе

Пошто се формирају ентитети (који дефинишу објекте) и везе у моделу објекти-везе, неопходно је додати и атрибуте ентитетима модела. Атрибути се додају ентитетима у складу са описом система на основу кога се формира модел. У табели (слика 18) су приказани атрибути ентитета формираног модела.

	Ентеитет	Атрибути
(А)	Autor	email, ime, prezime, srednje_slovo, biografija
	Izdavach	naziv, adresa, websajt, email
	Oblast**	naziv, opis
	Format*	***naziv, velichina
	Knjiga	ISBN, naslov, datum_izdavanja, obim, format*, opis, oblast**, cena
(Б)	Kupac	ime, prezime, email, adresa
	Sluzbenik	JMBG, ime, prezime, zanimanje
	Narudzbenica	datum-vreme, artikli, cena, adresa_dostavljanja

Сл. 18: Атрибути објеката модела објекти-везе

Пошто постоје тачно одређени формати - ***naziv (A4, B4, ...), који су јединствени за много књига, не би била лоша идеја да се формат издвоји као ентитет - Format*, јер би се тиме избегла редунданса (слика 18). Могло би да се постави питање да ли књига може бити штампана и у више различитих формата. Због једноставности ово

неће бити подржано у примеру.

Постоји доста области у које се књиге могу сврстати (слично као и за формате књига). Тако у једној области може бити сврстано више књига. Уколико се за област књиге може повезати више атрибута (опис области, кључни аутори у тој области...) добра идеја може да буде да област добије независтан ентитет - Oblast** (слика 18).

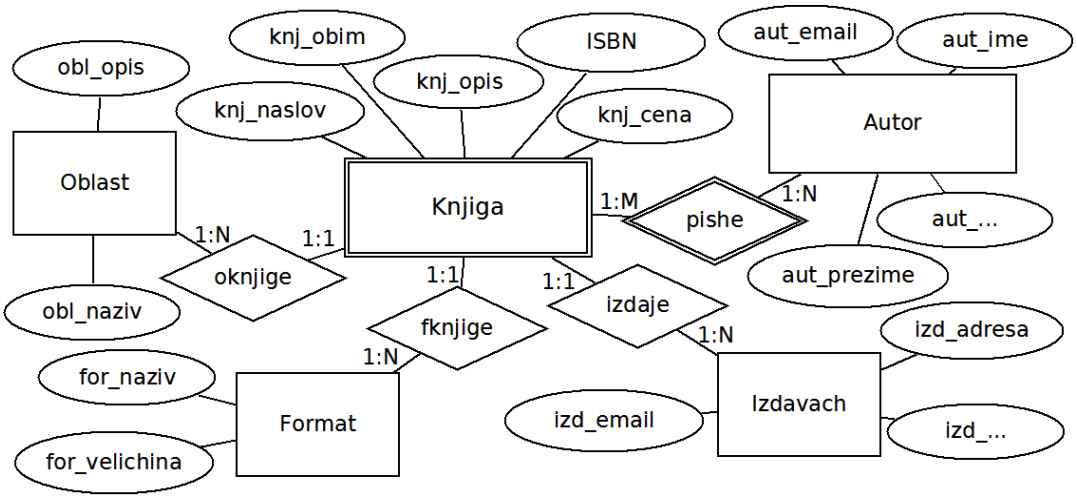
05.02.4
Дијаграм модела објекти-везе

Да би модел објекти-везе био јаснији уобичајено је приказати га и графички - E/R дијаграмом.

Обзиром на то да је задатак подељен у две целине, дијаграм се може поделити такође.

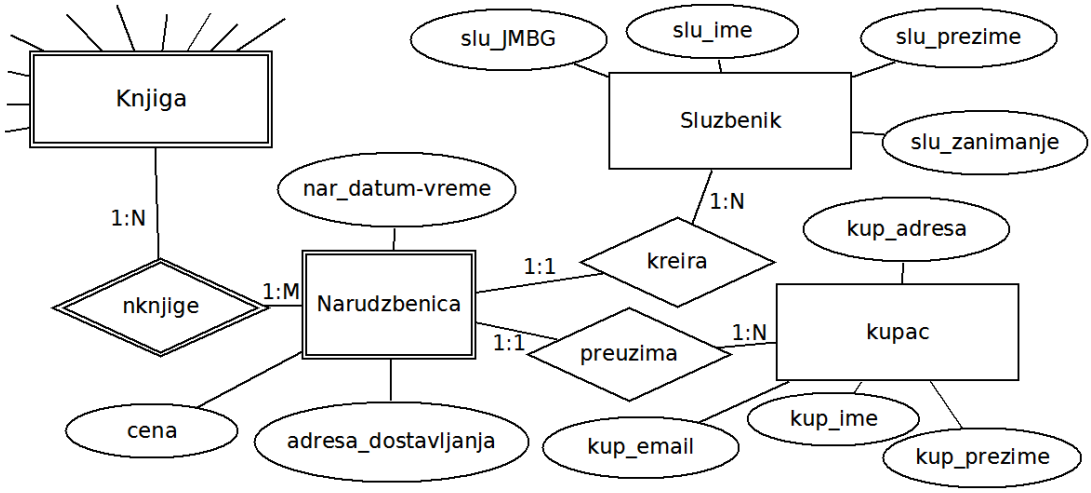
Ентитет Knjiga у оквиру модела (слика 19) је слаб. Разлог за то је што овај ентитет зависи од других ентитета, односно што сваки објекат ентитета Knjiga (свака књига) зависи од постојања неких других објеката (издавача, формата, области). Та особеност ентитета Knjiga може се назначити и двоструком линијом правоугаоника.

Веза између ентитета Knjiga и Autor такође је означена дуплом линијом јер ће због своје природе (кардиналности више према више) условити стварање новог слабог ентитета - релације у релационом моделу (односно табеле у бази података).



Сл. 19: E/R дијаграм модела - (А) Унос нових књига

У оквиру другог дела задатка (слика 20) уочљиво је да је Narudzbenica слаб ентитет, као и то да је однос између ентитета Narudzbenica и Knjiga такав да ће условити стварање новог слабог ентитета - релације у релационом моделу.



Сл. 20: E/R дијаграм модела - (Б) Наручивање књиге

Дуплом линијом ентитета Narudzbenica, као и дуплом линијом везе

између ентитета Narudzbenica и Knjiga и графички је приказано постојање слабих ентитета.

05.03

Релациони модел

Модел објекти-везе омогућава једноставно креирање релационог модела. Основна правила за превођење модела објекти-везе у релациони су:

- Дефинисати примарни кључ за све ентитете - у свим релацијама.
- Сваки ентитет (модела објекти-везе) постаје релација (релационог модела), а сваки атрибут постаје поље (атрибут) релације.
- Слаби ентитети такође постају релације али ће за сваки ентитет који је у надређеном (родитељском) односу према њима, дати слаби ентитети добити по један страни кључ. Сваки страни кључ који бива уграђен у дати слаби ентитет, једновремено је кључ кандидат надређеног ентитета (уобичајено примарни кључ тог ентитета).
- N:M везе више према више ће бити реализоване тако што ће се увести нове референцијалне (слабе) релације, односно ентитети, које ће служити само за остваривање веза више према више између релација које оне везе повезују. Атрибути, односно поља таквих релација биће страни кључеви - кључеви кандидати (уобичајено

примарни кључеви) релација које они повезују и заједно ће формирати примарне кључеве таквих релација. Свака N:M веза, односно веза више према више, овим се раздваја на две 1:N и 1:M, односно две везе један према више.

- N:1 везе више према један биће реализоване тако што ће се у прву релацију респективно уградити страни кључ - један кључ кандидат друге (углавном примарни кључ друге релације). Пример за реализовање ове везе је Narudzbenica - Sluzbenik веза која је претходно коментарисана.

05.03.1 Дефинисање примарних кључева релација

За ентитете формиране у моделу објекти-везе потребно је дефинисати примарне кључеве. За то се користе атрибути (или скупови атрибута) који једнозначно одређују дате ентитете.

За релацију Knjiga погодно је за примарни кључ поставити ISBN (број који јединствено одређује књигу).

За релацију Autor адекватан примарни кључ је email (адреса електронске поште аутора).

Као примарни кључ релације Izdavach може послужити email (адреса електронске поште издавача).

У релацији Oblast примарни кључ може бити naziv (назив области којој књига припада).

За релацију Format примарни кључ може бити naziv (назив

формата књиге).

За релацију Kupac најзгодније је да примарни кључ буде email (адреса електронске поште или број телефона купца, када би и то био атрибут).

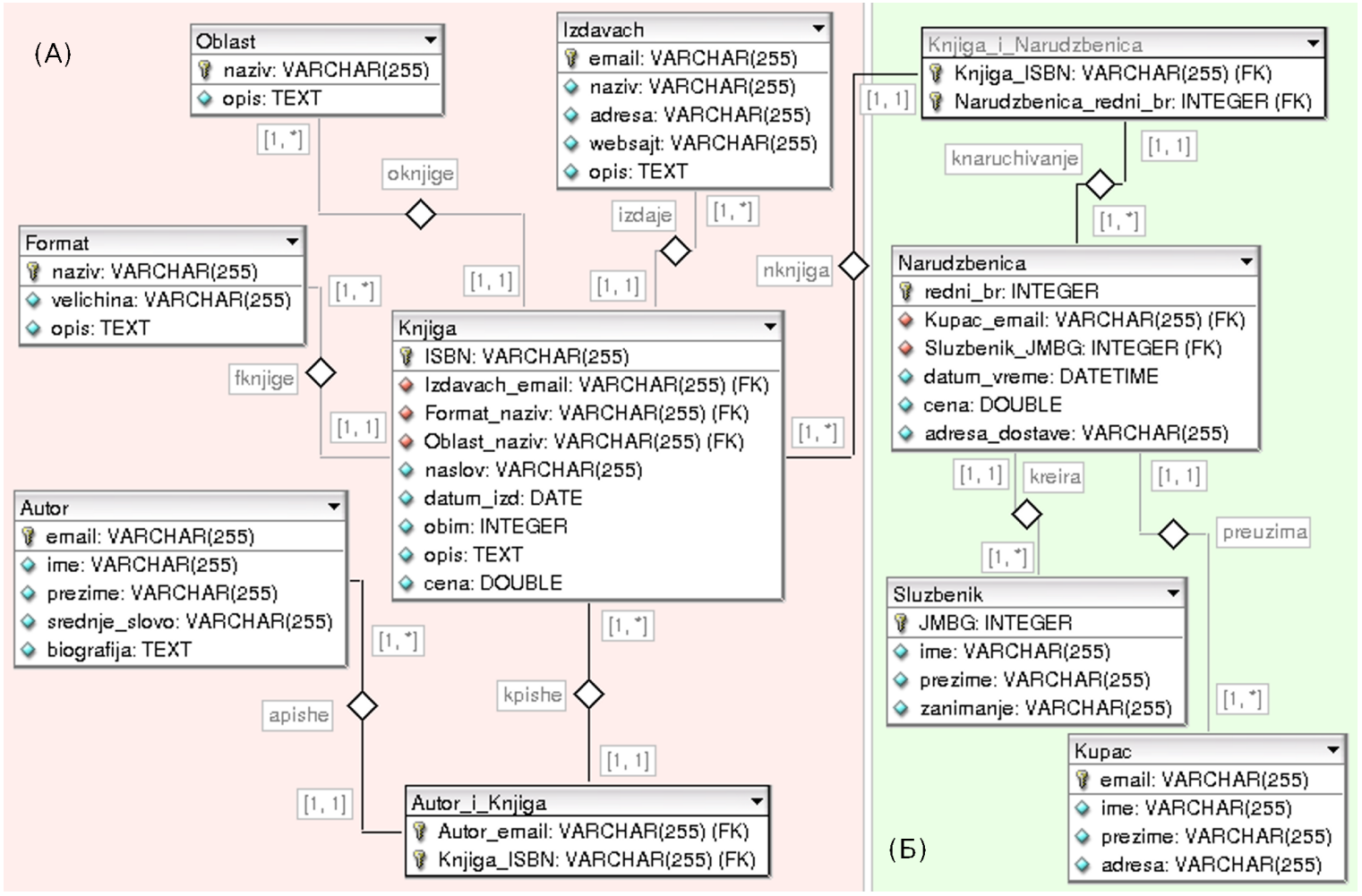
У релацији Sluzbenik примарни кључ може бити JMBG (јединствени матични број особе која је службеник).

