Факултет инжењерских наука

Универзитета у Крагујевцу

Назив студијског програма: Машинско инжењерство
Ниво студија: Мастер академске студије
Модул: Индустријски инжењеринг
Предмет: Пројектовање информационих система и базе података
Број индекса: 377/2012

Урош М. Пејчиновић

Складиштење података и агенти за претраживање

Мастер рад

Комисија за преглед и одбрану:

1.Проф. др Милан Ерић

2._____ Датум одбране: _____

3._____ Оцена: _____

У оквиру овог мастер рада кандидат треба да:

проучи и презентира развој складишта података (*Data Warehouse*), односно проблематику складиштења података као и истраживање истих у оквиру анализе пословних података за подршку одлучивању и управљању перформансама пословања. Рад треба да обухвати уводна разматрања, основна својства, структуру и компоненте складишта података, развој складишта података, истраживање података и агенте за претраживање, као и студијски пример са димензионалном анализом. На крају рада дати закључна разматрања.

Препоручена литература:

1. Chris Todman: **Designing a Data Werehouse**, Hewlett-Packard Company, Prentice-Hall International (UK), London 2007.
2. Claude S.: **Data Mining with Macrosoft SQL Server 2000-Tehnical Reference**, Microsoft Press, 2001.
3. Лазаревић Б., и др.: **Базе података**, ФОН Београд, Београд 2003.
4. Полишчук Ј.: **Пројектовање информационих система**, Електротехнички факултет, Подгорица, 2007.
5. Rainer K., Turban E.: **Introduction to Information Systems**, John Wiley & Sons, Inc., 2009
6. Ангелина Његуш, Алемпије Вељовић: **Основе релационих х аналитичких база података**, Универзитет Мегатренд, Београд, 2004.

Крагујевац, 28.05.2014

Име и презиме ментора

_____

Изјава о самосталној изради рада

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

Број индекса Име и презиме

_____

Резиме рада: Данашњи пословни информациони системи не могу да се замисле без уграђене пословне интелигенције. Софтверска индустрија се фокусира на пословну интелигенцију и алалитику услед потребе организација за доношењем одлука на свим нивоима менаџмента. Уводи се интелигенција у све пословне процесе, пружајући праву информацију, правим људима у право време.

Кључне речи: Систем, складиште података, информација, складиштење, агенти за претраживање, интелигенти агенти, процеси.

Summary of work: Today's business information systems can not be imagined without the built-in business intelligence. The software industry is focusing on business intelligence due to the needs of organizations for decision making at all levels of management. Introduces intelligence into all business processes, providing the right information to the right people at the right time.

Keywords: System, warehouse, information, storage, searching agents, intelligent agents, processes.

Садржај

1. Уводна разматрања

Данас више него икад, менаџерима су потребни лако доступни и конзистентни подаци представљени тако да у исто време, прецизно и сажето дају приказ организације у целини као и њеног окружења. Међутим, сложени услови пословања генеришу сваким даном све већи број пословних догађаја у оквиру предузећа и изван њега, а добијени подаци најчешће су смештени у оперативним базама података. Због величине таквих база није их могуће претраживати у реалном времену, а кад се и добије коначан одговор на питање, обично су то извештаји у дводимензионалном облику на великом броју страница и представљају селектовано преписивање података из базе.

Будући да је правовремено добијање квалитетних информација битно за остварење предности пред конкуренцијом, менаџер их мора добити што пре и у облику прилагођеном његовим потребама. Из тога произилази да се од данашњих информационих система предузећа, очекује да осигурају информације чији садржај, брзина приступа и начин приказа одговарају тренутним потребама менаџера у процесу одлучивања. Док се за потребе оперативног вођења пословања користе класичне базе података, засноване на релационом моделу, које одржавају ажурно, стварно стање пословног система, а одређеним се подацима након ажурирања губи траг, за доношење правилних пословних одлука потребно је имати увид у временски ток дешавања пословних догађаја, па такве базе података не представљају задовољавајуће решење. Ради тога се пришло креирању нових облика организовања података у рачунарским меморијама информационих система. Развијена је нова генерација рачунарских система која се темељи на концепту складиштења података. Складиште података садржи податке прикупљене из различитих извора, историјске, о пословању предузећа као и податке из спољног окружења, а дизајнирано је тако да омогућава претраживање података, *online* аналитичку обраду, извештавање и подржавање процеса доношења одлука.

Нова генерација рачунарских система се састоји од два дела, оперативног (трансакционог) и складишта података (аналитичког), чиме се постиже издвајање процеса за генерисање информација који се по својој природи разликују од оперативних процеса.[1]

2. Складишта података

Складиште података (енгл. *Data Warehouse*) је основ пословне интелигенције. Представља систем који екстрактује, пречишћава, стандардизује и учитава податке из постојећих продукционих система предузећа у димензионе шеме и подржава, па имплементира корисничке алате за постављање питања над подацима којима је циљ да омогуће крајњим корисницима (који по струци не морају бити информатичари) лако и ефикасно сналажење и претраживање великих количина података. [2]

Складиште података скуп је података организације на којем се темељи систем основе одлучивања. Из сврхе произлази да складиште података треба подацима потпуно "покрити" једно или више пословних подручја (нпр. набавке, продаје), да подаци у складишту треба да буду свеобухватни, тј. "интегрисани" од унутрашњих података организације, али и података из њеног окружења.

Подаци морају обухватити дужи временски период (пет, десет или више година), јер су временске анализе пословно врло значајне. [1]

Субјективна усмереност - подаци се организују око предмета, на начин да дају информације о тачно одређеним предметима у оквиру функционалних подручја уместо о текућим операцијама предузећа.

Интегрисаност - подаци се скупљају у бази података из различитих извора и смештају увек у истом формату, конзистентни су и приказују се на доследан начин.

Везаност уз време - сви подаци у складишту података везани су и идентификују се уз одређени временски период, што значи да имају историјски карактер. За разлику од њих, у оперативним базама података складиштени су само актуелни, најсвежији подаци.

Садржајна непромењивост - подаци у складишту су стабилни и када се једном учитају у складиште по правилу се не мењају. Тиме се омогућује да менаџмент, или неко други ко користи складиште података може бити сигуран да ће добити једнак одговор, независно од времена, или учесталости постављања питања.

Мање складиште података које обухвата податке само једног пословног подручја назива се подручним складиштем података (енгл. *Data mart*).

Складиште података намењено је менаџерима, али и свима који у свом послу обављају различите аналитичке задатке, као што су послови праћења и извештавања, који се темеље на примени различитих пословних правила, а обављају се постављањем усмерених питања и анализом добијених резултата, послови анализе и дијагнозе, који се темеље на умешности, а обављају се итеративним проналажењем и анализом добијених информација, и послови планирања и симулације, који се темеље на знању, а обављају се моделирањем и извршењем израђеног модела.

Складиште података настало је као систем намењен подршци пословном одлучивању (*DDS – Decision Support System*). У основи овог концепта лежи идеја о постојању засебне базе података (упоредно с продукцијском базом података) чије су основне карактеристике следеће:

- Модел података је врло једноставан, интуитиван, па самим тим, лако разумљив крајњем кориснику.
- Модел података је прилагођен искључиво читању података.
- Складиште података се у договореним временским интервалима пуни подацима прикупљеним у продукцијском систему (аутоматизовани масовни унос података из једног или више продукционих система)

При изградњи складишта података се сусреће са специфичним проблемима. Већина потешкоћа везана је за изградњу система за извлачење података односно уз периодички аутоматизовани пренос података из изворних продукционих система у одређено складиште података.

Током последњих двадесетак година документоване су се многобројне методологије изградње складишта података, али све почивају на два најпознатија приступа: одозго-доле и одоздо-горе.

Настанак димензионалног моделирања узрокован је недостацима сложених, нормализованих релационих структура, изграђеним техникама ER моделирања. Димензионални модел, због своје једноставности узорковане контролном денормализацијом даје могућност једноставног и брзог добијања информације, иначе скривене у сложеним структурама продукцијског система базираног на нормализованом, релацијском моделу. Сваки димензионални модел састоји се од две врсте објеката: мере и димензије. Неке ситуације на које се наилази у пракси захтевају употребу посебних димензијских структура као што су: пахуљаста структура, мосна структура, итд.[4]

Неки од проблема на које се наилази су:

- Обједињавање података из више различитих извора реализованих на различитим платформама (неинтегрисани информациони систем).
- Освеживање складишта података тј. брзо откривање промена насталих у изворном систему у периоду између претходног и тренутног извлачења (екстракције) података.
- Итеративни карактер изградње складиште података.

Проблематика везана уз изградњу модела података је врло добро обрађена у литератури те не представља проблем. Но проблем представљају потешкоће везане уз екстракцију података, због чега поступак изградње екстракцијског система одузима између 70% и 90% укупног времена потребног за изградњу складишта података. Проблем изградње произлази из чињенице да корисник није у стању да довољно прецизно и јасно опише структуру и садржај информација које жели да користи при доношењу пословне одлуке. То је један од разлога због којег је препоручљиво аутоматизовати поступке везане уз екстракцију података(њену изградњу и њен рад) у што је могуће већој мери.

2.1.Циљеви складиштења података

Традиционалне релационе базе података, као и различити системи за управљање пословањем предузећа (енгл. *Enterprise Resource Planning*, скраћено *ERP*) представљају неопходан алат за пословање једног предузећа, али не и довољан. Релационе базе су погодне за масовно уношење података, јер су засноване на нормализованом моделу података у коме је отклоњена редундантност.

Међутим, за крајњег корисника је важно да подацима на одговарајући начин приступи, да их прочита и да их протумачи. Такође крајњи корисник жели брзо и ефикасно да дође до одговора на питања која га занимају, па чак и да „у лету" поставља питања на основу одговора које је добио од претходно постављених питања. Међутим, перформансе традиционалних релационих база су веома лоше када је извршавање сложених упита у питању. Такође, корисник нема могућности да на лак и брз начин динамички поставља упите већ мора да познаје сложене упитне језике што му знатно отежава и успорава рад. Отуда се појавила потреба за развијањем технологија које ће омогућити крајњем кориснику да на лак и брз начин претражује податке и долази до одговора који га занимају.

На енглеском се овакво претраживање обележава термином „*slice and dice*" који буквално значи сечење на кришке и коцкице, а односи се на растављање информације у мање делове да би се боље или лакше разумела.

Ово је уједно и један од основних циљева складиштења података који се сматра остваренима када су подаци крајњем кориснику разумљиви, читљиви, смислени; он подразумева да су информације представљене комплетно и систематично без обзира на то из ког су извора потекле, нпр. из релационих база, *excel* табела или магнетних трака *mainframe* рачунара.

Поред доступности информација, од складишта података се очекује конзистентност у представљању информација без обзира на то из ког су пословног процеса или тематског система потекле.[4]

2.2.Методологија изградње складишта података

Током протеклих двадесет година документоване су многобројне методологије изградње складишта података, али генерално посматрано све имају исход у два најпознатија приступа:

- одозго-доле (*top-down*) – старији приступ који је увео *B. Inmon*. Корпорацијско складиште података користи нормализовани ЕР модел и садржи најнижу потребну ширину података, као и метаподатке. Подаци се пуне из расположивих извора посредством централизованог средишњег складишта података (*DSA - Data Staging Area*). Како би се подацима у складишту података приступало, креирају се независни издвојени сегменти складишта података односно тематска складишта података која садрже углавном сумарне податке и то најчешће у форми звездасте шеме.

- одоздо-горе (*bottom-up*) – нешто новији приступ изградњи складишта података предложен од стране *R. Kimballa* као одговор на велике трошкове и дуго време потребно за развој складишта у складу одозго-доле архитектуром. Обично се користи димензијска звездаста шема ради побољшања перформанси и разумљивости модела од страна крајњих корисника. Издвојена складишта података ће садржавати податке и на основној, као и сумарне податке.

Приступ одоздо-горе даје брже резултате, али проблеми често настају када издвојена подручја складишта података треба да постану системски модули корпорацијског складишта података. Уколико се од почетка није водило рачуна о усклађеним метаподатцима, димензијама и сл. односно ако на почетку нису дефинисани услови циљане архитектуре складишта података, овакав приступ ће највероватније резултирати промашајем.

Зато се данас најчешће примењују различити хибридни (комбиновани) приступи, примера *middle-out* – код којих се димензијски модел израђује уважавајући потребе крајњих корисника (*одозго-доле*), али и реалност расположивости појединих података (*одоздо-горе*). Неопходни су и робусни метаподаци који ће касније увелике олакшати интеграцију појединих фаза и делова пројекта у циљано складиште података.

2.3.Структура система складишта података

Структура система складишта података се може грубо поделити (слика 1) на четири основна подручја:

- Изворни систем/системи (*Source systems, Legacy Systems*)
- Привремено подручје (*Data Storing Area*)
- База складишта података тј. подручје доступних података (*The Data Warehouse – Presentation Servers*)
- Подручје корисничког приступа подацима тј. корисничке апликације (*End User Data Access*)

Слика 1. Структура система складишта података[7]

Када говоримо о систему складишта података (складиште података у ширем смислу) тада говоримо о целокупном систему почевши од изворишних система до складишних корисничких апликација тј. свим елементима који су укључени у процесе стварања односно коришћења података смештених у самој бази складишта података. База складишта података (складиште података у ужем смислу) је подручје у којем су смештени подаци намењени крајњем кориснику складишта података.

База складишта података (складиште података у ужем смислу) је подручје у којем су смештени подаци намењени крајњем кориснику складишта података.

2.4.Изворни систем/системи

Изворни систем је подручје у којем се врши унос података који настају као резултат пословног процеса. Обично се назива продукционим информационим системом. У већини случајева у срцу продукционог система ће бити смештена релацијска база података. Главни приоритети оваквог система су брз и несметан унос података односно непрекидна доступност оперативи подузећа.

Због такве намене, постављање сложених SQL (*Structured Query Language*) питања тј. извештавање, постаје оптерећење и можемо га посматрати као додатни посао који продукциони систем мора обављати поред оног основног. Такође, може се рећи да изворни системи чувају малу количину података што произлази из основне намене продукционог система, а то није извештавање и анализа података.

У пракси изворни системи често ће бити остварени на различитим платформама те у различитим облицима. Најчешће ће то бити релацијска база података, што је уједно и најквалитетнији извор података. У другим случајевима, извор може бити систем којем се не може једноставно приступити. У тим случајевима подаци се претварају нпр. у ASCII датотеке с одређеном структуром које се уз коришћење за то предвиђених алата уносе у одредјену базу података (у овом случају у привремено подручје). Као пример можемо навести податке смештене у *Excel* Табелама или податке прикупљене од система за праћење телефонских позива (који могу бити припремљени као ASCII датотеке одређене структуре које обрађују записе о појединим телефонским позивима) итд.[6]

2.5.Привремено подручје

Привремено подручје је део система складишта података у којем се врши обрада пристиглих изворних података како би се припремило за њихов унос у базу складишта података. Ова обрада укључује поступке чишћења, измене, надоградње, обједињавања продукционих шифрарника те сваку другу обраду која ће прилагодити изворне податке структури складишта података које их очекује на крају процеса преноса података.

Што је већи број изворних система то ће обрада података унутар привременог подручја бити сложенија.

Привремено подручије није доступно кориснику, односно подаци које привремено подручје садржи нису намењени приказивању корисницима. Ово подручје доступно је искључиво људима који су укључени у изградњу складишта података тј. изградњу екстракциског система.

Платформа привременог подручја може бити релациона база података, али може бити и датотечни систем који очитава нпр. ASCII датотеке над којима се коришћењем одговарајућих алата врше поступци обраде.

2.6.База складишта података

База складишта података је одређена тачка прикупљених изворних података. Пре уласка у базу складишта података, изворни подаци су прошли кроз поступке чишћења и обједињавања унутар привременог подручја. У подручју базе података подаци постају доступни крајњим корисницима.

Када говоримо о подручју базе складишта података тада се заправо мисли на скуп појединих компоненти складишта података које заједно чине једно велико складиште података. Те компоненте називамо тематско складиште података (*Data Mart*). У тематском складишту података подаци се скупљају у димензионалном облику. Уколико је реч о релационој бази података димензионални облик ће значити коришћење звездастог модела података. Такође, могуће је да ће неко тематско складиште података бити нерелациона, мултидимензионална OLAP (*Online Analytical Processing*) база података. У том случају сама природа OLAP базе података намеће димензионалну структуру података.[4]

2.6.1. Тематско складиште података

Тематско складиште података је компонента великог складишта података док је складиште података унија ових компоненти. Као компонента складишта, тематско складиште података је по својој функционалности комплетно, те може постојати само за себе. Оно обично покрива један одређени део пословања компаније, а усмерен је ка одређеној групи корисника. У правилу би засебна подручна складишта података могла егзистирати на засебним рачунарима-послужитељима.

Свако тематско складиште података мора задовољити одређене критеријуме који му омогућују неометано уклапање међу остала подручна складишта података. Критеријуми су следећи:

- Свако тематско складиште података је изграђено на димензионалном моделу података.
- Унутар једног складишта података, свако тематско складиште података мора се темељити на скупу складних димензија и складних мера.

