

Факултет инжењерских наука Универзитета у Крагујевцу

Марија Јоковић

**Развој информационог система и базе података
за предшколску установу**

Мастер рад

Крагујевац, септембар 2019.



Универзитет у Крагујевцу
Факултет Инжењерских наука



Назив студијског програма: Индустијско инжењерство

Ниво студија: Мастер академске студије

Модул: Пословни информациони системи

Предмет: Пројектовање информационих система и база података

Број индекса: 628/2016

Марија Б. Јоковић

**Пројектовање информационог система и базе
података за предшколску установу**

мастер рад

Комисија за преглед и одбрану:

1. Др Милан Ерић - ментор

2. _____

3. _____

Датум одбране: _____

Оцена: _____

Изјава о самосталној изради рада

Изјављујем да сам овај мастер рад урадила самостално, служећи се стеченим знањем и искуством као и наведеном литературом.

628/2016 Марија Јоковић

(потпис)

Садржај

1. Увод	6
2. Пројектовање и израда информационих система	8
3. Базе података	10
3.1. Моделовање ентитета и веза	12
3.1.1. Ентитети и атрибути	12
3.1.2. Везе	13
3.2. Организација података	13
3.3. Креирање база података	14
3.4. Модели података АНСИ архитектура	16
3.5. Релациони модел база података	17
3.5.1. Представљање података.....	19
3.5.2. Приступ подацима.....	19
3.5.3. Захтеви који треба да испуни језик за рад са релационим базама података	21
3.6. Софтверски пакети за рад са базама података	27
4. Приступи, упити и претраживање база података	29
4.1. Напредне анализе података – интелигентно пословање	31
4.1.1. ОЛТП	34
4.1.2. Дефиниција ОЛАП-а	35
4.1.3. ОЛАП коцка	36
4.1.4. Операције над ОЛАП коцком	37
5. Анализа реалног система “Предшколска установа”	40
5.1. Идентификација кандидата за ентитете	41
5.2. Идентификација веза	43

5.3. Дефинисање атрибута	49
5.4. Дефинисање кључа.....	49
5.5.ЕР дијаграм модела базе података.....	48
5.6.Релациони модел - модел друге генерације.....	51
5.6.1. Структура релационог модела	51
5.6.2. Превођење ЕР модела у Релациони модел	53
6. Приказ логичког модела применом алата Erwin Data Modeler	60
7. Апликативно моделирање и фаза имплементације.....	62
8. Закључак	70
9. Литература	71

Апстракт

У првом делу рада су приказане основне фазе у пројектовању информационих система. Описана је улога и начин креирања база података као и добробити које оне пружају. Објашњени су начини приступа подацима и њихово представљање у базама података.

Значајан део рада се односи на интелигентно пословање и алате за приступ и обраду података. Предности које интелигентно пословање доноси су: анализа пословања, анализа клијената, боља контрола трошкова пословања, надзор пословања, брзо реаговање, предвиђање будућих трендова, једноставни графички прикази, једноставно прављење и коришћење извештаја...

У наставку рада се на основу анализе реалног систем Предшколака установа објашњавају основне фазе у моделирању и развоју информационог система. Описано је функционално и информационо моделирање и креиран је ЕР дијаграм. На основу дела ЕР дијаграма креирана је Му SQL база података “predskolska_ustanova” и веб апликација која се односи на сегмент пословања - дневну евиденцију. Замисао је да се представи значај примене информационих технологија на једноставном примеру као што је вођење дневника. Презентација добијених резултата говори да је добит вишеструка и код овако једноставних пословних процеса.

На крају рада је описана послења фаза развоја информационог система, а то је фаза имплементације која подразумева увођење, тестирање и договор о одржавању креираног ИС-а.

1. Увод

Живимо у времену у коме је информација главни ресурс. Карактеристике информације као ресурса су надмоћне у односу на ресурсе који су се користили у претходним добима (машинско доба, електрично доба, нуклеарно доба..) јер се трошењем информације не уништава њен садржај, већ се њена вредност повећава. Такође, неисцрпна је као ресурс и трпи мноштво конзумента истовремено. Једино је људска способност ограничавајући фактор њеног универзалног коришћења. Да би се могло манипулисати са информацијама и користити наведене предности информације као ресурса неопходно је да она буде сачувана у облику података у базама података.

Да би се информацијама могло манипулисати на што квалитетнији начин неопходно је креирање адекватних информационих система (ИС) и одговарајућих система за управљање подацима (СУБП). Информациони систем представља модел реалног система у коме делује. Развојем информационог система треба дефинисати што објективнију слику реалног света, његових бивших и садашњих стања, као подлогу за процену будућег понашања и наравно, подлогу за даљи развој и примену информационе технологије.

Приликом описа рада пословних система не можемо се користити природним језицима већ се врши поступак моделирања. Применом моделирања могућа је ефикасна и једнозначна комуникација између пројектанта и корисника. У процесу моделирања, елиминишу се детаљи, чиме се умањује видљива комплексност система који се проучава. Графичке презентације, правоугаоници и линије, користе се да би већина људи размишља о процесу моделирања као о сликовитој презентацији. Поред графичког приказа, потребно је дати и прецизне дефиниције предмета који се појављују у моделу, као и пропратни текст, који је критичан према моделу који има своју улогу, као средство комуникације.

Модел се сматра успешним ако се једнако понаса као оригинал и допушта да на основу његове структуре и његовог понашања пронађемо допунске црте оригинала. Наравно, модел никада не може бити идентичан оригиналу, увек је само апроксимација објекта из реалног света. Његова сврха је да занемари споредне и за посматрача тренутно неважне одлике оригинала, а нагласи битна својства.

Систем база података се може посматрати као хијерархијска апстракција. Сваки ниво у тој хијерархији апстракција је једна врста модела и представља скуп објеката (ентитета) и операција (радњи) над тим објектима као и веза између објеката. За сваки ниво апстракције можемо поставити питање могућности имплементације на нижем нивоу апстракције. Задатак пројектовања је да омогући физичку реализацију најнижег нивоа апстракције и омогући трансформацију модела на вишем нивоу апстракције у модел на нижем нивоу апстракције.

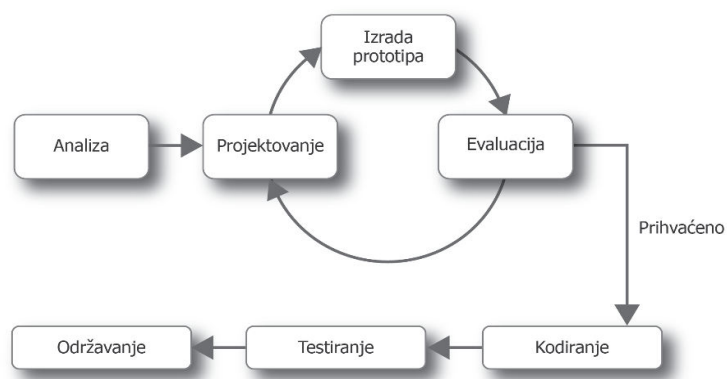
Један ниво хијерархијске апстракција је модел података, ЕР дијаграм, који ћемо креирати у овом раду за Предшколску установу. На нивоу непосредно испод модела података налази се репрезентација модела погодних структура података тј. релациони модел података. Карактеристика другог нивоа је потпуно одсуство било каквих утицаја особина рачунарског система на модел података. На трећем нивоу су те особине присутне па се због тога тај ниво зове ниво физичке репрезентације.

Креирање релационог модела за Предшколску установу, правила за превођење ЕР дијаграма у релациони модел података, креирање базе података у My SQL-у и израда веб апликације за евиденцију свакодневних активности деце у предшколској установи ће бити основна тема овог рада.

2. Пројектовање и израда информационих система

Данас управљање пословним системима подразумева примену информационог система који је у стању да ефикасно испоручује поуздане информације на свим местима и свим нивоима управљања, чиме се обезбеђује квалитетно и правовремено доношење и извршавање одлуке неопходних за успешно пословање.

Под пројектовањем информационог система (ИС), Слика 1., обично се подразумева аутоматизација и реорганизација постојећег ИС-а или пројектовање и израда потпуно новог информационог система.



Слика 1. Пројектовање информационог система.

Под развојем ИС-а се најчешће подразумевају четири фазе:

- Функционално моделирање,
- Информационо моделирање,
- Апликативно моделирање и
- Имплементација.

Функционално моделирање подразумева функционалну декомпозицију, дефинисање захтева корисника и анализу техничких предуслова.

Информационо моделирање подразумева дефинисање детаљних захтева, креирање ЕР дијаграма, креирање атрибута и дефинисање пословних правила.

Апликативно моделирање подразумева дефинисање физичког дизајна, генерисање шеме базе података и израду имплементације.

Имплементација се односи на увођење, тестирање и одржавање информационог система.

Захтеви који се постаљају пред пројектанте информационог система су да систем буде кориснички оријентисан, лак за употребу, флексибилан, и да подржава реализацију циљева организације.

Кад постојећи информациони систем постане уско грла у обради информација што изазива застоје и кашњења у изради извештаја (опсолентност система), као и одсуство савремених метода планирања и управљања, јавља се потреба за пројектовањем новог информационог система.

Израда новог информационог система подразумева формирање координационог тела које треба да донесе одлуку о оправданости пројектовања новог ИС-а, прати и контролише рад на реализацији пројекта. Координационо тело чини особље из највишег руководства организације које у сарадњи са пројектантима ИС-а, аналитичарима ИС-а, програмерима ИС-а и разним специјалистима за хардвер, софвер, базе података, мрежну комуникацију обезбеђује квалитетну израду ИС-а.

За планирање ИС-а неопходно је анализирати :

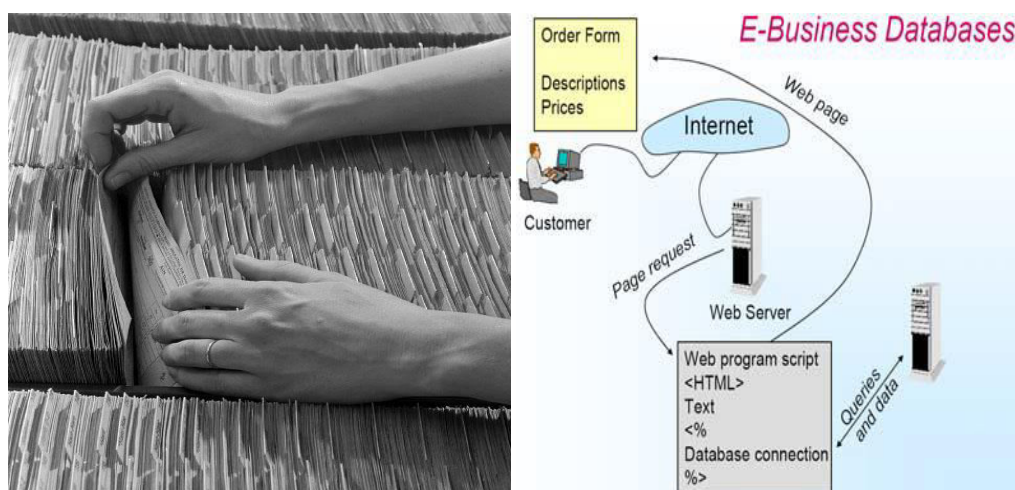
- недостатке у постојећем ИС-у, уколико постоје,
- решења која предлаже пројектни тим и приказ планираних ефеката тих решења,
- организационе измене које захтева нови систем,
- неопходну обуку особља које ће користити пројектовани систем.

У фази изградње ИС-а, на основу детаљне анализе врши се набавка одговарајућег хардвера који ће технички подршати будуће софтверско решење. Некада је потребно обезбедити и потпуну мрежну инфраструктуру (развлачење каблова, набавку свичева, рутера, сервера, прављење сервер сале...). Након потпуне израде или набавке већ постојетег апликационог софтвера врши се обука кадрова, тестирање апликације у реалном окружењу, пуштање у рад, израда пројектне документације и израда уговора о одржавању.

3. Базе података

Данас је основна улога рачунара чување и брза обрада различитих врста података. Подаци и информације су постали основни ресурс савременог друштва и највећа покретачка снага модерног пословања у целом свету. Када желимо да имамо квалитетне информације о свим сегментима нашег пословног или чак и приватног живота најбоље је да на одређени начин организујемо све податке чијом обрадом долазимо до информација које су нам потребне.

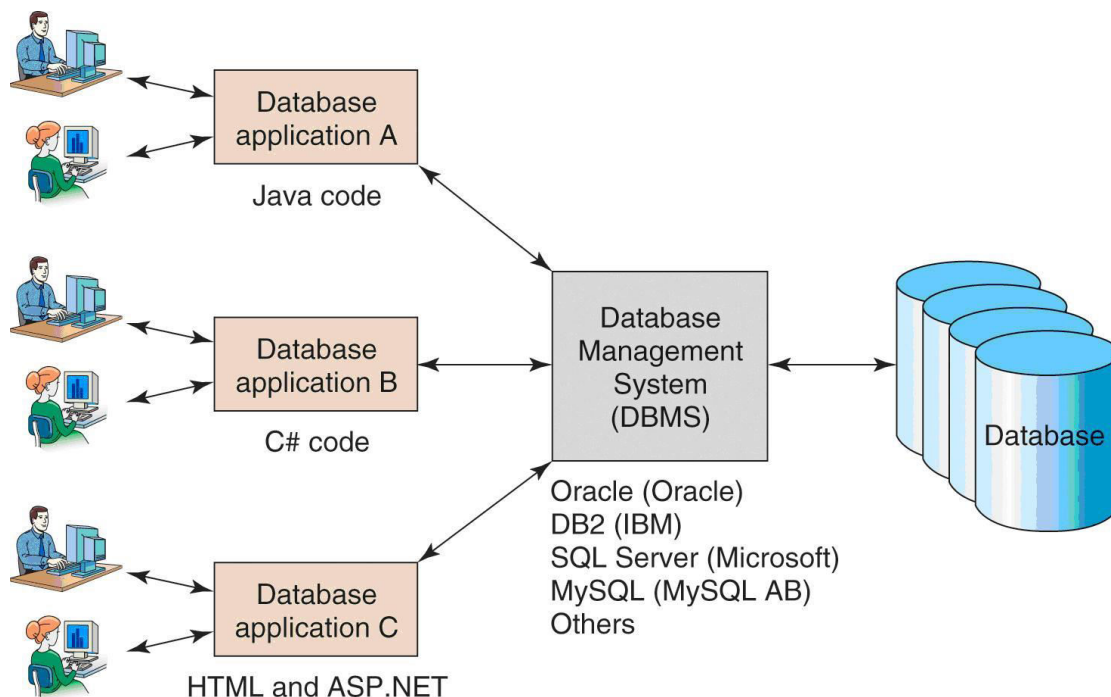
Најбољи начин је да подаци за сваки поједини елемент буду организовани тако да се могу сместити у табеле са истоврсним заглављем или у више табела које би обухватиле све сегменте нашег интересовања. Скуп више таквих табела које служе једном заједничком циљу, заједно са њиховим везним елементима назива се базом података.



Слика 2. Базе података некад и сад

Када користмо MS Word, или вршимо неке табеларне прорачуне у Excel-у више табела онда имамо додира са базом података и проблемом организације наших података смештајући их по одређеним директоријама. У случају када наше база постане толико компликована да нисмо више у стању да једноставно контролишемо ток и развој података потребно је прећи на виши ниво организације података и почети размишљати о систему за управљање базом података СУБП (енгл. *DataBase Management System, DBMS*). У најпознатије програме из ове области спадају: Oracle, DB2, Microsoft Access, Microsoft SQL Server, My SQL (који ће бити коришћен за израду информационог система за Предшколску установу).

Једноставно речено, база података је софтверска конструкција намењена за чување, анализу и претраживање групе сродних и повезаних података, као што су подаци о купцима, пацијентима, ученицима, телефонским бројевима и сл.



Слика 3.Пословни систем база података

База података састоји се од једне или више (дводимензионалних) табела које међусобно могу бити повезане. Свака табела чува истоврсне податке (нпр. податке о некој особи, предмету и сл.). Сваки ред у табели представља један слог у табели (најмања група података у бази која у потпуности описује неки од концепата које база моделира), а свака колона једно од поља унутар тог слога.

Слог може бити група података која описује нпр. неку особу, а поља унутар тог слога могу садржати име, презиме, адресу становања или датум рођења те особе. Слог се у литератури назива и ентитет, а поље се назива атрибут. Сваки слог табеле се може јединствено идентификовати путем једне или комбинацијом вредности неког од поља тог слога. То поље или комбинацију поља тада називамо кључем.

Једна табела може имати више поља или комбинација поља који могу бити као примарни кључ. Поред тога што примарни кључ има улогу јединственог идентификовања слога он игра улогу и у повезивању табела. Узмимо да наша табела уствари представља листу писаца. Поред те табела имамо и листу књига, те је потребно ове две табеле повезати како би смо знали који је писац написао коју књигу. Ако у слог

књиге убацимо поље које садржи вредност примарног кључа писца, ове две табеле су повезане. Ово ново поље (које искључиво служи за повезивање две табеле) у табели се зове страни кључ. Овакав начин повезивање података називамо релациони модел база података.

3.1 Моделовање ентитета и веза

Како обликовати шему за базу података, усклађену с правилима релационог модела? У стварним ситуацијама доста је тешко директно погодити релациону шему. Зато се служмо једном помоћном фазом која се зове моделовање ентитета и веза тј. ЕР моделовање (енгл. *Entity-Relationship Modelling*). Реч је о обликовању једне мање прецизне, концептуалне шеме, која представља апстракцију реалног света.

ЕР - шема се даље применом одређених правила која ће бити објашњена у наставку рада претвара у релациону. Моделовање ентитета и веза захтева да се све посматра преко три категорије:

- **ентитети** : објекти или догађаји који су нам од интереса;
- **везе** : односи међу ентитетима који су нам од интереса;
- **атрибути** : својства ентитета и веза која су нам од интереса.

3.1.1. Ентитети и атрибути

Ентитет је нешто по чему желимо памтити податке, нешто што је у стању постојати или не постојати, те се може идентификовати. Ентитет може бити објект или биће (на пример кућа, студент, дете, аутомобил), односно догађај или појава (на пример утакмица, празник, екскурзија).