JMBG би (пошто му намена и јесте да једнозначно одреди једну особу) наравно могао бити и примарни кључ за остале релације које се односе на особе (купци, аутори...) али уобичајено није тако из разлога што у тим случајевима јединствени матични бројеви нису познати.

Примарни кључ за релацију Narudzbenica може бити једна целобројна вредност - редни број наруџбенице (која се аутоматски повећава са додавањем новог реда). У виду страних кључева у релацију Narudzbenica морају ући и примарни кључ релације Kupac, као и примарни кључ релације Sluzbenik, али због природе везе са релацијом Knjiga не и примарни кључ ове релације (веза више према више).

05.03.2 Дијаграм IDEF1X

Пошто се формирају релације на основу ентитета и веза које су настале у формирању E/R модела, могуће је направити IDEF1X дијаграм (слика 21). На овом дијаграму уочљиве су све релације, везе и атрибути као и сви домени атрибута.



Сл. 21: IDEF1X дијаграм базе података

05.04

Имплементација у систему за управљање базама података

IDEF1X је целовит графички начин приказивања релационог модела базе података на основу кога је једноставно вршити имплементацију у систем за управљање релационим базама података. Програми који се користе за исцртавање IDEF1X дијаграма често имају могућност и да на основу дијаграма генеришу SQL упите за имплементацију приказаног модела у систем за управљање релационим базама података. Мада SQL имплементација може бити различита у зависности од конкретног система за управљање базама података, те разлике су (када постоје) веома мале. За модел који је приказан на претходној слици у тексту који следи дата је SQL имплементација у оквиру MySQL система за управљање базама података.

Употребом следећег упита креира се нова база података KNJIZARA.

```
CREATE DATABASE KNJIZARA;
```

Након креирања нове базе података, неопходно је доделити базу на употребу клијентима. Упитом који следи база се даје на располагање клијенту рачунара на коме се DBMS налази са именом 1klijent. Након овог упита дати клијент базе ће имати све привилегије над овом базом података (што значи да ће моћи да креира нове релације, односно табеле у оквиру базе, брише постојеће, уноси и требује податке и тако даље). Приликом рада са базом клијент 1klijent ће морати да се идентификује шифром shifrajednogklijenta.

```
GRANT ALL PRIVILEGES ON KNJIZARA.* TO
'1klijent'@'LOCALHOST' IDENTIFIED BY 'shifrajednogklijenta';
```

Да би у базу могли да се уносе и ћирилични карактери неопходно је кориговати карактере са којима база ради. Наредним упитом врши се промена карактера у UTF-8. То је стандард који дозвољава унос разноврсних слова.

```
ALTER DATABASE KNJIZARA CHARACTER SET UTF8;
```

Упитом који следи ресетују се привилегије у оквиру система за управљање садржајима.

```
FLUSH PRIVILEGES;
```

Након креирања нове базе података у оквиру система, и доделе клијената, потребно је да клијент базе активира (употреби) одговарајућу базу, упитом који следи.

```
USE KNJIZARA;
```

Затим се креирају планиране табеле у бази података на основу релација из релационог модела редом употребом приказаних упита.

```
CREATE TABLE Sluzbenik (
  JMBG INTEGER UNSIGNED NOT NULL,
  ime VARCHAR(255) NOT NULL,
  prezime VARCHAR(255) NOT NULL,
  zanimanje VARCHAR(255),
  PRIMARY KEY(JMBG) );

CREATE TABLE Kupac (
  email VARCHAR(255) NOT NULL,
  ime VARCHAR(255) NOT NULL,
  prezime VARCHAR(255),
  adresa VARCHAR(255),
  PRIMARY KEY(email) );
```

```
CREATE TABLE Oblast (
  naziv VARCHAR(255) NOT NULL,
  opis TEXT,
  PRIMARY KEY(naziv) );

CREATE TABLE Izdavach (
  email VARCHAR(255) NOT NULL,
  naziv VARCHAR(255) NOT NULL,
  adresa VARCHAR(255) NOT NULL,
  websajt VARCHAR(255),
  opis TEXT,
  PRIMARY KEY(email) );

CREATE TABLE Autor (
  email VARCHAR(255) NOT NULL,
  ime VARCHAR(255) NOT NULL,
  prezime VARCHAR(255) NOT NULL,
  srednje_slovo VARCHAR(255),
  biografija TEXT,
  PRIMARY KEY(email) );

CREATE TABLE Format (
  naziv VARCHAR(255) NOT NULL,
  velichina VARCHAR(255),
  opis TEXT,
  PRIMARY KEY(naziv) );

CREATE TABLE Narudzbenica (
  redni_br INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
  Kupac_email VARCHAR(255) NOT NULL,
  Sluzbenik_JMBG INTEGER UNSIGNED NOT NULL,
  datum_vreme DATETIME NOT NULL,
  cena DOUBLE NOT NULL,
  adresa_dostave VARCHAR(255) NOT NULL,
  PRIMARY KEY(redni_br),
  FOREIGN KEY(Sluzbenik_JMBG)
    REFERENCES Sluzbenik(JMBG),
  FOREIGN KEY(Kupac_email)
    REFERENCES Kupac(email) );

CREATE TABLE Knjiga (
  ISBN VARCHAR(255) NOT NULL,
  Izdavach_email VARCHAR(255) NOT NULL,
  Format_naziv VARCHAR(255) NOT NULL,
  Oblast_naziv VARCHAR(255) NOT NULL,
```

```
  naslov VARCHAR(255) NOT NULL,
  datum_izd DATE,
  obim INTEGER UNSIGNED,
  opis TEXT,
  cena DOUBLE NOT NULL,
  PRIMARY KEY(ISBN),
  FOREIGN KEY(Oblast_naziv)
    REFERENCES Oblast(naziv),
  FOREIGN KEY(Format_naziv)
    REFERENCES Format(naziv),
  FOREIGN KEY(Izdavach_email)
    REFERENCES Izdavach(email) );

CREATE TABLE Knjiga_i_Narudzbenica (
  Knjiga_ISBN VARCHAR(255) NOT NULL,
  Narudzbenica_redni_br INTEGER UNSIGNED NOT NULL,
  PRIMARY KEY(Knjiga_ISBN, Narudzbenica_redni_br),
  FOREIGN KEY(Knjiga_ISBN)
    REFERENCES Knjiga(ISBN),
  FOREIGN KEY(Narudzbenica_redni_br)
    REFERENCES Narudzbenica(redni_br) );

CREATE TABLE Autor_i_Knjiga (
  Autor_email VARCHAR(255) NOT NULL,
  Knjiga_ISBN VARCHAR(255) NOT NULL,
  PRIMARY KEY(Autor_email, Knjiga_ISBN),
  FOREIGN KEY(Autor_email)
    REFERENCES Autor(email),
  FOREIGN KEY(Knjiga_ISBN)
    REFERENCES Knjiga(ISBN) );
```

Након што се креирају табеле, база је потпуно празна (без икаквих података) и спремна је за употребу. Треба приметити да су прво направљене јаке (независне) табеле, а затим слабе (зависне), да би на адекватан начин били дефинисани страни кључеви. Употребом неколико додатних SQL упита биће демонстрирано како је могуће користити направљену базу података. Употребом NOT NULL у оквиру модела спецификује се да домен датог атрибута мора имати неку вредност.

Упитом који следи додаје се нови аутор, односно нови ред у

табелу Autor (нова торка у релацију Autor, односно нови објекат ентитета Аутор) у бази података.

```
INSERT INTO Autor (email, ime, prezime, srednje_slovo, biografija)
VALUES ('ppetrovic@asdf.com', 'Петар', 'Петровић', 'А','Петар
Петровић чувени је аутор свемирских опера. Рођен је у
непознатом месту 1910. године. И тако даље.');
```

Уколико је потребно могу се излистати сви аутори из базе. Упитом који следи систем за управљање базом података генерисаће табелу у којој ће писати имена и презимена свих аутора из базе.

```
SELECT ime, prezime FROM Autor;
```

Упитом који следи биће генерисана табела која садржи наслове свих књига чија је цена већа од 900.

```
SELECT naslov FROM Knjiga WHERE cena > 900;
```

06 Основе управљања пројектима

Велики број сасвим различитих послова (изградња моста, изградња информационог система, разна истраживања...) у времену у коме живимо реализује се у форми пројекта. Ова форма подразумевано се користи приликом израде информационих система за потребе савремених компанија. То је разлог што је у овој књизи на елементарном нивоу посвећена пажња класичном управљању и раду на пројектима. Текст који следи треба повезати са првом и другом главом ове књиге.

Реч пројекат преузета је из латинског (lat. projectum) и њоме се истиче планирање пре самог посла. Пројекат сачињава низ активности које су пажљиво планиране. Свака од активности на пројекту је ограничена (временом, тимом, финансијским и другим ресурсима). Након што се успешно реализују све планиране активности реализује се и циљ пројекта.

Различити пословни подухвати данас изводе се у форми пројеката зато што та форма олакшава рад особама које реализују посао (пројектном тиму) као и особама које управљају послом (пројектном менаџменту). Готово за било какве врсте послова у модерно доба подразумевају се тимови сачињени од особа различитих струковних профила, а често и из различитих држава (различитих култура).

Једна јединствена стандардизована форма која се свуда употребљава у послу (форма пројекта) омогућава да се, без обзира на поменуте разноликости, чланови пројектног тима лако усагласе у раду. Та стандардизована форма пројекта и менаџерима омогућава лакше управљање послом, јер се они приликом доношења одлука могу ослањати на стандардне нормативе пројекта (о којима ће бити речи касније).

06.01

Опште карактеристике

Рад на пројектима није једноставан јер се, готово подразумевано, у њиховој реализацији јављају проблеми. Уобичајени проблеми у раду на пројектима су:

- кашњење;
- повећање трошкова у односу на замишљене;
- неодговарајући резултати;
- незадовољни клијенти;
- неадекватна радна атмосфера пројектног тима.