2.7. Улога складишта података

Главни циљ складишта података је ослободити информације које су „закључане у оперативним базама података и „помешати“ их са са информацијама из осталих, по правилу екстерних извора података.

Да би складиште података могло испунити циљ и сврху свог постојања морају пре свега бити испуњени следећи предуслови:

- складиште мора садржати велику количину детаљних података, што значи да све пословне трансакције битне за доношење пословних одлука, које су настале у процесима предузећа морају бити евидентиране у складиште података.
- ажурирање новим подацима мора бити континуиран процес, по могућности треба да се одвија у стварном времену, практично одмах након што се неки пословни догађај одиграо или одмах по завршетку неког процеса.
- мора увек бити расположиво и обликовано на начин да може послужити свакој сврси коју није увек могуће унапред предвидети
- треба предвидети могућност издвајања и међусобног повезивања података у смислу добијања свих мера и показатеља пословања у предузећу
- подаци у складишту који се прикупљају из различитих извора, чисте се уз осигурање квалитета и само такви су доступни корисницима.
- мора бити проширио да би могло следити стратегију проширења пословања компаније.[1]

2.8. OLAP и OLTP (Аналитичке и трансакционе базе података)

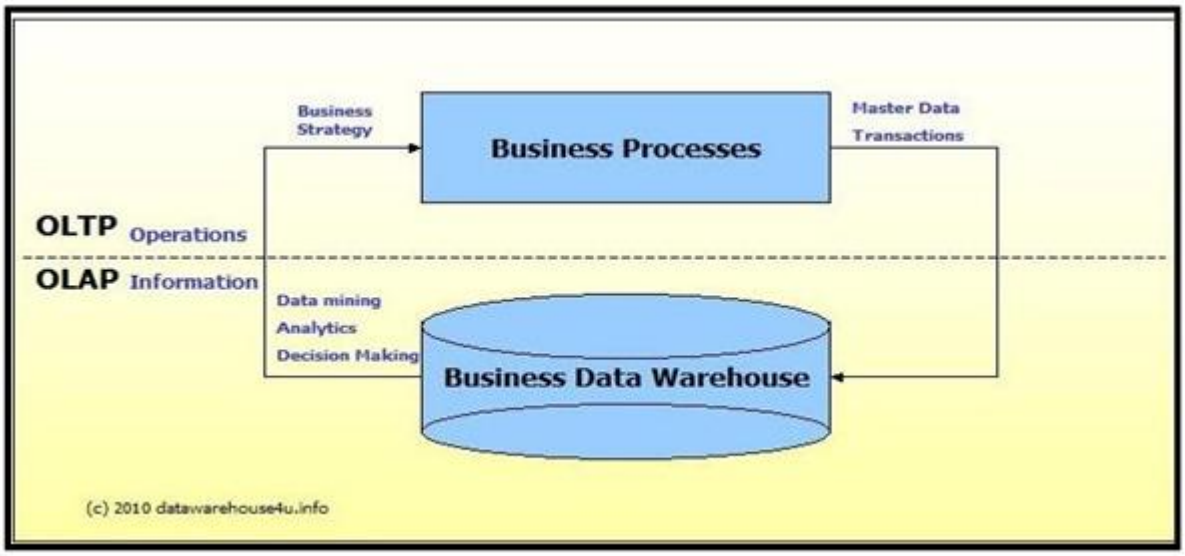
За разумевање суштине концепта Data Warehouse-а од великог значаја је уочавање карактеристика OLTP трансакционе и OLAP аналитичке базе података (слика 2).

OLTP (On Line Transaction Processing) описује начин на који рачунарски системи и крајњи корисници обрађују податке. Усмерен је на детаље, са честим ажурирањем од стране крајњих корисника. OLTP системи се заснивају на релационим базама података, изграђени су у складу са Код-овим правилима нормализације, да би се обезбедио инегритет и конзистентност података.

OLAP (On Line Analitical Processing) је врста технологије која омогућава аналитичарима и менаџерима увид у податке кроз брз, конзистентан и интерактиван приступ великом броју разноврсних извештаја, сачињених на основу добијених трансформацијом сирових података.
Data Warehouse подразумева овај приступ. Једна од карактеристика која раздваја OLTP од OLAP јесте дизајн базе података:

- ***трансакциони системи*** су дизајнирани тако да преузимају податке, врше измене над постојећим подацима, дају извештаје, одржавају интегритет података и управљају трансакцијама што је брже могуће

- ***аналитички системи*** су дизајнирани за велики број података намењених само за читање, обезбеђујући информације које се користе за доношење одлука.[7]



Слика 2. OLTP и OLAP базе података

Крајњи корисник захтева следеће:

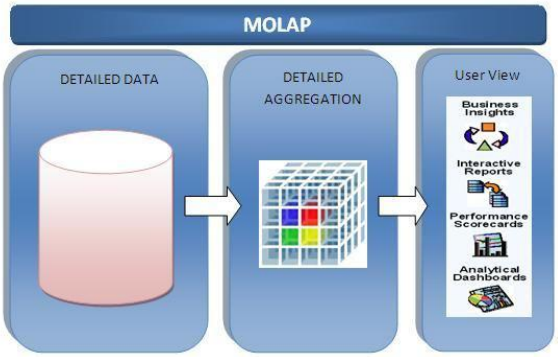
- да може да постави било које пословно питање
- да било који податак из предузећа користи за анализу
- да добијени подаци буду интегрисани и поуздани
- могућност неограниченог извештавања

У почетку су упити корисника били једноставни. Међутим, временом су постали толико сложени да релациони алати (OLTP) нису били у могућности да дају одговоре у прихватљивом временском периоду. Управо у ту сврху се користе OLAP системи. Они омогућавају једноставну синтезу, анализу и консолидацију података. OLAP је начин обраде података који карактеришу ад-хоц упити, слабо структуирани извештаји и анализа која обухвата релативно мали број трансакција али која укључује велики број табела и записа у њима. OLAP системи подржавају комплексне анализе које спроводе аналитичари и омогућавају анализу података из различитих перспектива (пословних димензија). OLAP системи као складишта података користе мултидимензионалност и денормализацију.

Постоје следеће архитектуре OLAP система:

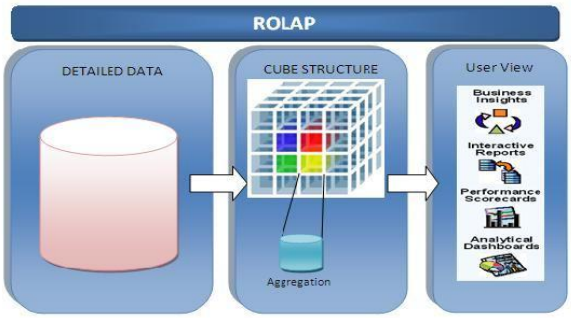
- ***вишедимензиони OLAP (MOLAP),***
- ***релациони OLAP (ROLAP),***
- ***хибридни OLAP (HOLAP).***
- ***Еластицни OLAP (EOLAP)***

MOLAP и ROLAP (слика 3, слика 4) се разликују по начину физичког чувања података. Код MOLAP система подаци се чувају у вишедимензионој структури, а у случају POLAP система подаци се чувају у релационим базама података. ***Предност MOLAP система*** је што обезбеђују одличне перформансе система када се ради са већ срачунатим подацима (агрегацијама). ***Недостатак MOLAP*** система је тешкоћа додавања нових димензија.



Слика 3. MOLAP систем

Подаци из различитих трансакционих система учитавају у вишедимензиону базу података помоћу батцх рутина. Када се заврши са учитавањем података атомског нивоа, прелази се на креирање агрегација, након чега је база података спремна за рад. Корисници задају своје захтеве за OLAP извештајима путем интерфејса.

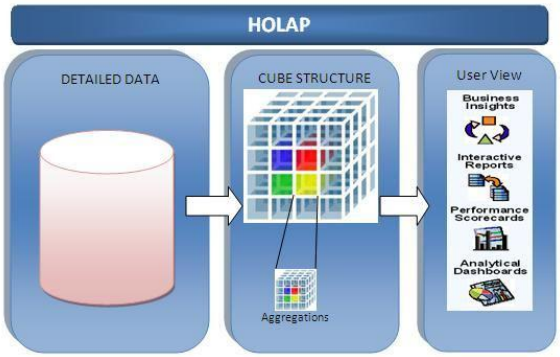


Слика 4. Релациони OLAP (ROLAP)

ROLAP системи приступају подацима директно из складишта података и раде са релационим базама података. ROLAP системи могу да раде са великим скуповима података. Чим се одреди извор података, корисник може започети анализу. С обзиром да се ради директно над базом података, кориснику су увек на располагању текући подаци. Код ROLAP система не постоје ограничења по питању броја димензија која постоје у случају MOLAP система.

HOLAP алати **(слика 5.)** могу приступати и релационим и вишедимензионим базама података. Циљ коришћења HOLAP алата јесте да се искористе предности MOLAP алата (кратко време одзива система и аналитичке могућности) и ROLAP алата (динамички приступ подацима).

При томе се не може рећи да је HOLAP прост збир MOLAP-а и ROLAP-а. То је заправо ROLAP који има могућност извршавања врло сложених SQL наредби. Циљ је био да се задрже све предности ROLAP-а, али да се при томе додају и неке нове могућности за рад са вишедимензионим базама података.



Слика 5. HOLAP систем

3. *Развој складишта података*

За разлику од трансакционих система који су орјентисани пословним процесима, складишта података су субјективно орјентисана, што значи да су фокусирана на субјекте у пословним процесима. Интегрисаност података у складиштима података обезбеђује да се подаци представљају у конзистентним форматима коришћењем конвенција при задавању имена и ограничења над доменима, атрибутима и мерама. Подаци у складиштима података су временски зависни, што значи да је сваки податак у вези са неким временским тренутком.При изградњи складишта података најбитнији су сами подаци, а не пословни процеси као што је то случај са трансакционим системима. Базе података намењене системима за подршку одлучивању могу бити веома велике, при чему неке табеле могу садржати и гигабајт података. Зато се величина базе мора узети у обзир при планирању складишта података.

За изградњу ***"DW" битни су сами подаци*** и потребно је:

- *извршити анализу извора података,*
- *припремити податке,*
- *изградити складиште података.*

3.1.Анализа извора података

Основни извори података за концепт складишта података су оперативни (трансакциони), тзв. OLTP (*On Line Transaction Processing*) подаци, као и спољне информације настале као историја пословања или индустријски и демографски подаци узети из великих јавних база података. Анализа изворних података се сматра кључним елементом и одузима 80% времена, јер је потребно дефинисати одговарајућа правила за преузимање података из изворних података. Знања везана за ову област су најчешће у главама оних који треба да користе складиште података.

Анализа извора података пролази кроз следеће фазе:

- ***Прикупљање захтева,***
  - ✓ ***прикупљање изворних захтева***
  - ✓ ***прикупљање корисничких захтева***
- ***Избор технике анализе података.***

У фази ***прикупљање захтева*** разматрају се пословне потребе и захтеви будућих корисника система.

Прикупљање изворних захтева је метода базирана на дефинисању захтева коришћењем изворних података у производно-оперативним системима. Ово се ради анализирањем ЕР- модела(концептуалан модел података који реалан свет "види" кроз ентитете и њихове односе) изворних података. Главна предност су подржавање свих података и свођење на минимум време потребно кориснику у раним фазама (стањима) пројекта док су недостаци: умањивањем корисниковог учешћа повећава се ризик од промашаја испуњења захтева корисника и одузима доста времена.

Прикупљање корисничких захтева је метода која се базира на дефинисању захтева истраживањем функција којима корисник тежи, односно које корисник извршава. Ово се обично постиже кроз серију састанака и/или интервјуа са корисником. Главна предност овог приступа је што се концентрише на оно што је потребно, а не на оно што је доступно.

***Поступак прикупљања захтева*:**

- интервјуисање кључних људи у организацији, нпр: аналитичари, менаџери и извршиоци.
- утврдити проток информација у и из сваког оделења (који извештаји и документација пристижу у оделење, како се користе, ко их користи, колико често пристижу итд).
- добијене податке организовати у неколико секција, као што су:
  - ✓ подаци о анализи (подаци о свим врстама анализа које се трентуно користе)и
  - ✓ захтеви везани за податке (опис свих поља података која се користе, извори).
- организоване податке проследити свим учесницима интервјуа ради мишљења и евентуалних корекција.

Постоји неколико ***техника анализе података***:

- Упити и извештаји,
- Вишедимензионалне анализе и
- *Data Mining*-циљ *Data Mining*-а јесте откривање скривених веза, предвидивих секвенци и тачних класификација у улазним подацима.

3.2.Припрема података

У процесу развоја складишта података припрема података је једна од најбитнијих активности. Даљи процес развоја складишта података биће успешан само ако је ова активност успешно завршена.

Припрема података се врши на основу раније одређеног извора података, правила за преузимање тих података, процедуре припреме и захтева корисника. Припрема се врши одређеним екстракционо- трансформационим алатима кроз следеће кораке:

- ***екстракција и чишћење података,***
- ***трансформација података.***

3.3.Изградња складишта података обухвата следеће задатаке:

- ***денормализација података,***
- ***дефинисање хијерархија,***
- ***креирање агрегација,***
- ***креирање физичког модела,***
- ***генерисање базе података,***
- ***учитавање података.***

3.4.Архитектура складишта података

Посматрајући појам могуће је разликовати архитектуру тока података (*енг. data flow*) и архитектуру система. Архитектура тока података говори о томе на који су начин подручна складишта уређена у самом складишту података, и на који начин подаци дотичу из изворних система до својих дестинација у складишту.

Дизајнирање физичке архитектуре складишта података, или једноставније архитектуре система, следи тек након оптимизације „*data flow*“-а. У пракси се архитектура често прилагођава ограничењима у инфраструктури или у буџетима нарушитеља система, па се у складу с тиме дефинишу послужитељи (*енг. software*), мрежа, програмска подршка (*енг. software*) и слично.

Оваква се једноставна архитектура најчешће користи с нагласком на употребу одвајања развојног, тестног и продукцијског послужитеља, међутим с обзиром на ограничења то понекад није могуће. Архитектура система приказана је на слици 6, испод.

Слика 6. Физичка архитектура система[5]

Архитектура *Data Warehouse*-а описује елементе и услуге које складиштење пружа, са детаљним приказом интеграције и оптимизације компонената, као и потенцијалног раста развоја.

Постоје два доминантна правца када је реч о архитектури система за складиштење података. Први правац пропагира Б. Инмон, који се сматра „оцем“ складиштења података. Други правац пропагира Р. Кимбалл, данас засигурно најутицајнија особа у области складиштења података. Ова два доминантна приступа развоју архитектуре складишта података позната су под називом ЦИФ и БУС архитектура.

У ЦИФ **(*енг. Corporate Information Factory*)** архитектури, подаци се из операционалних, изворних система преводе, трансформишу и сакупљају у свеобухватно, корпоративно складиште података. Складиште података садржи историјске податке који се не ажурирају, као и неке изведене податке и служи као извор сваком појединачном Дата Март-у. Корисници ЦИФ архитектуре приступају Дата Март- овима преко корисничких алата и податке прилагођавају својим потребама.

У БУС архитектури складиште података се посматра као скуп различитих Дата Март подсистема, који поседују конформисане (усаглашене) димензије. Овакви Дата Март-ови се граде тако што се директно из операционалних система издвајају подаци који су значајни за поједини пословни процес. Кимбалл сматра да се Дата Март-ови морају градити тако да задовољавају принцип конформности димензија и чињеница, како би се омогућила њихова интеграција. На тај начин добијени подсистеми продаје, набавке, финансија, магацина деле заједничке информације између себе, јер су повезани скупом заједничких или усагалашених димензија. За такве подсистеме каже се да се налазе на заједничкој „магистрали“ (енг. *Bus*) и сви заједно чине део архитектуре.