Ентитет је описан атрибутима (на пример атрибути куће су: адреса, број спратова, боја фасаде, неки од атрибута за ентитет дете ће бити име, презиме, име родитеља, ЈМБГ..).

Уколико неки атрибут и сам захтева своје атрибуте, тада га радије треба сматрати новим ентитетом (на пример модел аутомобила). Исто правило важи и ако атрибут може истовремено имати више вредности (на пример квар који је поправљен при сервисирању аутомобила).

Име ентитета, заједно са припадним атрибутима, заправо одредује тип ентитета. Може постојати много примерака (појава, инстанци) ентитета задатог типа (на пример СТУДЕНТ је тип чији примерци су Петровић Петар, Марковић Марко, . . .).

Кандидат за кључ је атрибут, или скуп атрибута, чије вредности једнозначно одређују примерак ентитета задатог типа. Дакле, не могу постојати два различита примерка ентитета истог типа с истим вредностима кандидата за кључ. (На пример за тип ентитета АУТО, кандидат за кључ је атрибут РЕГ_БРОЈ). Уколико један тип ентитета има више кандидата за кључ, тада бирамо једног од њих и проглашавамо га примарним кључем (нпр. примарни кључ за ентитет СТУДЕНТ могу бити атрибути БРОЈ_ИНДЕКСА или ЈМБГ).

3.1.2. Везе

Везе се успостављају између два или више типова ентитета (на пример веза ИГРА_ЗА између типова ентитета ИГРАЧ и ТИМ). Заправо је реч о именованој бинарној или к-нарној релацији између примерака ентитета задатих типова

Веза може имати и своје атрибуте које не можемо приписати ни једном од типова ентитета (на пример веза УПИСАО може имати атрибут ДАТУМ УПИСА).

Ако сваки примерак ентитета неког типа мора суделовати у задатој вези, тада кажемо да тип ентитета има обавезно чланство у тој вези. Иначе тип ентитета има необавезно чланство. О везама и кардиналности ће бити речи у поглављу 5.

3.2. Организација података

Организација података је од веома великог значаја када желимо радити са базом података. Један од кључних аспеката, доброг креирања базе података јесте како ће подаци бити организовани у бази података. Добро креирана база података подразумева лаку доступност подацима и омогућава лако одржавање.

Треба одредити који ће подаци улазити у базу података, затим који ће се подаци сместити у одређене табеле међу којима ће бити успостављен однос, те какав је однос међу тим подацима. Потребно је смањити могућност, колико је то могуће, да се исти

податак записује више пута (редундација), јер вишеструким записивањем настају проблеми очувања стварне, јединствене вредности свих података при ажурирању. То утиче и на поузданост информација које се добију из тих података. Потребно је управљати смештањем података и чувањем тих података од намерних и ненамерних уништења тј. да не дође до губитка интегритета података. Неке податке треба заштитити од тога да их неовлашћени корисници не мењају што се зове тајност или приватност података.

3.3. Креирање базе података

У свакодневном животу да би почели нешто правити, креирати потребно је да унапред одредимо дизајн, тј. нацрт. Ако хоћемо да припремимо неко јело, потребан нам је рецепт, ако хоћемо градити кућу потребан нам је нацрт како ће изгледати и сл. При креирању базе података, потребно је организовати податке, одредити циљеве, елиминисати сувишне податке, омогућити брзо проналажење појединачни података, сачувати једноставно одржавање базе података..

Кључне активности при креирању базе података су:

- Дефинисање података неопходни за апликацију,
- Организовање података у табелама,
- Успостављање међусобних веза између табела,
- Успостављање захтева индексирања и вредновања података,
- Израда и сниммање свих потребних упита у вези са апликацијама.

Моделирање апликације се односи на поступак при којем дефинишемо задатке које апликација треба да обави. Било би добро да дефинисане задатке спецификујемо у одређени документ који нам може помоћи да будемо усредсређени на задатак нашег програма. Приликом организирања података у табеле можемо лако одредити да ли неки податак припада тој табели или не, нпр. ако неки клуб жели да прати информације о својим члановима и запосленима, управа клуба ће у једну табелу ставити и запослене и чланове. Обе групе захтевају информацију о имену, адреси, телефонском броју, док запослени захтевају и информације о броју социалног осигурања, висини плате и сл.

Али ако направимо више табела где се сваки податак појављује само једном и ми ћемо приликом измена нову информацију уписивати само једном.

Други начин да се подаци нормализују јесте да се формира тзв. табела дете. Табела дете је табела у којој сви унети подаци деле заједничку информацију која је смештена у некој другој табели. Ако узмемо за пример неку породицу где су код свих иста презимена, иста адреса, исти бр. телефона, а различита имена, њихове заједничке податке ставимо у једну табелу, а имена у другу у том случају табела која садржи имена чланова је табела дете. Табела претраживања је још један приступ смештају информација с циљем спречавања појаве сувишних информација и повећање тачности уношења података. Она се обично користи за смештај општеважећих улазних података.

Када неко у току апликације унесе такву ознаку, програм проверава у одговарајућој табели да ли та ознака стварно постоји. Када нормализујемо податке, међусобно повезане информације обично смештамо у неколико табела. Међутим, када је потребно да приступимо подацима желимо да видимо информације из свих табела на једном месту. Да бисмо то постигли морамо формирати скупове записа који обједињују међусобно повезане информације из неколико табела. Тај скуп записа се формира употребом SQL (енгл. *Structured Query Language*) наредбе у којој наводимо жељена поља, локације поља и однос између табела и те наредбе можемо сместити као упит у базу података.

Употреба упита има неколико предности:

- премештање апликације у сервер окружење је једноставније,
- уношење измена у SQL наредбу је једноставније,
- SQL наредбу можемо лакше користити на више места у програму или у више програма.

Ови упити раде брже од упита који се израђују задавањем наредбе програма кода.

Добро креирана база података омогућава:

- минимално време потребно за проналажење записа,
- смештање података на најефикаснији начин тако да база не постане сувише велика,
- најједноставније ажурирање података,

- захваљујући својој флексибилности укључивање нових функција које би се од програма могле захтевати.

3.4. Модели података АНСИ архитектура

Архитектура највечег броја система база података одговара предлогу АНСИ/СПАРЦ студијске групе Америчког националног института за стандарде, и позната је као АНСИ архитектура. Ова архитектура представљена је хијерархијом апстракција, при чему сваки ниво хијерархије укључује специфични начин представљања, репрезентацију, објеката, односа међу објектима и операција над објектима.

Хијерархијска архитектура омогућује природну декомпозицију и ефикасни развој система за управљање базама података. Најнижи ниво АНСИ архитектуре је унутрашњи ниво. Он је најближи физичкој репрезентацији базе података, која у рачунарском систему једина заиста постоји. Због тога се унутрашњи ниво често и зове “ниво физичке базе података”. Следећи ниво АНСИ архитектуре је концептуални (логички) и представља начин на који се подаци из физичке базе података представљају кориснику у општем случају. Највиши ниво АНСИ архитектуре је спољашњи ниво који представу о подацима из базе прилагођава потребама специфичних корисника или група корисника.

Репрезентација која се налази на “средњем”, концептуалном нивоу АНСИ хијерархије зове се модел података. Моделом података представља се логичка структура свих података у бази и скуп операција које корисник може извршити над тим подацима. То значи да се на концептуалном нивоу могу “видети” сви подаци из физичке базе података, само што је њихова презентација погоднија за корисника од физичке (на вишем је нивоу апстракције).

Поједини корисници или групе корисника могу имати своја сопствена специфична гледања на модел података (нпр. из разлога заштите или удобности), па се погледи (подмодел, спољашњи ниво хијерархије) налазе изнад модела у хијерархији апстракција. Исти подаци из физичке базе података (и са концептуалног нивоа), на овом нивоу могу се разним корисницима представити на разне начине, док се постојање неких података може од неких корисника и сакрити. Зато постоји тачно један

модел података у систему, који се односи на целокупност базе података, и већи број спољашњих погледа, од којих се сваки састоји од апстрактне слике дела базе података.

На пример, база података о студентима може да садржи податке из досијеа студената, са личним подацима и подацима о успеху (предметима, оценама, датумима полагања, наградама и казнама) сваког студента. На концептуалном нивоу подаци могу бити представљени табелама ДОСИЈЕ, НАСТАВНИ ПЛАН, НАГРАДЕ И КАЗНЕ и ПОЛАГАЊЕ. На спољашњем нивоу, корисницима статистичких обрада ова база података може се представити као да садржи само податке о успеху (нпр. у виртуелној табели УСПЕХ ПО ПРЕДМЕТИМА); остали, лични подаци који могу да идентификују студента, скривају се од статистичких апликација. Различитим подгрупама корисника могу одговорати још апстрактније представе о подацима, које одговарају вишим нивоима подмодела.

Примарни циљ модела података у контексту база података јесте да обезбједи формални систем за представљање података и манипулисање подацима у бази података.

3.5. Релациони модел база података

Релациони модел података, који се заснива на математичкој теорији скупова, предложен је још 1969. године од стране Е.Ф.Цодд-а, тада истраживача у ИБМ-овим лабораторијама. Овај модел је својим особинама превазишао тада актуелне хијерархијске и мрежне моделе података.

Релационе базе података се пројектују у виду табела, индексираних по различитим појмовима (пољима) и са више кључева. Већ при њиховом креирању могу се успоставити логичке везе између појединих табела преко заједничких поља тако да се на лак и једноставан начин долази до свих потребних података, било да се ради о подацима само из једне табеле или се подаци добијају спајањем више табела у једну нову, која је привременог карактера, а садржи само поља која су потребна кориснику. До податка се у табели долази директно позивањем одговарајућег индексног кључа при чему се показивач аутоматски позиционира на слог са траженим податком.

Оснивач релационог модела, у циљу заштите епитета “релациони”, који се све чешће из комерцијалних разлога приписивао и неким нерелационим системима за управљање базама података, дефинисао је 12 строгих правила које мора поштовати један РДБМС

(енгл. *Relation DBMS*). Он је такође и аргументовао постојање низа позитивних ефеката које у пракси постиже систем уколико задовољава дефинисане критеријуме.

Основне предности потпуно релационог система за управљање базом података су:

- администраторима базе омогућено је да у сваком тренутку знају које се врсте података налазе у бази и како се њима може приступити;
- продуктивност програмера је побољшана с обзиром на чињеницу да је подржано итеративно тестирање програма и наредби у јединственом окружењу, унутар једног језика;
- општа независност апликационих програма од података олакшава развој и одржавање информационих система (последица физичке, логичке, интегритетске и дистрибуционе независности);
- смањени су трошкови измена описа података у бази, приликом прилагођавања програма новим захтевима (промене перформанси система, организационе промене, промене правила пословања)
- трошкови комуникација у дистрибуираној бази података су смањени захваљујући особини додавања, брисања и модификовања на високом нивоу;
- обрада свих језика за манипулацију подацима у релационом систему ослања се на запамћене дефиниције података, укључујући правила интегритета и ограничења;

Познавање релационог модела основни је предуслов за успешно дизајнирање базе података која ће подржавати рад прилагодљиве и лако доградиве апликације.

Релациони модел одваја базу података од оперативног система као и од апликације. Када се да захтев за информацијама, систем направи табелу која садржи те информације. Стандардни програмски језик за изражавање приступа подацима и манипулацију са табелама у релационој бази података се назива SQL (енгл. *Structured Query Language*). У овом језику, питања на једноставном енглеском језику се аутоматски превode у SQL. У овом случају софтверски програм, који се зове природни језик (енгл. „Natural language“) и који дозвољава упите у ограниченој форми природног језика, анализира корисников упит, преводи га у упит на SQL, преноси SQL захтев DBMS-у и даје на дисплеју податке кориснику.

3.5.1. Представљање података

Сви подаци у релационој бази представљају се на логичком нивоу на јединствен начин - преко вредности у табелама.

Под појмом “сви подаци” подразумевају се подаци које је корисник дефинисао и унео у базу без обзира на то да ли је реч о:

- основним подацима (који се односе на вредности атрибута),
 - тзв. мета-подацима (којима су дефинисани називи табела, колона и др.),
- или
- дефиницији различитих правила као што су правила интегритета.

Наиме, независно од тога на шта се подаци односе, они су у релационој бази доступни као неке вредности у одређеним табелама. На тај начин обезбеђена је јединствена структура података и мета-података у систему.

3.5.2. Приступ подацима

Сваком основном податку (конкретној вредности колоне у неком слогу табеле) у релационој бази увек се може приступити коришћењем комбинације имена табеле (релације), вредности примарног кључа и имена колоне (атрибута).

Последица наведене особине је да релациони модел омогућава лако руковање подацима на логичком нивоу, с обзиром на то да се до података долази без примене поступка секвенцијалне претраге или неког сличног, у сваком случају мање ефикасног поступка.

Многи системи који се данас на тржишту софтвера рекламирају као релациони показују извесне неправилности у односу на описано правило. Наиме, они дозвољавају креирање табела без дефинисаног примарног кључа, тј. табела које садрже слоге са потпуно идентичним вредностима у свим колонама. У таквим случајевима, табеле без дефинисаног примарног кључа губе смисао релације дефинисане у релационом моделу. Приступ подацима у таквим табелама је отежан, а самим тим отежано је и руковање подацима.

Наведени проблем могао би се избећи увођењем “дисциплине” у поступку креирања релација, тј. обавезном дефиницијом примарног кључа за сваку табелу.

Потпуно релациони систем за управљање базом података обавезно системски подржава представљање информација које недостају у бази, употребом тзв. NULL вредности, независно од типа података. NULL вредност је специфичан индикатор различит од празног низа карактера или низа “бланкова” и различит од нуле или било ког другог броја.

РДБМС обезбеђује јединствену презентацију NULL вредности, а на тај начин и једноставно руковање тим вредностима. Вредност NULL може бити потенцијална вредност сваке колоне без обзира на њен тип, осим у случају кад се изричито захтева да вредности у некој колони не смеју бити недефинисане (додатно ограничење NONULL). Изузетак представљају колоне примарног кључа које, као идентификатори записа у табели, не смеју узети NULL вредност.

Релациони систем поседује каталог (речник) података који се на логичком нивоу представља на исти начин као и сами подаци, тако да овлашћени корисници могу примењивати јединствени релациони језик за претрагу ових мета-података.

Под појмом "каталог" подразумева се директоријум у којем се налазе системске дефиниције података. Увидом у каталог података може се сазнати које колоне поседују одређене табеле, која су ограничења дефинисана, као и друге информације које се односе на системске табеле. У случају дистрибуираних база података, каталог треба да садржи податке о локацији сваког дела базе.

Каталог називамо “динамичким” због чињенице да дефиниције података које се могу виђети одговарају управо текућем стању у бази података. Без обзира на то који корисник, на којој локацији, или у ком ђелу описа базе направи неку измену, ново стање ће бити доступно за читање свим овлашћеним корисницима.

“Он лине” приступ каталогу практично значи да је тренутном стању описа базе обезбеђен непрекидан приступ. Међутим, све док се стање у опису базе коначно не промени, овлашћеним корисницима је видљиво стање пре промена. Наиме, у релационој бази све операције које мењају опис података започињу и завршавају се одређеним наредбама којима се систем обавештава да је ажурирање комплетно. “On line” увидом не види се ни једна измена која није коначна.

Веома је важна чињеница да право приступа опису базе имају само за то овлашћени (ауторизовани) корисници.

3.5.3. Захтеви које треба да испуни језик за рад са релационим базама података

Без обзира на то колико језика и колико начина коришћења терминала подржава, релациони систем мора поседовати бар један језик чије се наредбе могу изразити као низ карактера са добро дефинисаном синтаксом, а који свеобухватно може да подржи:

- 1) дефиницију података
- 2) дефиницију погледа
- 3) руковање подацима
- 4) дефиницију правила интегритета
- 5) права у систему (ауторизацију)
- 6) границе трансакције (BEGIN, COMMIT, ROLLBACK).

У наведених 6 ставки обухваћене су све функције једног РДБМС-а, које се сагласно суштини овог правила, морају одвијати у јединственом окружењу, унутар једног језика.

То подразумева да се сваки задатак може комплетно извршити, без напуштања таквог свеобухватног окружења.

За администраторе система постојање свеобухватног језика омогућава да се, приликом руковања подацима, избегне низ сувишних корака.

Илустрације ради, кад је потребно сачувати неке међурезултате или слоге, избегавају се следећи кораци:

- напуштање датог окружења,
- улазак у другу апликацију и дефинисање нове табеле у њој,
- повратак у старо окружење и унос података у креирану табелу.

Ажурирање погледа

Релациони систем поседује ефикасан алгоритам за ажурирање свих погледа који се теоријски могу ажурирати. Резултат овог алгоритма смешта се у каталог (речник) базе података.

Погледи представљају виртуелне релације које физички не постоје али се у сваком другом смислу понашају као релације. Може се рећи да су то изведене табеле над којима је могуће дефинисати упите као и над било којим постојећим табелама у бази. Поглед се понаша као “динамички прозор” кроз који се гледају релације базе и путем којег је могуће вршити промене које ће се аутоматски рефлектовати на релацијама базе.

Сваки пут када корисник чита податке кроз поглед, систем формира резултујући скуп слогова на основу дефиниције погледа и података у табелама базе. Резултујући скупови слогова, односно посебне копије података који се виде кроз поглед, не чувају се у систему, већ се само у каталогу базе чувају дефиниције погледа.

Под ажурирањем погледа подразумевају се операције брисања и додавања слогова, као и модификовање постојећих слогова. Код наглашава да је “поглед теоријски могуће ажурирати ако постоји временски независан алгоритам који недвосмислено одређује низ измена у базним релацијама, чији је коначни ефекат управо захтевана измена над погледом.”

Додавање, брисање и модификовање на високом нивоу

Не само за издавање података, већ и за операције додавања, брисања и модификовања података постоји могућност да се рукује базним и изведеним релацијама у целости као операндима.