Особе које учествују у реализацији пројекта чине пројектни тим. Пројектни тим уобичајено, али не и нужно, сачињавају особе различитих пословних профила. Њихов задатак је да реализују пројекат (да остваре пројектни циљ). Сваки члан пројектног тима треба да допринесе његовој реализацији својим знањима и вештинама.

Пројектом управља менаџер пројекта. Он руководи свим његовим аспектима и одговоран је за његову реализацију. Без обзира на тип пројекта, грубо га карактеришу:

- заинтересоване стране;
- циљ;
- финансије;
- време трајања;
- одређена доза несигурности (ризика) да ли ће се све реализовати на планирани начин.

Са свим горе наведеним ставкама менаџер пројекта мора да се суочи. Због тога је управљање пројектом веома захтеван посао.

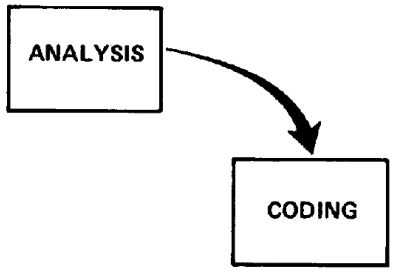
Природа управљања пројектима је таква да особа никада не може бити савршено оспособљена за ту делатност. Начин руковођења пројектима у модерно доба прилично је стандардизован тако да је менаџерима у знатној мери олакшан рад.

У зависности од самог пројекта (и мноштва за то везаних фактора) менаџер се опредељује за начин по коме ће се реализовати пројекат. Мада постоји више различитих начина који су у употреби данас, у овој књизи ће на елементарном нивоу бити појашњен класичан приступ (традиционални приступ) у управљању пројектима. Овај начин може се сматрати полазном основом свих савремених начина у управљању пројектима. Због тога је класичан начин примењив на било какав тип пројекта (не нужно IT пројекат).

06.02

Водопадни распоред фаза у реализацији пројекта

Да би неки (било какав проблем) било могуће решити, иницијално је неопходно добро га дефинисати. Да би рад на пројекту био добро дефинисан потребно је пројекат поделити у фазе. Овом поделом комплетан посао дели се у мање смислене целине. То олакшава рад у свакој од целина, а значајно олакшава и управљање пројектом - поготово због лакше прегледности.

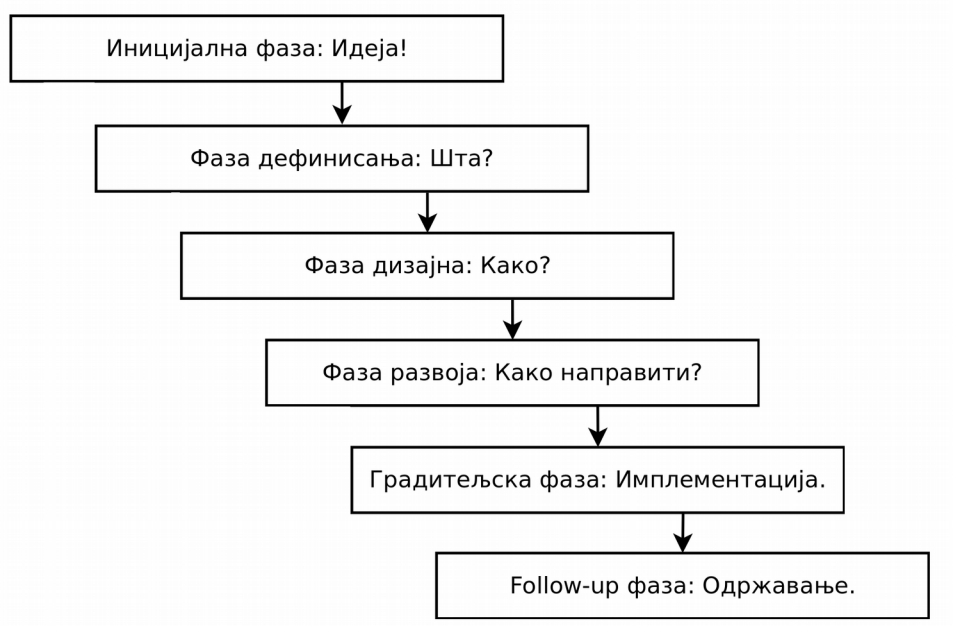


Сл. 22: Прелаз фаза водопадног модела
[R.Winston, 1970.]

Рад на пројекту, према класичном моделу, дели се у више фаза које се надовезују једна на другу. Након што се успешно заврши једна фаза прелази се на следећу - у идеализованом случају без повратка на претходну (слика 22). Због тога се подела на фазе у класичном приступу често назива и каскадни или водопадни модел, а прелази између фаза означавају се стрелицама чиме се акцентује само један смер (налик на каскадни ток воде).

Водопадни модел настао је услед потреба у развоју софтвера у прошлом веку, а и у модерно време уобичајено се повезује са пројектима развоја софтвера и IT пројектима генерално.

У зависности од интерпретације класичног приступа у раду на пројектима број фаза (каскада) може да варира.



Сл. 23: Фазе у реализацији пројекта
[W.Baars, 2006]

У складу са препорукама холандске организације DANS (eng. Data Archiving and Networked Services) у овој књизи пројекат према класичном моделу биће подељен у 6 фаза [W.Baars, 2006.]. Фазе на које се дели пројекат у класичној интерпретацији приказане су на претходној слици.

Мада је принципијелно намењен за IT пројекте, овакав модел може се применити и на друге типове пројеката те се и упутства која следе могу третирати као генерална.

06.02.1
Иницијална фаза

У иницијалној фази неформално почиње пројекат. Идеја се истражује и документује. Разматра се изводљивост пројекта и формира се одлука о томе ко би могао да изнесе замишљени пројекат.

Руководилац пројекта у овој фази (углавном уз асистенцију сарадника) пише пријаву пројекта у којој се специфицирају горе наведена разматрања. Пошто тип пројекта може бити сасвим различит и пријава може обухватати различите врсте докумената, скица, софтверског кода и тако даље. У зависности од конкретног примера често је случај да и сама пријава пројекта мора да испоштује одређену форму (примера ради пријаве пројеката које финансира Европска комисија). Ако је тако, тим који пише пријаву ограничен је нормативима за писање пријаве те мора да на оптималан начин уклопи своје планове са датим нормативима.

На основу пријаве пројекта, његови потенцијални финансијери разматрају у којој мери би им се исплатила таква инвестиција. У том контексту треба да буде јасно да то у зависности од конкретног случаја може бити прилично различито, јер финансијер пројекта може бити (већ постојећи) послодавац пројектног тима, а може бити и сасвим други ентитет (примера ради независна компанија или организација).

Ваљана пријава пројекта је искра живота за целокупан пројекат пошто тек након њеног прихватања пројекат официјелно (формално) почиње са радом. Уколико је пријава написана добро дешава се да и

пројекти са идејом која није толико добра бивају одобрени, а исто тако уз лошу пријаву могу бити одбијени пројекти са веома добром идејом. Због важности које има сама пријава пројекта често компаније које реализују одређени тип пројеката имају у свом саставу тим чија је основна (некада и једина) улога писање пријава пројеката.

Основна питања, на које је неопходно пружити одговоре у иницијалној фази, невезано за тип пројекта су:

- Зашто овај пројекат?
- Да ли је овај пројекат изводљив?
- Које институције би требало да изнесу овај пројекат?
- Који су очекивани исходи пројекта?
- Које су границе пројекта? Докле треба ићи, односно шта превазилази оквире пројекта?

Питања која су наведена су генералне смернице у овој фази и прилично су само-дескриптивна. Битно је напоменути да је важна особина коју руководилац пројекта мора поседовати способност да каже „не”. У било којој фази пројекта заинтересоване стране лако постају „похлепне” и уколико руководилац није у стању да ваљано постави границе онога што треба да буде урађено може се угрозити целокупан пројекат.

У иницијалној фази пројекта партнерске институције улазе у почетни вид сарадње. Такав вид сарадње уобичајено је мање формалан, јер у тој фази нема финансијских средстава за пројекат. Да би предупредили сва прекомерна очекивања, када су исходи пројекта у питању, потребно је да се сви сложе око тога да ли су

очекивани исходи пројекта:

- истраживање;
- истраживање и стварање прототипа нечега;
- истраживање, стварање прототипа али и производа који оперативно ради.

У складу са пословицом: „Непотпуно знање је опасније од незнања“, треба да буде јасно да уколико буде остављено довољно недоумица у иницијалној фази пројекта, велика је вероватноћа да дође до несугласица између заинтересованих страна касније.

06.02.2 Фаза дефинисања

Након прихватања плана пројекта из иницијалне фазе (пошто се одобре финансије за пројекат), прелази се у фазу дефинисања пројекта. Тада и официјелно (формално) пројекат почиње јер су одобрене финансије. Заинтересоване стране у овој фази формирају чвршће везе (праве документе који јасно дефинишу њихову сарадњу).

Најбитније код ове фазе је дефинисати резултате пројекта, што је прецизније могуће. У том контексту неопходно је идентификовати очекивања свих заинтересованих страна по питању резултата пројекта и ваљано их документовати. Уколико се овај корак у еволуцији пројекта не реализује на правилан начин, готово подразумевано пројекат мора да претрпи велики број проблема. Правилан начин: „И после вишечасовне полемике, заинтересоване

стране су усагласиле шта су очекивани резултати пројекта и све је документовано“.

Основне смернице, које принципијелно треба поштовати у овој фази пројекта су [G.Wijnen..., 2010.]:

- предуслови реализације пројекта;
- функционални захтеви;
- операциони захтеви;
- ограничења.

У зависности од типа пројекта предуслови његове реализације могу да варирају. Без обзира на тип пројекта, уобичајено се морају задовољити неки законски предуслови. На ову групу проблема се у току пројекта, према дефиницији, не може утицати. Уколико је резултат пројекта изградња информационог система, примера ради, предуслов у реализацији може бити да по закону систем не сме да садржи здравствене информације о запосленима.

Под функционалним захтевима се подразумева дефинисање захтева (квалитета) које производ, који настане у пројекту, мора да задовољи. Уколико је изградња информационог система курирске службе пројектни циљ, пример за функционални захтев може бити да систем аутоматски буде способан да процени време приспећа пакета када му се дефинише одредиште и његова тренутна локација.

Да би резултат пројекта могао да ради на предвиђени начин, потребно је задовољити неке услове. Ти услови спецификују се у оквиру фазе дефинисања као операциони захтеви. Уобичајено се ови захтеви везују за питања која се тичу резултата пројекта. Примери оваквих питања могу бити:

- Где ће се користити систем?
- Да ли сваки корисник мора имати кориснички налог?
- Да ли систем може радити без везе са Интернетом?

На свако од наведених или њима сличних питања у овом делу фазе дефинисања мора се дати и документовати јасан одговор. Примера ради: „Мобилни уређај који користи курир у доставном возилу мора поседовати GPS (eng. Global Positioning System) модул да би информациони систем радио на одговарајући начин“; „Уколико су временски услови непогодни (велика облачност) систем не може радити коректно“.