Делови архитектуре су:

- ***Извори података (Data Sources)***
- ***ЕТЛ процеси (Extraction Transformation Loading)***
- ***Модел базе података (logički i fizički)***
- ***OLAP сервер***
- ***Метаподаци (Metadate)***
- ***Складиште оперативних података (Operational Data Storage)***
- ***Дата Март-ови (Data Marts)***
- ***Алати за извештавање и анализу (Reporting and Analitycal tools)***

3.5.Извори података

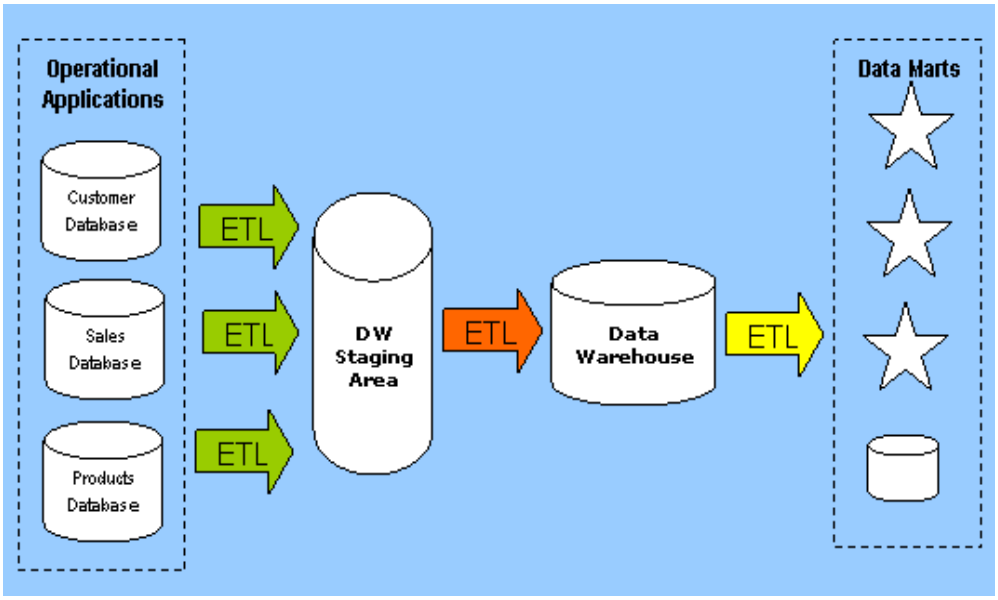
Постоје две врсте извора података: спољашњи и унутрашњи. Унутрашњи подаци припадају компанији и генерисани су путем трансакциског система, и описују активности које су се догодиле у предузећу. Спољашњи подаци се прикупљају изван компаније посредством специјализованих функција које се баве прикупљањем и дистрибуцијом информација. Могу се налазити на разним платформама које садрже структуиране податке, као што су табеле, или неструктуиране податке као што су текстуални фајлови, фотографије и др. Од суштинске важности су за стратешке одлуке, јер помоћу њих организација уочава повољне могућности као и претње.

3.6.ЕТЛ процеси

Као што је већ речено, подаци улазе у складиште података из различитих извора, најчешће из трансакционих система предузећа (слика 7).Најопсежнији посао у активностима складиштења података представља процес интегрисања података и организовања њиховог садржаја. Скуп процеса има задатак да изврши целовито трансформисање и пуњење из једног или више трансакционих система у складиште података.

Пре почетка ЕТЛ процеса потребно је извршити припремне активности везане за складиштење података. Изворне податке унешене из различитих датотека потребно је стандардизовати, односно превести их у стандардни формат. У том формату подаци ће се користити у свим даљим фазама обраде. Осим што се у информационом сисрему исти подаци могу појавити на више места, они могу бити различити, односно њихове вредности нису исте на свим местима на којима се ти подаци јављају. Због тога је потребно извршити њихово усклађивање.

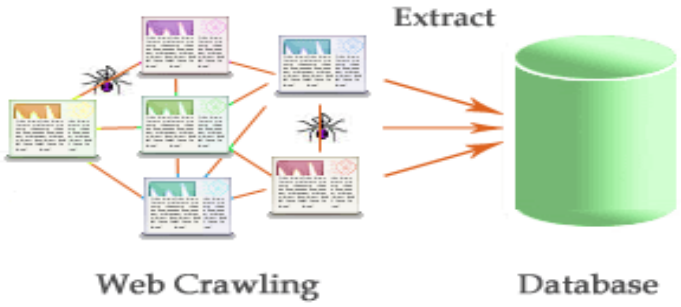
Чишћење има задатак да уклони све оне податке који се појављују као последица ранијих грешака у раду информационог система или уношења нетачних и лажних података у систем**.**



Слика 7. Етл процеси[3]

3.6.1. Екстракција података

Процес екстракције података потребно је извести на начин да при том редовни оперативни послови што мање трпе. Екстракција представља процес прикуплања података из различитих извора и платформи и смештање тих података у Data Warehouse-у. Екстракција података је много више од простог копирања података са једног система на други. Програми и алати за екстракцију су обликовани тако да ЕТЛ процес могу обављати што продуктивније уз настојање да потребне податке из оперативних процеса преузимају што је могуће брже (слика 8). При том се као проблем може јавити потенцијално висок степен редундансе података у трансакциским системима и зато је неопходно изабрати такав приступ екстракцији којим се врши узимање само оних података који ће се користити у апликацијама пословне интелигенције.



Слика 8. Екстракција података[4]

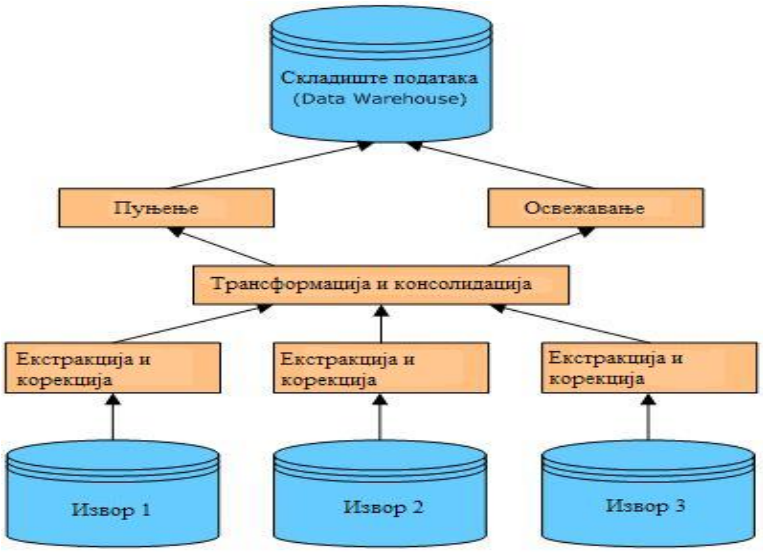
3.6.2. Процес трансформације података

У оквиру ЕТЛ процеса највише времена се троши на поступак трансформације података, према стручним проценама он траје и до 80% од укупног ЕТЛ процеса. У поступку трансформације могу се појавити различити проблеми који успоравају процес а као најчешћи се издвајају: неконзистентне вредности података, неподударност примарних кључева, нетачност података, различити формати података. Неке од метода трансформације су:

- Селектовати само одговарајуће табеле за уношење
- Превођење кодираних података
- Стварање нове вредности
- Спајање података из разних извора
- Сумирање више редова података

3.6.3. Пуњење

За процес пуњења складишта података (слика 9.) користе се више врста ЕТЛ алата као што су алати за иницијално пуњење, пуњење историјских података и програми за пуњење. Карактеристика програма за иницијално пуњење складишта података је да садрже рутине за чишћење и усклађивање података, да би се из података елиминисале грешке. Код историјских података понекад није могуће применити поступке чишћења који се примењују за „живе податке“, јер је од времена настанка тих података до данас можда дошло до различитих промена у слоговима и форматима података. За разлику од ажурних, историјски подаци су статичног карактера и чине само садржај архивских датотека. Трећу врсту представљају програми за инкрементално пуњење података, а активирају се временски након предходна два. Покрећу се периодично и имају улогу стално активног механизма пуњења складишта одговарајућим садржајем.



Слика 9. Место и ток ЕТЛ процеса у креирању концепта пословне интелигенције[2]

3.7.Модел базе података

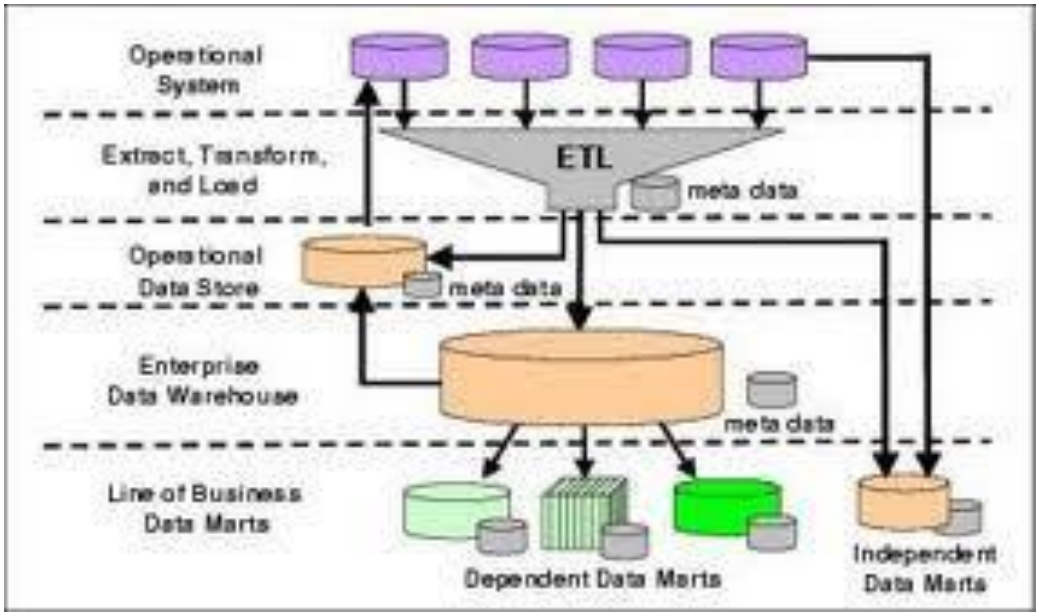
Приликом креирања складишта података данас у пракси сусрећемо три основна модела:

- Двослојна архитектура са једним заједничким складиштем података
- Двослојна архитектура са више независних локалних складишта података
- Трослојна архитектура са заједничким складиштем података и више повезаних локалних складишта података

Двослојну архитектуру са једним заједничким складиштем карактерише јединствено, централизовано складиште података. Таква складишта су великог обима и врло сложена и у њима се по правилу складишти огромна количина података. Трошкови одржавања овакве архитектуре су веома високи и уз то захтевају већи анганжман људства на одржавању складишта.

Двослојну архитектуру са више независних локалних складишта карактерише постојање већег броја независних локалних складишта података намењених за рад појединачних апликација по организационим јединицама предузећа. Резултат овакве архитектуре је велики број система у којима се посебно уносе подаци из различитих трансакционих база података.

Трослојна архитектура складишта података састоји се од већег броја локалних складишта података(***Дата Март-ова***) и једног заједничког складишта (*Дата Wарехоус-а*) које је смештено између складишта података и различитих извора података унутар и изван предузећа (слика 10). Складишта података се ослањају на централно складиште података које им испоручује податке у облику који даје уједначен увид у све сегменте пословања предузећа.



Слика 10. Трослојна архитектура са заједничким складиштем података и више повезаних локалних складишта података[3]

3.8.Метаподаци

Метаподаци описују информације и податке унутар складишта , инегришу долазеће податке, представљају алат за редефинисање и ажурирање одређеног модела *Data Warehouse*-а. Они служе да пруже информације о подацима који су смештени у Data Warehouse-у. Укључују описе елемената података, као што су описи типова података, описи атрибута, описи домена, затим називе, величину и дозвољене вредности. Предметно су орјентисани, дефинишу начин на који ће се трансформисани подаци инерпретирати, пружају информације о сродним информацијама у Data Warehouse-у и предвиђају време одзива приказујући број слогова који треба да се обради у упиту. Са становишта администрадора Data Warehouse-а, Метаподаци представљају складиште података и документацију о садржају и процесима у Data Warehouse-у. Са друге стране, са становишта корисника Метаподаци представљају мапу за кретање кроз податке.

3.9.Складиште оперативних података

Традиционална *Data Warehouse* архитектура није у складу са потребама менаџера за "*up-to-the-minute*" подацима, неопходним у одлучивању у стварном времену. У таквој ситуацији кад перфомансе постају критичне, јавља се идеја о креирању живих, оперативних складишта података, тзв. "репорт сервер" или "огледало базе". ОДС је предметно орјентисан(на једном месту су сви подаци), инегрисан(представља инегрисану слику предметно орјентисаних података извучених из било ког дела оперативног системе),орјентисан на тренутну вредност(осликава тренутни садржај његових изворних система, при чему се тренутна вредност може дефинисати на различите начине, за различите изворе, у зависности од захтева имплементације), променљив(како је ОДС орјентисан на предмет, он је подложан променама онолико често колико је то потребно за осликавање тренутног стања. То значи да се подаци мењају у стилу OLTP система, па ће један исти упит дати различите вредности у резличитим тренуцима времена, јер су се подаци у међувремену променили) и брзина ажурирања се одвија у краћим временским интервалима него код *Data Warehouse*-а.

3.10. Data Mart-ови

Дата Март-ови су подскупови података складишта података и место где се одвија највише аналитичких активности. Подаци у сваком Дата Март-у су уобичајено креирани за одређену могућност или функцију. Сваки специфични Дата Март је оптимизован за унапред дефинисано подручје и не мора бити одговарајући за друге употребе. Најчешћи облик Дата Март-а је мулти-димензионалан, што омогућава лак приступ, брзу и квалитетну анализу података. Проблем који се може јавити у организацији која је имплементирала неколико Дата Март-ова пре имплементације централног складишта података, је интеграција постојећих Март-ова у целовит систем. Потребно је непрестано правити баланс између тежње да се они креирају као одвојени силоси или одељења, и потребе за успешним функционисаним складишта на глобалном нивоу. Међусобно усклађени и кординирани Дата Март-ови се називају супер Март-ови.

Сваки Дата Март се састоји из низа табела чињеница, чији је кључ системљен од више спољних кљиучева који долазе из табела димензија.

Конформисана димензија је она која има потпуно исто значење у свакој табели чињеница са којом је повезана. Зато је та димензија идентична у сваком Дата Март-у. Управо то доводи до интеграције Дата Март-ова, а међусобне везе се успостављају преко дељених димензија. Може бити реализован као:

- *независни Дата Март* » изолован од других Дата Wapexoyc система
- *зависни Дата Март* » наслоњен на друге Дата Wapexoyc системе (слика 11)

Слика 11. Зависни дата март[3]

3.11. Алати за извештавање и анализу

Да би се испунио примарни циљ пословања *Data Warehouse*-а подаци морају бити на располагању за анализу, извештавање, постављање упита и сличне процесе. Постоје многе апликације које могу обављати ове функције а неке од њих су:

- **Business Intelligence алати**
- **Извршни информациони системи**
- **OLAP алат**
- **Аналитичке апликације**
- **Data Mining**

Са аспекта крајњег корисника овај слој је најбитнији слој у *Data Warehous* архитектури.

Како би се пронашли одговарајући презентациони алати за информационе захтеве крајњих корисника, претпоставка је да постоје четири категорије корисника: "моћни корисници" који су спремни и способни да користе комплексније алате за креирање сопствених извештаја, "повремени корисник" који директно није заинересован за детаље о *Data Warehouse*-у, али му је повремено потребан приступ информацијама, "корисници који имају потребу за статичким информацијама" који имају потребу за прецизно дефинисаним подацима у одређеном временском интервалу и "корисници који захтевају ад хоц упите и аналитичке могућности алата" а то су углавном аналитичара којима свака информација у Data Warehouse-у може бити значајна у неком тренутку. Различите врсте корисника захтевају различите презентационе алате, али сви могу да приступају заједничком Data Warehouse-у. Такође, различите способности корисника одређују и разне начине презентације резултата обраде, од табеларних приказа до најразличитијих графичких приказа.

4. Кориснички приступ подацима

У подручју корисничког приступа подацима налазимо скуп алата за приступ и анализу података. Оно што корисник очекује је брзина одазива (пожељно у секундама) тако и једноставност рада.

Алате за кориснички приступ можемо грубо поделити у следеће категорије:

- ***Алати за једноставно извештавање*** – алати који омогућују преглед предефинисаних извештаја које није могуће мењати (нпр. *Oracle Reports*)
- ***Алати за постављање „adhoc" питања (Ad Hoc Query Tools)*** – алати који омогућавају постављање питања на једноставан начин без познавања SQL-а у сврху прегледа података постављених у складишту. Такође, омогућују преобликовање постојећих извештаја применом поступака као што су мењање оси односно положаја димензијских атрибута на екрану, кретање по хијерархијским нивоима димензија тзв. бушење (*drill down*). Неки од алата из овог скупа су: *Business Objects, Oracle Discoverer, Oracle Express Objects.*
- ***Алати за напредне анализе*** – алати који омогућују стварање модела за предвиђање, односно откривање скривених односа међу подацима. Неки од алата су: *Business Objects, Oracle Express*, те алати компаније *SAS*.

Када је у питању врста корисника која ће користити поједине групе алата тада ће вероватно само мали број корисника обављати самосталну изградњу извештаја над базом складишта података. У већини случајева, извештаји се морају израдити, а затим дати кориснику на употребу. Другу и трећу врсту алата користе аналитички оријентисани корисници. Трећа врста алата често ће црпити податке из мултидимензионалних OLAP база података које омогућују напредније анализе.

4.1.Димензионални модел података

Настанак димензионалног моделирања узрокован је недостацима сложених, нормализованих релационих структура, изграђеним техникама ER моделирања.