Главни објект у релационој бази података је релација. Све операције релационе алгебре рукују релацијама као операндима, а резултат сваке операције је такође нека релација. Ако издвојимо један конкретан слог из релације можемо га посматрати као релацију са једним слогом. Конкретан податак (основна вредност) издвојен из базе, такође представља релацију над само једним атрибутом са само једним слогом (табела са једном колоном и једним редом). У том смислу у бази података се увек рукује са релацијама у цјелости.

Ова чињеница позитивно утиче на оптимизацију поступка спровођења операција у систему. Илустрације ради, посматрајмо читање целе релације. Уколико за сваки слог који се чита морамо систему да задамо посебну команду, систем мора појединачно да оптимизује и извршава сваку од њих. У случају читања цијеле релације једном операцијом, систем одређује најбољи начин приступа слоговима у глобалу за целу релацију. Исте предности односе се и на операције брисања, модификовања и додавања слогова.

Описано правило има значајну улогу у обради трансакције над дистрибуираном базом података. Чињеница да се релацијама рукује као операндима, рефлектује се на ефикаснију обраду података. Уштеде се остварују не само у времену већ и у трошковима комуникација. Употребом оваквих упита на високом нивоу избегава се пренос појединачних захтева, за сваки слог, према удаљеном систему.

Физичка независност

Релациони системи у обавези су да задовоље правило физичке независности чија је суштина у следећем: апликациони програм и активности на терминалима остају неизмењени када се промени физичка организација базе или физички метод приступа подацима.

Апликације и програми рукују подацима искључиво на семантичком и логичком нивоу. Сама физичка организација података представља интерну ствар система и има утицаја једино на перформансе (брзину приступа подацима, ожив система). Неопходно је да у систему постоји оштра граница између логичког модела података и физичког размештаја података на медију.

Креирање и брисање индекса над неком табелом из базе, не сме имати утицаја на логичку структуру програма и наредби. Код нерелационих система за управљање базом података сами програми би морали претрпети измене у зависности од постојања индекса над одређеним табелама.

Наведено правило омогућава да се уоче две основне групе корисника у систему:

- пројектанти и програмери,
- администратори базе података.

Улога пројектаната и програмера је да се баве логичким дизајнирањем и програмирањем, док се администратори базе, између осталог, баве и одржавањем интерне организације података у систему. Основна компонента која у актуелним релационим системима обезбеђује физичку независност је тзв. оптимизатор SQL наредби.

Оптимизатор за сваку SQL наредбу одређује оптималан план извршења у смислу времена ожива и укупног утrophка ресурса система. Сам процес оптимизације одвија се у тренутку извршења наредбе, а састоји се од анализе доступних индекса и начина приступа подацима. План извршавања наредбе за који се предвиђају најбоље

перформансе у обради бира се на основу хеуристичких правила или статистичких података о бази.

Логичка независност

Када се на табелама базе изврше измене које не изазивају губљење података или логичка оштећења, апликациони програми и активности на терминалима остају логички неоштећени. Уколико РДБМС испуњава услов ажурирања погледа онда ће без проблема моћи да задовољи и правило логичке независности.

Када спајамо две табеле могућност ажурирања погледа над новом табелом услов је за очување логичке независности.

Логичка, као и физичка независност, за последицу имају једну веома важну предност коју релациони системи показују у односу на нерелационе. Предност се састоји у томе што грешке у почетном дизајну модела података немају тешке последице и дозвољавају накнадне корекције. Пројектанти у том смислу могу да без превеликог оптерећења креирају иницијалне моделе података, а да их касније, чак и у поодмаклој фази имплементације апликације, поново дорађују и прилагођавају новим захтевима.

Интегритетска независност

Сва правила интегритета дефинишу се у оквиру дефиниције базе података и чувају се у каталогу података, а не у апликационим програмима.

Интегритет података дели се на 4 категорије:

- интегритет ентитета,
- интегритет домена,
- референцијални интегритет, и
- кориснички дефинисан интегритет.

Интегритет ентитета подразумева да свака вредност примарног кључа мора бити јединствена на нивоу целе релације. Осим тога, мора бити испуњен услов да ни једна ставка у колони примарног кључа не сме имати NULL вредност.

Интегритет домена одређује скуп дозвољених вредности колоне. Одржавање интегритета домена постиже се навођењем дозвољеног типа података, дозвољеног формата за унос података, или задавањем скупа могућих вредности.

Референцијални интегритет одражава дефинисане односе (релације) међу табелама када се у њих додају или се из њих бришу записи.

Кориснички интегритет омогућава дефинисање специфичних пословних правила. Наиме, за базу података могу постојати додатна ограничења, формирана на основу одређених правила пословања, која дефинишу важеће податке у бази.

РДБМС треба да обезбеди дефинисање свих наведених категорија интегритета пођезиком релационих података. Када се правила интегритета једном дефинишу она спречавају упис података који не задовољавају те дефиниције (неконзистентни подаци).

У случају промене правила интегритета долази до измене једног или више исказа ограничења у каталогу података, али логичка структура програма и активности на терминалима остају неизмењени.

Једино могуће средство за чување интегритета базе података, код постојећих релационих система, је коришћење тзв. окидача (Triggers). Окидач је покретач процедуре која се аутоматски позива кад год дође до додавања, ажурирања или брисања података у табели. Окидач је објект базе који се чува у каталогу података и сваки је појединачно придружен једној табели, при чему се наводи које од операција додавања, ажурирања или брисања изазивају његово извршење. Тако се, на пример, за базу података текућих рачуна грађана у некој банци може дефинисати окидач који се извршава приликом брисања слогова из табеле, а има улогу да онемогући брисање рачуна чије стање није равно нули.

Дистрибуциона независност

У зависности од тога колико примерака вредности логичких објеката из базе постоји и на колико локација се они налазе, разликујемо два типа архитектуре система базе података:

1. централизовани систем, и
2. дистрибуирани систем.

У централизованом систему база података је скуп логичких објеката и њихових вредности, где постоји само један примерак вредности сваког логичког објекта базе.

У дистрибуираном систему вредност неког логичког објекта може имати више примерака, меморисаних у више локалних база, на различитим локацијама.

Е.Ф.Цодд је дефинисао дистрибуциону независност РДБМС-а као његову способност да апликациони програми и активности на терминалима остану логички неоштећени:

- када се први пут уведе дистрибуција података, и
- када се подаци предистрибуирају.

Први случај подразумева да је систем првобитно управљао само недистрибуираним подацима, а онда је дошло до њихове дистрибуције. У другом случају систем је већ управљао дистрибуираним подацима, али је дошло до промене њиховог распореда.

Релациони систем који је дистрибуционо независан омогућава да једна трансакција рукује подацима са више удаљених система, а да корисник при томе има утисак да се све одвија у оквиру локалног система. Ова особина даје велику моћ наредбама језика података на високом нивоу, које могу унутар једне команде комбиновати податке са различитих места.

Рад са језицима ниског нивоа

Иако у обављању својих функција користи језике високог нивоа (принцип групног приступа слоговима), систем за управљање базом података може да садржи и неке од језика ниског нивоа (појединачни приступ слоговима) и да подржава њихов рад.

Услов који РДБМС треба да задовољи је да се у раду са језицима ниског нивоа не смеју нарушити или заобићи правила интегритета и ограничења изражена релационим језиком високог нивоа.

Код нерелационих система чест је случај да језици ниског нивоа заобилазе дефиниције из каталога података. У том случају процес обраде се не ослања на запамћене дефиниције података, правила интегритета и ограничења.

Високе могућности SQL-а у погледу руковања релацијама као операндима (у целости), углавном нису примењљиве у језицима као што су C и PASCAL, из једноставног разлога што у окружењу ових језика релација нема одговарајућу презентацију.

У циљу превазилажења овог проблема морао се обезбедити приступ и руковање подацима на нивоу слогова применом концепта курсора (cursor).

Курсор у суштини представља објекат дефинисан SELECT наредбом, над којим се могу реализовати три операције:

- отварање курсора (OPEN),

- дохватање следећег слога из курсора (FETCH), и
- затварање курсора (CLOSE).

Отварањем курсора формира се активни скуп слогова (нека врста интерне копије слогова који се смештају у меморију). Над тим скупом слогова извршавају се операције дохватања једног по једног слога све док се не дође до последњег из издвојене групе. Затварањем курсора ослобађа се меморијски простор активног скупа слогова.

3.6. Софтверски пакети за рад са базама података

Базе података се у правилу реализују коришћењем неког од проверених софтверских пакета. Табеларни приказ 1, даје преглед софтвера који у овом тренутку имају значајан удео на светском тржишту.

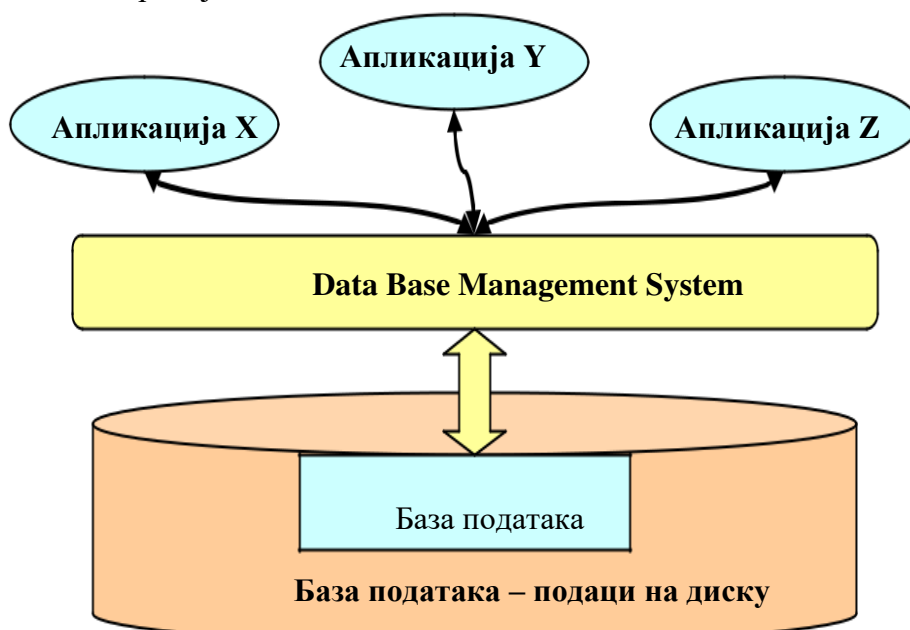
Proizvođač	Produkt	Operativni sistem	Jezici
IBM Corporation	DB2	Linux, UNIX (razni), MS Windows NT/2000/XP, VMS, MVS, VM, OS/400	SQL, COBOL, Java, ...
Oracle Corporation	Oracle	MS Windows (razni), Mac OS, UNIX (razni), Linux i drugi	SQL, Java i drugi
IBM Corporation (prije : Informix Software Inc.)	Informix	UNIX (razni), Linux, MS Windows NT/2000/XP	SQL, Java i drugi
Microsoft	MS SQL Server	MS Windows NT/2000/XP	SQL, C++, ...
MySQL AB	MySQL	Linux, UNIX (razni), MS Windows (razni), Mac OS	SQL, C, PHP, ...
Sybase Inc.	Sybase SQL Server	MS Windows NT/2000, OS/2, Mac, UNIX (razni), UNIXWare	SQL, COBOL, ...
Hewlett Packard Co.	Allbase/SQL	UNIX (HP-UX)	SQL, 4GL, C, ...
Cincom Systems Inc.	Supra	MS Windows NT/2000, Linux, UNIX (razni), VMS, MVS, VM	SQL, COBOL, ...
Microsoft Corporation	MS Access	MS Windows (razni)	Access Basic, SQL

Табела 1. Софтверски пакети за рад са базама

Готово сви данашњи софтверски пакети подржавају релациони модел и СQL. Сваки од њих садржи свој ДБМС, уобичајене клијенте (нпр. интерактивни интерпретер СQL), те библиотеке и алате за развој апликација. Сваки пакет испоручује се у верзијама за разне рачунарске платформе (операционе системе). Конкуренција међу произвођачима софтвера за базе података је изузетно велика, тако да је последњих година често долазило до њиховог нестанка, спајања или преузимања. Листа релевантних софтверских пакета зато је сваке године све краћа.

4. Приступи, упити и претраживање база података

SQL омогућава пуни приступ подацима у релационим базама података (као што су: Oracle, SQL Server, Acces, и др.) тако што корисник опише податке које жели да види. SQL омогућава и дефинисање и модификацију изгледа табела унутар базе података. Нити једна операција на базама података, извршена директно из ДБМС-а или преко корисничке апликације, не би могла бити извршена без директне или индиректне употребе SQL језика. Често, сам SQL није довољан уколико је потребно извршити неки упит у тачно одређено време или га поновити неколико пута. Због тога сваки ДБМС поред SQL подржава бар још један програмски језик помоћу којег се могу извршити комплексне операције.



Слика 4. ДБМС је интерфејс између (апликација) корисника и записа базе података на диску

Релациони систем за управљање базом података омогућава обављање најразличитијих операција над релацијама и комбиновање релација да би кориснику обезбједио одговоре на сва питања која имају смисла с обзиром на садржај базе података. Оснивач релационе теорије Е.Ф.Цодд, још 1985. године, дефинисао је 12 строгих правила која мора задовољити РДБМС да би с правом носио епитет “релациони”. Основни принцип на којем се заснивају правила гласи: “Сваки систем који тврди да је релациони систем

за управљање базом података, или се тако рекламира, мора бити у могућности да у потпуности управља базом података својим релационим способностима."

Database API (енгл. **application programming interface**) :

ODBC (енгл. *Open Database Connectivity*) – представља најстарији Мицрософтов стандард за повезивање апликационог софтвера са различитим релационим базама података који је развијен 1992 од SQL Access Group. Старији COM интерфејси као што су DAO (енгл. *Data Access Objects*) и RDO (енгл. *Remote Data Objects*) заснивали су се управо на моделу ODBC преко кога су приступали одређеној бази података. Помоћу њега програмери могу радити са табеларним подацима, као што су SQL базе података или са мултидимензионалним подацима, као што је OLAP (енгл. *On-Line Analytical Processing*) коцке.

OLE DB (енгл. *Object Linking and Embedding Database*) – Мицрософт стандард који дефинише јединствено подржавање апликација које приступају релационим и нерелационим базама података, појавио се као одговор проблему приступања комплексније организације података, као што су текст фајлови, е - маил системи и др. То је нова верзија ODBC-а.

JDBC (енгл. *Java Database Connectivity*) – је еквивалент ODBC технологије намењен за повезивање на различите релационе базе из Јава кода.

DAO (енгл. *Data Access Objects*) – се јавља као решење у прављењу објектног модела за приступ бази података који спречава да дође до евентуалних грешака при програмирању ODBC, OLE DB и JDBC. Он је саставни део Visual Basica. Служи за приступ MS Access базама података. DAO такођер омогућава приступ ODBC изворима података и ту лежи и његов највећи недостатак. Пошто се ослања на Access, приликом приступа ODBC базама података све наредбе за базу података и сви подаци из ње морају проћи кроз овај додатни слој, што може знатно угрозити перформансе апликације.

ADO (енгл. *ActiveX DataAccess Objects*) – представља Microsoftov COM базирани објектни модул који омогућава да управљамо OLE DB изворима података

WEB/DataBase систем код клијент сервера - Данас, готово све информације добијамо онлине, путем Интернета. На овај начин, проналажење информација постаје брзо и лакше. Очигледно је да велики део тога добијамо путем многих веб страница. Линкови на почетној страници обезбеђују ткз. Корпоративни Интранет, који омогућује велики

број различитих информација из организације. Да би се обезбедио приступ великим базама података потребно је да се веб сервиси, путем сопственог ДБМС, повежу са базом података. Клијент који ради као веб претраживач издаје захтев за добијање информација у облику URL (енгл. *Uniform Resource Locator*) референцу. Ова референца покреће програм на веб серверу, који сада издаје исправну команду бази података и упућује је ка серверу базе података. Подаци се враћају веб серверу, конвертују се у HTML и шаљу клијенту.

Предности WEB/DataBase система су :

- Администрација: једина веза са базом података је веб сервер. Свако додавање новог типа сервера базе података не захтева конфигурацију нових драјвера и интерфејса за сваког клијента већ само на веб серверу.
- Развој: Веб претраживачи су доступни скоро на свим платформама. Они ослобађају програмере апликација да имплементирају графички кориснички интерфејс за различите рачунаре и оперативних система (ОС). Већина веб претраживача подржава нове веб сервисе што ослобађа клијенте од инсталације и синхронизације.
- Брзина: Велики део нормалног развојног циклуса, као што су развој и дизајн клијента, не односе се на пројекте везане за веб апликације.
- Презентација: Мултимедијална особина веб сервиса омогућује приказ података у великом броју различитих формата који су најбољи за податке.

Недостаци Web/DataBase система у односу на традиционални приступ:

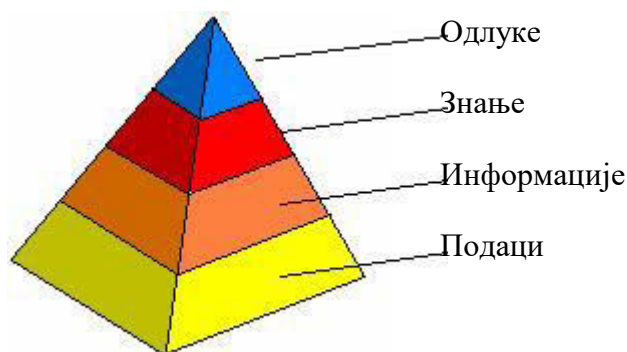
- Функционалност: У поређењу са функционалношћу класичног приступа у графичком приказу података типичан веб претраживач је ипак ограничен.
- Операције: природа HTTP је таква да свака интеракција између претраживача и сервера представља посебну трансакцију која је независна од претходног или наредног захтева. Веб сервер не прати стања корисника.

4.1. Напредне анализе података – интелигентно пословање

Интелигенција је ментална особина која се састоји од више способности: учење из искуства, адаптирање на нове ситуације, схватање и разумевање нових ситуација и коришћење стеченог знања у интеракцији са окружењем. Коришћењем овог тумачења,

појам „пословне интелигенције“ односно „интелигентног пословања“ се може дефинисати као процес прикупљања најразличитијих података, њихово претварање у корисне информације, које могу да помогну пословним корисницима у доношењу одлука.