Често би се пројекат могао квалитетније реализовати употребом техника које из неког разлога не могу да се користе. Због тога се у оквиру смерница у фази дефинисања спецификују и ограничења. Разлози ових ограничења могу да варирају, али без обзира на то ограничења морају бити дефинисана. Примера ради: „Систем није у стању да самостално врши расподелу запослених по сменама“.

Све заинтересоване стране треба да сарађују у овој фази пројекта и да што прецизније дефинишу све аспекте да не би дошло до колизије између њих у некој од каснијих фаза.

06.02.3
Фаза дизајна

Дизајн пројекта у овом контексту подразумева „линију по којој ће пројекат да еволуира до успешног завршетка“.

На основу спецификација из претходне фазе (фазе дефинисања) у овој фази препоручљиво је разрадити неколико различитих варијанти дизајна од којих би сваки морао да доведе до успешне реализације пројекта.

У зависности од типа пројекта формални резултат ове фазе су дијаграми и документација (HTML документација доступна свим заинтересованим странама, UML дијаграми...) којом се дефинише рад на пројекту.

Након што се направи више варијанти дизајна менаџер пројекта се одлучује за једну варијанту по којој ће пројекат бити реализован.

06.02.4 Фаза развоја

Све што је потребно прибавити за реализацију пројекта у фази развоја мора се организовати. Са свим добављачима као и извођачима (подизвођачима) одређених делова посла морају се направити распореди обавеза и потписати одговарајући документи. Потребни алати или материјали морају се наручити (изнајмити). Због тога је само по себи јасно да тек када се ова фаза заврши имплементација може да почне.

Уколико је израда информационог система пројектни циљ, пример за ову фазу може бити договор са подизвођачем послова. Ако је у примеру програмерска компанија А та која реализује пројекат (гради информациони систем), онда подизвођач може бити компанија Б која се бави постављањем и одржавањем рачунарских сервера.

Тада би се у овој фази направио договор између компанија А и Б, који би укључио и детаљну спецификацију рачунарског сервера за који би била задужена компанија Б.

Битно је приметити да у зависности од типа пројекта, може да се догоди да нема формалне потребе за овом фазом (поготово код IT пројеката), јер се често дешава да све што је неопходно за реализацију пројекта запослени једне компаније могу самостално да реализују.

06.02.5 Градитељска фаза

У овој фази пројекта заправо се врши рад на конкретном резултату пројекта. У зависности од типа пројекта у овој фази програмери програмирају, дизајнери раде на дизајну различитих варијанти интерфејса, амбалаже и тако даље.

За посматраче са стране, од ове фазе резултати постају видљиви. Ова фаза подразумева заједно са имплементацијом и упоредно тестирање. Након завршеног посла (или етапе посла) у оквиру градитељске фазе врши се и евалуација резултата у складу са спецификацијама које су направљене у фази дефинисања. Важно је доследно спроводити ову евалуацију да би се крајњи резултат на прави начин постигао. Фаза је завршена када се утврди да резултат задовољава комплетну спецификацију и када се постави у рад.

Понекад је случај да је у фази дефинисања немогуће све било предвидети те да се дословно спецификација из једног или више

разлога не може реализовати. Пошто то оставља простор за конфликт код заинтересованих страна, битно је да менаџер пројекта увери све заинтересоване стране да је постигнуто решење оптимално и задовољава све прохтеве. Све евентуалне промене у односу на спецификацију која је претходно била дефинисана такође је потребно документовати. У супротном се поново оставља простор за конфликт.

06.02.6 Follow-up фаза

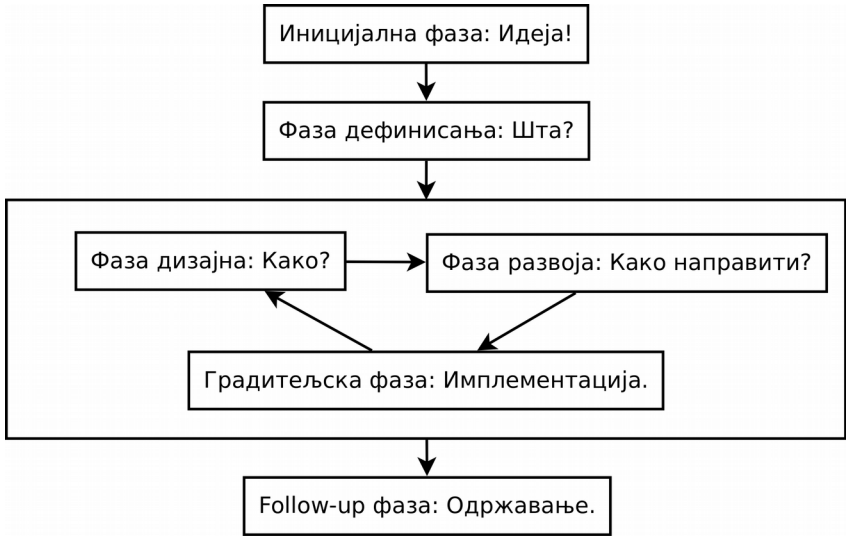
Follow-up фаза пројекта подразумева да се све уреди на начин да пројекат буде доведен до задовољавајућег краја. Мада на први поглед може деловати чудно, у овој фази основна ствар је правилно дефинисати крај пројекта. Тиме се подразумева писање извештаја као и писање упутства за употребу резултата пројекта (примера ради упутство за употребу софтвера).

Често се у овој фази формира техничка подршка, и у складу са додатним захтевима клијената врше дораде и одговарајуће одржавање.

06.03 Алтернативни распоред фаза у реализацији пројекта

Основна идеја пројекта може се представити изреком: „Think before you act“ (размисли пре активности). Поента је детаљно планирати пре извршења посла. Фазе у реализацији пројекта служе

да целокупан посао буде подељен у више смислених радних пакета. Тиме се расподељују посао и одговорности и у далеко већој мери је олакшано управљање пројектом.



Сл. 24: Алтернативни распоред фаза пројекта
[W.Baars, 2006.]

У зависности од конкретног пројекта, може се догодити да се након фазе имплементације не добије жељени резултат (који је планиран у фази дефинисања). У том случају класични водопадни распоред фаза мора се преусмерити другачије. Тада се у фази дефинисања подразумева глобално специфицирање резултата пројекта, а након те фазе се итеративно долази до одговарајућег решења (слика 24).

Осим приказане постоје и друге варијанте расподеле фаза пројекта, али превазилазе оквире ове књиге. Приступ који је овде приказан је класични, и полазна је основа свих осталих приступа у

раду са пројектима.

06.04 Контролни фактори у раду на пројекту

Менаџер се у раду на пројекту суочава са:

- пројектним тимом;
- циљем;
- ограниченим ресурсима;
- несигурности (ризиком) да ће све бити према плану.

У складу са пројектним циљем, менаџер пројекта треба да на адекватан начин примени своје вештине да организује дати пројектни тим на оптималан начин. Осим тога сваки пројекат осуђен је на борбу са ограниченим ресурсима. Чак и пројекти са најбољим условима рада морају да се боре са ограниченим новцем и временом. Тим изазовима пројектни менаџер такође мора адекватно да одговори.

Константа у раду на свим пројектима је ризик. Једноставно готово да не постоји пројекат који је апсолутно сигуран. Увек постоји одређена доза несигурности у његовој реализацији. Чак и уколико пројектни тим у својој историји већ има успешне пројекте истог типа, увек постоји доза ризика да ли тим може да реализује дати пројекат у предвиђеном времену са предвиђеним средствима. Увек се може десити нешто непредвиђено. Примера ради кључни члан пројектног тима изненадно је онемогућен за даљи рад услед саобраћајне

незгоде.

На основу до сада наведених ствари јасно је да менаџери пројекта морају да се суоче са мноштвом различитих елемената у руковођењу пројектом. Контролни фактори који увек фигуришу у плановима пројекта (eng. project plans), праћењу прогреса (eng. progress monitoring) и извештајима (eng. project reports) су [W.Baars, 2006]:

- Време;
- Новац;
- Квалитет;
- Организација;
- Информација.

Без обзира на тип пројекта поменути контролни фактори су од изузетне важности за пројекат. Због тога је уобичајени фокус менаџера пројекта је на ових пет контролних фактора.

06.04.1 Време

Управљање временом у раду на пројекту је од круцијалне важности. Овај контролни фактор се манифестује у раду на пројектима кроз временска ограничења за реализацију одређених активности. За сваку активност увек се предвиђа максимални временски интервал - рок (eng. deadline) за који се дата активност мора завршити успешно.

У плановима пројекта време утиче на:

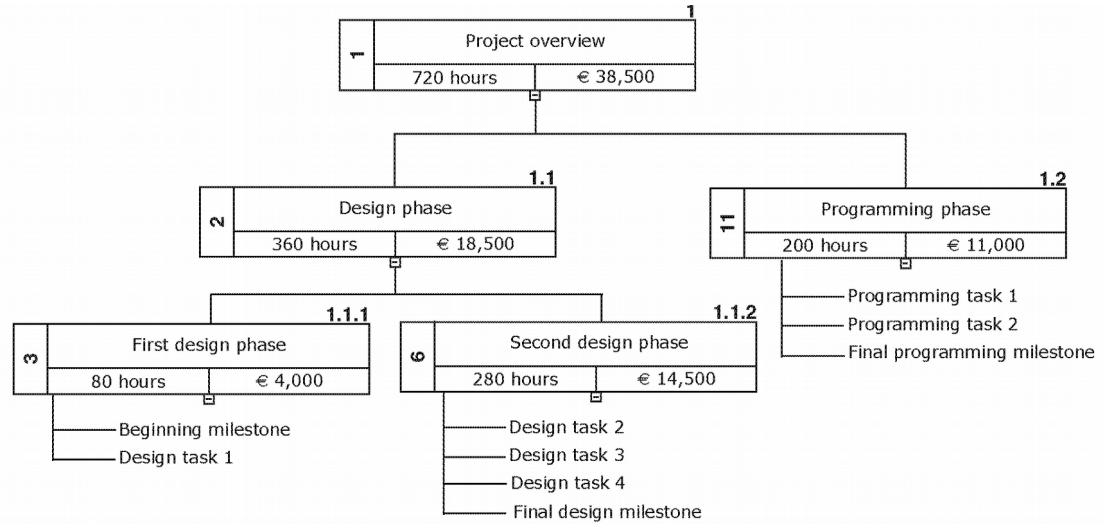
- одлучивање које активности треба реализовати у којој фази;
- одлучивање којим временским редоследом активности морају бити успешно завршене;
- распоред особља, материјала и алата према активностима;
- распоред активности у времену у оквиру сваке фазе;
- дефинисање рокова за сваку активност појединачно.

Када је реч о праћењу прогреса, са контролним фактором време повезано је:

- праћење прогреса свих активности;
- праћење рокова свих активности;
- прилагођавање распореда у складу са плановима на једној страни и тренутној ситуацији на пројекту на другој страни.

У извештајима пројекта време се везује за:

- извештај о тренутном распореду на пројекту;
- анализама и објашњењу зашто неке активности бивају завршене брже или спорије у односу на планирани распоред.