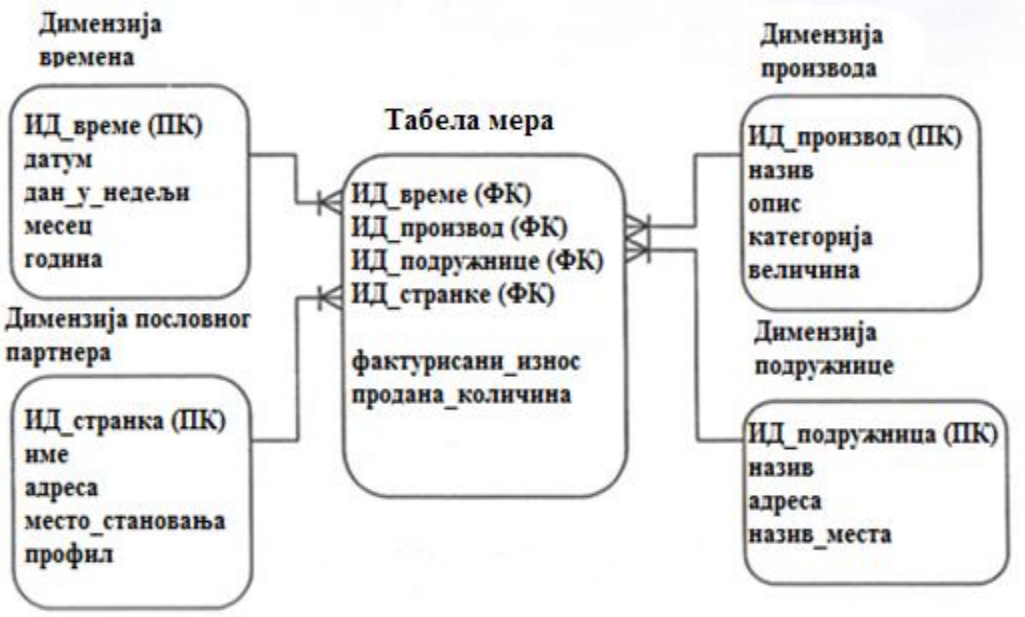
Основна тежња ER моделирања је уклањање редунданције што резултира сложеном структуром неприкладном за прави приступ од стране корисника те спорим одазивом при постављању сложених задатака.

С друге стране, димензионални модел, због своје једноставности узроковане контролисаном денормализацијом даје могућност једноставног и брзог добијања информације, иначе скривене у сложеним структурама продукцијског система базираног на нормализованом, релацијском моделу.

4.2.Основни елементи димензионалног модела

Сваки димензионални модел састоји се од две врсте објеката (слика 12):

- Мере – у релацијској бази смештене у табелу мера
- Димензије – у релацијској бази смештене у Табелама димензија



Слика 12. Пример димензионалног модела[8]

4.3.Табела мера

Табела мера обухвата мере те поседује сложени примарни кључ системљен од комбинације страних кључева који показују на димензијске табеле.

Табеле мера можемо класификовати у односу на њихову адитивност и то на следећи начин:

- Адитивне мере
- Полуадитивне мере
- Неадитивне мере

Адитивна мера је она која подноси операцију здруживања по свим димензијама. Овакву врсту мера најлакше је приказивати и вршити манипулације над њом, помоћу корисничких алата за претраживање.

Полуадитивна мера је она која подноси операцију здруживања по неким димензијама. Нпр., уколико би имали меру која бележи количину производа у складишту на крају месеца, тада би ова мера била адитивна по свим димензијама осим по временској (ако узмемо да се запис у димензионалној табели времена односи на јединицу мању од једног месеца). Кориснички алати за претраживање нису у могућности да воде бригу о полуадитивности мере па ће врло лако доћи до погрешних резултата не водећи бригу о димензији по којој мера није адитивна.

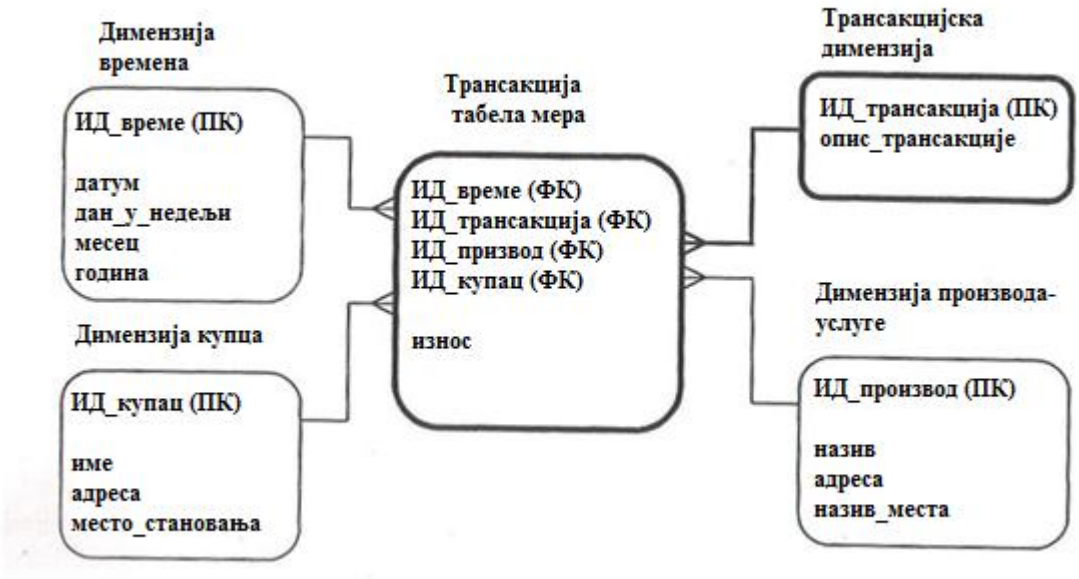
Неадитивна мера би била она која не подноси сумирање ни по једној димензији. Пример би могла бити мера *измерена-температура-просторије*. У овом случају сумирање оваквог податка по било којој димензији нема смисла. Овде би се могла тражити нпр. средња вредност, број проведених мерења и слично, али операција здруживања, нема смисла.

Важна карактеристика табеле мера је и однос текстуре. Текстура говори на ком односу детаља су подаци постављени у табели мера и Табелама димензија. Димензије су обично на свом атомичном односу јер је то најприроднији начин. Када је о мерама реч, препоручује се да буду у најмањој могућој ширини текстуре која одговара најнижим односима димензија о којима мера зависи. Дакле, износ мере може одговарати нпр. појединој ставци у фактури, појединим телефонском позиву којег бележи централа, а не сумарном износу с фактуре на ширини државе становања пословних партнера и слично. Овај принцип је од изузетне важности јер Табела мера која бележи мере на најнижој могућој раздаљини текстуре пуно лакше подноси измене у модулу. Надаље, подаци најниже раздаљине текстуре су неопходни при поступцима тражења скривених корелација међу подацима. Текстура података даје робусан дизајн који се лакше прилагођава променама у моделу података те боље одговара на корисничке захтеве за новим и можда неочекиваним задацима и извештајима. Важно је нагласити да све мере унутар једне табеле мера морају имати исту раздаљину текстуре.

Одабирајући текстуру табеле мера одлучујемо се и за један од три могућа типа праћења података у табели мера. Разликујемо три врсте Табела мера:

- Трансакцијска Табела мера (*Transaction Fact Table*)
- Временска слика мера (*Snapshot Fact Table*)
- Табела ставки (*Line Item Fact Table*)

Трансакцијске табеле мера садрже у себи вредности везане уз одређени догађај који се догађа као део пословног или неког другог процеса. Примери трансакција могли би бити нпр. појединачни догађаји подизања новца или полагања новца у банци, појава телефонског позива којег бележи телефонска централа, куповина производа или услуге на продајном месту итд. Трансакцијски модел захтева постојање тзв. трансакцијске димензије (слика 13) која говори о каквој је трансакцији реч.[8]



Слика 13. Пример трансакцијске табеле мера[7]

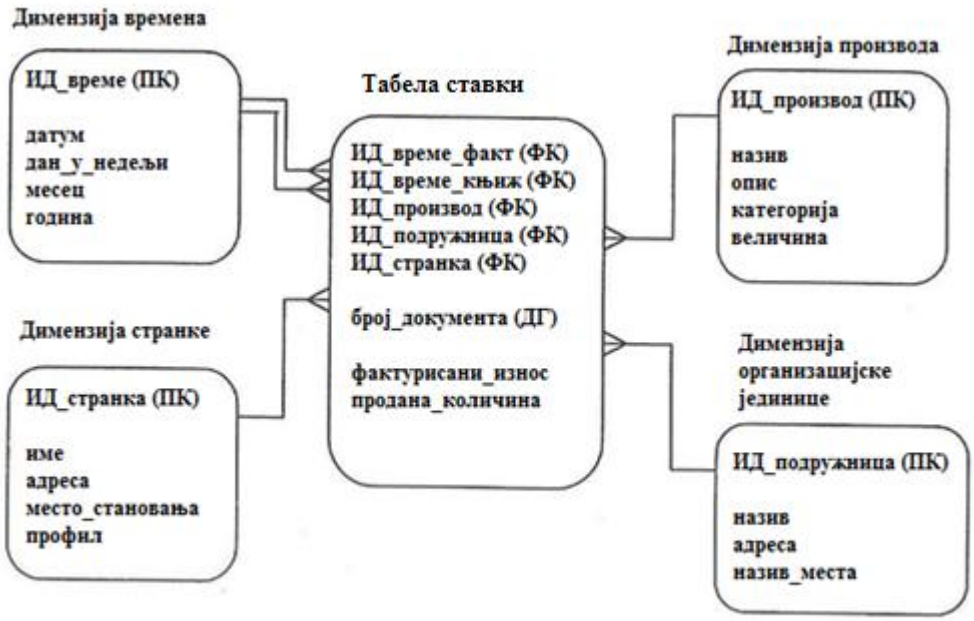
Друга врста табеле мера је временска слика. У овој табели мера не прате се поједини догађаји односно уз њих везани износи, као што је случај с трансакцијским Табелама мера. Временска слика обухвата дужи временски период (нпр. једну недељу или месец).

Мера у временској слици приказује обављени посао на сумираној раздаљини унутар задатог периода. Оваква Табела најчешће ће бити системљена од већег броја мера.

Поједине мере унутар једне временске слике ће бити резултат различитих пословних трансакција (уговорени прилив, фактурисани прилив, фактурисани одлив, наплаћени прилив, наплаћени одлив, утржена потраживања итд.). Пример модела с временском сликом приказан је на слици 14.

Слика 14. Пример временске слике мера

Трећа врста табеле мера је тзв. Табела ставки. Мере из табеле ставки добијају се из износа везаних уз поједине ставке (*Line items*) документа као што су нпр. рачун (фактурисани износ везан уз ставку рачуна), полиса осигурања (уговорени и осигурани износ везан уз поједина покрића) итд. На слици 15. дан је пример овакве табеле мера.



Слика 15. Пример табеле мера као табеле ставки с једном дегенерисаном димензијом

4.4.Сложене димензијске структуре

Неке ситуације које наилазимо у пракси захтевају употребу посебних структура података. При томе мислимо на надоградњу Табела димензија. У поглављима која следе дат је опис неких познатих сложених структура које се користе при моделирању звезда модела.

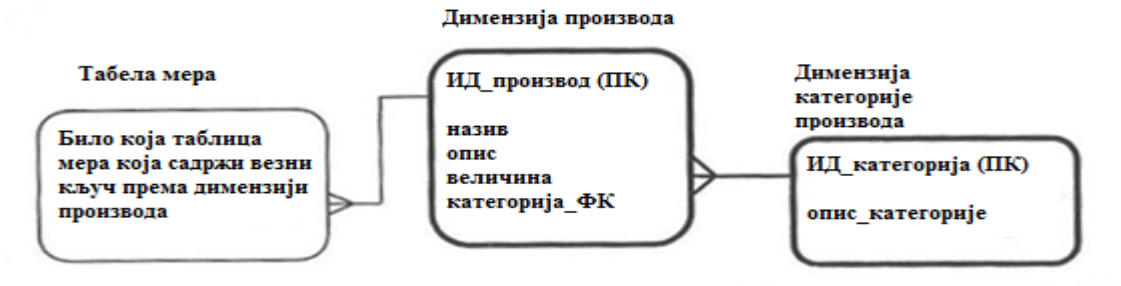
4.4.1. Пахуљаста структура

Пахуљаста структура (*Snowflake*) настаје када се над одређеном димензијом изврши поступак делимичне или потпуне нормализације. На пример, уколико се из димензије производа издвоји категорија којој производ припада те се стави у засебну димензијску табелу, настаје пахуљаста структура, у случају са слике 16 приказана је димензија производа над којом је извршен овакав поступак. У овом случају димензију производа можемо назвати главном димензијом пахуљасте структуре док димензију категорије производа можемо назвати пахуљастом димензијом.

Важно је напоменути да стварање овакве структуре није препоручљиво. Описана структура отежава кориснику разумевање пакета података, отежава приказ података на извештају те успорава SQL задатке.

Истина, пахуљастом структуром остварује се уштеда на физичком простору али је она занемарљива (Уштеда од неколико мегабајта не значи ништа у складишту података које се мери у гигабајтима).

Још један од разлога који се наводи против коришћења оваквих структура је и могућност коришћења тзв. бит-мапираних индекса који изузетно добро функционишу када су постављени над пољима ниске кардиналности. Када се ствара пахуљаста структура тада се обично одстрањују атрибути ниске кардиналности. Тиме се уклања и могућност постављања тзв. бит-мапираних индекса над посматраним пољем.[9]

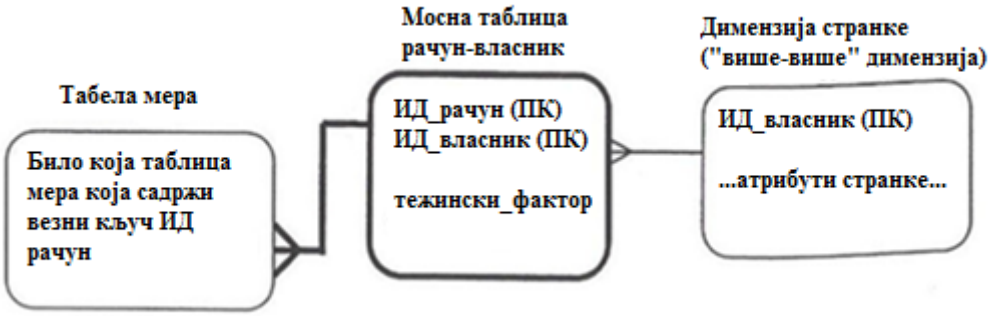


Slika 16. Pahuljasta struktura

4.4.2. Мосна структура

Мосна (*bridge*) структура појављује се у случајевима када једном запису мора бити придружено више записа неке димензије. Примери могу бити преглед пацијената којем се на једном прегледу пружа дијагноза или банковни рачун који има више записа димензије дијагноза, односно димензије власника рачуна.

Овакву димензију називамо димензијом више-према-више (*manu-to-manu dimension*). На слици 17. дан је пример банковног рачуна који има више власника.



Слика 17. Мосна структура[7]

Између димензије више-према-више власника рачуна и табеле мера налази се мосна Табела рачун-власник коју карактерише следеће:

- Примарни кључ мосне табеле системљен је од две компоненте – од примарног кључа групе (кључ рачуна) и примарног кључа димензије више-према-више (кључ власника).
- Страни кључ којим се Табела мера веже на мосну табелу системљен је само од кључа групе (кључа рачуна) – тиме је остварена веза више-према-више
- Мосна Табела може поседовати тежински фактор који сваком запису из више-према-више димензије додељује одређени део мереног износа из табеле мера – сваком власнику посматраног рачуна прописује се одређени проценат од износа мере.
- Уколико мосна Табела поседује тежински фактор, тада збир тежинских фактора свих мосних записа с истим групним кључем мора износити „1”. То је неопходно како би зброј делимичних износа додељених појединим записима из више-према-више димензије унутар једне групе био јединак укупној вредности придруженој посматраној групи.