Интелигентно пословање је посебно почело да се развија када су предузећа почела да аутоматизују своје пословне процесе. Постојећи трансакциони системи су се показали као савршени генератори великих количина података. Подаци су се временом све више gomилали, али се јавио велики проблем - до тих података је било све теже доћи. Било је јасно да у тим подацима лежи огроман потенцијал, па се јавила потреба да се они некако обједине, обраде и ставе на располагање руководству. Тако је настало интелигентно пословање, које се, са техничке стране, најједноставније може описати као процес у коме се подаци претварају у информације. Информације су резултат анализе и организовања података тако да се од њих добија неко ново сазнање. Знање је одговарајући скуп информација. На основу постојећег знања, могу се доносити пословне одлуке.



Слика 5. Процес доношење одлука од расположивих података

Предности које доносе системи интелигентног пословања су: анализа пословања, анализа клијената, боља контрола трошкова пословања, напор пословања, брзо реаговање, предвиђање будућих трендова, једноставни графички прикази, једноставно прављење и коришћење извештаја... На тржишту доношење правовремених одлука омогућава пружање бољих услуга клијентима, издизање изнад конкуренције, а на основу тога и повећање профита.

Систем интелигентног пословања омогућује историјске, садашње и будуће погледе на пословне процесе. Заједничке функције система пословне интелигенције су: извештавање, ОЛАП (енгл. *Online Analytical Processing*) аналитике, истраживање

података, управљање пословним процесом, тестирање карактеристика система итд. Циљ система интелигентног пословања је да помогну приликом одлучивања.

Језгро сваке пословне активности је обрада информација. То укључује скупљање, чување и обраду података. Значај добре информације је у разлици између доношења исправне и погрешне одлуке. Добра информација мора да буде тачна, благовремена и разумљива.

Пословни процес располаже великим бројем података, који су обично у различитим базама, а често и на различитим физичким локацијама. Подаци су или доступни или архивирани, али се догађа да су у различитим форматима. Приликом доношења одлука, кориснику је потребно да ти подаци буду на једном месту (бар виртуално) и да приступ њима буде брз упркос величини података и колико год да су они стари. У те сврхе се користи тзв. складиште података (*data warehouse*) које обједињује различите изворе података. Ажурирање складишта се врши само додавањем нових података, док постојећи подаци најчешће остају непромењени. Основни извори података за концепт складишта су: оперативни подаци (транзакциони, тзв. ОЛТП (енгл. *On-Line Transaction Processing*)), спољне информације настале као историја пословања и разни други подаци узети из јавних база података. Најнижи ниво обраде података је транзакционо оријентисан и подржава обраду свакодневних оперативних послова. Основна карактеристика транзакционог приступа су нормализовани модели података, кратко време обраде, велики број транзакција које раде са релативно малим бројем табела и са релативно малим бројем операција над њима. Дубља анализа података коришћењем ОЛТП система је отежана и непогодна и своди се на анализу оперативних података и на коришћење извештаја који се директно генеришу над оперативним подацима.

Анализа података је значајан део интелигентног пословања, јер коришћењем одговарајућих алата за анализу може да се дође до релевантних информација потребних за доношење нових одлука у пословном процесу. Различите технике анализе података су: ОЛАП, истраживање података (*Data mining* - ДМ), симулације, коришћење упитних језика, итд.

Истраживање података је процес који пружа информације из велике количине података. Идентификују се везе између наизглед неповезаних података. Основна циљ је да се из великог броја оперативних података и веза које не могу одмах да се сагледају, дефинишу одговарајуће релације и обрасци понашања, на основу чега би се добиле

потребне информације. Симулационим алатима се тестирају будућа могућа стања на основу трендова у пословању и омогућује се формирање нових пословних правила. Упитни језици представљају стандардни упитно - извештајни алат који омогућава корисницима да претражују, анализирају и узимају појединачне податке из постојећих база ради формирања различитих извештаја. Приступ подацима путем упитног језика омогућава детаљан преглед података и њихових односа у трансакционој бази.

ОЛАП омогућава брз, интерактиван приступ подацима, пружајући широк спектар различитих мултидимензионих погледа на податке. Повезани пословни појмови се анализирају преко димензија. Идеја овог приступа је у томе да пословање предузећа, а и функционисање људског ума почива на вишедимензионалности. Рецимо, ако је потребно да се анализира продаја, онда желимо да је анализирамо у времену, по регијама, по продајним местима, по купцима, по артиклима... Наведени појмови су, у ствари, различите димензије појма продаја. Основна разлика између ОЛАП-а и истраживања података је у самом приступу подацима. ОЛАП алати омогућавају вишедимензионе анализе, које обухватају рашчлањавање података на ситније детаље, као и њихово сумирање. Истраживање података је, с друге стране, проучавање односа међу подацима и поналажење неких образаца и међусобног утицаја података. ОЛАП и истраживање података могу заједно да се користе као веома моћно средство за анализу.

4.1.1. ОЛТП

ОЛТП (енгл. *On – Line Transsaction Processing*) означава обраде при којима систем у кратком временском периоду одговара на захтев корисника. У процесу пословања, јавља се потреба за изградом извештаја који би помогли руководству у доношењу одлука. Проблеми код ОЛТП система приликом извештавања су следећи: приступ подацима је компликован, обрада података успорава пословне трансакције, подаци су различити и сложени. Извештаји се израђују помоћу упитног језика. За то је потребно одлично познавање базе података и потпуно владање упитним језиком. За реализацију врло сложених упита је потребно дуго време оцива, а обим података који се могу узети у обзир је ограничен.

Едгар Код, аутор релационог модела база података, је дефинисао појам ОЛАП-а 1993. године. Сврха ОЛАП алата је анализа и обрада података и израда извештаја који ће бити од користи за стратешко управљање. Долази се до новог погледа на постојеће

податке. Разлике између дотадашњег и новог приступа су приказане у наредној табели 2.

	ОЛТП	ОЛАП
корисници	službenici	„pametan“ korisnik
функција	свакодневне операције	podrška odlučivanju
дизајн базе података	апликативно орјентирана, високо нормализована база са више табела	орјентисана према области коришћења, денормализована база са мање табела
подаци	оперативни, оригинални извор	скупљени из различитих ОЛТП база података
коришћење	често	ad hoc
приступ	Читање/писање	много прегледа
Јединица посла	Кратка, једноставна	сложени упити
Број слогова којима се приступа	десетине	милиони
Број корисника	хиљаде	стотине
Величина базе	100 MB - GB	100 GB- TB

Табела 2. Поређење ОЛТП и ОЛАП система

4.1.2. Дефиниција ОЛАП-а

ОЛАП (енгл. *On - Line Analytical Processing*) представља скуп алата који омогућују анализу података сачуваних у бази података. Помоћу ОЛАП-а, руководиоци и аналитичари могу да имају брз, интерактиван приступ подацима кроз широк спектар различитих углова посматрања. ОЛАП припада категорији апликација и технологија за сакупљање, управљање, обраду и представљање мултидимензионих података за потребе анализе. Најшире усвојену дефиницију ОЛАП-а, која се данас користи, дао је Нигел Пендсе и описана је помоћу следећих 5 речи: Брза Анализа Дељених Мултидимензионих Информација:

- **Брза** - односи се на брзину којом ОЛАП може да донесе што већи број одговора својим крајњим корисницима.

- **Анализа** - односи се на способност ОЛАП система да савлада пословну логику и статистичке анализе релевантне за апликацију и корисника и да их довољно поједностави за разумевање крајњем кориснику.

- **Дељених** - омогућено је да се подаци деле између више корисника.

- **Мултидимензионих** - односи се на на концепт који је примаран захтев ОЛАП-а. ОЛАП систем мора да омогући мултидимензиони поглед на податке и да укључи хијерархије и вишеструке хијерархије димензија.

- **Информација** - представља све податке и изведене податке који су релевантни за апликацију.

Појам ОЛАП обухвата више целина између којих не постоји јасна граница. Неке од њих су:

- ОЛАП концепти који обухватају идеју о вишеструким димензијама са хијерархијама.

- ОЛАП формални језици који обухватају: језик за дефинисање података (*Data Definition Language* - ДДЛ), језик за обраду података (*Data Manipulation Language* - ДМЛ), језик за приказивање података (*Representation Language* - ДРЛ) и повезане анализаторе (опционо и компајлере). ОЛАП језици могу да се користе за описно моделовање, или за трансакциону подршку или подршку у одлучивању.

- ОЛАП производи који се ослањају на релациону базу података. Сваки упит који корисник постави ОЛАП систему се трансформише у низ SQL наредби. Чување података и приступ њима су дефинисани на нивоу базе.

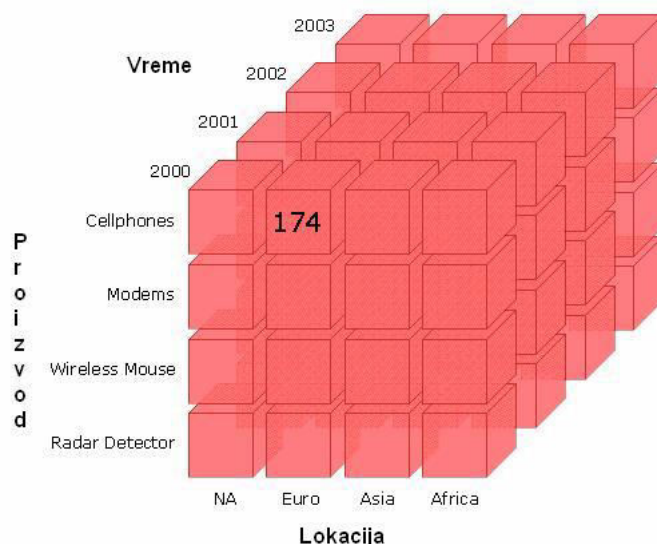
- ОЛАП производи који садрже компајлер, методе чувања и приступа подацима. Ови производи су оптимизовани за брз приступ подацима и брза израчунавања и користе се у системима за подршку одлучивању.

4.1.3. ОЛАП коцка

Према самој дефиницији ОЛАП-а, кључни захтев је вишедимензионалност. ОЛАП постиже своју вишедимензиону функционалност коришћењем структуре која је названа коцка. Коцка омогућује мултидимензиони поглед на податке, а, са друге стране, може

да се пореди са табелом у бази податка. Специфичан дизајн ОЛАП коцке гарантује оптимизацију извештаја.

Вишедимензионалне структуре података (описане у претходном поглављу) се најбоље визуелизују као коцке података које се састоје из мањих, „јединичних“ коцки. Свака страна коцке је једна њена димензија. Димензија представља скуп категорија података исте врсте. Нпр. у коцки продаје једна димензија је локација, њене различите категорије могу бити градови, окрузи, државе... Свака ћелија те коцке садржи агрегиране податке (мере о којима је раније било говора) који су у вези са димензијама. Нпр. једна ћелија може да садржи податке о укупној продаји за да ти производ и регион у току једне године.

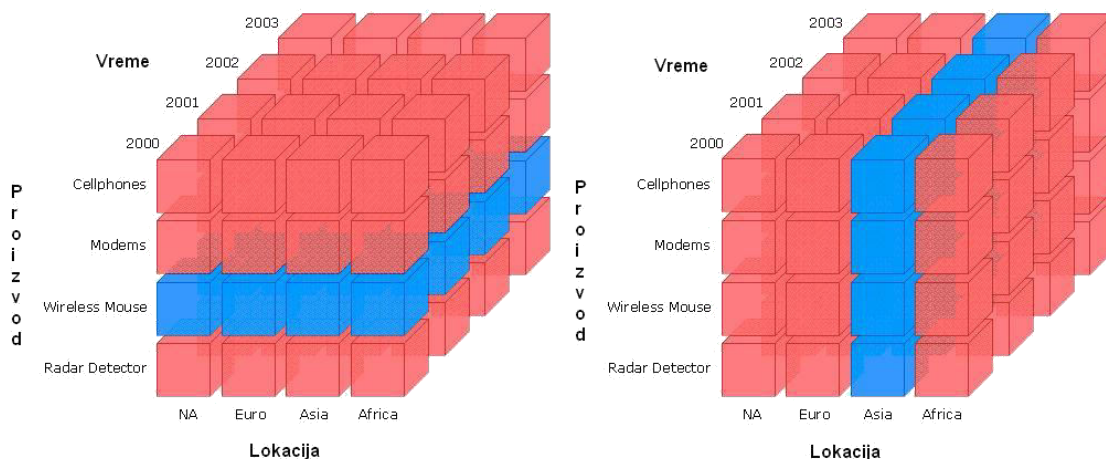


Слика 6 . ОЛАП коцка са подацима о продаји

4.1.4. Операције над ОЛАП коцком

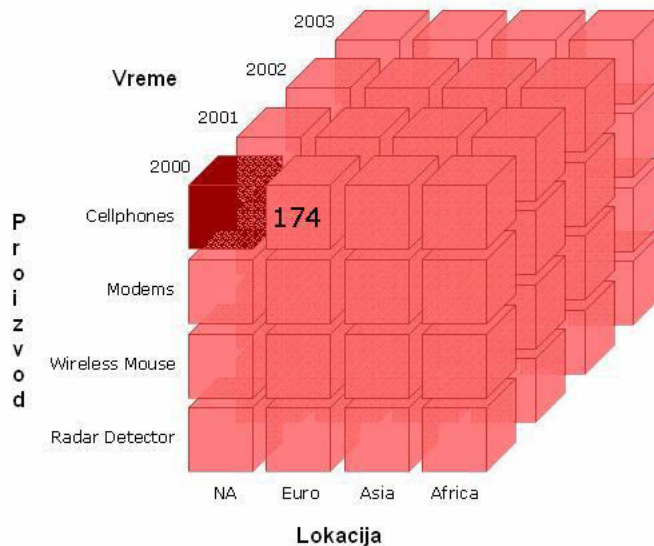
Основне операције које су битне за анализу података помоћу ОЛАП коцке су: сечење на појасеве, комадање, бушење, завијање и ротација.

Сечење на појасеве представља издвајање података за дати услов по једној димензији. На следећој слици је приказана продаја за конкретан производ - бежични миш, односно локацију – Азија.



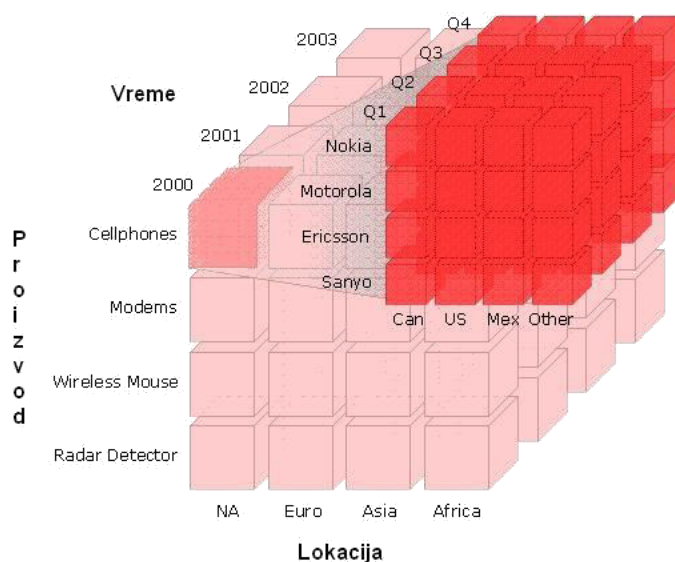
Слика 7. Операција сечење ОЛАП коцке на појасеве

Комадање је издвајање података за услове по две или више димензија. На следећој слици је приказана продаја телефона за 2000. годину у Северној Америци.



Слика 8. Операција комадања ОЛАП коцке

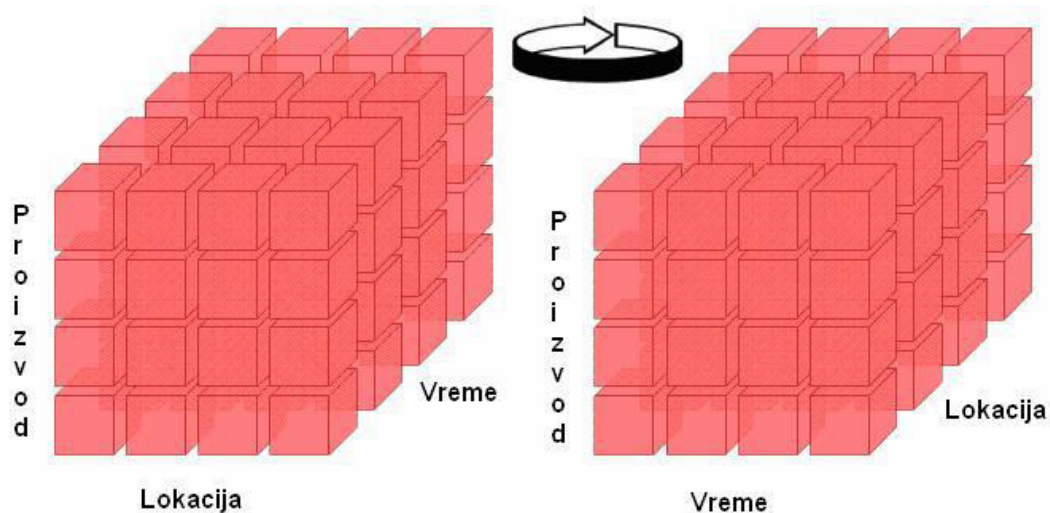
Бушење представља детаљизацију коцке, спуштањем по хијерархији димензије. На следећој слици је приказана комбинација операција комадања (приказ продаје телефона у Северној Америци 2000.године) и бушења (временска димензија је приказана по категорији квартала, локација по државама, а групе производа по појединачним врстама).



Слика 9. Комбинација операција комадања и бушења ОЛАП коцке

Завијање је супротна операција од бушења, тј. представља издвајање података пењањем по хијерархији неке од димензија.

Ротација се у литератури назива и пивотирање. Представља визуелизацију коцке окретањем димензионих оса ради алтернативног приказа података (Слика 10).