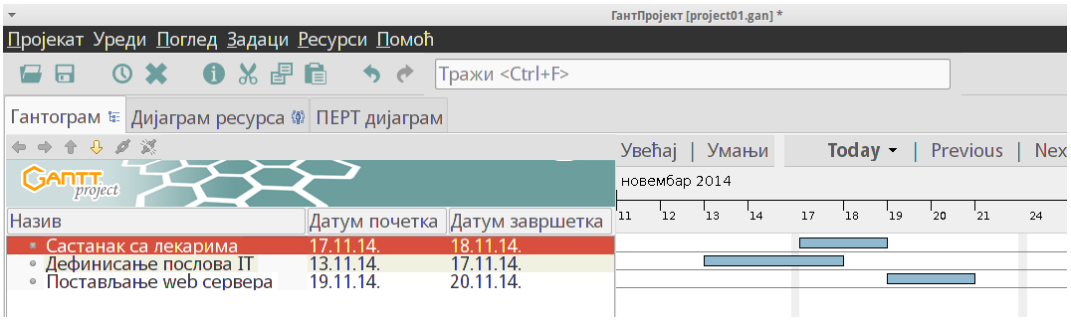
Сл. 25: Пример декомпозиције активности по времену - WBS IT пројекта [W.Baars, 2006.]

Временски распореди активности (послова) на пројекту се дефинишу употребом WBS (eng. work breakdown structure). WBS је временско декомпоновање активности на пројекту, које мора бити испуњено у целости да би пројекат био завршен успешно (слика 25).

Да би декомпоновање послова по времену могло да се уради на прави начин, особа која га ради мора имати довољно стручности да процени колико је времена потребно за сваку активност појединачно. У случају да особа која врши ово декомпоновање нема стручности из одређене области мора се консултовати са стручним саветником (из дате области) да би декомпоновање било извршено на адекватан начин.

Као што се за креирање текстуалних докумената данас користе за то специјализоване апликације - текст процесори (Microsoft Word, Libreoffice Writer...), тако се уобичајено за креирање WBS пројекта, као и планирање пројекта генерално, користе за то намењене

апликације (Microsoft Project, GanttProject...).



Сл. 26: Прозор програма GanttProject

Интерфејс свих апликација за планирање пројекта је сличан. Када корисник научи да употребљава једну од њих, лако може употребљавати и било коју другу (слично као што возач аутомобила може возити било коју марку аутомобила). На следећој слици приказан је интерфејс софтвера GanttProject[7].

Уобичајено се активности у времену у апликацијама за вођење пројеката приказују као тракасти графици[8] (слика 26).

06.04.2 Новац

Новац на пројекту се манифестује кроз пројектни буџет. Ваљано управљање пројектом подразумева да сва плаћања остану у

7 Ова апликација бесплатна је и може радити под различитим оперативним системима (MS Windows, Linux, MacOSX). Има мањи број опција од осталих апликација исте намене, али се често користи у великом броју IT компанија јер се њена једноставност тумачи као предност. Ова апликација је доступна и на српском језику. Може се преузети са ове адресе: www.ganttproject.biz

8 Препоручени видео материјал: www.youtube.com/watch?v=5rHCSa5ad34

границама пројектног буџета. Када се каже плаћања подразумева се финансирање особља, опреме, подизвођача...

Практично увек је управљање новцем у спрези са управљањем времена на пројекту (примера ради кроз број радних сати). Због тога се на WBS често приказују и плаћања за декомпоноване активности (слика 25).

У плановима пројекта новац утиче на:

- одређивање хонорара особља које учествује у реализацији пројекта;
- одређивање броја радних сати особља на пројекту;
- додељивања финансијских средстава тимовима у оквиру пројекта за извршење специфичних задатака;
- одређивање цене алата, материјала и услуга везаних за реализацију пројекта.

Приликом праћења прогреса пројекта, када је реч о новцу, врши се:

- праћење тока финансија;
- преговарање са добављачима у вези цена;
- провера да ли су планом предвиђена плаћања испоштована и у којој мери (да ли се цене поклапају са реалним);
- прилагођавања буџета, у складу са потребама;
- преговарање са заинтересованим странама на пројекту око усклађивања буџета.

Рад на пројекту подразумева и писање финансијских извештаја у

дефинисаним временским интервалима, као и писање финалног финансијског извештаја (по завршетку пројекта).

06.04.3 Квалитет

Резултати пројекта морају испунити договорене захтеве у погледу квалитета. У том контексту у раду на пројекту је важно дефинисати, усагласити (са заинтересованим странама) и документовати, у фази дефинисања пројекта, жељене стандарде квалитета. Да би резултати пројекта били одговарајућег квалитета и велики број међукорака (међурезултата) такође мора да задовољи одговарајуће стандарде у погледу квалитета.

Након фазе имплементације листа са стандардима у погледу квалитета мора се проверити да би било јасно да ли је све урађено на адекватан начин.

У току рада на пројекту, према потреби, могу се дефинисати и додатни стандарди који се тичу квалитета.

У плановима пројекта документује се очекивани квалитет резултата пројекта, као и квалитет међурезултата. Приликом планирања такође се према потреби одређује и квалитет појединачних активности на пројекту.

Приликом праћења прогреса на пројекту квалитет се проверава и документује (потврђује) за сваки међурезултат. Уколико из неког разлога за неки од резултата квалитет није на планираном нивоу, прави се адекватан извештај да би та чињеница била транспарентна.

У складу са евентуалним проблемима у задовољењу квалитета, рад на пројекту се у одређеној мери може благовремено кориговати.

У извештају пројекта квалитет се потврђује за коначне резултате пројекта, а према потреби и за међурезултате. Било какве примедбе на квалитет које се након фазе имплементације усвоје условљавају да се резултат пројекта коригује, у складу са њима, у Follow-up фази.

06.04.4
Организација

У раду на пројекту мора се управљати тимом. Колико је то важно сликовито казује популарна изрека француског државника и војсковође Наполеона Бонапарта (fr. Napoléon Bonaparte): „Умеће одабирања људи није ни издалека толико тешко колико умеће омогућавања одабранима да постигну своју пуну вредност“.

У основној форми организација подразумева одређивање тога шта ће ко морати да уради од активности на пројекту.

У широј форми организација укључује и „немушти језик“ - мотивационе способности, комуникационе способности, различите приступе руковођењу и тако даље. Циљ употребљавања ових особина је да тим успешније направи жељени резултат, без обзира на то унапређење ових особина превазилази ову књигу.

У плановима пројекта генерално организација подразумева:

- формирање квалитетног пројектног тима;

  додељивање руководиоца тима, или делова тима;

- додељивање задатака члановима тима;
- правилну размену/расподелу чланова тима различитих група на пројекту, а у складу са пројектним задацима и потребама.

У току рада на пројекту, приликом праћења прогреса, када је реч о организацији битно је вршити адекватно руковођење тимом у складу са тренутном ситуацијом на пројекту. Осим тога организација у том контексту подразумева правилну и сврсисходну интеракцију између различитих тимова (или заинтересованих страна) који учествују на пројекту.

Организација подразумева и формирање тима за писање финалних извештаја о реализацији пројекта, као и формирање тима за одржавање и надоградњу у Follow-up фази.

06.04.5 Информација

Када је реч о контролном фактору информација на пројекту, основно што треба да буде решено је ко одлучује о одређеним питањима и на који ће начин да се архивирају документи (чувају информације). Потребно је да буде познато да за одређено питање одлучује руководилац, члан пројектног тима који је експерт из одређене области или можда клијент. На који начин архивирати информације (примера ради web апликација), у којој форми (можда криптовати) и ко све треба да има приступ информацијама су детаљи који такође морају бити унапред познати.

У плановима пројекта, што се тиче контролног фактора информација, потребно је:

- да се дефинише ко треба да обезбеди коју врсту информација/директива и у којој форми;
- које информације ће морати да се архивирају, а које да се дистрибуирају;
- како технички треба да се изведе чување информација.

У праћењу прогреса подразумевају се:

- периодичне консултације;
- да се обезбеди да адекватне информације стижу до одговарајућих особа, као и да одговарајуће информације буду у тајности;
- да се, како посао одмиче, поштују договорена правила.

Формално гледано завршни извештај пројекта, као и сваки периодични извештај је скуп релевантних информација о пројекту.

07 Увод у системе пословне интелигенције

Савремено пословање (у било којој области) подразумева генерисање веома велике количине података. Колико је то изражено лепо осликава изјава Ерика Шмита (eng. Eric Schmidt) који је од 2001. године био председавајући у Board of directors (телу које је налик на управни одбор) чувене компаније Google: „У сваких 48 сати генеришемо толико података колико смо генерисали од зоре (почетка) човечанства до 2003. године“ (techcrunch.com, 2010.).

Да би се од података који су на располагању добиле информације, које су употребљиве, потребно је да се подаци анализирају. Број аналитичара ни приближно брзо не расте као количина расположивих података. За описивање дефицита аналитичара у односу на количину расположивих података користи се термин data-gap (eng.). Data-gap дословно значи „раскорак са подацима“. Овим термином указује се на немоћ у разумевању постојећих података.

У класичном приступу (класични информациони системи) клијент базе података (особа - директно, или индиректно употребом апликације) употребљава податке из базе коришћењем упита у циљу добијања жељених информација. То практично значи да се подаци (односно објекти) у оквиру базе претражују систематски у складу са претходно стриктно дефинисаним кључем (који је формално

дефинисан упитом). Након што се установи поклапање конкретних података из базе са дефинисаним кључем, сматра се да је пронађена жељена информација. То и јесте разлог што класични информациони системи нису у могућности да адекватно одговоре изазову анализе велике количине података које генерише савремено пословање. За такве количине података уобичајено се користи термин big data (eng.). Big data се може превести као „веома велика количина података“, али се и у домаћој литератури углавном употребљава енглески термин.

07.01
Системи за ефикасно проналажење информација

Да би било могуће одговорити изазову рада са великим количинама података у савремено доба користе се системи за ефикасно проналажење информација. Овакви системи могу бити категорисани у системе чија ефикасност лежи у:

- интелигентном понашању
  (интелигентни информациони системи);
- начину претраге (представљања) података.

Постоји велики број различитих дефиниција интелигентног понашања које се користе за дефинисање интелигентних система. Пошто тема везана за интелигентне системе превазилази оквире ове књиге, за интелигентно понашање биће употребљена дефиниција која следи.

Понашање неког система сматра се интелигентним, уколико је

систем способан да га адаптира у складу са окружењем зарад остваривања свог циља. Уколико се интелигентно понашање уочава код софтвера или машина, каже се да је реч о вештачкој интелигенцији.

Након што корисник добије жељене информације употребом класичног информационог система, он користи своју интелигенцију за додатно анализирање тих информација. Примера ради нека рецепционар жели да нађе одређеног госта хотела за кога зна да је из Холандије. Рецепционар тада може да изврши претрагу свих гостију хотела, али да подеси да се филтрирају сви који су имали држављанство Холандије. Након што из информационог система добије жељену листу гостију, на основу додатних података који су му на располагању и своје интелигенције врши додатно анализирање података и проналази жељену особу. За далеко веће количине података овакав процес није могућ, те је једна од варијанти да се превазиђе проблем поменуте анализе да се у систем угради вештачка интелигенција која треба да ради на сличан начин као човек.