5. Агенти за претраживање web-а

Пораст популарности и броја доступних садржаја, који су у потпуности неорганизовани, руши основни концепт *"web"*-а - једноставно проналажење информација. Због тога се почевши од 1994. године развија низ алата, чији је једини задатак проналажење жељених садржаја на *"web"*-у. Ти алати се развијају у два основна смера, па разликујемо каталоге и претраживаче (*eng. search engines*). Једни и други садрже базу *"web"* страница, индех-а, УРЛ-ова (*eng. Uniform Resource Locator* - адреса која јединствено указује на неки садржај на Интернету), а основна разлика је у начину прикупљања тих података. Код каталога, садржај базе искључиво зависи од људи, тј. људи претражују *"web"*. То раде тако што за интересантне странице напишу сажетак и сажетак се уз УРЛ адресу странице памти у бази. Каталози се користе и данас, а један од најпопуларнијих је *Yahoo*. За разлику од каталога, прикупљање података код претраживача је у потпуности аутоматизовано и обављају га агенти за претраживање *"web"*-а (*eng. "web" Searching Agent*) који су још познати под називом пауци, пузачи, црви (*eng. spider, crawler, worm*).[12]

Агенти за претраживање *"web"*-а „лутају" *"web"*-ом у потрази за новим страницама, и када их пронађу „довлаче" их и снимају у базу. Мада разни називи попут паука, луталица, пузача, црва стварају привид да се агенти крећу, то није истина, они су стационирани на рачунару и на том рачунару „довлаче" странице. Они „скачу" с *"web"* странице на *"web"* локацију преко веза, прикупљајући наслове свих локација, УРЛ, и део њихових текстуалних садржаја. Када нађу место, они прегледавају (*eng. Scan*) *"web"* странице тога места и записују све информације у индекс.

У случају агената који претражују *"web"*, њихова околина је *"web"*, а комуникацију са корисником омогућује терминал рачунара. Оно што агент сагледава су речи ХТМЛ документа научене коришћењем програмских детектора (сензора) повезаних кроз целу мрежу (Интернет) уз помоћ ХТТП-а. Задатак агента је да установи да ли је постигнут циљ, тј. да ли је пронађена "web" страница која садржи кључну реч или фразу, и ако није, пронаћи друга места која ће посетити и претражити у сврху задовољења циља, тј. обављања задатака. Агент делује на околину користећи излазне методе како би обавестио корисника о статусу претраживања или крајњим резултатима, који би требали представљати постигнут циљ.[15]

6. Истраживање података

6.1.Интелигентни агенти за претраживање web-а

Интелигентним агентима за претраживање *"web"*-а називају се рачунарски програми који самостално изводе неки претраживачки посао "у име и за рачун" корисника. Смештени су у рачунару власника, што не мора нужно бити (а најчешће и није) рачунар крајњег корисника, већ неко *"web"* место. Корисник их мора "напунити" информацијама о доменима свог интересовања, правилима претраживања, приоритетима, и евентуалним временским ограничењима. Након што агент обави постављени задатак, анализирају се резултати, а агент исправља сам себе ако ти резултати нису задовољавајући према неком унапред постављеном критеријуму.

Интелигентни агенти за претраживање *"web"*-а могу имати различите степене самосталности у извршавању задатака у односу према кориснику и његовим потребама. Тако, неки од њих могу "само" прикупљати информације, неки могу филтрирати поруке примљене електронском поштом, а неки сарађивати са другим агентима, и на на тај начин обављати врло сложене и специфичне задатке.

Развијене су три врсте таквих агената:

- ***"web" crawler*** (Покушај давања кориснику целовит преглед информација, "шетајући" *"web"*-ом и извештавајући корисника о ономе шта су пронашли.)
- ***"web" пауци (Web spider)***,(То су програми који се помоћу "црвића" (*eng. worm*) "увлаче" у дубину хипермедијских докумената, односно *"web"* страница, и пронађене информационе садржаје индексирају и чувају у властитој бази података, коју корисник касније може једноставно локално (на своме рачунару) претраживати и анализирати.)
- ***"web" роботи (Web robot)***.(То су програми који могу готово независно од корисника извршавати комплетне трансакције, попут куповине на даљину, резервације авио карата, новчаних трансакција итд., у складу са инструкцијама које им је корисник раније дао.)

Интелигентни агенти имају широку област примене у претраживању тако да се они користе за:

- ***статистичке анализе*** - агенти могу рачунати просечан број докумената по серверу, однос различитих типова датотека, просечан број *"web"* страница, број *"web"* сервера на Интернету итд.

- ***освежавање УРЛ адреса*** - главни проблем код одржавања *"web"* страница је у томе што адресе на друге странице могу постати неважеће, тзв. 'мртве везе', ако се та страница премести или чак уклони. Стога су развијени агенти, чији је задатак провера исправности адреса.

- ***mirroring*** - је поступак пресликавања садржаја *"web"* сервера на други рачунар с циљем смањења промета на неком делу мреже. Агенти се могу користити за успостављање миррорне слике *"web"* страница, али ова метода још није у потпуности развијена тако да кад дође до промене неке *"web"* странице морају се освежавати све миррорне странице, а не само оне које су промењене.

- ***индексирање*** - прикупљање кључних речи из *"web"* страница како би претраживање било што лакше и брже. Најчешће се агенти користе за прикупљање наслова и неколико првих параграфа текста са странице. О овој функцији интелигентних агената биће више речи у наредном делу.

- ***проналажење података*** - интелигентни *"web"* агенти имају велики потенцијал у проналажењу података (*eng. data mining* - процес проналажења узорка у великој количини података кроз низ претраживања), уз то могу доносити одлуке засноване на претходним претраживањима, да би обавили што комплексније претраживање.

- ***комбинована употреба*** - један агент може извршавати и више споменутих задатака.[16]

6.2.Алгоритми и основни проблеми у имплементацији web претраживача

Основу *World Wide "Web"-a* (који називамо и само *"web"*), представља једноставни, отворени, клијент-сервер дизајн:

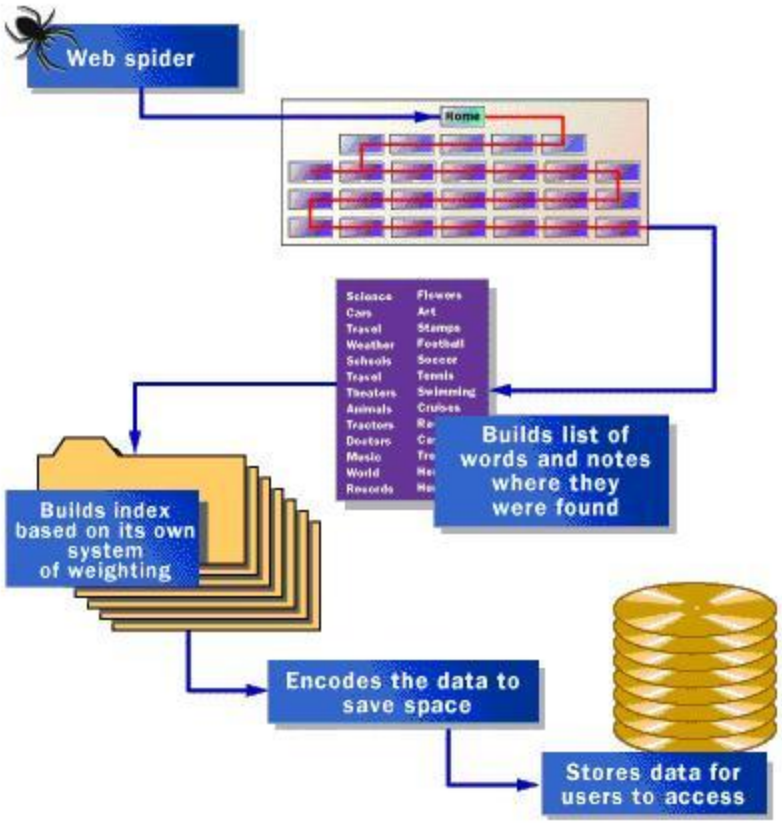
(1) сервер комуницира са клијентима користећи *http* протокол (*lightweight*, асинхрони протокол који омогућава пренос разних врста информација - текста, слике, медија - као што су аудио и видео фајлови, енкодовани у оквиру *хтмл* маркуп језика),
(2) клијент (*browser*), врши парсирање и графички приказ добијених *html* страница.

Сам хтмл језик, у оквиру дефинисаног маркуп-а, подразумева и дефинисање тзв. хиперлинкова *(hyperlinks)*, који представљају референцу једног документа ка другом, од којих је сваки јединствено дефинисан својим УРЛ-ом (*uniform resource locator*). На овај начин, *"web"* (у најужем смислу) можемо посматрати као мрежу, УРЛ-ова, међусобно повезаних хиперлинковима.

Најефикаснији начин приступа информацијама на *"web"*-у јесте коришћењем *"web"* претраживача. У општем случају, претраживач се састоји из три дела :

• *crawler* , који је као што је у претходном делу описано задужен за аутоматско прикупљање страница са *"web"*-а и њихово смештање у индекс претраживача
• *indexer* , који обезбеђује креирање одговарајуће структуре (*inverted index*), која омогућава ефикасну репрезентацију и претраживање архивираних страница.
• *query handler*, који прихвата корисничке упите и одговара на њих коришћењем индекса претраживача

Шематски приказ сваког савременијег претраживача је дат на (Сл.18.).



Сл.18. Шематски приказ "web" претраживача

Основни принцип који је узроковао експлозивни раст *"web"*-а – децентрализовано и неконтролисано публиковање садржаја – се показало и показује као највећи изазов за "web" претраживаче, у њиховом настојању да индексирају и архивирају садржај на *"web"*-у. Будући да је публиковање доступно практично сваком кориснику, *"web"* странице показују хетерогеност у великом броју кључних информационих аспеката (истина, неистина, контрадикције…), што доводи до проблема одређивања страница које садрже релевантне и објективне информације, везано за задату тему.

Додатни проблем, представља димензија и структура самог *"web"*-а. У пракси, чак није ни могуће дати једноставан одговор на питање "колико је велики *"web"*. Најрелевантнији одговор на ово питање би се могао дати посматрањем величине индекса неких од највећих *"web"* претраживача. Крајем 1995, Алтависта је пријављивала величину индекса од 30 милиона статичких *"web"* страница, док је на јесен 2005. године, Гоогле пријављивао величину индекса од око 8 милијарди страница. Под овим подразумевамо само статичке *"web"* странице (странице чији се садржај не мења у периоду између два приступа).

Услед наведеног, сваки *"web"* претраживач, се пре или касније (у зависности од величине), суочава са неким од следећих проблема:

- ✓ Брзина раста *"web"*-а је знатно већа него што је постојећа технологија у стању да индексира.
- ✓ Велики број *"web"* страница ажурирају свој садржај веома често, што захтева да их претраживачи чешће посећују, да би имали ажурне копије у индексу.
- ✓ Динамичке странице се или споро и тешко индексирају или могу резултовати у прекомереном броју резултата.
- ✓ Велики број динамички генерисаних *"web"*-сајтова није уопште могуће индексирати коришћењем стандардних *"web"* претраживача (ови сајтови чине тзв. "*невидљиви "web""*).
- ✓ Secure странице (*xmmnc*), могу представљати проблем *Crawler*-има јер им не могу приступити или из техничких разлога или их не индексирају из разлога приватности.
- ✓ Релевантност страница, поред тога што се тешко одређује, може бити и двосмислена, односно корисник и претраживач могу имати различита "схватања" релевантности.
- ✓ Кориснички упити су ограничени искључиво на кључне речи, што може резултовати у великом броју лажних позитивних резултата (странице које садрже дате кључне речи, али нису оно што је корисник тражио).
- ✓ Количина релевантних резултата у односу на задати упит јесте обично већа него што је корисник у могућности да прегледа (ово у пракси резултује у чињеници да се најчешће прегледа само првих пар страница са резултатима).
- ✓ Квалитет садржаја на "web"-у варира, тако да су неопходне технике које одвајају "*сигнал од шума*", односно странице ниског квалитета од страница високог квалитета.
- ✓ Велики број страница на *"web"*-у садржи валидне информације, али није структуиран у складу са дефинисаним конвенцијама.
- ✓ Велики број страница, представља дупликате, односно странице истог или сличног садржаја, али са различитим УРЛ адресама. Неопходно је елиминисати дуплирање индексирања оваквих страница.
- ✓ Неки претраживачи не рангирају странице по релевантности, већ по количини новца коју оглашавачи плаћају да се нађу међу резултатима са највећим скором.
- ✓ На *"web"*-у, могу постојати групе сајтова, креиране искључиво са циљем манипулације функцијом рангирања страница (*linkspam*).
- ✓ Такође, могу постојати и други облици спам-овања "web" претраживача, као што су *cloaking* (враћање различитог скупа страница у зависности да ли приступа корисник или *Crawler*), *doorway pages (* странице креиране у односу на одређени упит, које при учитавању врше редирекцију на страницу потпуно различитог садржаја).[14]

6.3.Crawling

Web Crawling представља процес прикупљања страница са *"web"*-а, ради њиховог индексирања у оквиру *"web"* претраживача. Циљ *Crawling*-а јесте прикупљање што већег броја *"web"* страница, заједно са информацијама о њиховој међусобној повезаности, у што краћем временском периоду и на најефикаснији могући начин. Функционалности *Crawlinga* се могу поделити на оне које мора и које би требало да обави.

Да би *"web"* претраживач могао да врати неки документ као резултат упита он га најпре мора пронаћи. Да би нашли информације које се налазе на милијардама *"web"* страна "web" претраживачи користе спајдере.

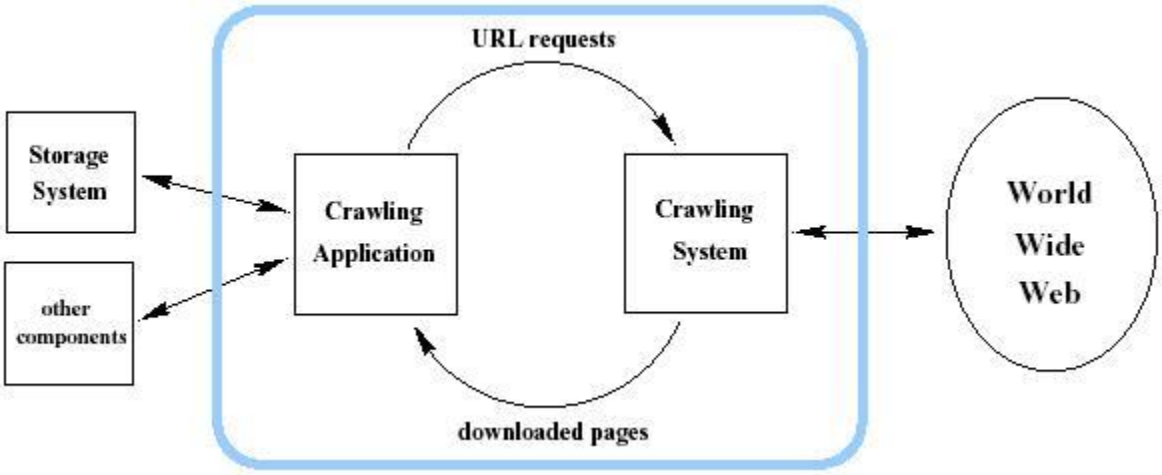
"Web" спајдер (у употреби су још изрази *Web Crawler, Web robot, Web bot*) представља програм или скрипт који аутоматизовано крстари "web"-ом прикупљајући информације о странама. Овај процес прикупљања информација се назива *Veb Crawling* или *spidering*.

Спајдери обично крећу своје крстарење *"web"*-ом са најпопуларнијих сајтова и сервера и даље претећи линкове обилазе све остале странице. У зависности од имплементације самог спајдера они шаљу различите информације матичним серверима. Неки од њих шаљу само УРЛ адресу странице коју посете, други шаљу наслов или садржај тачно одређених ХТМЛ тагова док трећи шаљу целокупан садржај.

Две веома битне карактеристике "web"-а диктирају понашање спајдера и њихов задатак чине веома тешким:

- *Велики број страница*. Ово има за последицу да спајдери могу само да посете делић "web"-а, што значи да тај делић треба да буде посебно одабран.
- *Брзина промене*. Док спајдер посети последњу страницу на сајту, веома је вероватно да су у међувремену неке стране додате, неке обрисане, а неке измењене. Ово је поготово карактеристично за велике сајтове.

Да би спајдер био ефикасан мора имати и изузетно оптимизовану архитектуру као што је приказано на (Сл.19). Веома је лако направити спајдера који ће *download*–овати пар страница у секунди и који ће радити кратко време, међутим направити ефикасног и робусног спајдера који ће скинути на хиљаде милиона страница за неколико недеља представља велики изазов.