Слика 10. Операција ротације ОЛАП коцке

5. Анализа реалног система “Предшколска установа”

У наставку рада ће бити поступно приказан процес креирања информационог система за Предшколску установу. Дефинисање граница система представља тачку од које се креће при изради неког информационог система. При утврђивању граница система треба јасно дефинисати циљеве који морају да садрже следеће елементе:

- Зашто се процес моделира
- Шта ће процес да прикаже
- Шта ће корисник модела да направи са њим и
- Чему служи модел

Како се у предшколској установи већ писмено документују активности деце по питању успешности храњења, спавања и организовање разних дечијих представа и екскурзија дошли смо на идеју да би се креирањем ИС-а трајно документовали ови подаци чиме би они постати доступни родитељима свуда (родитељи би им могли приступити и од куће), а могле би се вршити и разне анализе и контроле од надређеног кадра. Све укупно васпитни кадар се не би оптеретио додатним послом, већ би уместо ручног попуњавања вршио унос податка у рачунар, а расположивост подарака би тиме тренутно постала већа.

У предшколској устанви деца су распоређена по узрастним групама. Свака група има одређени број деце које могу бити уписане у њу и надлежне васпитаче. Свака група има и своју собу у којој се одвијају свакодневне активности, а које свакодневно надлежни васпитачи воде у дневнику активности. У дневнику активности се уписује да ли је тог дана дете доручковало, ужињало, ручало и одморило. Овим се родитељима даје обавештење о очекиваним потребама детета ван предшколске установе.

У предшколској установи су запослени васпитачи који су распоређени по групама, али има и васпитача који нису надлежни ни за једну групу. Они могу бити резервни васпитачи (у случају болести неког од надлезних васпитача за групу), водити децу на екскурзије или их обучавати за представе. За сваког васпитача је познато име, презиме и ЈМБГ (Јединствен Матични Број Грађана).

За свако дете је познато име, презиме, ЈМБГ, име једног од родитеља и његов број телефона. Када се дете упише у вртић отвара се картон детета, где се уписују примљене

вакцине, нутритивне алергије (уколико их дете има) и посебне потребе (уколико их дете има). Посебне потребе се представљају посебном дијагнозом по МКБ10 шифарнику дијагноза, која је од значаја за посебан запослени кадар.

Током године у вртићу се организују разне представе. Свака група припрема по једну из скупа познатих представа, а обучава их један или више васпитача. То може бити и васпитач који није надлежан за ту групу. Такође, током године се организују и разне екскурзије на које могу ићи деца и за које је познат тачан термин поласка. Неопходно је да се неко дете пријави како би један или више васпитача водили децу на ту екскурзију.

Даље у раду ћемо се позивати на описани начин функционисања реалног система и на основу њега креирати ентитете, везе, дефинисати кардиналност веза и одредити ограничења.

5.1. Идентификација кандидата за ентитете

Након дефинисања граница система који ће се посматрати и за који ће се пројектовати информациони систем је идентификација кандидата за ентитете. Идентификација кандидата за ентитете се врши разговором са запосленима тј. методом одозго надоле.

Ентитет је елеменат који можемо једнозначно одредити и на тај начин га издвојити у скупу. Објекти модела који су једнозначно одређени називају се јединствено ентитети. Ентитет се графички представља правоугаоником у коме се уписује назив типа ентитета у једнини као на слици 11.



Слика 11. Графичко представљање ентитета

Разликују се у општем случају самостални и описни ентитети. Самостални ентитет је од интереса сам за себе, његова егзистенција у бази не зависи од другог ентитета. Описни ентитет описује друге ентитете и одговара атрибутима тих ентитета чији домен није прост, па је и егзистенција у бази зависна од тог ентитета.

Дефинисањем граница система долазимо до обеката који ће бити од интереса за овај семинарски рад, а ови објекти представљају истовремено и кандидате за ентитете: ДЕТЕ, ГРУПА, ВАСПИТАЧ, КАРТОН, ЕКСКУРЗИЈА, ПРЕДСТАВА, ВАКЦИНАЦИОНИ КАРТОН, КАРТОН АЛЕРГИЈЕ, ПОСЕБНЕ ПОТРЕБЕ. У даљем раду ће бити дефинисано који од ових, наведених ентитета спадају у самосталне, а који у описне (зависне) ентитете.

5.2. Идентификација веза

Ентитети који су елементи истог скупа, као и ентитети из различитих скупова могу међусобно стајати у различитим односима, тј. повезани различитим везама. Карактеристика веза између два или више ентитета је поседовање одређених односа односно својстава. Свака веза може бити одређена једнозначно и према томе може се сматрати ентитетом што ће се у неким случајевима користити у раду. Графички се веза приказује ромбом у коме се уписује назив типа веза као на слици 12.



Слика 12. Графичко представљање концепта везе

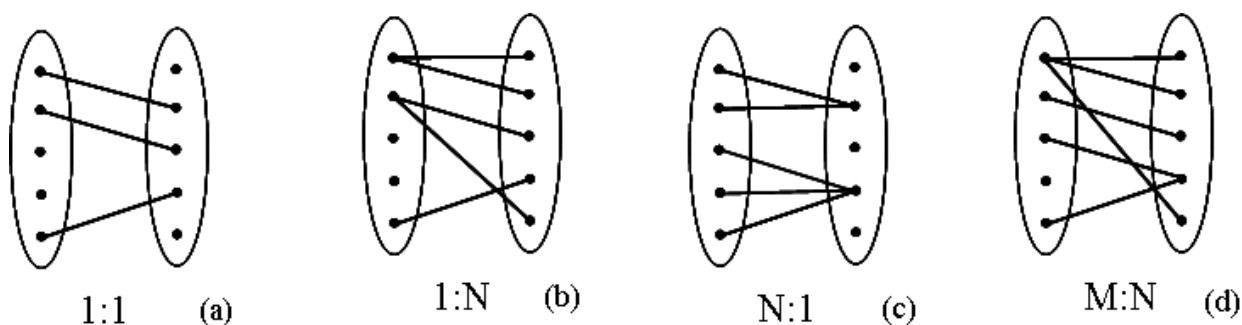
Представљање везе ентитетом, добија се мешовита класа ентитет - веза, уводи се да би се решио управо проблем представљања везе између веза. Врста везе се тада представља као класа ентитета што омогућава да се може успоставити веза између релација. Графички се оваква класа представља на следећи начин, слика 13.



Слика 13. Мешовита класа

У овом случају се скуп ентитета и њихових веза третира као један ентитет на вишем нивоу апстракције. Ентитет у овом случају има улогу и везе и ентитета и назива се мешовити тип (класа) ентитет - веза.

Степен релације, кардиналност, односно степен везе између два ентитета је веома важна особина везе. Постоје три различита степена веза које могу постојати између два ентитета слика 15.



Слика 14. Графички приказ кардиналности везе

1:1 веза (један према један), означава да се један ентитет једног типа придружује једном ентитету другог типа.

1:N веза (један - према - више), означава да се један ентитет првог типа придружује већем броју ентитета другог типа, а један ентитет другог типа само једном ентитету првог типа.

M:N веза (више - према - више), означава да се једном ентитету једног типа придружује више ентитета другог типа и обрнуто.

Ентитети могу бити и у вези сами са собом и таква веза се назива унарна. Ретко, али ипак могуће, јављају се тројне и вишеструке везе, које повезују више ентитета. Најчешће се вишеструке везе преводе у бинарне увођењем новог ентитета уместо посматране везе.

Веза може бити тотална (обавезна, мандаторна) и парцијална (опциона). Један тип ентитета има тоталну везу, ако сваки ентитет тога типа учествује у бар једној вези, иначе, веза је парцијална. Тотална веза се графички означава тачком (звездицом) на темену везе према објекту који има тоталну везу.

Поред до сада дефинисаних типова веза и то чисте везе дефинисане називом, затим мешовите, постоје и специјалне везе којима се дефинише:

- Егзистенцијална зависност где је веза физичка (пример везе ентитета РАДНИК и ДЕТЕ. Ова веза се означава са Е).
- Идентификациона зависност где је кључ једног ентитета део кључа слабог ентитета.

- Идентификационо - егзистенцијална зависност.

У оквиру дефинисања специјалних веза дефинише се и појам слабог ентитета који се графички представља двоструким правоугаоником. Класа слабог ентитета је врста ентитета који је на неки начин завијан од другог ентитета. То су ентитети који за једнозначно дефинисање тј. кључ захтевају и атрибуте (примарне кључеве) других ентитета са којима су у вези. Други извор слабих ентитета су ентитети настали заменом вишеструких веза бинарним, јер обично тако формиран ентитет захтева атрибуте других ентитета са којима је у вези за свој примарни кључ.

Зависност подтипа и зависност претхођења су још два типа везе између ентитета. Генерализациона зависност или зависност подтипа је зависност између ентитета подтипа и њему надређеног ентитета. Ова веза (зависност) се означава са С или “ис” или “ис а” и графички се приказује троуглом.

У овом раду се јављају ентитети: ВАКЦИНАЦИОНИ КАРТОН, КАРТОН АЛЕРГИЈЕ, ПОСЕБНЕ ПОТРЕБЕ који су слаби ентитети јер су зависни од ентитета КАРТОН.

Идентификација веза и дефинисање кардиналности за “Предшколска установа”

Анализирајући пословна правила, тј. односе између објеката које ћемо разматрати у раду долазимо до дефинисања веза између ентитета и одговарајућих кардиналности која важе за те везе.

Прво ћемо поћи од ентитета ДЕТЕ и његове везе са ентитетом ГРУПА и дефинисати везу УПИСАНО У као на слици 16. Једно дете може бити уписано у једну и само једну групу, одакле следи да је кардиналност везе са стране ентитета ДЕТЕ (1,1), док у једној групи може бити уписано више деце, али мора постојати бар једно дете да би група била оформљена одакле следи да је кардиналност везе УПИСАНО У са стране ентитета ДЕТЕ (1,М).



Слика 15. Приказ везе УПИСАН У

За једну групу надлежна је једна или више васпитачица. У вртицу може постојати васпитач који није распоређен, тј. не води ни једну групу или може бити надлежан за само једну групу. Веза НАДЛЕЖАН и кардиналност ове везе између ентитета ГРУПА и ВАСПИТАЧ су графички приказане на слици испод.



Слика 16. Приказ везе НАДЛЕЖАН

За свако дете уписано у вртић се отвара картон у првих месец дана боравка у вртићу. У картону се воде разне ствари везане за дете, о којима ће касније детаљније бити речи. У овом тренутку ће бти дефинисана веза између ентитета ДЕТЕ и КАРТОН. Ова веза се зове ИМА и њена кардиналност са стране ентитета ДЕТЕ је (0,1) јер дете након првих месец дана мора имати један и само један картон. Такође, један картон се односи на једно и само једно дете одакле следи да је кардиналност везе ИМА са стране ентитета КАРТОН (1,1), што је приказано на слици број 17.



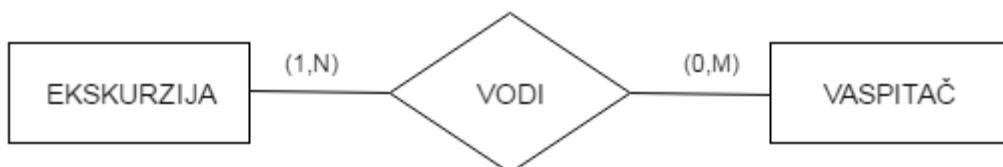
Слика 17. Приказ везе ИМА

Објекат (ентитет) из реалног света који смо узели у разматрање је и ЕКСКУРЗИЈА. Да би се ишло на неку екскурзију мора се пријавити бар једно дете, а може их ићи и више. Такође, не мора свако дете ићи на екскурзију, а једно дете се може пријавити да иде и на више екскурзија. Тако да, између ентитета ЕКСКУРЗИЈА и ДЕТЕ је креирана веза ИДЕ НА са кардиналношћу (0,N) са стране ентитета ДЕТЕ и (1,M) са стране ентитета ЕКСКУРЗИЈА.



Слика 18. Приказ везе ИДЕ НА

На екскурзију децу води бар један васпитач, а најчешће иде више васпитача. Неки васпитач не води децу ни на једну екскурзију, а може водити децу и на више различитих екскурзија. Из оваквог пословног правила втрића закључујемо да су ентитети ЕКСКУРЗИЈА и ВАСПИТАЧ повезани везом коју називамо ВОДИ, док је кардиналност те везе са стране ентитета ЕКСКУРЗИЈА (1,M), а са стране ентитета ВАСПИТАЧ (0,M) као што је приказано на слици испод.



Слика 19. Приказ везе ВОДИ

Повремено се у предшколској установи организују представе, као што је нпр. Новогодишња представа. Тада свака група изводи једну представу. Децу за извођење представе уче васпитачи (један или више). Један васпитач може учити и више група да изводе одређену представу, а не мора учити ниједну групу. Веза између ентитета ГРУПА и ентитета ПРЕДСТАВА је ИЗВОДИ, кардиналност везе са стране ентитета ГРУПА је (1,1) (једна група организује само једну представу), а са стране ентитетаа ПРЕДСТАВА је (0,1) (постоји представа коју нико не организује у том тренутку или је организује само једна група). Веза између ентитета ПРЕДСТАВА и ВАСПИТАЧ је ОРГАНИЗУЈЕ, кардиналност ове везе са стране ентитета ПРЕДСТАВА је (1,M), а са стране ентитета ВАСПИТАЧ је (0,N). Везе ИЗВОДИ и ОРГАНИЗУЈЕ са својим кардиналностима су приказане на сликама испод.

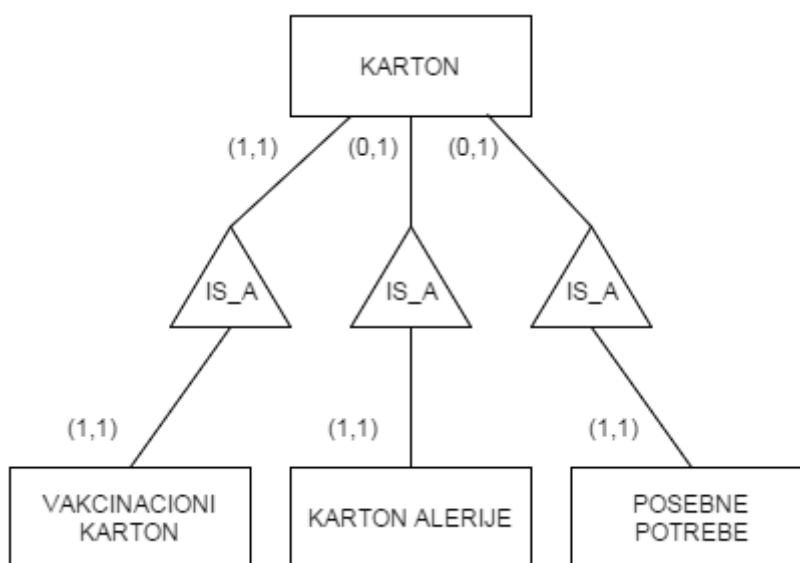


Слика 20. Приказ везе ИЗВОДИ



Слика 21. Приказ везе ОРГАНИЗУЈЕ

Ентитети ВАКЦИНАЦИОНИ КАРТОН, КАРТОН АЛЕРИЈЕ, ПОСЕБНЕ ПОТРЕБЕ су зависни ентитети од надређеног ентитета КАРТОН. Ова врста везе малопре је објашњена и означено је са IS_A. Проблем са алергијом и посебне потребе има само неко дете, тако да је кардиналност везе између надређеног ентитета КАРТОН и подређених ентитета КАРТОН АЛЕРГИЈЕ и ПОСЕБНЕ ПОТРЕБЕ са стране надређеног ентитета (0,1). Правило предшколске установе (Вртића) је да свако дете мора имати вакцинациони картон, одакле следи да је кардиналност везе између ентитета ВАКЦИНАЦИОНИ КАРТОН и КАРТОН са стране надређеног ентитета (1,1).



Слика 22. Приказ везе КАРТОН и њему подређених ентитета

За свако дете се води свакодневно дневник. У њему васпитач уноси да ли је дете тог дана доручковало, узинало, ручало и одморило. Тако да ће између ентитета ДЕТЕ и ВАСПИТАЧ постојати веза ВОДИ ДНЕВНИК која је са стране ентитета ВАСПИТАЧ кардиналности (0,N), а са стране ентитета ДЕТЕ кардиналности (1,M). Васпитач води дневник за више деце, али ако није надлежан ни за једну групу не води ни за једно дете. За једно дете обавезно води дневник бар један васпитач, а може водити и њих више (нпр. васпитач који ради прву смену и васпитач који ради другу шмену).



Слика 23. Приказ слике ВОДИ ДНЕВНИК

5.3. Дефинисање атрибута

Сваки објекат реалног света је дефинисан особинама, па по тој аналогiji и ентитети имају атрибуте којима се описују карактеристична својства. Заједно, атрибут и вредност атрибута јесу опис одређене особине ентитета.

Интервијуом са корисницима се долази до објеката који ће се разматрати, ђ. дефинисање ентитета се спроводи поступком одозго надолe. Док се атрибути одређују из докумената који се прикупљају у посматраном предузећу, тј. поступком одоздо нагоре.

Основна правила која се користе при дефинисању атрибута су:

- Сваки ентитет има произвољан број атрибута, што значи да нема ограничења у броју атрибута.
- Одређени атрибут припада једном и само једном ентитету, тако да исти атрибут не може бити описан у оквиру два или више ентитета.
- Свако појављивање ентитета има вредности за све атрибуте тог ентитета.
- Атрибут одређеног појављивања ентитета може имати само једну вредност, па примерак ентитета за одређени атрибут има једну вредност.
- Сваки атрибут представља једну одређену чињеницу, тако да и свако значење вредности атрибута мора имати једно доследно значење.