За анализирање велике количине података се поред употребе система са вештачком интелигенцијом користе и системи чија ефикасност лежи у начину представљања података. Један од најпознатијих типова оваквих система је OLAP (eng. online analytical processing). OLAP системи креирају различите извештаје у зависности од перспективе на историјске податке које класификују у OLAP-кубне вишедимензионе матрице. Мада на „први поглед“ може изгледати као нешто компликовано, овакви системи намењени су менаџерима и веома су једноставни и захвални за употребу. Постоји више варијанти оваквих система од којих ће неке бити поменуте у даљем тексту.

07.02

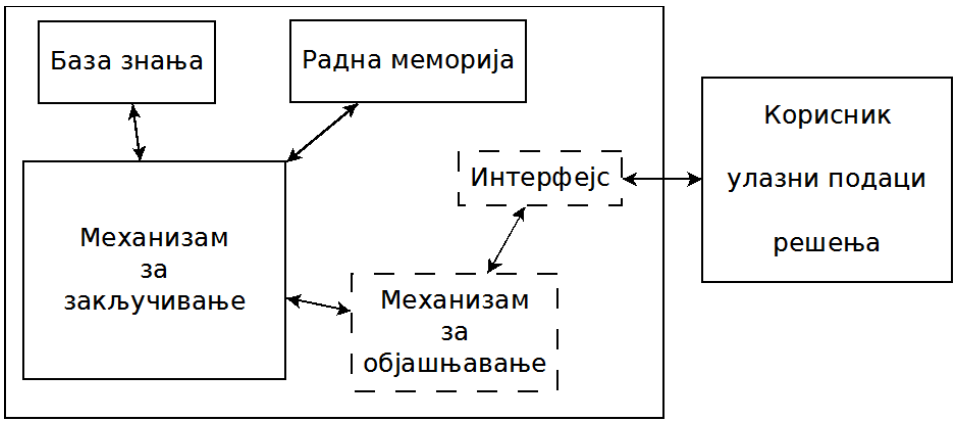
Интелигентни системи

Да би интелигенти систем могао да одговори задатку, потребно је иницијално у њега уградити знање. Ради поједностављења, може се рећи да је знање је разумевање неке тематске области. Знање особе (експерта из дате области) могуће је формализовати и уградити у софтвер. Уколико за пример експерта послужи аутомеханичар, онда се може рећи да експерт има знања из области ауто-механике. У модерно доба познате компаније за производњу аутомобила готово подразумевано имају и софтвер у кога уграђују знања оваквих експерата у комбинацији са конкретним случајевима са сервиса својих аутомобила. Након тога софтвер се, уз употребу уграђеног знања, може понашати као интелигентни систем. У том случају употребом уграђеног знања и података које добија, софтвер (интелигентни систем) самостално доноси нове закључке. У датом примеру то би практично значило да се након што се аутомобил који има неки проблем довезе у сервис, на основу дијагностике (података са датог возила) и знања (са којим систем већ располаже) аутоматски врши закључивање, односно систем аутоматски идентификује шта треба и на који начин поправити на датом аутомобилу.

Основне компоненте интелигентног система су (слика 27):

- база знања – у којој се налази знање које је формализовано (да би систем могао да га користи);
- радна меморија – која садржи чињенице и закључке (закључци су чињенице које су настале као последица резоновања);

- механизам за закључивање – комбинује чињенице из радне меморије и знање из базе знања, а у том процесу ствара закључке (решава проблем и ажурира стање).



Сл. 27: Општа схема структуре интелигентног система

Део интелигентног система углавном је и механизам за објашњавање. Када га систем поседује он је посредник између система и интерфејса преко кога корисник система и систем комуницирају. Ова компонента система користи се да објасни:

- Зашто систем поставља одређено питање?
- Како је систем дошао до решења?

07.03 Интелигентни информациони системи

Интелигентни информациони системи могу се дефинисати као нова генерација информационих система. Ова група информационих система настала је као резултат интеграције вештачке интелигенције

и класичних информационих система (технологија које укључују базе података). Интелигентни информациони системи, за разлику од класичних, поседују знање које им омогућава да испоље интелигентно понашање, да сарађују са корисницима и другим системима у решавању проблема [R.W.Zbigniew - T.Li-Shiang, 2010].

Сл. 28: Интелигентни информациони системи

Интелигентно понашање у оквиру информационог система може бити реализовано на различите начине. Примера ради интелигентни систем може вршити оптимизацију упита над базом података или може бити део корисничког интерфејса система. Област „интелигентни информациони системи“ нема јасно дефинисане границе. Неке примене оваквих система су: системи пословне интелигенције, технолошка подршка и тако даље.

Да би се направила јасна разлика између класичних и интелигентних информационих система може се замислити једноставан пример. Нека постоји семафор који обавештава возаче на аутопуту о попуњености паркинг места у гаражи која је у близини [В.Девеџић, 2000.]. Уколико систем обавештава:

1. о тренутном стању – тренутној попуњености места, реч је о класичном информационом систему, јер тада возач користи своју интелигенцију да процени колико му у дато доба дана треба времена да стигне до гараже, колико других возача у то доба дана долази у гаражу, а на основу

тога процењује да ли да скрене са аутопута или не;

2. да ће када возач дође до гараже имати одређени број слободних места (различит од тренутног броја слободних паркинг места), реч је о интелигентном информационом систему, јер тада систем користи своју уграђену интелигенцију и, примера ради, узима у обзир тренутну гужву у саобраћају, доба дана, временске прилике, да ли је радни дан или викенд и било какве сличне податке (поред тренутног броја слободних места у гаржи).

07.04 Системи пословне интелигенције

Историјски посматрано системи пословне интелигенције први пут су формално дефинисани средином прошлог века као системи који имају „способност запажања веза међу подацима на начин који помаже активностима које воде ка одређеном пословном циљу“ [H.P.Luhn, 1958.].

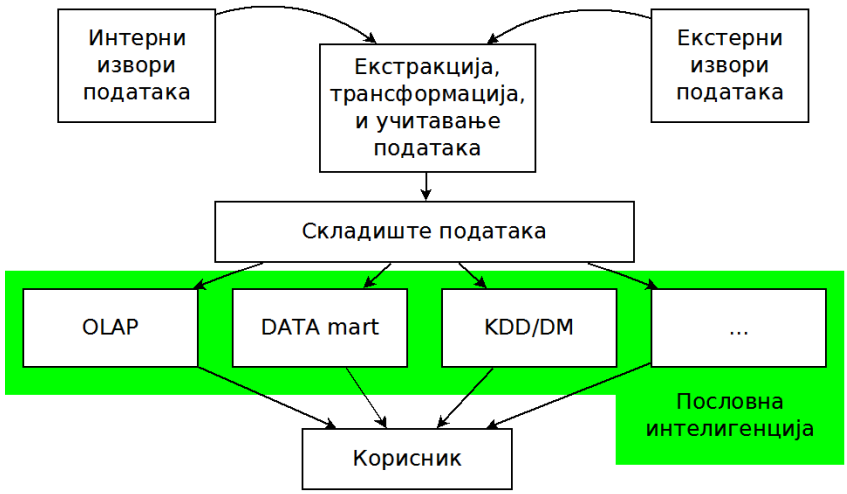
Суштински пословна интелигенција подразумева скуп метода и софтверских алата који омогућавају ефикасну употребу података и њихову анализу у циљу добијања информација које се могу употребити као помоћ при одлучивању у пословању. Циљ алата пословне интелигенције је да уз употребу екстерних и интерних података у оквиру неке компаније пронађу законитости и везе између њих, на основу којих могу да генеришу информације које помажу менаџерима да донесу праве пословне одлуке.

Савремени алати пословне интелигенције тако су направљени да менаџери могу без великих техничких потешкоћа сами да их користе у циљу самосталне анализе пословних података. Углавном резултати овакве анализе помажу менаџерима у доношењу нових стратешких одлука. Стереотипан пример за ово може бити анализа продаје различитих производа у великим ланцима ресторана неке компаније. У том случају менаџерима алати пословне интелигенције помажу да донесу праву одлуку о томе какве би нове производе требали да додају у мени својих ресторана или на којим би локацијама требало отворити нове ресторане. На тактичком нивоу такође овакви алати могу бити од помоћи, примера ради у оптимизацији неких пословних процеса.

07.04.1 Складиште података

Концепт складишта података (eng. data warehouse) осмислио је Вилијам (Бил) Инмон (eng. William H. Inmon). Према овом научнику складиште података је предметно оријентисана, трајна, интегрисана, сукцесивна у времену (историјска) колекција података чија је сврха подршка менаџменту у доношењу одлука.

Складиште података је кључни чинилац система пословне интелигенције.



Сл. 29: Систем са складиштем података

Складишта података су дизајнирана да помогну у анализи велике количине података. Примера ради могу асистирати менаџерима у анализи продаје различитих производа њихове компаније - у том случају било би потребно формирати складиште података које садржи релевантне податке везане за продају. Обзиром на то да се складиште података формира тако да садржи податке везане за одређену намену има каже се да има особину предметне оријентисаности (eng. subject oriented).

За разлику од класичних база података код којих се врши и освежавање (промена) података (eng. update), подаци који једном буду унети у складиште података више се не мењају и такви подаци служе само за читање (eng. read only). Ова особина складишта података назива се трајност (eng. nonvolatile). Обзиром на то да је сврха складишта података да омогуће анализу тога шта се догодило ова његова особина је сасвим логична.

Особина интеграције (eng. integrated) је тесно повезана са предметном оријентацијом. Обзиром на то да складиште података готово увек користи различите изворе података, сви подаци морају бити похрањени конзистентно (морају имати исти формат). Поступак интеграције података често подразумева решавање конфликта везаних за називе, конверзију јединица мера и тако даље.

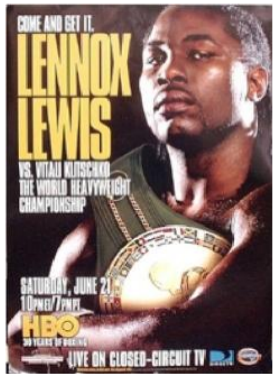
У раду са складиштем података фокус се ставља на променама у времену, због тога овакав систем и има особину сукцесивног уношења података са временом (eng. time variant). Ова особина је један од водећих фактора због којих складишта података садрже веома велике количине података.

Складиште података се уобичајено технички реализује као дистрибуирана релациона база података код које се независно одвајају рачунарски ресурси за анализирање од оних за трансакције. Код оваквих система готово подразумевано се омогућава прикупљање података са више различитих извора [P.Lane - V.Schupmann, 2002.].

Складиште података је основа система пословне интелигенције пошто је готово увек складиште података једини извор података који се у оваквим системима користе.

Пошто се за улазне податке складишта података користе различити извори, подразумевано се врши екстракција информација и форматирање података пре уноса у складиште. У процесу екстракције информација - IE (eng. information extraction) се врши формирање структурираних (информација, односно података) на основу неструктурираних података. Овај процес подразумева додељивање дескриптора неструктурираним подацима, на основу којих се касније врши ефикаснија претрага и анализа.