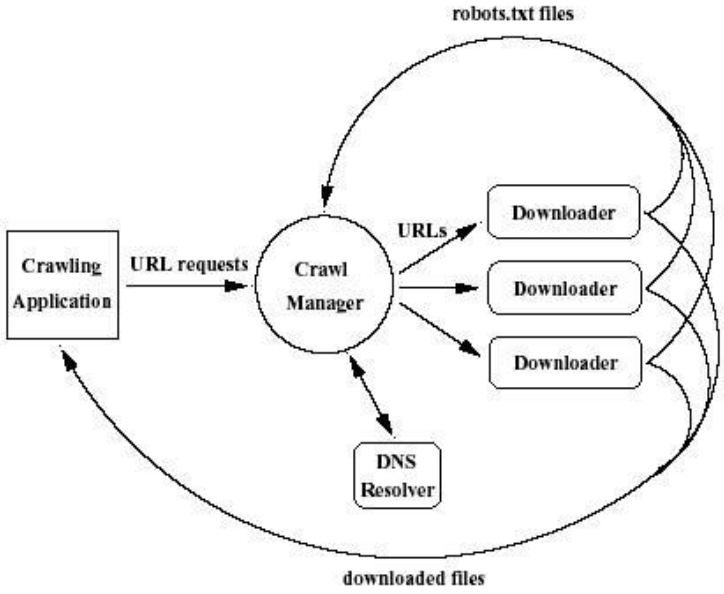


Сл.19.Архитектура спајдера

Shkapenyuk i Suel су у свом раду представили пример архитектуре једног спајдера. Према њима сваки спајдер се састоји од две главне компоненте:

- ***Crawling апликација (eng. Crawling Application)***
- ***Crawling систем (eng. Crawling System)***

Crawling апликација има задатак да донесе одлуку коју следећу адресу (УРЛ) треба *Crawling* систем да посети. Она има могућност да сваку скинуту (*Download*) страницу парсира у потрази за линковима, провери да ли је тај УРЛ већ посећен и уколико није проследи га до Crawling система. Који следећи линк ће бити посећен се одређује на основу неке од бројних стратегија селекције и стратегија поновне посете. Архитектура *Crawling*-а је дата на (Сл.20.).



Сл.20. Архитектура crawling система

Crawling систем има задатак да скине тражену страницу и проследи је *Crawling* апликацији на анализу и складиштење. Он се састоји од неколико специјализованих компоненти.

-Crawling менаџер (eng. Crawl manager) је задужен за примање УРЛ адресе од *Crawling* апликације и њено прослеђивање слободном dowлоадер-у обраћајући пажњу на правила која се могу наћи у *robots.txt* фајлу.

- Једног или више downloader-а. *Downloader* је веома ефикасан асинхрони ХТТП клијент способан да *download*-ује стотине "web" страница паралелно.

-Једног или више система за ДНС разрешавање резолвера (eng. DNS resolvers). Он претвара УРЛ адресу у ИП адресу (уколико тај ДНС већ није кеширан) и на тај начин још додатно убрзава рад *download*-ера.

На основу и уз помоћ свих компоненти *Crawling* мора да обави одређене функционалности:

- Робустност – На "web"-у, постоје сервери који креирању тзв. „*spider traps*", злонамерно генерисане "web" странице које имају за циљ онемогућавање рада *Crawler*-а, тако што настоје да креирају бесконачну петљу унутар домена у којој се *Crawler* може „заглавити". *Crawler*-и морају бити дизајнирани тако да буду отпорни на овај вид „*замки*" (посебно из разлога што могу постојати и грешке овога типа које се не јављају намерно, већ услед грешака).

- Пристојност – "web" сервери имају имплицитне и експлицитне полисе по питању *rate*-а којим их *Crawler* може посећивати. Ове полисе се морају поштовати.

Такође постоје функционалности које би *Crawler* требало да обезбеди :

-Дистрибуираност – *Crawler* би требао да буде у стању да се извршава дистрибуирано на више рачунара.

-Скалабилност – Архитектура *Crawler*-а би требало да подржава повећавање опсега кравл-рате-а додавањем додатних рачунара и *bandwidth*-а

-Перформансе и ефикасност – *Crawl* систем би требао да максимално искористи различите системске ресурсе, као што су процесор, стораге и мрежни bandwidth.

-Квалитет – *Crawler* би требао да буде оријентисан ка *феч*-овању прво "*кориснијих*" страница

-Ажурност резултата – *Crawler* би требао да функционише непрекидно, поновно феч-ујући свеже копије претходно фечованих страница. У принципу, *Crawler* би требао да кравлује страницу са фреквенцијом која апроксимира фреквенцију промене дате странице.

-Проширивост – *Crawler* би требао да буде проширив у више аспеката: да се снађе са новим форматима података, новим феч протоколима и друго, што захтева модуларну архитектуру *crawler*-а.

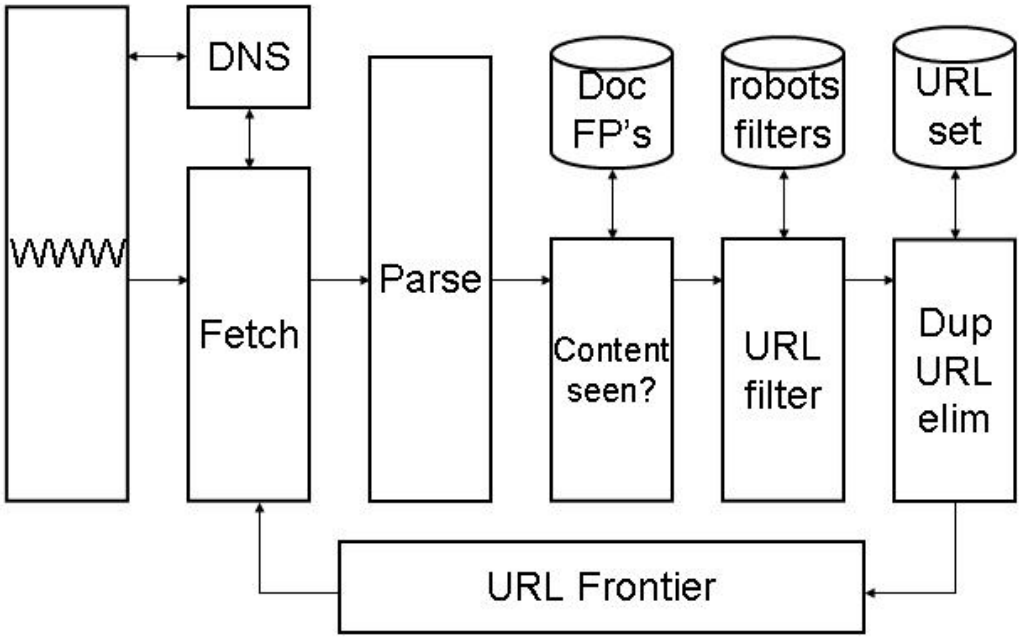
Основана операција сваког *hypertext crawler*-а може се евентуално посматрати и као обилазак "web" графа. *Crawler* почиње скупом од једног или више урл-а (*seed set*), одабира и фечује страницу на том УРЛ-у. Парсира фечовану страницу да би издвојио текст и линкове из странице.

Овако добијени текст се предаје индексеру, док се сви УРЛ-ови додају у УРЛ фронтиер (структура редова за чекање) који је састоји од УРЛ-ова чије странице требају бити феч-ован од стране *Crawler*-а.[10]

Модули који чине *crawler* су:

1. *URL frontier*, који садржи УРЛ-ове који ће бити феч-овани у текућем *crawl*-у.
2. *DNS resolution* модул који одређује адресу "web" сервера на коме се налази УРЛ који феч-ујемо (овај модул се може налазити и ван модула, тј. може се користити системски сервис за ДНС резолуцију).
3. *феч* модул – који враћа страницу на датом УРЛ-у.
4. *parsing* модул – који екстрахује скуп линкова са задате "web" стране.
5. модул који одређује да ли се екстраховани линк већ у УРЛ фронтиер реду или је недавно фечован.

Комплетна структура *Crawler*-а са наведеним модулима приказана је на (Сл.21.).



Сл.21. Структура Crawler-а са наведеним модулима[1]

6.4.Индексирање

Инвертовани индекс, представља основну структуру података која се корисити у оквиру "web" претраживача и *information retrieval* (ИР-област која се бави изучавање метода за проналазак информација у оквиру докумената и ван њих) софтвера уопште.

Под инвертованим индексом, подразумевамо индекс структру која садржи пресикавања између кључних речи и њихових локација у скупу докумената, и коришћењем које се омогућава ефикасно претраживање посматраног скупа.

Постоје две основне варијанте реализације инвертованих индекса: на нивоу записа *(record level inverted index)*, који се још назива *inverted file* индех и који садржи листу референци на документ за сваку реч која се у оквиру њега јавља макар једанпут и на нивоу речи (ворд левел инвертед индех) који додатно у оквиру индекса, садржи и информације о позицији сваког јављања дате речи у оквиру одговарајућег документа.

Код *"web"* претраживача, паралелно са кравл операцијом, обавља се и операција индексирања феч-ованих страница, управо коришћењем структуре инвертованог индекса. За задати корпус докумената, пролази се кроз сваки документ и за сваки токен, врши се његово ажурирање у оквиру индекса:
-уколико већ постоји, додаје се текући документ као локација у којој се налази, док уколико не постоји, креира се нови улаз у индексу, за задати токен и текући документ се поставља за прву локацију у којој се наведени токен налази.

Фокус у пракси јесте процењивање величине сегмента *"web"*-а који индексира један претраживач. Ово се показало као веома тешко за одређивање с обзиром да постоји велики број динамички креираних страница. Због свега тога се уводи питање одређивања величине индекса за два задата претразивача. Одређивање ове величине се показало такође компликованим и то због:

1. Претраживачи могу да врате *"web"* стране чији садржај нису (потпуно или чак ни парцијално) индексирали. У принципу, претраживачи индексирају само првих пар хиљада речи на web страни. У неким случајевима, претраживач је свестан странице на коју линкују странице које он индексира, али сама страница није индексирана. Ово ипак омогућава да претраживач врати смислене резултате у *search results*.

2. Претраживачи у принципу, организују индексе у виду више партиција, од којих се не разматрају све у сваком претраживању. На пример, *"web"* страна, дубоко у оквиру *"web"* сајта може бити индексирана али се не разматра у случају генералног *web search*-а, док се разматра у случају специфицних упита везаних за посматрани *"web"* сајт.

Претраживачи могу садржати вишеструке класе индексираних страна, тако да не постоји јединствена мера њихових целокупних индекса. Као одговор на наведене проблеме, развијен је велики број техника које омогућавају процену *односа* величина индекса два претраживача, Основна хипотеза од које се полази јесте да сваки претраживач индексира онај сегмент *"web"*-а, који се одабира независно и случајно са униформном расподелом.[11]

6.5.Рангирање

Механизми одређивања релевантности страница у оквиру индекса, које одговарају задатом упиту, се показују као изузетно ефикасни у оквиру класичног *information retrieval* домена.

Међутим, у оквиру проблематике *"web"* претраживача, дати методи се често показују као недовољно ефикасни, управо услед специфичности садржаја на *"web"*-у, која се огледа у постојању структуре, индуковане хиперлинковима између страница.

Услед овога, целокупан информациони садржај о страницама на *"web"*-у се не исцрпљује искључиво посматрањем садржаја самих страница, већ захтева и посматрање информација о односима између страница.

6.6.Clustering

Груписање података (*clustering*) представља начин обраде података, којим се у самим подацима откривају тзв. "групе" (*clusters*) података које показују известан степен "*природне блискости*". Под алгоритмима за груписање података, подразумевамо алгоритме који аутоматски откривају групе ("*кластере*") у задатом скупу података. Овакав вид обраде података у пракси има низ примена, посебно у области "истраживања података" (*data mining*).

Два најчешћа приступа проблематици груписања резултата претраживања поклапају се са приступима у оквиру *data mining* методологије и то су *supervised* и *unsupervused learnung.* Код *supervised learning*-а користе се алгоритми машинског учења (*SVM* , Неуронске мреже), који на основу задатог скупа "ручно" класификованих докумената, након процеса "учења" врше класификацију страница на основу њиховог садржаја. Други приступ у оквиру дата мининг методологије представља тзв. *"unsupervised" learning*, који врши класификацију докумената без претходног "знања", користећи само информације прусутне у самим подацима (*k-nearest neighboor*). Ова методологија се често користи у "истраживању података", када не постоји претходно дефинисани корпус података или када нисмо сигурни шта тачно тражимо у подацима.Томе сходно, резултате претраживања можемо посматрати као обичне документе и примењивати стандардне методе машинског учења за њихово груписање (supervised – у односу на задату тему или unsupervised – по сличности докумената).

Проблематика *"web"* претраживача је специфична по томе што сами документи садрже поред садржаја и информацију о међусобној повезаности. Сам *link graph* садржи количину информација која омогућава њихово ефикасно ранкирање јер се *pagerank* алгоритам заснива управо на *link infromation*. Претпоставља се да се *link information* може ефекасно искористити и у циљу груписања *search* резултата.[11]

7. Методе претраживања

Скуп свих могућих решења проблема назива се простор претраживања (*eng. search space*). У систему постоји почетно стање из којег систем креће, проласком кроз одређени број стања долази у коначно стање које је решење проблема. Чест је случај да скуп стања између почетног и коначног стања није једнозначно одређен, него постоји више различитих, често и различито ефикасних путева од почетног до крајњег стања. За рад таквих система потребно је уградити алгоритме за тражење споменутих путева. Због непостојања формалних поступака који би на темељу почетног и крајњег стања система одређивали скуп стања на путу између њих потребно применити најпримитивнију, али зато и најмање ефикасну методу, претраживања простора стања. Претраживање простора стања је поступак који пролази кроз стања у којима би се могао налазити систем, и упоређивањем тренутног стања с циљним стањем утврђује да ли је поступак дошао до краја.[14]

Постоје две основне класе алгоритама за претраживање:

- ***Blind search or uninformed search*** (неинформисано претраживање)
- ***Heuristic or informed search*** (информисано претраживање)

Неинформисано претраживање је претраживање код којег нема никаквих информација о броју корака или вредности путање (вредност удаљености чвора од почетног чвора) од почетног до крајњег стања, тј. циља. У ову групу претраживања спадају:

− претраживања по дубини (*eng. Depth-first search*)
− претраживања по ширини (*eng. Breadth-first search*)
− претраживања с једнаком ценом (*eng. Uniform-cost search*)
− претраживање до одређене дубине (*eng. Depth-limiting search*)
− итеративно претраживање по дубини (*eng. Iterative deeping search*)
− двосмерно претраживање (*eng. Bidirectional search*)

Информисано претраживање је претраживање које има додатне информације о циљу, цену путање или броју корака. Те информације чине ова претраживања бољим од неинформисаних па им омогућују готово рационално понашање. У ова претраживања спадају:

− претраживање најбољим првим (*eng. Best first search*)
− претраживање пењањем (*eng. Hill-climbing search*)
− A* претраживање (*eng. A* search*)
− ограничено претраживање по ширини (*eng. Beam search*)
− ИДА* претраживање (*eng. Iterative deeping A* search*)[10]

7.1.Претраживачи сличности и разлике

Задатак претраге је да што ефикасније врати што квалитетнији резултат. Ефикасност се мери брзином која протекне од клика на дугме до појаве резултата, а квалитет се мери са бројем резултата (страница) које претраживач врати и њиховом релевантношћу. За брзину је задужен процес индексирања и алгоритми који сортирају податке приликом парсирања. Број резултата зависи од броја индексираних страница, односно броја страница које *"web"* спајдер претраживача посети и преузме у базу. Релевантност представља повезаности садржаја сајта са траженим појмом и зависи од квалитета алгоритама које претраживач користи.[14]

Након добијања идентификатора за речи које се налазе у фрази извршава се упит над *inverted index* табелом који враћа све документе који их садрже. Овде се процес претраге завршава за оне претраживаче који се не баве релевантношћу резултата. За све остале следећи корак је управо оно што разликује добре од лоших – ранкирање резултата.

Ранкирање резултата представља процес сортирања на основу релевантности. Најпре се за сваки од докумената који садржи тражене речи броји број појављивања фразе на истом. Затим за свако појављивање речи гледа се позиција у документу (наслов, *anchor, URL, H1,H2,H3 tag…*) и да ли је та реч посебно наглашена (подебљана, закривљена, подвучена). Свака од позиција и наглашавање речи носи одређени тежински фактор. Сабирањем тежинских фактора свих појављивања добија се само делић вредности по којој се после документи сортирају. Остали фактори релевантности зависе од самог "web" претраживача.

Након завршеног сортирања докумената (свих оних који садрже унету фразу) по релевантности "web" претраживач корисника пребацује на нову страницу на којој се налазе излистани сортирани линкови ка документима.

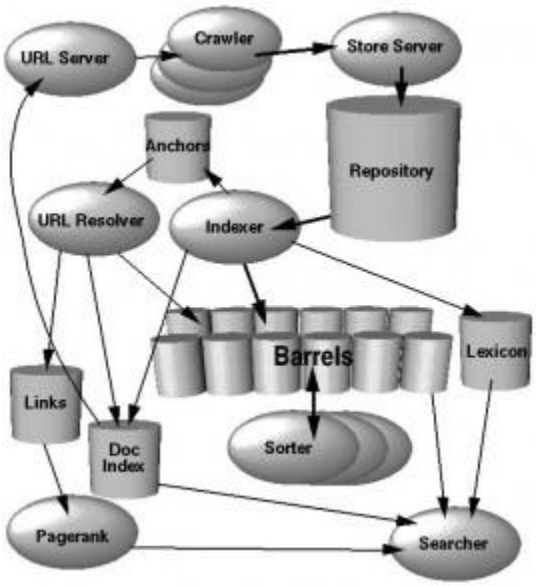
"Web" претраживачи и *"web"* каталози врло су популарне услужне *"web"* странице које скупљају информације о другим, постојећим *"web"* страницама. Корисници их користе да би пронашли жељене информације, компаније, производе и услуге. Притом користе кључне речи које најбоље описују тражене информације или, у случају каталога, претражују категорије организиране према делатности (као у жутим страницама). Неки од најпознатијих *"web"* претраживача су: *Google, Live Search (MSN), Yahoo, Altavista*. Неки од њих су описани у наставку. Осим разних генеричких претраживача постоје и специјализовани претраживачи за одређена подручја или врсте садржаја, разни директоријуми "web" страница од којих неке, поред рачунара, уређују и људи.[16]

7.2.Google

За високо рангирање на претраживачу *Google*-у најважнији су следећи фактори:

1) улазни линкови
2) старост
3) садржај
4) успех на резултатима претраге

На (Сл.22.) је дата архитекрура *Google* претраживача



Сл.22. Архитектура Google система[8]

Улазни линкови – линкови који са других страница воде на "web" средиште. Некада је био битан само број тих линкова тј. што је већи број линкова, виша је позиција. Данас је, међутим, знатно другачија.