5.4. Дефинисање кључева

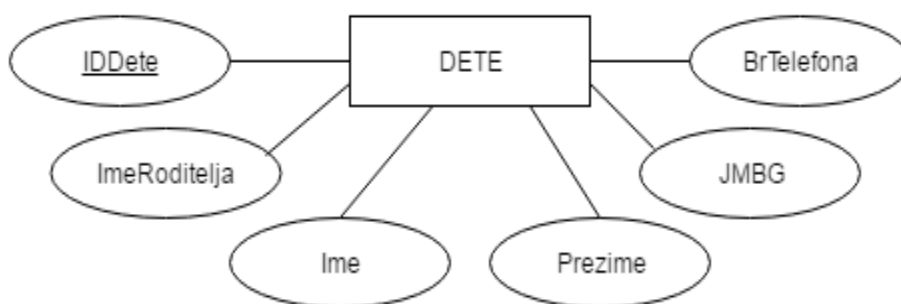
Кључ је атрибут или комбинација атрибута који једнозначно одређују ентитет из скупа. Обично је то један (прост или елементаран кључ) или комбинација два или више атрибута (сложен кључ). Избор кључа је битан за пројектовање целе базе података јер мора једнозначно да идентификује чињенице сачуване у бази података и једино помоћу њега се сигурно може приступити жељеном ентитету.

У једном ентитету може постојати више комбинација атрибута који задовољавају дефиницију кључа. На пример ентитет ДЕТЕ се може једнозначно одредити са

стрибутом ЈМБГ, али ће се ипак као кључ за овај ентитет увести атрибут “ИДДете” како би се избегао проблем уколико неко дете нема ЈМБГ (страни држављанин).

Идентификација атрибута и кључева за пример Предшколска установа

Поћићемо од ентитета ДЕТЕ. За свако дете је познато његово име, презиме, ЈМБГ, име једног од родитеља и његов број телефона. Тако да ове наведене ствари ће бити атрибути ентитета ДЕТЕ. Поред ових атрибута (као што је већ споменуто) уводи се и атрибут ИД_Дете који ће бити примаран кључ овог ентитета. Графички, примарни кључ се од осталих атрибута разликује по томе што је подвучен, као на слици испод.



Слика 24. Графички приказ ентитета ДЕТЕ са атрибутима

Ентитет ГРУПА ће имати следеће атрибуте ИДГрупа , БрСобе и БрМеста. ИДГрупа је атрибут који је уведен и који ће бити примарни кључ ентитета ГРУПА.

Ентитет ВАСПИТАЧ ће имати атрибуте ИД-Васпитач, ИмеВаспитач, ПрезимеВаспитач, ЈМБГВаспитач. Као и код ентитета ДЕТЕ уведен је атрибут ИДВаспитач који ће имати улогу примарног кључа.

Ентитет ЕКСКУРЗИЈА ће имати атрибуте ИДЕкскурзија и Место.

Ентитет ПРЕДСТАВА ће имати атрибуте ИДПредстава и Назив.

Ентитет КАРТОН ће имати атрибут ИДКартон, ДатумОтварања.

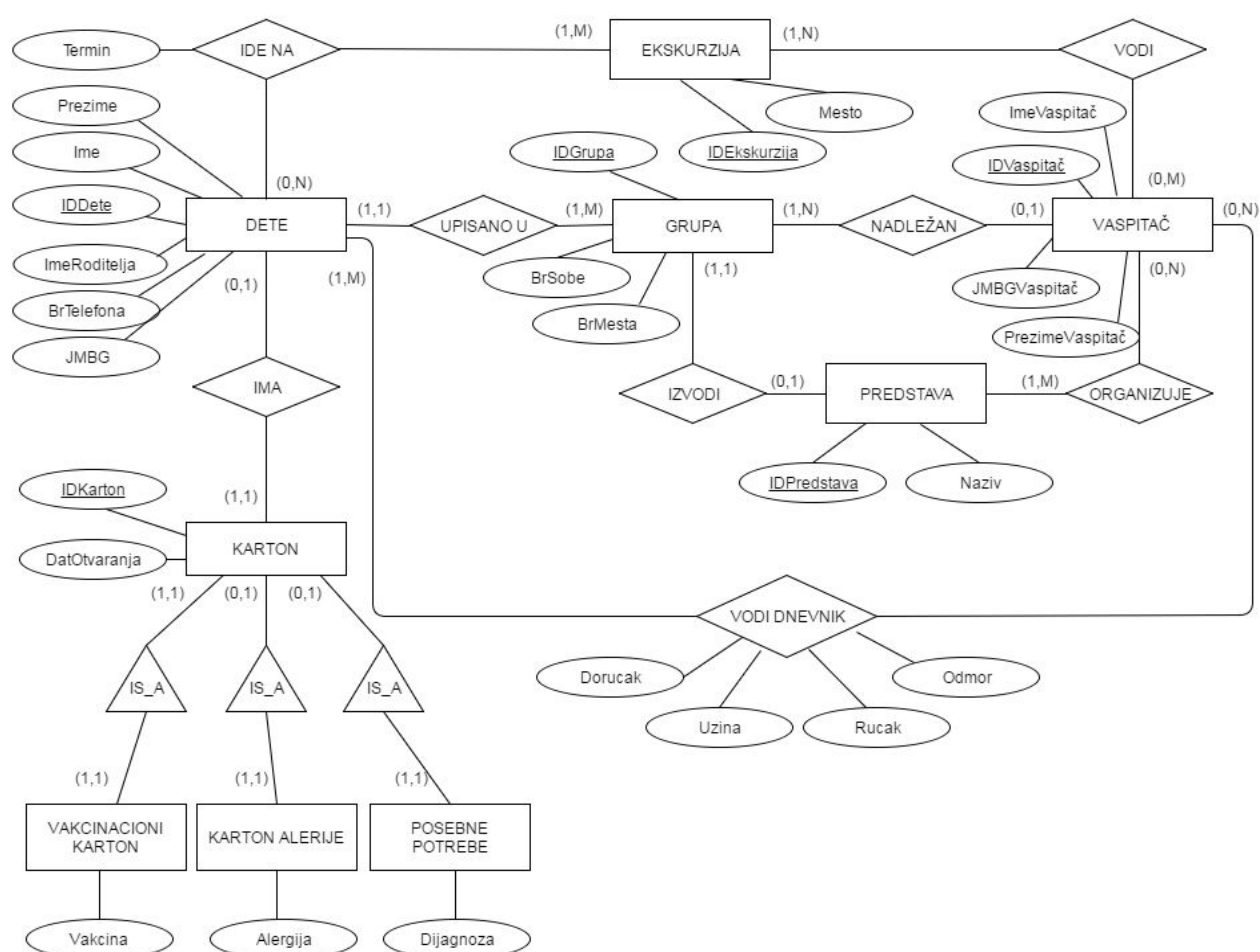
Ентитет ВакцинациониКартом ће имати атрибут Вакцина.

Ентитет КартонАлергије ће имати атрибут Алергија.

Ентитет ПосебнеПотребе ће имати атрибут Дијагноза.

5.5. ER dijagram modela baze podataka

У претходним поглављима су идентификовани ентитети и креиране везе између ентитета. На основу пословних правила су дефинисане кардиналности између веза, а анализом докумената су дефинисани атрибути за ентитете. Тако да ЕР дијаграм модела базе података за Вртић има изглед као на слици испод.



Слика 25. ЕР модел базе података за Предшколску установу

У даљем тексту ће се описати релациони модел базе података и правила за преводјење ЕР модела у релациони модел. На основу тих правила ће се извршити преводјење ЕР модел базе података у релациони модел базе података за пример предшколске установе.

5.6. Релациони модел – модел друге генерације

Релациони модел за управљење базама података је једна врста модела податка, који је базиран на рачуну и теорији скупова. Данас је најраспрострањенији модел за моделирање реалних проблема и динамичко управљање подацима.

Основне карактеристике релационог модела су:

- једноставна и природна репрезентација (модел је представљен скупом табела, а табеле су структура најприхватљивија за комуникацију са корисницима);
- могућа је формално-математичка интерпретација модела (табела се може дефинисати као математичка релација, а затим искористити све предности ове формалне дефиниције);
- богат математички аларат служи као основа за развој базе података.

5.6.1. Структура релационог модела

Концепти који карактерише структуру Релационог модела су атрибути и домен. Атрибут је било шта што описује ентите, а домен је скуп дозвољених вредности неког типа.

Релација се представља у облику дводимензионалне табеле у којој н атрибута одговара колонама табеле, док шема релације се прецизира име релације, листа њених атрибута и њихови домени.

ИД Васитач	Име Васпитач	Презиме Васпитач	ЈМБГ Васпитач
15	Бојана	Пејић	1231231231222
7	Милена	Марковић	1111111111111
11	Ирена	Јарић	2222222222222
10	Иван	Марић	2333312345675

Слика 26. Пример шеме релације Васпитач

Степен релације/кардиналност је степен веза између два ентитета. Н – торка је један ред у табели која представља релацију, која је дефинисана не само доменом већ и именованим индексом примарни кључ.

Свака релација мора имати примарни кључ. X је кључ релације P , ако испуњава следеће услове:

- X функционално одређује све атрибуте релације P
- ни један прави подскуп од X нема ту особину

Страни кључ је атрибут који у датој релацији није примарни кључ, али је примарни кључ у некој другој релацији. Страни кључ једне релације не мора да буде кандидат за кључ те исте релације.

У даљем раду ће се дефинисати домени атрибута који се појављују код ентитета анализираног ЕР дијаграма.

Домени појединих атрибута за ентитет ДЕТЕ:

Име – скуп коначних низова алфаветских знакова

Презиме – скуп коначних низова алфаветских знакова

ИмеРодитеља – скуп коначних низова алфаветских знакова

ИДДете – јединствена петоцифрена шифра

ЈМБГ – тринаестоцифрени број који може почети са 0

БрТелефона – скуп бројева

Домени појединих атрибута за ентитет ГРУПА:

ИДГрупа – скуп ненегативних јединствених бројева

БрСобе – скуп двоцифрених бројева

БрМеста – скуп двоцифрених бројева

Домени појединих атрибута за ентитет ВАСПИТАЧ:

ИДВаспитач – скуп ненегативних јединствених бројева

ИмеВаспита – скуп коначних низова алфаветских знакова

ПрезимеВаспитач – скуп коначних низова алфаветских знакова

ЈМБГВаспитач – тринаестоцифрени број који може почети са 0

Домени појединих атрибута за ентитет ЕКСКУРЗИЈА:

ИДЕкскурзија – скуп ненегативних јединствених бројева

Место – скуп коначних низова алфаветских знакова

Домени поједих атрибута за ентитет ПРЕДСТАВА :

ИДПредстава – скуп ненегативних јединствених бројева

Назив – скуп коначних низова алфаветских знакова

Домени појединих атрибута за ентитет КАРТОН:

ИДКартон – скуп троцифрених бројева

ДатОтварања – датум

Домени поједих атрибута за ентитет ВАКЦИНАЦИОНИ КАРТОН:

Вакцина – скуп алфаветских знакова

Домени појединих атрибута за ентитет АЛЕРГИЈА:

Алергија – скуп алфаветских знакова

Домени поједих атрибута за ентитет ПОСЕБНЕ ПОТРЕБЕ:

Дијагноза – дијагноза по МКБ10 шифарнику дијагноза

Атрибути омогућавају да се релације приказу у виду табела. Да би табела била релација, она мора да задовољи следеће услове:

- нема дуплираних врста (зато што је Р скуп)
- распоред врста је небитан
- распоред колона је небитан
- све вредности у табели су атомичне

Уколико релација Р задовољава дате услове, она се може приказати у облику Прве нормалне форме (1НФ).

5.6.2. Превођење ЕР модела у Релациони модел

ЕР модел података креиран је у почетној фази дизајна базе података и није га могуће директно имплементирати коришћењем релационог система за управљање базама података. У фази дизајна базе података ЕР модел података потребно је превести у

релациони модел податка, који је могуће директно имплементирати у фази имплементације базе података. Превођење једног модела податка у други не значи да сваком концепту структуре једног модела мора одговарати концепт структуре другог модела, да свакој операцији једног модела да мора одговарати операција другог модела и сваком експлицитном ограничењу једног модела мора одговарати експлицитно ограничење другог модела. Модели објекти - везе је семантички богатији модел, модел треће генерације, од релационог модела који представља модел друге генерације.

Превођење ЕР модела у релациони модел података изводи се на следећи начин:

- Део структуре ЕР дијаграма се представља релационом шемом
- Ограничења, операције и део структуре ЕР модела се представља операцијама за очување интегритета дефинисања над релационим моделом, а имплементацију коришћењем средстава одабраног релационог СУБП.

Поступак превођења

1.Правило за објекте (ентитете):

1.1. Сваки објекат (ентитет) из ЕР дијаграма постаје шема релације. Назив типа ентитета постаје назив шеме релације. Обележја (атрибути) објеката(ентитете) постају обележја шеме релације. За основне објекте идентификатор објекта постаје примарни кључ шеме релације.

1.2. Сваки слаб објекат постаје шема релације. Назив типа објекта постаје назив шеме релације. Обележја објекта су обележја шеме релације. Примарни кључ надређеног ентитета постаје једно од обележја шеме релације, која одговара слабом типу ентитету.

1.3. Објекат надтипа постаје шема релације. Назив надтипа постаје назив шеме релације. Обележја надтипа постају обележја шеме релације. Идентификатор надтипа постаје кључ шеме релације.

1.4. Објекат подтипа постаје шема релације. Назив подтипа постаје назив шеме релације. Обележја подтипа постају обележја шеме релације. Идентификатор надтипа постаје кључ шеме релације

2. Правила за превођење бинарне веза

2.1 Везе кардиналности 1:1 по правилу немају обележја. Обележја која би могл бити приписана самој вези, у суштини су обележја једног од објеката који учествују у тој вези.

2.1.1 Веза кардиналности $(1,1):(1,1)$ и оба објекта која у њој учествују преводимо у једну шему релације, чија су обележја сва обележја једног и другог објекта. Кандидат за кључ у овој шеми релације су идентификатори једног и другог објекта који су у вези.

2.1.2 Везе са кардиналношћу $(0,1) : (1,1)$ и објекте у вези преводимо у две шеме релације. За сваки објекат у вези по једна шема релације, с тим што се идентификатор једног од објеката који су у вези убади за обележје друге шеме релације. Веза се представља спољњим кључем.

2.1.3 Везе са кардиналношћу $(0,1) : (0,1)$ креирају се у три шеме. Обележја у шеми релације која одговарају вези су и идентификатори објекта који су у вези и оба су кандидати за кључ.

2.2 Везе кардиналитета 1 : N

2.2.1. Веза кардиналитета $(1,1) : (0,M)$ и $(1,1) : (1,M)$ не постаје посебна шема релације. Идентификатор објекта са стране за коју је горња граница кардиналитета пресликавања M постаје обележје шеме релације која одговара објекта са чије стране максимални кардиналитет 1.

2.2.2. Веза између надређеног и слабог објекта као и веза између подтипа и надтипа не постоји посебне шеме релације. Оне су већ остварене правилима 1.2 ,1.3 и 1.4

2.2.3. Везе кардиналитета $(0,1) : (0,M)$ и $(0,1) : (1,M)$ постаје посебна шема релације. Обележја ове шеме релације су идентификатори обележја који су у вези док кључ шеме релације је идентификатор обележја са чије стране максимални кардиналитет је 1.

2.3. Везе кардиналитета N:M

2.3.1. Везе кардиналитета $(0,M) : (0,M)$, $(1,M) : (0,M)$ и $(1,M) : (1,M)$ постаје посебна шема релације. Обележја ове шеме релације су идентификатори објекта који су у вези, а кључ шеме релације је сложен кључ који се састоји од идентификатора објеката који су у вези.

2.4. Герунд или мешовити тип објекта - везе се посматра на исти начин као и одговарајућа веза. Уколико веза постаје обележје, она постају обележја шеме релације

веза када се веза преводи у посебну шему релације или се укључи у шему релације у коју се уписује спољњи кључ.

2.5. Правила за унарне везе

Унарном везом се сматра веза између два објекта истог типа. Код унарних веза типа (1:1) парцијалност само на једној страни везе, односно тоталност само на једној страни везе, не би имало смисла. При преводјењу унарних веза с обзиром да би спољни кључ у шеми релације имао исти назив као и примарни кључ, потребно је његово преименовање.

Преводјење ЕР модела Предшколска установа у Релациони модел

На основу правила 1.1 сви типови ентитета постају шеме релације. Име типа ентитета постаје име шеме релације. Обележја типа објекта постају обележја шеме релације. За основне објекте идентификатор објекта (ентитета) постаје примарни кључ шеме релације.

ДЕТЕ (ИДДетета, Име, Презиме, ЈМБГ, ИмеРодитеља, БрТелефона)

ГРУПА (ИДГрупа, БрСобе, БрМеста)

ВАСПИТАЧ (ИДВаспитач, ИмеВаспитач, ПрезимеВаспитач, ЈМБГВаспитач)

ЕКСКУРЗИЈА (ИДЕкскурзија, Место)

ПРЕДСТАВА(ИДПредстава, Назив)

КАРТОН (ИДКартон, ДатОтварања)

Према правилу 1.2 сваки слаб објекат постаје шема релације. У нашем случају је то ентитет КАРТОН који постаје шема релације.

КАРТОН (ИДКартон, ДатОтварања, ИДДете)

Према правилу 2.1.1 веза између ентитета КАРТОН и ВАКЦИНАЦИОНИ КАРТОН и наведене ентитете преводимо у једну шему релације и то КАРТОН коме додајемо једини постојећи атрибут из ентитета ВАКЦИНАЦИОН КАРТОН - Вакцина.

КАРТОН (ИДКартон, ДатОтварања, Вакцина)

Према правилу 1.3 и 1.4 објекат надтипа постаје шема релације. Назив надтипа постаје назив шеме релације. Обележја надтипа постају обележја шеме релације. Идентификатор надтипа постаје кључ шеме релације. Док објекат подтипа постају

шема релације. Назив подтипа постаје назив шеме релације. Обележја подтипа постају обележја шеме релације. Идентификатор надтипа постаје кључ шеме релације

ПОСЕБНЕ ПОТРЕБЕ (ИДКартон, Дијагноза)

КАРТОН АЛЕРГИЈЕ (ИДКартон, Алергија)

Превођење веза

У зависности од кардиналности, на типове веза се примењују различита правила за превођење.

На слици број 15, представљена је веза ентитета ДЕТЕ и ГРУПА и означена са „УПИСАНО У“, кардиналности (1,1) : (1,М), која по правилу 2.2.1 не постаје посебна шема релације. Идентификатор објекта ГРУПА овом приликом постаје атрибут шеме ДЕТЕ.