Британски боксер Ленокс Луис одбранио је назив првака света у супертешкој категорији у верзијама WBC и IBO. Луис је 21. јуна 2003. у Лос Анђелесу савладао Украјинца Виталија Кличка прекидом после шесте рунде. Меч је прекинут због обилног крварења из леве аркаде украјинског боксера.



IE (дескриптори):
Ленокс Луис, Бокс, првак света, WBC, IBO, Витали Кличко

Сл. 30: Екстракција информација - IE

Да би дескриптори били више употребљиви у анализи често се за сваки дескриптор у процесу екстракције информација додељује и број (тежински фактор) који ближе одређује дати дескриптор. Уколико се у некој вести на пример 15 пута понови име Владимир Путин (рус. Влади́мир Влади́мирович Пу́тин), а само 3 пута Барак Обама (енг. Barack Hussein Obama II), дескриптор са првим именом (именом председника Русије) имаће већи тежински фактор те ће за дати скуп дескриптора документ који они описују бити прецизније одређен. Неке примене процеса екстракције информација су подаци са берзе, огласи...

Када се подаци сниме у складиште података они су истог формата без обзира на то из ког извора су стигли и спремни су за употребу - анализу неким од алата пословне интелигенције.

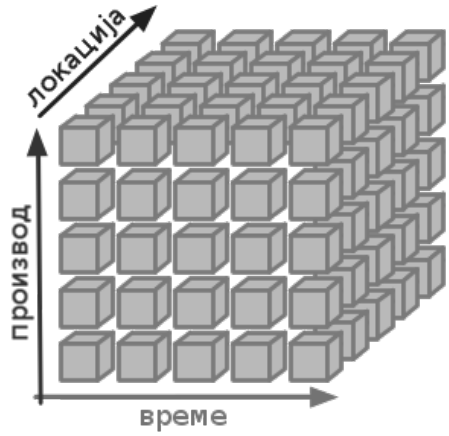
07.04.2
OLAP

OLAP системи креирају различите извештаје у зависности од перспективе на историјске податке које класификују у OLAP кубне вишедимензионе матрице. Овакви системи имају концепт који им омогућава да ефикасно одговоре на мултидимензионе упите [E.F.Codd.., 1993.].

Мада претходни пасус може деловати компликовано због терминологије која је употребљена суштински се може илустровати једноставним примером. Уколико менаџер неке велике компаније разматра продају неких производа свакако ће користити податке о продајама са којима располаже његова компанија (који су врло вероватно похрањени у складишту података). У тим подацима могу бити забележене све продаје свих производа те компаније у свим радњама у којима се продаја врши. У зависности од тренутне инспирације менаџеру, примера ради, може бити занимљиво да погледа колико је производа типа А продато у периоду од 2011. до 2015. године у некој одређеној радњи или у некој географској регији. За такве врсте упита OLAP је право решење.

Као што је већ поменуто у претходном тексту, ефикасност OLAP система лежи у начину на који овакви системи раде са подацима. Подаци се снимају у OLAP кубне вишедимензионе матрице. Основна јединица овакве матрице је мера (eng. measure). Упити се врше над мерама. Суштинска идеја је да подаци буду упаковани у модел који одговара кориснику (eng. business user), а не у модел који има везе са структуром података у бази података.

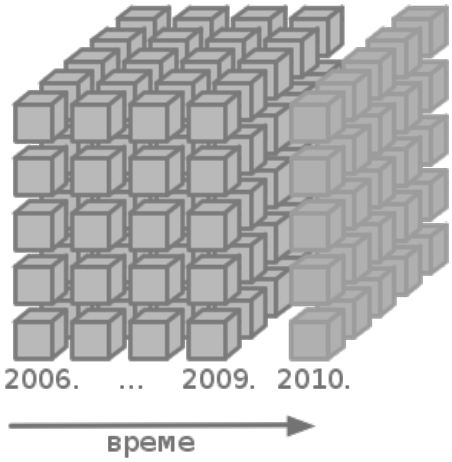
Димензија може имати и више или мање од 3 (димензије матрице нису нужно повезане са димензијама коцке). Уобичајено су за димензије фиксиране локације, типови производа, време (у зависности од тога како корисник жели). Димензије служе као филтери (димензија треба да филтрира вредност мера по пољима матрице). У примеру приказаном на наредној слици као димензије су фиксиране време, производ и локација (град у коме су производи продати).



Сл. 31: Димензије OLAP кубне вишедимензионе матрице (okfnlabs.org)

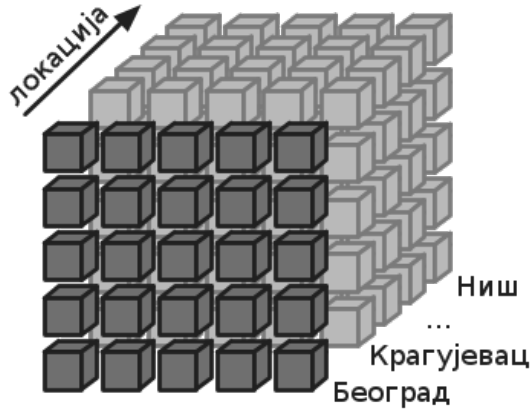
У пољима матрице записане су вредности мере (која је иницијално дефинисана). Мере могу бити: износ новца, попуст, трајање телефонског разговора, количина неког производа... У овом примеру вредност мере која записана је у пољима матрице може бити број продатих производа.

Корисник OLAP система је уобичајено заинтересован за агрегацију. Агрегација даје одговор на питања као што су: колико је укупно новца зарађено, колико је укупно минута разговора било, колико производа је продато...



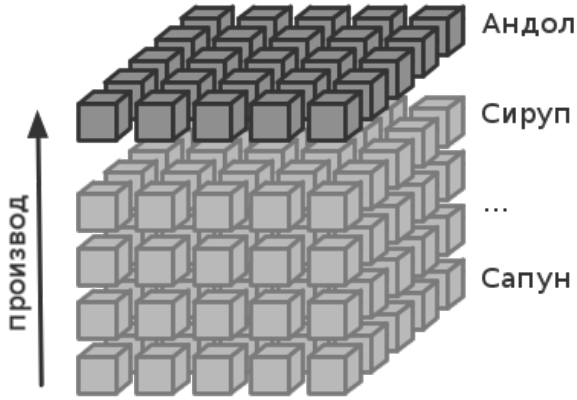
Сл. 32: OLAP - филтрирање по времену
(okfnlabs.org)

Осим тога што уобичајено захтева агрегацију, корисник система користи и димензије као филтере (слика 32). Тако се питање (агрегација) на које корисник добија одговор своди филтером, примера ради на: колико је укупно новца зарађено у 2010. години, колико је укупно минута разговора било у 2010. години, колико производа је продато у 2010. години...



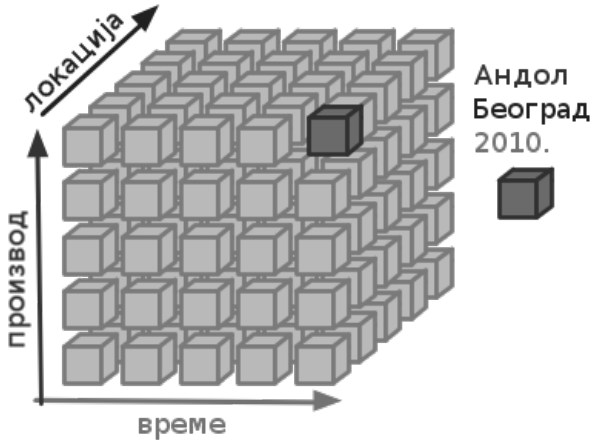
Сл. 33: OLAP - филтрирање по локацији
(okfnlabs.org)

Корисник може да „ротира” и пресеца OLAP кубну матрицу у складу са потребама, те да на тај начин добија различите резултате.



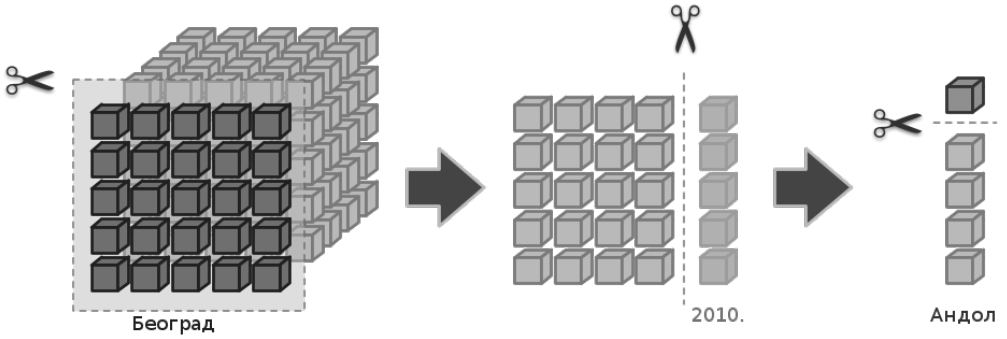
Сл. 34: OLAP - филтрирање по типу пословања
(okfnlabs.org)

У примеру приказаном на сликама (слике 32, 33, и 34), корисник је употребио 3 димензије OLAP кубне матрице за филтрирање. Резултат је приказан на слици која следи. У овом случају резултат је број продатих производа типа Андол у Београду 2010. године.



Сл. 35: OLAP - филтрирање по типу пословања,
времену и локацији
(okfnlabs.org)

Корисник система треба да употребљава матрицу тако да добије што прецизније одговоре у складу са тренутним потребама. Употреба овакве матрице слична је као играње Рубиковом коцкицом.



Сл. 36: Употреба OLAP матрице
(okfnlabs.org)

07.04.3
Data mart

Уколико би складиште података замислили као велики магацин различитих производа („од игле до локомотиве“) неке велике компаније, онда би data mart систем могли да замислимо као продавницу одређене групе тих производа (примера ради продавницу алата за шивење) исте те компаније. Продавници у том случају приступају купци (који немају приступ магацину) који су заинтересовани за одређену групу производа. Data mart је систем који се ослања на складиште података (мада постоје и независни data mart системи који имају складишта података која не користе други системи), а његова сврха је да испоручује податке који су везани за одређену тему (примера ради продаја) корисницима. Корисници data mart су уобичајено департмани (пословне јединице) неке велике фирме (продаја, маркетинг, рачуноводство), а концепт је такав да сваки департман има свој независтан data mart, а да сви користе исто складиште података.

Обзиром на то да се за чување информација користи складиште података, подаци се никада не мењају и никада се не бришу већ се само додају нови.

Data mart креира се у складу са потребама његових корисника, те корисници оваквог система немају тешкоће (са компликованим упитима) да пронађу податке који су им потребни чак и у великим складиштима података.

07.04.4
Откривање знања у базама података

Откривање знања у базама података – KDD (eng. knowledge discovery in databases) односи се на откривање законитости, правилности (eng. pattern) и индиректних садржаја у подацима (уобичајено великим количинама података). KDD се често поистовећује са техником истраживања података – DM (eng. data-mining). У наредном тексту биће појашњен однос ових појмова. Овакви системи настали су услед потребе да се ефикасније употребљавају-анализирају подаци похрањени у велике базе података [В.Девеџић, 2000.]. Често се називају KDD/DM системи због технологија које их чине.