Сваки линк добија одређену тежину према следећим параметрима:

- ***Старост линка*** – што је линк старији, тежина му се повећава. Већину тежине линкови постижу тек за неких 5 до 6 месеци од објављивања на Интернету.

- ***Локација линка*** – позиција линка на страници одређује његову тежину. Већу тежину ће имати линкови при врху странице јер је већа вероватноћа да ће их посетиоци видети. Такође, битан је и садржај који се налази око линка. Ако је линк унутар садржаја, имаће већу тежину него ако је само одвојено наведен на страници.

- ***Назив линкова и њихово форматирање*** – коришћење одређених кључних речи у називу линка даће додатну релевантност "web" средишта са тим кључним речима.

Ако је линк истакнутији, добиће већу тежину па је због тога важно да ли је у његовом форматирању употребљено задебљање и/или *italic* слова.

- ***Релевантност*** – релевантност странице која води на "web" средиште је од кључне важности. Што је та страница сличнија кључним речима које описују "web" средиште, линк више добија на тежини.

- ***PageRank*** – што је већи *PageRank* странице са линком на "web" средиште, линк има већу тежину.

- ***Старост*** – 2004. године Google уводи архиву имена домена. По томе може видети старост домена и остале важне податке везане уз неки домен. Што је домен старији, то је већа вероватноћа да ће имати већи ранг.

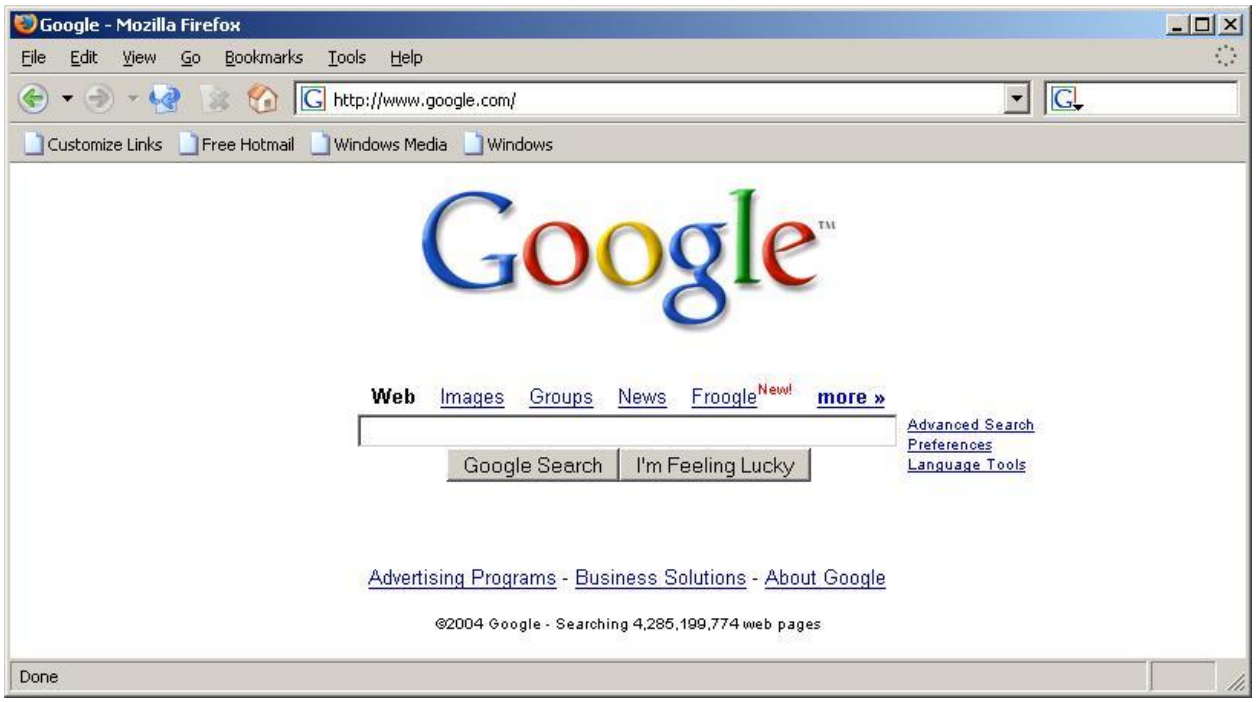
- ***Садржај*** – у општем случају, што више садржаја има "web" страна, то је већа вероватноћа да ће посетиоц наћи оно што је тражио. Код тог садржаја битна је и густина кључних речи која мора бити оптимална, а није добро претерати са кључним речима и занемарити остатак садржаја. Један од бољих начина за додавање већих количина релевантног садржаја је писање блога који је тематски повезан са "web" страном.

- ***Успех на резултатима претраге*** – параметар који показује колико често корисници кликну на "web" страну ако се појави као резултат претраге за одређене кључне речи. Наравно, ако је посетиоц кликнуо на "web" страну, али се након кратког времена вратио назад на резултате, што значи да није пронашао оно што је тражио то ће довести до смештања "web" стране ниже у рангу. Овај параметар „приморава" "web"мастере да направе функционалне и пре свега *user-friendly* "web" стране. Google има најкомплекснији алгоритам од три најпосећенија претраживача и због тога покушаји „варања" могу функционисати само у кратком временском периоду. За успех на Googley важно је направити квалитетну, *user-friendly* "web" страну са добром навигацијом, оптимизовати густину кључних речи и осигурати квалитетне линкове до "web" стране.

Појам *„Google"* базира се на игри речи (неки извори тврде да се ради о правописној грешци) код америчког изговора речи *googol. Milton Sirotta* (нећак америчког математичара *Edwarda Kasnera*) наведени је појам измислио 1983. године, како би броју са јединицом и сто нула дао назив. Оснивачи *Google*-а слично томе били су у потрази за појмом који би означио суперлативну количину информација коју би њихов претраживач требао пронаћи.

„Google" је назив познатог интернет претраживача који моментално доминира информатичким тржиштем. Пробна верзија претраживача *„Google"* покренута је 7. септембра 1998.

Изглед Googlea се од тада практично није променио. Захваљујући успеху претраживача, компанији *Google Inc*. омогућено је финансирање даљњих програма; приступ њима омогућиће се преко службених страница *Google*-а. Успркос свему, језгро *Google*-а је интернет претраживач (слика 23). Такође је вредно напоменути да конкурент *Yahoo* повремено користи *Google*-ове базе података у властите потребе.



Слика 23. „Google“ претраживач

Google је као претраживач постао толико популаран да су одређени правописи почели користити појам „гооглати“. Наведени глагол више не представља само претраживање помоћу претраживача *Google*, већ је синоним за претрагу интернета било којим претраживачем. Од августа 2006. *Google* жели да правно заштити појам „гооглати“, тако да би се наведени појам смео користити само у вези са интеренет претраживачем *„Google“*. Google жели да спречи развој догађаја који се одвијао са појмовима као што су нпр. *Tempo, Nutella, Jeep, Linoleum, Aspirin, Gramofon*, итд. – наведени појмови пре су представљали заштићене логотипе, а данас су синоними за разне производе.[17]

7.3.Yahoo

За високо рангирање на претраживачу *Yahoo* najvažniji су следећи фактори:

1) густина кључних речи
2) структура “web” странице
3) улазни линкови
4) старост

Густина кључних речи – не постоји оптимална густина кључних речи за сва подручја већ се она дефинише кроз време и с обзиром на врсту и подручје “web” странице. Најбољи приступ је анализирати густину на неколико прво-рангираних “web” страница па према томе прилагодити густину на властитим страницама.

Структура “web” странице – Код *Yahoo*-а је овај параметар веома важан док је за *Google* и *MSN* практично занемарив. Садржај који се налази при врху кода “web” странице има већу тежину од садржаја који се налази на дну због тога што је већа вероватноћа да ће одговарати посетиоцу.

Ово се може оптимизовати коришћењем ЦСС-а (*Cascadung Style Sheets*) и осталих метода које доводе до смањења кода и самим тим довођења садржаја према врху.

Улазни линкови –На основу овог параметра може се закључити да је *Yahoo* је сличнији *Google*-у него *MSN*-у. Параметри које узима у обзир су:

- ✓ ***квалитет “web” странице*** – нешто слично као и Googleов *PageRank, Yahoo* води рачуна о квалитету линкова, а не само о њиховом броју. Што је линк квалитетнији, то је већа његова тежина.

- ✓ ***локација линка*** – линк при врху странице има већу тежину од линка на дну странице. Такође ако се линк налази на страници са пуно линкова тежина му се пропорционално смањује са укупним бројем линкова јер је мања вероватноћа да ће посетиоц одабрати баш тај линк који води на одређену “web” страну.

- ✓ ***текст линка*** – важно је да текст линка садржи кључне речи везане уз страницу на коју води. Такође, ако је линк унутар садржаја, имаће знатно већу тежину у односу на линк који се налази са стране.

- ✓ ***линкови који нису реципрочни*** – реципрочни линкови су и даље важни *Yahoo*-у, али је важно имати и линкове који су једносмерни.

Старост – нове “web” странице и линкови на њима немају исту тежину као старије странице и старији линкови у оквиру њих. Иако је критеријум *Yahoo*-а по овом питању блажи од Googlea ипак ће новим линковима ће требати ти до четири месеца да постигну пуну тежину.[18]

7.4.MSN

За високу позицију на MSN-у (*Microsoft Network*) важни су следећи фактори:

1) садржај странице
2) структура унутрашњег повезивања
3) број страница и релевантност
4) наслови, заглавља и посебни формати

Садржај странице – за високи положај на MSN-у важна је добра прилагођеност садржаја алгоритмима претраживача, исто тако садржај мора бити добро прилагођен и самим посетиоцима "web" странице. Поред тога важан је и проценат присуства кључних речи. Густина кључних речи се мења са сваком надоградњом и побољшањем алгоритама, али најчешће се налази између 3,5 и 4 посто.

Структура унутрашњег повезивања – начин на који су странице интерно повезане пуно говори о садржају тако да је то један од важнијих фактора. Ако је навигација базирана на сликама или скриптама, важно је додати и обичне текстуалне линкове јер њих претраживачи лакше прате, а и сам текст линка говори о чему се ради на страници до које линк води. Самим тим могуће је ефикасно употребити кључне речи тамо где су најважније.

Број страница и релевантност – MSN настоји да осигура да посетиоци нађу тачно оно што им је потребно кад кликну на неки резултат претраживања. Због тога веће "web" странице које се баве једном темом заузимају више позиције од мањих које обрађују различите теме. За постизање жељене величине, добро решење може бити увођење блога или форума.

Наслови, заглавља и посебни формати – наслов је најважнији део кода сваке "web" странице.

- ✓ Као прво, он има јако велику тежину код алгоритама претраживача.
- ✓ Као друго, код приказа резултата, претраживач најчешће приказује наслов "web" странице као линк према њему - квалитетан наслов мора привући посетиоца јер у супротном високо рангирање не доноси жељене резултате.

Заглавља дефинишу велики део текста тако да и она носе велику тежину. Ипак, с њима не треба претеривати, оптимално је имати једно заглавље по страници. Посебни формати се односе на форматирање текста тако да се тај текст истиче у односу на мање битан садржај. То се може постићи коришћењем различитих боја, подебљаног текста, као и уметањем линкова на други садржај на истој страници у сам текст.[19]

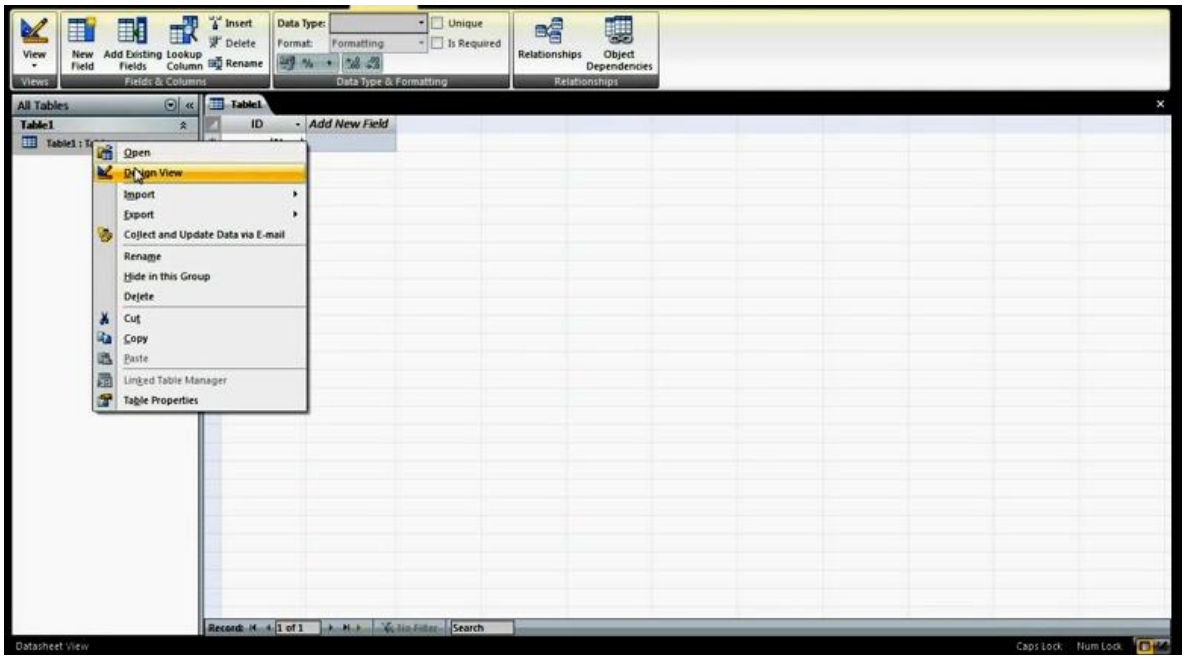
8. Студијски пример

8.1.Пример складиштења података

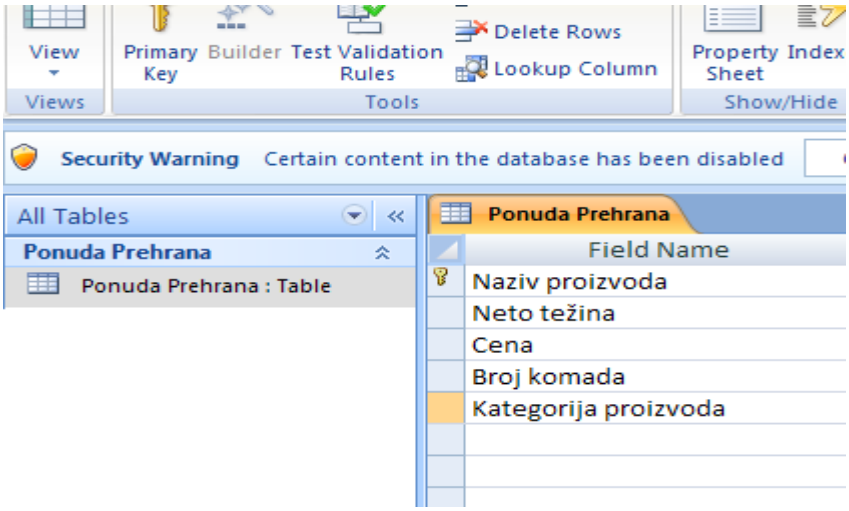
Израда базе података и употреба основних наредби (табеле, форме, извештаји, упити, полазни образац..). Као пример, узета је база података за виртуелно предузеће „Прехрана“.