ДЕТЕ (ИДДетета, Име, Презиме, ЈМБГ, ИмеРодитеља, БрТелефона, ИДГрупа)

ГРУПА (ИДГрупа, БрСобе, БрМеста)

На слици број 16, представљена је веза ентитета ГРУПЕ и ВАСПИТАЧ и означена са „НАДЛЕЖАН“, кардиналности (1,Н) : (0,1) која по правилу 2.2.3 постаје посебна шема релације. Обележја ове шеме релације су идентификатори објекта који су у вези, а кључ шеме релације је идентификатор објекта за које је ГГ=1.

НАДЛЕЖАН (ИДВаспитач, ИДГрупа)

ВАСПИТАЧ (ИДВаспитач, ИмеВаситач, ПрезимеВаспитач, ЈМБГВаспитач)

ГРУПА (ИДГрупа, БрСобе, БрМеста)

Веза „ИМА“

На слици број 17, предсављена је веза између ентитета КАРТОН И ДЕТЕ и означена са „ИМА“, кардиналности (1,1) и (0,1) коју по правилу 2.1.2 преводимо у две шеме релације, с тим што се идентификатор једног од објеката који су у вези убаци за обележје друге шеме релације. Код ентитета КАРТОН је убачено обележје ентитета ДЕТЕ (ИДДете), као што је приказано у шемама испод.

КАРТОН (ИДКартон, ДатОтварања, ИДДете, Вакцина)

ДЕТЕ (ИДДетета, Име, Презиме, ЈМБГ, ИмеРодитеља, БрТелефона, ИДГрупа)

На слици број 18, представљена је веза између ентитета ДЕТЕ и ЕКСКУРЗИЈА и означена везом „ИДЕ НА“, кардиналности $(0,N) : (1,M)$ која по правилу 2.3.1 постаје посебна шема релације. Обележја ове шеме релације су идентификатори објекта који су у вези, а кључ шеме релације је сложени кључ. Ова веза има и атрибут Термин.

ИДЕ НА (ИДЕкскурзија, ИДДете, Термин)

ДЕТЕ (ИДДетета, Име, Презиме, ЈМБГ, ИмеРодитеља, БрТелефона, ИДГрупа)

ЕКСКУРЗИЈА (ИДЕкскурзија, Место)

На слици број 19, представљена је веза између ентитета ЕКСКУРЗИЈА и ВАСПИТАЧ и означена везом „ВОДИ“, кардиналности $(1,N) : (0,M)$ која по правилу 2.3.1 постаје посебна шема релације. Обележја ове шеме релације су идентификатори објекта који су у вези, а кључ шеме релације је сложен кључ.

ВОДИ (ИДЕкскурзија, ИДВаспитач)

ЕКСКУРЗИЈА (ИДЕкскурзија, Место)

ВАСПИТАЧ (ИДВаспитач, ИмеВаситач, ПрезимеВаспитач, ЈМБГВаспитач)

На слици број 20, представљена је веза између ентитета ГРУПА и ПРЕДСТАВА и означена везом „ИЗВОДИ“, кардиналности $(1,1) : (0,1)$ коју по правилу 2.1.2 преводимо у две шеме релације. За сваки објекат у вези по једна шема релације, с тим што се идентификатор једног од објеката који су у вези убаци за обележја друге шеме релације.

ГРУПА (ИДГрупа, БрСобе, БрМеста, ИДПредстава)

ПРЕДСТАВА (ИДПредстава, Назив)

На слици број 21, представљена је веза између ентитета ПРЕДСТАВА и ВАСПИТАЧ и означена са „ОРГАНИЗУЈЕ“, кардиналности $(1,M) : (0,N)$ која по правилу 2.3.1 постаје посебна шема релације. Обележја ове шеме релације су идентификатори објекта који су у вези, а кључ шеме релације је сложен кључ који се састоји од идентификатора објекта који су у вези.

ОРГАНИЗУЈЕ(ИДПредстава, ИДВаспитач)

ПРЕДСТАВА (ИДПредстава, Назив)

ВАСПИТАЧ (ИДВаспитач, ИмеВаситач, ПрезимеВаспитач, ЈМБГВаспитач)

На слици број 23, према правилу 2.3.1 представљена је веза између ентитета ДЕТЕ и ВАСПИТАЧ и означена са ВОДИ ДНЕВНИК, кардиналности (1,М) : (0,Н), из којег произилазе три релације. Веза ВОДИ ДНЕВНИК је дефинисана и својим атрибутима Доруча, Ручак, Ужина и Одмор.

ВОДИ ДНЕВНИК (ИДДете, ИДВаспитач, Доручак, Ужина, Ручак, Одмор)

ДЕТЕ (ИДДетета, Име, Презиме, ЈМБГ, ИмеРодитеља, БрТелефона, ИДГрупа)

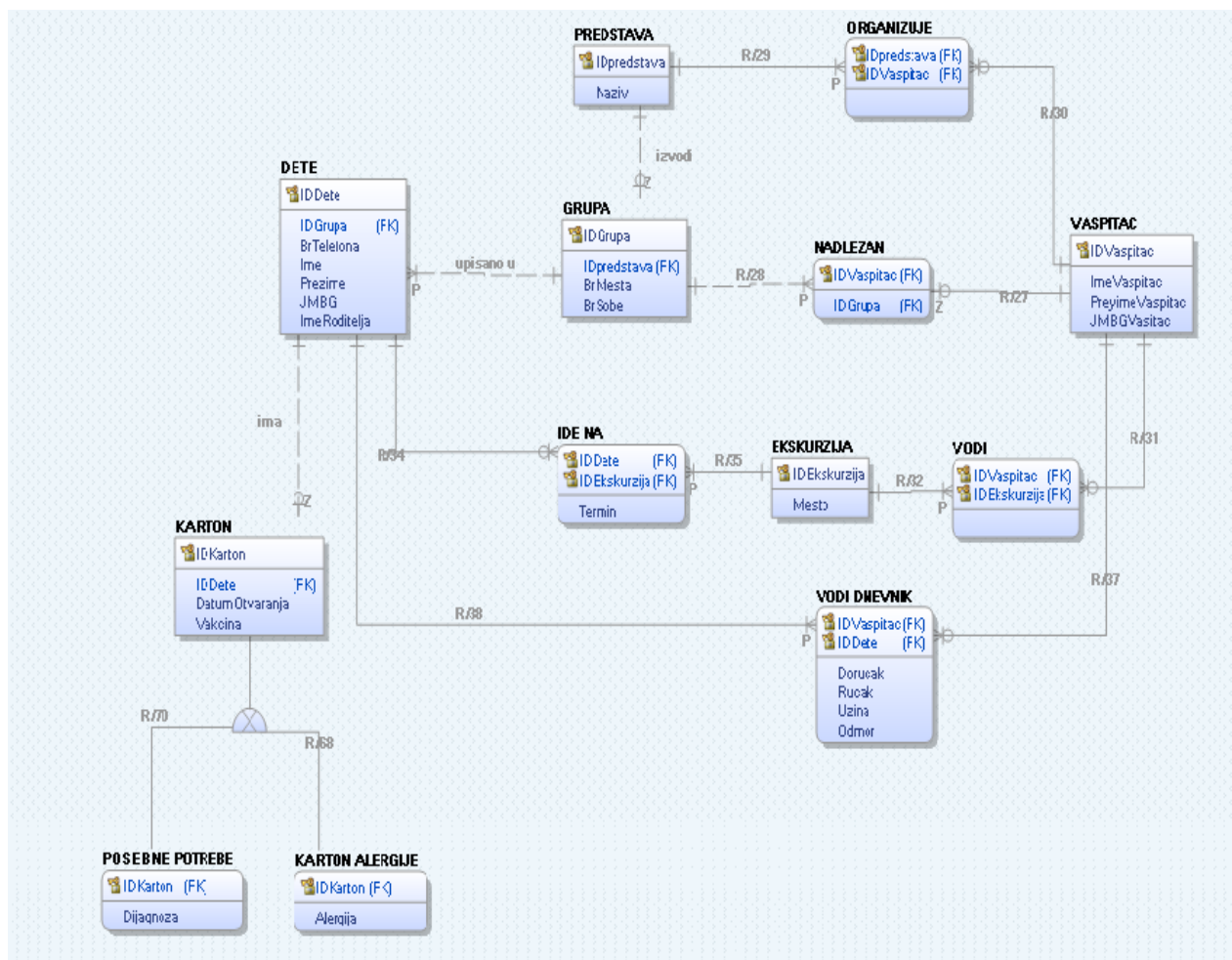
ВАСПИТАЧ (ИДВаспитач, ИмеВаситач, ПрезимеВаспитач, ЈМБГВаспитач)

6. Приказ логичког модела применом алата Erwin Data Modeler

Један од алата за моделовање података је Erwin Data Modeler, који служи за прављење и одржавање база и складишта података (Data Warehouse, Data Mart) и пословних модела података. Могуће је прављење одвојених логичких и физичких, али и јединственог логичко - физичког модела података. Основне карактеристике Erwin - а су:

- брзо прављење база података на основу направљеног модела. Може се увести синхронизација између базе и модела, чиме се олакшава процес одржавања. Подржане су следеће базе: Oracle, DB2/UDB (укључујући iSeries), SQL Server, Teradata, Sybase, Informix, Ingres, Progress, Access...
- олакшан тимски рад на моделу и његова реупотребљивост,
- механизми за детаљно поређење базе и модела,
- реверзни инжењеринг и
- дефинисање и одржавање конвенције о именима, атрибутима и типовима података.

У овом раду је креиран логички модел за Предшколску установу који је приказан на слици 27. При цртању модела коришћена је ИЕ (Information Engineering) нотација.



Слика 27. Логички модел Предшолске установе

7. Апликативно моделирање и фаза имплементације

База података која је коришћена за израду базе „предшколска_установа“ је MySQL. Ово је релациона, бесплатна и квалитетна база података која је погодна за веб програмирање. Физички модел је имплементиран кроз MySQL Workbench апликацију која представља графичко решење за рад са MySQL базом података. Скрипта која се извршава омогућава креирање базе података под именом ‘предшколска_установа’, креирање табела и њихових веза на основу већ дефинисаног ЕР модела.

У наставку је дат код у коме су креиране табеле: дете, дневник, група, надлежан и васпитач. Због ограниченог обима рада у наставку је развијен један модул апликације који се односи на дневне активности детета у вртићу. Тачније, евиденцију деце, група и васпитача и вођење евиденције свакодневних активности деце у дневнику од стране васпитача.

```
CREATE SCHEMA IF NOT EXISTS `predskolska_ustanova` DEFAULT CHARACTER SET latin1 COLLATE
latin1_general_ci ;
USE `predskolska_ustanova` ;

-----
-- Table `predskolska_ustanova`.`dete`
-----
CREATE TABLE IF NOT EXISTS `predskolska_ustanova`.`dete` (
  `id` INT(11) NOT NULL AUTO_INCREMENT,
  `ime` VARCHAR(45) NULL DEFAULT NULL,
  `prezime` VARCHAR(45) NULL DEFAULT NULL,
  `ime_roditelja` VARCHAR(45) NULL DEFAULT NULL,
  `broj_telefona` VARCHAR(45) NULL DEFAULT NULL,
  `jmbg` VARCHAR(45) NULL DEFAULT NULL,
  `grupa_id` INT(11) NULL DEFAULT NULL,
  PRIMARY KEY (`id`))
ENGINE = InnoDB
AUTO_INCREMENT = 25
DEFAULT CHARACTER SET = utf8
COLLATE = utf8_unicode_ci;

-----
-- Table `predskolska_ustanova`.`dnevnik`
-----
CREATE TABLE IF NOT EXISTS `predskolska_ustanova`.`dnevnik` (
  `id` INT(11) NOT NULL AUTO_INCREMENT,
  `aktivnost` VARCHAR(255) NULL DEFAULT NULL,
  `opis` VARCHAR(255) NULL DEFAULT NULL,
  `dete_id` INT(11) NULL DEFAULT NULL,
  `vaspitac_id` INT(11) NULL DEFAULT NULL,
  `datum` DATE NOT NULL,
  PRIMARY KEY (`id`),
ENGINE = MyISAM
AUTO_INCREMENT = 24
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8_unicode_ci;

-----
-- Table `predskolska_ustanova`.`grupa`
-----
CREATE TABLE IF NOT EXISTS `predskolska_ustanova`.`grupa` (
  `id` INT(11) NOT NULL AUTO_INCREMENT,
  `br_mesta` VARCHAR(255) NULL DEFAULT NULL,
```

```

    `br_sobe` VARCHAR(255) NULL DEFAULT NULL,
    PRIMARY KEY (`id`))
ENGINE = MyISAM
AUTO_INCREMENT = 9
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8_unicode_ci;

-----
-- Table `predskolska_ustanova`.`nadlezan`
-----

CREATE TABLE IF NOT EXISTS `predskolska_ustanova`.`nadlezan` (
  `vaspitac_id` INT(11) NOT NULL,
  `grupa_id` INT(11) NOT NULL,
ENGINE = MyISAM
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8_unicode_ci;

-----
-- Table `predskolska_ustanova`.`vaspitac`
-----

CREATE TABLE IF NOT EXISTS `predskolska_ustanova`.`vaspitac` (
  `id` INT(11) NOT NULL AUTO_INCREMENT,
  `ime_vaspitac` VARCHAR(255) NULL DEFAULT NULL,
  `jmbg_vaspitac` VARCHAR(255) NULL DEFAULT NULL,
  `prezime_vaspitac` VARCHAR(255) NULL DEFAULT NULL,
  PRIMARY KEY (`id`))
ENGINE = MyISAM
AUTO_INCREMENT = 16
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8_unicode_ci;

ALTER TABLE predskolska_ustanova.dete
ADD CONSTRAINT FK_DeteGroupId
FOREIGN KEY (grupa_id) REFERENCES predskolska_ustanova.grupa(id);

ALTER TABLE predskolska_ustanova.nadlezan
ADD CONSTRAINT FK_NadlezanGrupaId
FOREIGN KEY (grupa_id) REFERENCES predskolska_ustanova.grupa(id);

ALTER TABLE predskolska_ustanova.nadlezan
ADD CONSTRAINT FK_NadlezanVaspitacId
FOREIGN KEY (vaspitac_id) REFERENCES predskolska_ustanova.vaspitac(id);

ALTER TABLE predskolska_ustanova.dnevnik
ADD CONSTRAINT FK_DnevnikVaspitacId
FOREIGN KEY (vaspitac_id) REFERENCES predskolska_ustanova.vaspitac(id);

ALTER TABLE predskolska_ustanova.dnevnik
ADD CONSTRAINT FK_DnevnikDeteId
FOREIGN KEY (dete_id) REFERENCES predskolska_ustanova.dete(id);

```

На основу претходно генерисане шеме базе података предшколска_установа и конкретних захтева будућих корисника врши се израда апликације. Поступак израде апликације поред избора технологије која ће се користити, подразумева дефинисање менија, дефинисање изгледа форме, дефинисање упута и дефинисање извештаја.

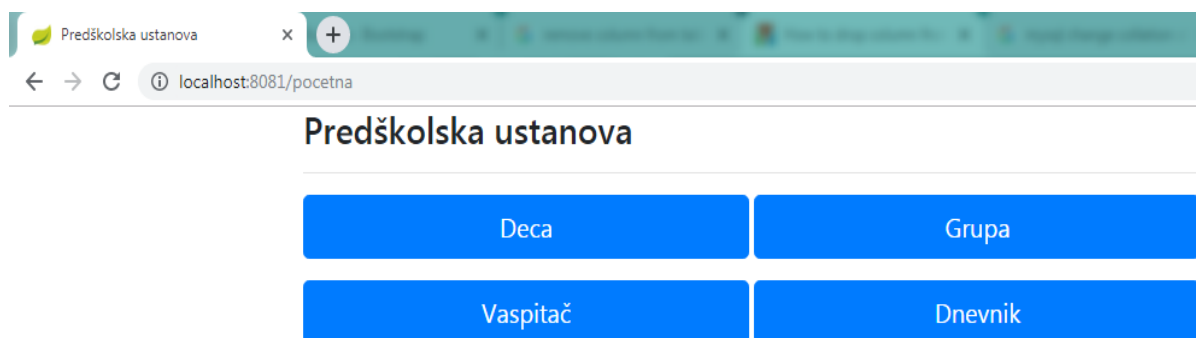
За израду апликације коришћен је Јава програмски језик, као један од најактуелнијих програмски језика данашњице за израду веб апликација. Веб апликација је софтвер који се налази на веб серверу и корисник приступа апликацији помоћу веб прегледача (енгл. *web browser*), преко мреже, најчешће интернета или интранета. Основе предности

веб апликација су доступност било где у било ком тренутку, лака прилагодљивост на различитим уређајима, лака инсталација и одржавање, централизовани приступ.

Оквир (енгл. *framework*) који је коришћен за лакши развој апликације је Spring Boot. Тај оквир нам омогућава лакшу конфигурацију и креирање инфраструктуре за самосталне апликације које су спремне за продукцију. Овај оквир има већ уграђен веб сервер (у овом раду Tomcat server), па је апликацију довољно покренути као Јава апликацију. Апликација за предшколску установу након покретања Јава апликације биће доступна на платформи где је покренута покретањем веб прегледача на адреси <http://localhost:8081/>.