Зашто употребљавати KDD/DM системе? - Традиционално проналажење знања у подацима подразумева мануелну интерпретацију података. Са великим количинама података овакав начин рада није ефикасан те су KDD/DM системи неопходност.

Примера ради у маркетингу, овакви системи користе се за анализирање база података муштерија/клијената. Овом анализом одређују се групе клијената и предвиђа њихово понашање (захтеви) у будућности. Након што клијенти, примера ради, направе куповину у некој великој Интернет радњи, web базирана апликација (Интернет сајт) одмах почиње да им нуди сличне производе. Систем зна које „сличне“ производе да понуди. То знање резултат је KDD/DM анализе из базе података (складишта података) свих дотадашњих куповина свих клијената. Систем „размишља“ на сличан начин као и човек, који би да је у улози продавца могао да препоручује производе својим купцима. Мада се може догодити да KDD/DM систем не предложи

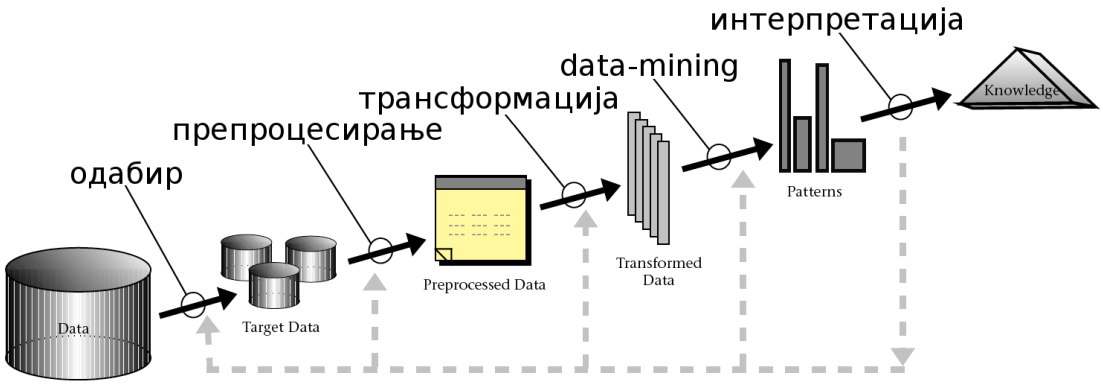
толико тачно (оптимално) додатне производе као што би то урадио човек - те то може бити мана оваквог система, велика његова предност у односу на људе је што је у могућности да у кратком временском интервалу изврши анализу велике количине података (у овом случају великог броја различитих производа и купаца).

Знање до кога анализом података долази KDD/DM систем формализује се у мустру, односно шаблон (eng. pattern). Термин мустра или шаблон у овом контексту ретко се користи - уобичајено је у употреби термин pattern. На горе поменутом примеру лако је направити дистинкцију између података и знања које је формализовано у pattern. Подаци који су релевантни за анализу у овом примеру су: називи производа, спецификације производа, цена производа, име купца, годиште купца, пол купца, листа производа које је дати купац купио... Анализом података KDD/DM систем долази до законитости између стереотипа купца и производа који би могли да му буду интересантни. Са временом складиште података се увећава, као и база знања коју KDD/DM систем формира те су листе препоручених производа све тачније. Знање би у овом примеру могло бити формализовано у pattern као што је: уколико је купац особа мушког пола која је купила производе X (патике за фудбал) и Y (фудбалску лопту), обавезно понудити производе Z1 и Z2 (фудбалски дрес и штитнике за потколенице).

Истраживање података

Мада се често поистовећује са откривањем знања у базама података истраживање података, односно рударење података - DM, је само један корак овог процеса (слика 37). Основни разлог што се ова два процеса поистовећују је тај што је истраживање података

практично основни корак процеса истраживања знања у базама података [U.Fayyad.., 1996.].



Сл. 37: Откривање знања у базама података
[U.Fayyad.., 1996.]

На претходној слици илустрована је поједностављена схема процеса откривања знања у базама података. Прво што се може уочити је да иницијалне податке (eng. data) у кораку одабира систем своди на податке који су од интереса (eng. target data). Анализом података од интереса систем долази до знања. Примера ради од свих података у складишту података велике компаније издвајају се подаци о козметичким производима и њиховој продаји. Циљ комплетног процеса мора бити узет у обзир да би се у кораку одабира од целог скупа података издвојио одговарајући подскуп - подаци од интереса.

У кораку препроцесирања систем додатно рафинише податке од интереса. Уклања се шум на подацима (уколико постоји), подаци се ређају у правилном временском распореду (уколико постоје временски кораци), попуњавају се сва поља са подацима

подразумеваним вредностима (уколико нека поља недостају)... Након овог корака подаци од интереса су униформно (правилно) структурирани, са свим попуњеним пољима (eng. preprocessed data) и спремни су за трансформацију.

У зависности од циљева процеса откривања знања у базама података у кораку трансформације препроцесирани подаци бивају додатно редуковани у складу са циљевима система. У наведеном примеру са козметичким производима у анализи може бити рецимо ирелевантан рок трајања производа и њихов састав. Да би се систем растеретио у анализи у кораку трансформације тада се „вишак“ података уклања. Стручно говорећи врши се смањење „димензија“ података. Ради илустрације можемо замислити пример у коме особа слаже намештај у некој високој просторији те занемарује висину свих предмета (пошто је плафон далеко виши од свих), а концентрише се само на њихову ширину и дужину. На сличан начин након корака трансформације систем ради са подацима (eng. transformed data) са мање димензија, јер неке занемарује „да би му било лакше“.

Над трансформисаним подацима примењују се технике истраживања података. Ово је по много питања сложен процес (компликована математика, различити алгоритми за различите врсте проблема, процесорски захтевна анализа...). Генерално у оквиру процеса истраживања података прво се мануелно дефинишу циљеви (класификовање, откривање односа, проналажење одступања), након тога бирају се (такође мануелно) оптимални алгоритми који ће бити употребљени, а након тога систем се оставља да самостално ради. Након корака истраживања података систем проналази мустре - pattern-е (законитости), а њиховом интерпретацијом се долази до знања.

Претходни опис система који врше откривање знања у базама података веома је груб и дат је ради илустрације, пошто даље описивање овог процеса превазилази оквире ове књиге. Знања до којих анализом овакви системи дођу користе се у великим фирмама - маркетинг, продаја, банке употребом оваквих система доносе одлуке да одобре кредите клијентима и тако даље.

Литература

[1] Д.Милашиновић: Основе пословне информатике; Факултет за хотелијерство и туризам у Врњачкој Бањи, Универзитет у Крагујевцу, 2014.

[2] В.Цвјетковић: Савремене информационе технологије; Природно-математички факултет, Универзитет у Новом Саду, 2008.

[3] В.Девеџић: Интелигентни информациони системи; Факултет организационих наука, Универзитет у Београду, 2000.

[4] А.Његуш: Информациони системи у туристичком пословању; Универзитет Сингидунум, Београд, ISBN: 978-87-7912-290-2, 2010.

[5] Организациони одбор научне конференције TISC 2016: Закључак Прве интернационалне научне конференције „Tourism in function of development of the Republic of Serbia" Spa tourism in Serbia and experiences of other countries, TISC 2016; Врњачка Бања, јун 2016.

[6] C.Zaniolo: A New Normal Form for the Design of Relational Database Schemata; ACM Transactions on Database Systems 7(3); 1982.

[7] E.F.Codd, S.B.Codd, C.T Salley: Providing OLAP (On-Line Analytical Processing) to user-analysts: An IT mandate;

Technical report, 1993.

[8] G.Wijnen, W.Renes, P.Storm: Projectmatig werken; Het Spectrum Utrecht, ISBN: 9789027469052, 2004.

[9] H.P.Luhn: A Business Intelligence System; IBM Journal of Research and development, Vol. 2(4), p314, 1958.

[10] J.Whitten, L.Bentley: Systems Analysis and Design Methods; McGraw-Hill/Irwin, ISBN: 978-0073052335, 2005.

[11] K.Al-Busaidi, H.Al-Shihi: Instructors' Acceptance of Learning Management Systems: A Theoretical Framework; Communications of the IBIMA, Vol. 2010, Article ID 862128, 10 pages, IBIMA Publishing, 2010.

[12] K.S.Proctor: Optimizing and Assessing Information Technology: Improving Business Project Execution; John Wiley & Sons, ISBN: 978-1-118-10263-3, 2011.

[13] N.Minić, A.Njeguš, J.Tulić Ceballos: The impact of Web 3.0 technologies on Tourism Information Systems; E-Business in tourism and hospitality industry, pp. 781 - 787, DOI: 10.15308/SInteZa-2014-781-787, 2014.

[14] P.Lane, V.Schupmann: Oracle9i Data Warehousing Guide; Oracle Corporation, 2002.

[15] R.Riordan: Designing Effective Database Systems; Addison-Wesley, 2005.

[16] R.W.Zbigniew, T.Li-Shiang: Advances in Intelligent Information Systems; Springer Publishing Company, ISBN: 978-3-642-05183-8, 2010.

[17] R.Winston: Managing the Development of Large Software Systems; Technical Papers of Western Electronic Show and Convention (WesCon), USA, 1970.

[18] U.Fayyad, G.P.-Shapiro, P.Smyth: From data mining to knowledge discovery in databases. AI Magazine, 17(3):37-54, 1996.

[19] U.Gretzel: Intelligent systems in tourism: A social science perspective; Annals of Tourism Research, 38(3), 757-779, 2011.

[20] W.Baars: Project Management Handbook; DANS, Royal Netherlands Academy of Arts and Sciences and the Netherlands Organisation for Scientific Research, 2006.

CIP - Каталогизација у публикацији -
Народна библиотека Србије, Београд

007:004(075.8)

МИЛАШИНОВИЋ, Данко, 1981-
Увод у информационе технологије /
Данко Милашиновић. - 1. изд. - Врњачка Бања :
Универзитет у Крагујевцу Факултет за хотелијерство
и туризам, 2016 (Краљево : Принт-промет). -
135 стр. : илустр. ; 25 cm

Тираж 300. - Напомене и библиографске референце
уз текст. - Библиографија: стр. 133-135.

ISBN 978-86-89949-14-8

а) Информациони системи

COBISS.SR-ID 226232332

Одлуком Комисије за издавачку делатност
Факултета за хотелијерство и туризам у Врњачкој Бањи,
Универзитета у Крагујевцу број 1444 (13.09.2016.) рукопис је
одобрен за штампу и употребу у настави као уџбеник.

ISBN 978-86-89949-14-8

Универзитет у Крагујевцу
Факултет за хотелијерство и туризам у Врњачкој Бањи

Co-funded by the
Erasmus+ Programme
of the European Union

MODERNIZATION AND
HARMONIZATION OF TOURISM
STUDY PROGRAMMES IN SERBIA
mhtsps

The publication has been funded within the framework of the European Union Tempus programme which is funded by the Directorate General for Development and Cooperation – EuropeAid and the Directorate General for Enlargement.

Project No. 544543-TEMPUS-1-2013-1-RS-TEMPUS-JPCR