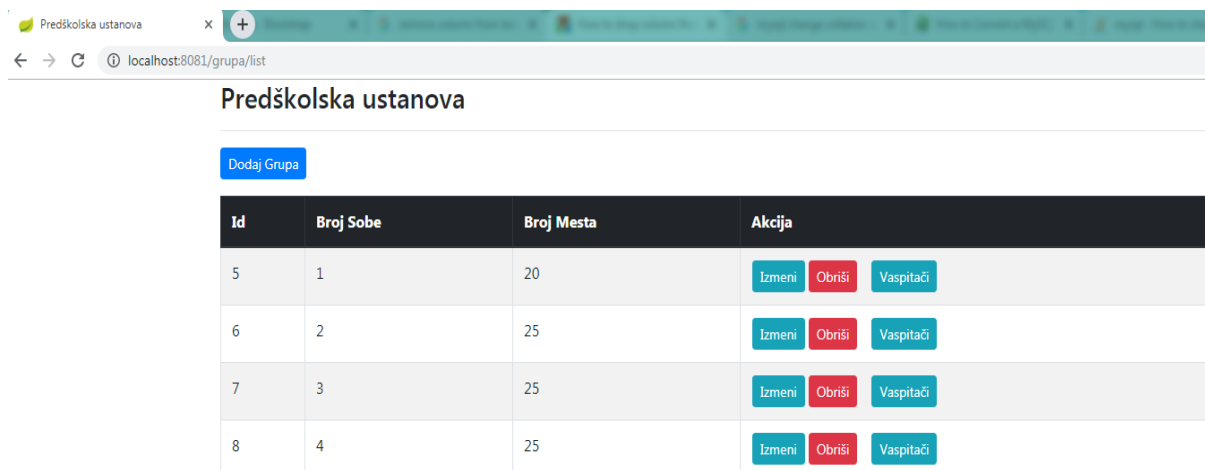
У даљем тексту ће бити приказани екрани веб апликација “Педшколска установа” која је ограничена на унос података о свакодневним активности деце у групи предшколске установе и на извештавање родитеља о тим активностима.

Почетни екран апликације је приказан на слици испод.



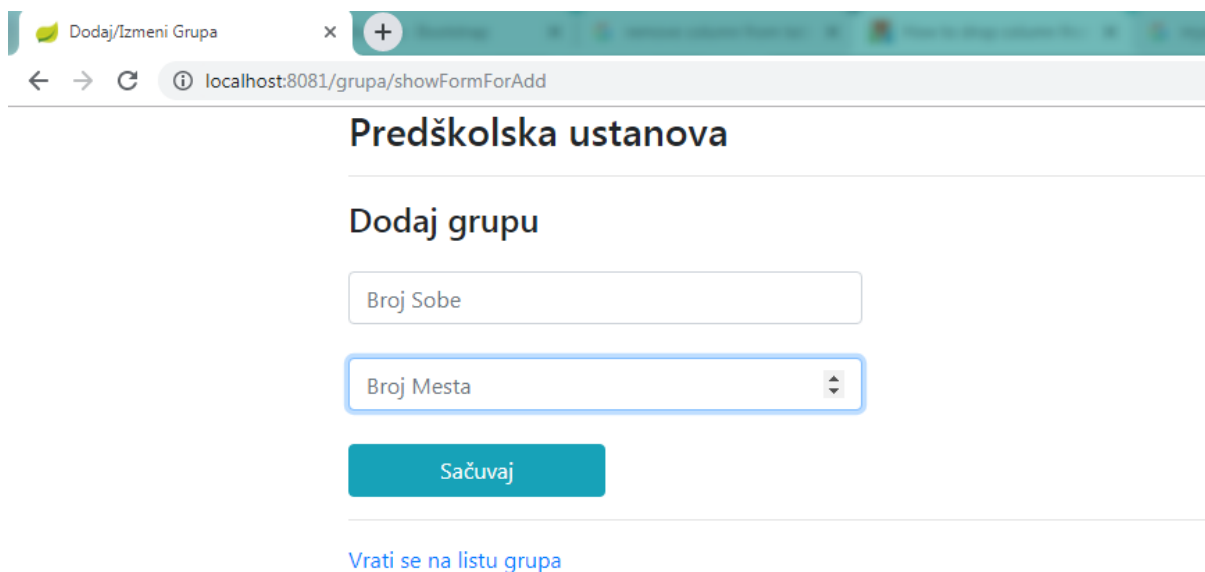
Слика 28. Почетни екран веб апликације

Одабиром форме Група приступа се екрану у коме су приказане већ евидентиране групе у бази података са својим атрибутима, тј. са бројем собе у којој се налазе и број места који је предвиђен за њу. Податке у групама је могуће ажурирати или обрисати. Такође, из овог екрана се приступа екранима Додај групу и Васпитач.



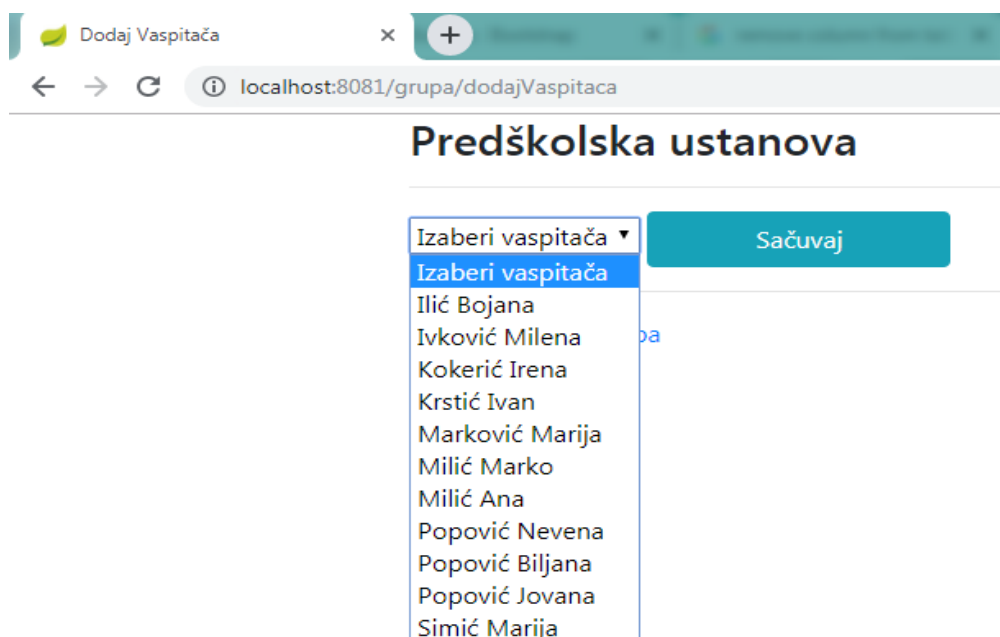
Слика 29. Екан група

Екран додај групу је приказан на слици испод и у овом екрану се дефинише број собе и број места креиране групе.



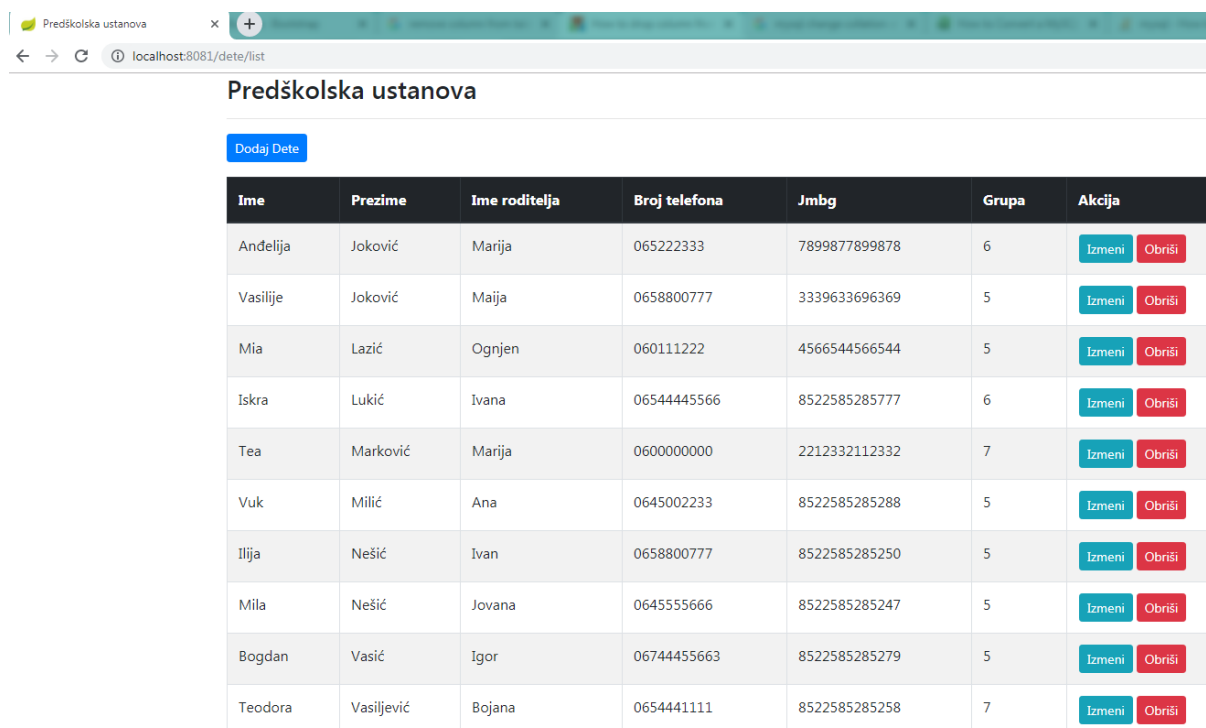
Слика 30. Екран за унос нове групе

Из екрана Група може се приступити екрану где се дефинише веза између ентитета група и васпитач, тј. где се дефинише који васпитачи су надлежни за одабрану групу. Описан екран је приказан на слици испод.



Слика 31. Екран за избор васпитача у групи

Са почетног екрана се приступа екрану у коме се врши евидентирање деце са њиховим подацима. Уносе се подаци који су на самом почетку израде информационог система одабрани за атрибуте, а то су: Име, Презиме, Име родитеља, Број телефона, ЈМБГ и Група којој дете припада. Податке је могуће изменити или обрисати дете из листе.



Слика 32. Екан за додавање и преглед унете деце

Predškolska ustanova

Dodaj dete

Ime

Prezime

Ime Roditelja

Broj Telefona

JMBG

Izaberi grupu ▼

Sačuvaj

[Vrati se na listu dece](#)

Слика 33. Екран за унос новог детета у апликацију

Поред групе и деце у апликацији се евидентирају васпитачи и битни подаци о њима. Листа васпитача и начин њиховог евидентирања су приказани на слици испод.

Predškolska ustanova

[Dodaj Vaspitač](#)

Id	Ime Vaspitača	Prezime Vaspitača	JMBG Vaspitača	Akcija
15	Bojana	Ilić	1231231231231	Izmeni Obriši
7	Milena	Ivković	3333333333333	Izmeni Obriši
11	Irena	Kokerić	7777777777777	Izmeni Obriši
10	Ivan	Krstić	6666666666666	Izmeni Obriši
5	Marija	Marković	1111111111111	Izmeni Obriši
9	Marko	Milić	5555555555555	Izmeni Obriši
12	Ana	Milić	7777777777777	Izmeni Obriši
6	Nevena	Popović	2222222222222	Izmeni Obriši
8	Biljana	Popović	4444444444444	Izmeni Obriši
14	Jovana	Popović	1234567891234	Izmeni Obriši

Слика 34. Екран за преглед, унос и ажурирање васпитача

Екрани који приказују уношење активности за дете од стране васпитача, претрагу по детету и датуму и звештај о активностима за тражено дете су приказани на сликама испод. Екран за приказ активности деце у предшколској установи је намењен родитељима, док су остали екрани предвиђени за запослене у предшколској установи.

Predškolska ustanova

Dodaj aktivnost

Izaberi dete ▼

Izaberi vaspitaca ▼

mm/dd/yyyy

Izaberi aktivnost ▼

Opis

Sačuvaj

[Vrati se na listu dnevnik](#)

Слика 35. Екран за евидентирање дневних активност деце

Predškolska ustanova

Pretraga aktivnosti

Joković Anđelija ▼

09/15/2019

Nadji

[Vrati se na listu dnevnik](#)

Слика 36. Екран за преглед дневних активности за конкретно дете и датум

Rezultati pretrage:

Dete	Vaspitac	Datum	Aktivnost	Opis
Joković Anđelija	Ilić Bojana	2019-09-15	Rucak	Neobavljen
Joković Anđelija	Ilić Bojana	2019-09-15	Odmor	Neobavljen
Joković Anđelija	Ilić Bojana	2019-09-15	Uzina	Potpuno obavljen
Joković Anđelija	Ilić Bojana	2019-09-15	Dorucak	Neobavljen

Слика 37. Резултат претраге за одабрано дете и датум

Фаза имплементације

Након успешно извршеног апликативног моделирања следи фаза имплементације. Имплементација подразумева увођење, тестирање и одржавање система.

У овој фази се врши коначно вредновање софтвера и ситне преправке у софтверу. Произвођач софтвера креира корисничка упитства и прави план обуке корисника. Обука корисника се прво врши на тестној бази, а након едуковања кадра прелази се на продукциону базу и рад са реалним подацима. Ако је број будућих корисника велики и ако се ради о кадру који претходно није радио у неком информационом систему фаза имплементације је процес који може доста времена одузети. Често може доћи до отпора корисника и проблема са синхронизацијом у обављању пословних процеса у новом пословном информационом систему. Истрајним приступом се долази до комплетне примене пословног информационог система, а након неког времена примене долази се до великог и свеобухватног унапређења пословања који је и крајњи циљ приликом одлуке о информатизацији пословања.

Након свих тестирања, обуке и преласка са тестне на продукциону базу података неопходно је да се дефинише уговор о одржавању система између продавца софтвера и наручиоца. Најчеште су информациони системи, као и реални системи, живи, склони променама и неопходно их је дорађивати и мењати. Некада је потребно и додавање нових функционалности или измене услед промена закона или надређених установа.

8. Закључак

У овом раду је разматран реални систем “Предшколска установа”. Полазећи од мисије, циљева и информационих потреба родитеља извршено је Функционално и Информационо моделирање једног реалног система. Креирана је база у My SQL -у и урађена је веб апликација која задовољава потребе за дневном евиденцијом активности деце у предшколској установи.

Исправно извршено пројектовање информационог система и његова имплементација штеди време и ресурсе, вишеструко побољшава пословање кроз обавезну прецизнос пословних процеса и омогућава извођење корисних закључака у сврху унапређења пословања.

Оно што се желело постићи овом веб апликацијом јесте једноставан начин за свакодневну евиденцију дневних активности деце и једноставан извештај унетих података. Као што је већ речено, апликација приказује дневне активности деце тј. да ли је дете доручковало, ужило, ручало и одморило. Ове информације су нешто што се свакодневно већ евидентира у папирном облику, чиме информације нису биле доступне онима којима су најпотребније тј. родитељима. Све информације васпитачи свакодневно преноси родитељима, за свако дете у групи, како би родитељима дали смернице о будућим активностима детета, и слободно могу рећи троше време на њихо усмено преношење. Овај вид преношења података је превазиђен у садашњем друштву, не само због уштеде времена већ и због поузданости података јер је најтачније приступити податку који је унет у тренутку радње. Такође, свакодневним евидентирањем, подаци постају лако доступни за обраду, извештавање и доношење корисних закључака о пословању.

9. Литература

- [1] James L. Johnson, *Database: Models, Languages, Design*, Oxford University Press, 1997., London.
- [2] Michael Kifer, A. Bernstein, P.M. Lewis, *Database, Systems*, Pearson, Addison Wesley, 2004.
- [3] S. Abiteboul, R. Hull, V. Vianu, *Fundation of Databases*, Addison Wesley, Boston, MA
- [4] Branislav Lazarević, Z. Marjanović, N. Aničić, S. Babarogić, *Baze podataka*, FON, Beograd, 2003.
- [5] Vladimir Blagojević, *Relacione baze podataka*, Klub Nikola Tesla, Beograd, 2001.
- [6] Jeffrey A. Hoffer, M.B. Prescott, F.R. McFadden, *Modern Database Management*, Pearson, Prentice Hall, 2005.
- [7] Prof. dr Mladen Veinović Igor Franc Aleksandar Jevremović, *Baze podataka*, Fakultet za poslovnu informatiku, Beograd 2006.
- [8] Nevena Joksić, *Primena OLAP tehnika u analizi otplate duga klijenata Banke Poštanske štedionice a. d.*, Magistarski rad, Univerzitet u Beogradu, Matematički fakultet, Beograd 2010.
- [9] B. Thalheim, *Fundamentals of ER Modeling*, Springer Verlag, Berlin
- [10] Craig S. Mullins, *Administracija baza podataka*, Kompjuter biblioteka, 2003.
- [11] Pavlović-Lazetić Gordana, *Uvod u relacione baze podataka*, drugo izdanje, Matematički fakultet, 1999.
- [12] MicroStrategy, *The Case for Relational OLAP*, 1995,
http://www.cs.bgu.ac.il/~dbm031/dw032/Papers/microstrategy_211.pdf
- [13] Star schemas, <http://publib.boulder.ibm.com/infocenter/rbhelp/v6r3/index.jsp?topic=/com.ibm.redbrick.doc6.3/wag/wag32.htm>
- [14] DB2 Cube Views A Primer, IBM Redbooks, SG24-7582-00 December 2008,
<http://www.redbooks.ibm.com/abstracts/sg247002.html>
- [15] DB2 Cube Views: Getting Started with Business Objects, REDP-3711-00 September 2003,
<http://www.redbooks.ibm.com/abstracts/redp3711.html>
- [16] Thomsen Erik, *OLAP Solutions: Building Multidimensional Information Systems*, Second Edition, John Wiley & Sons, 2002
- [17] Control Center overview, http://publib.boulder.ibm.com/infocenter/db2luw/v8/index.jsp?topic=/com.ibm.db2.udb.cc.doc/db2_udb/intro.htm
- [18] OLAP Market Share, <http://www.1keydata.com/datawarehousing/olap-market-share.html>
- [19] The FASMI test, <http://www.olapreport.com/fasmi.htm>
- [20] Business Intelligence: OLAP Tool Selection, <http://www.1keydata.com/datawarehousing/toololap.html>
- [21] DB2 for OS/390 V5 and Federated Systems,
<http://nlazovic.bravehost.com/Xephon/docs/DB2%20for%20OS390%20V5%2>
- [22] Star & snowflake shemas, <http://publib.boulder.ibm.com/infocenter/ablxhelp/8.3/index.jsp?topic=/com.ibm.db2.ax.cub.doc/sii-cubescema-32488.htm>
- [23] OLTP vs. OLAP, <http://datawarehouse4u.info/OLTP-vs-OLAP.html>