Креирање складишта података, нова база података:

Слика 24. Бирање фолдера и уношење назива“Прехрана“

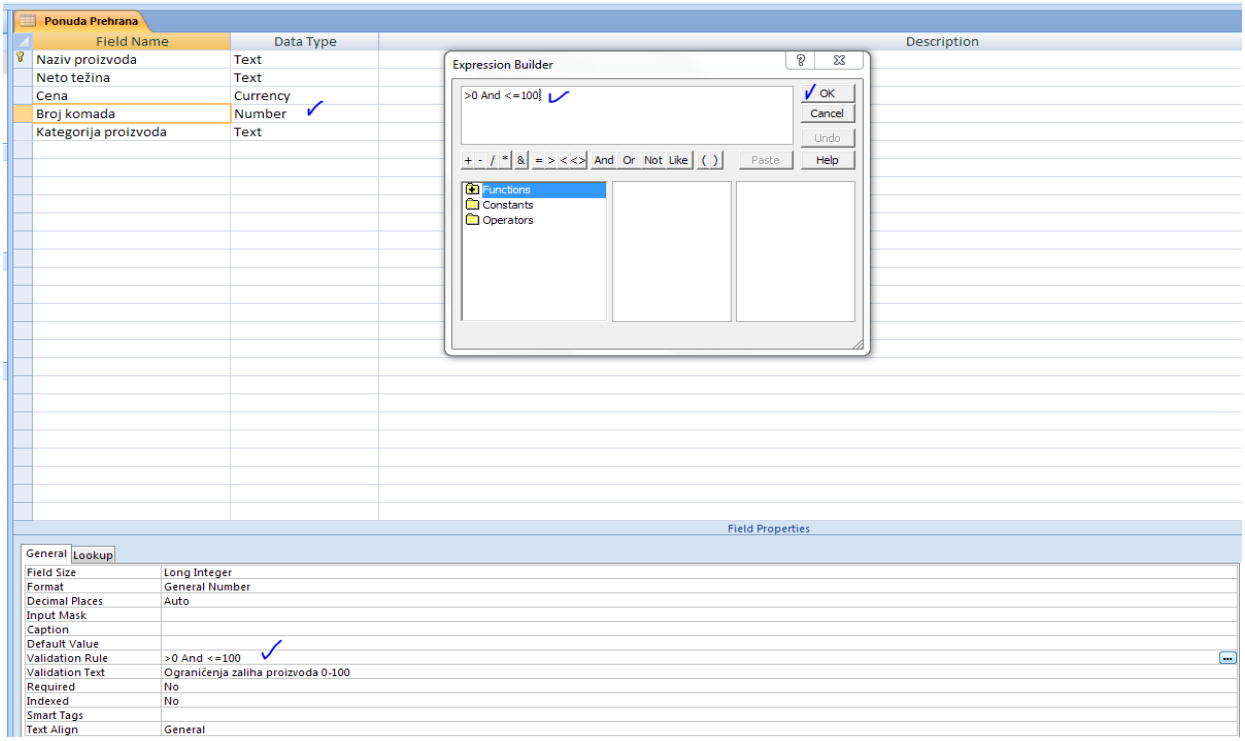


Слика 25. Почетак израде једноставног складиштења података



Слика 26.Уношење основних података у „Понуда Прехрана“

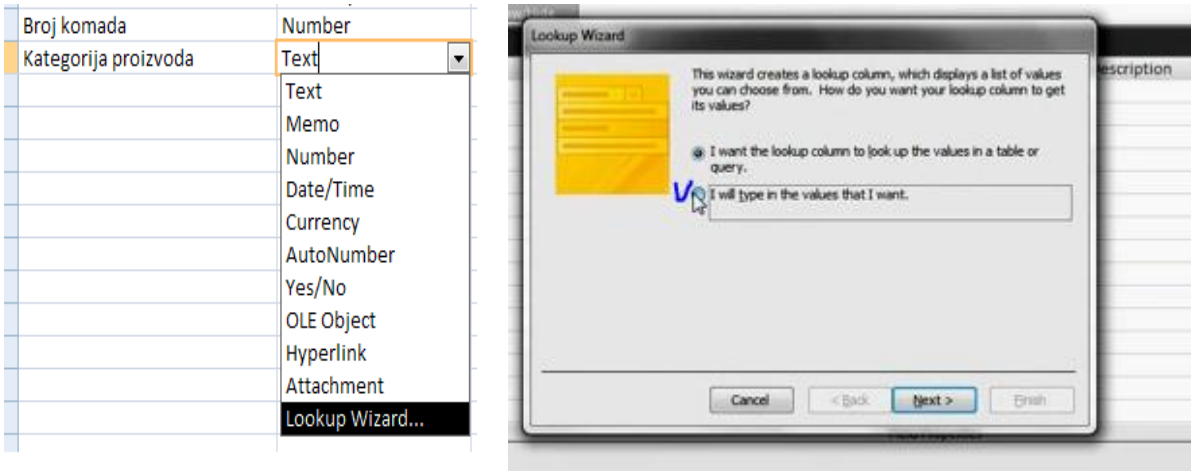
Након уношења основних података у табелу „Понуда Прехрана“ (слика 26.), приступа се категорисању основних података.



Слика 27. Уношење података кроз „Data type Expression Builder“

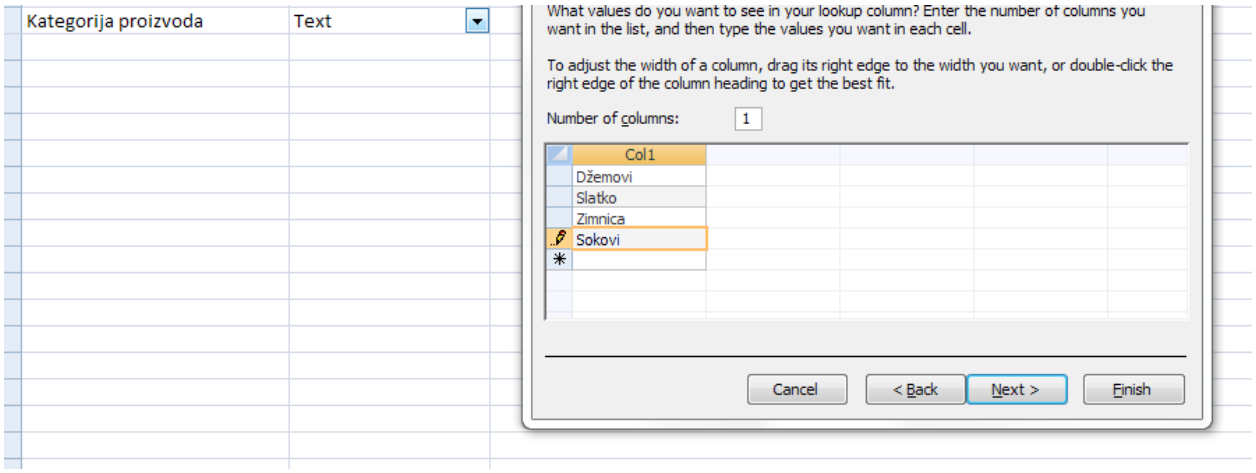
Приликом формирања „Броја комада“, кроз *„Data type“* се бира *„Number“*, који садржи додатне опције попут *„Validation Rule“* и *„Validation Text“*. *Validation Rule* је опција која, грубо написано, означава валидне границе при уношењу података и која садржи *“Expression Builder“*, као што је приказано на слици 27., док *„Validation Text“* носи име саме те границе.

У *„Expression Builder“* се уноси, као што је на слици приказано, почетна и крајња граница „Броја комада“ и тај број мора бити у том периоду, у супротном, систем ће пријављивати грешку,исти поступак је урађен и при формирању табеле добављача (слика 32а).

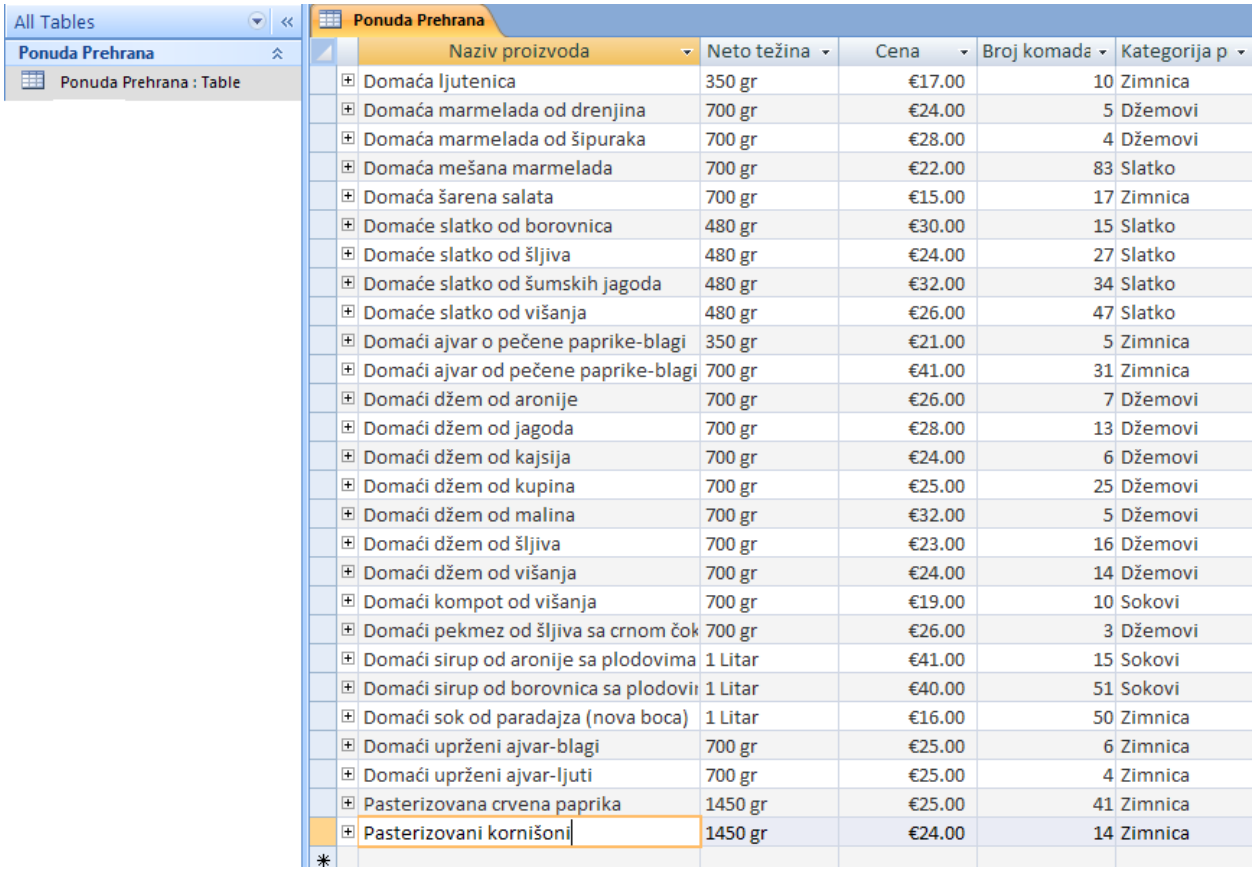


Слика 28. Формирање колона категорије производа

Приликом уношења категорије производа у табелу, програм пружа могућност формирања одређених колона вредности које су нам потребне, што су у овом случају категорије производа (слика 28 и слика 29).



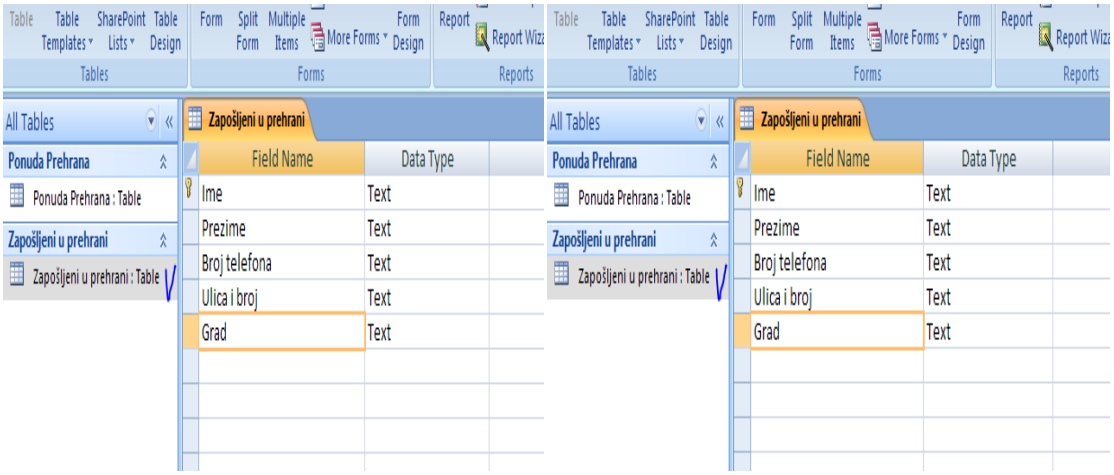
Слика 29. Табела са колонама категорије производа



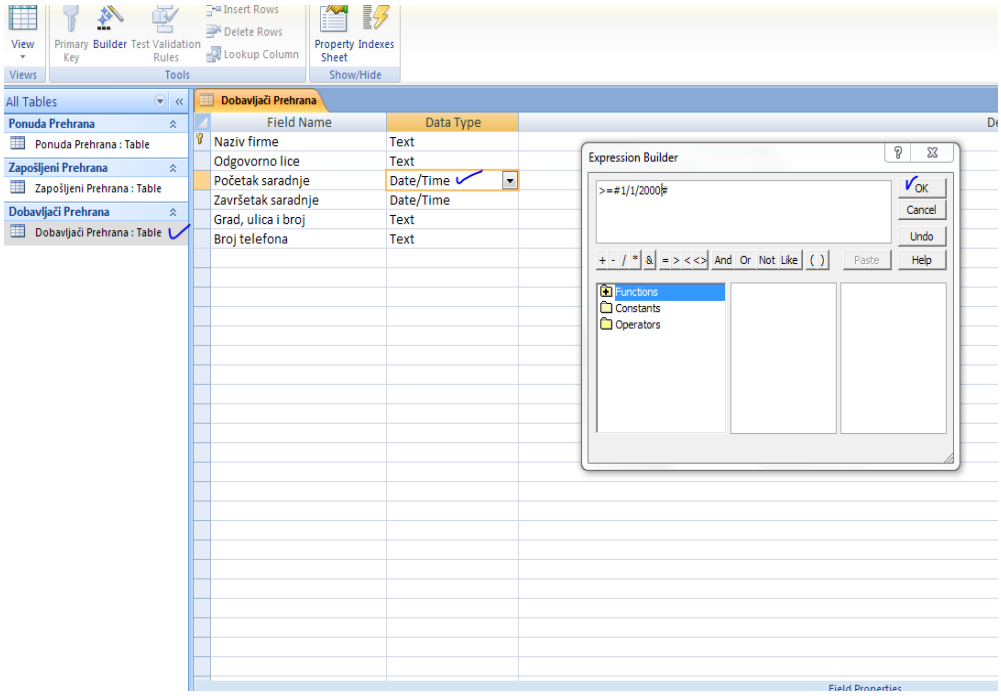
Naziv proizvoda	Neto težina	Cena	Broj komada	Kategorija p
Domaća ljutenica	350 gr	€17.00	10	Zimnica
Domaća marmelada od drenjina	700 gr	€24.00	5	Džemovi
Domaća marmelada od šipuraka	700 gr	€28.00	4	Džemovi
Domaća mešana marmelada	700 gr	€22.00	83	Slatko
Domaća šarena salata	700 gr	€15.00	17	Zimnica
Domaće slatko od borovnica	480 gr	€30.00	15	Slatko
Domaće slatko od šljiva	480 gr	€24.00	27	Slatko
Domaće slatko od šumskih jagoda	480 gr	€32.00	34	Slatko
Domaće slatko od višanja	480 gr	€26.00	47	Slatko
Domaći ajvar o pečene paprike-blagi	350 gr	€21.00	5	Zimnica
Domaći ajvar od pečene paprike-blagi	700 gr	€41.00	31	Zimnica
Domaći džem od aronije	700 gr	€26.00	7	Džemovi
Domaći džem od jagoda	700 gr	€28.00	13	Džemovi
Domaći džem od kajsija	700 gr	€24.00	6	Džemovi
Domaći džem od kupina	700 gr	€25.00	25	Džemovi
Domaći džem od malina	700 gr	€32.00	5	Džemovi
Domaći džem od šljiva	700 gr	€23.00	16	Džemovi
Domaći džem od višanja	700 gr	€24.00	14	Džemovi
Domaći kompot od višanja	700 gr	€19.00	10	Sokovi
Domaći pekmez od šljiva sa crnom čok	700 gr	€26.00	3	Džemovi
Domaći sirup od aronije sa plodovima	1 Litar	€41.00	15	Sokovi
Domaći sirup od borovnica sa plodovi	1 Litar	€40.00	51	Sokovi
Domaći sok od paradajza (nova boca)	1 Litar	€16.00	50	Zimnica
Domaći uprženi ajvar-blagi	700 gr	€25.00	6	Zimnica
Domaći uprženi ajvar-ljuti	700 gr	€25.00	4	Zimnica
Pasterizovana crvena paprika	1450 gr	€25.00	41	Zimnica
Pasterizovani kornišoni	1450 gr	€24.00	14	Zimnica

Слика 30. Листа унетих производа уз нето тежину, цену, број комада и категорију

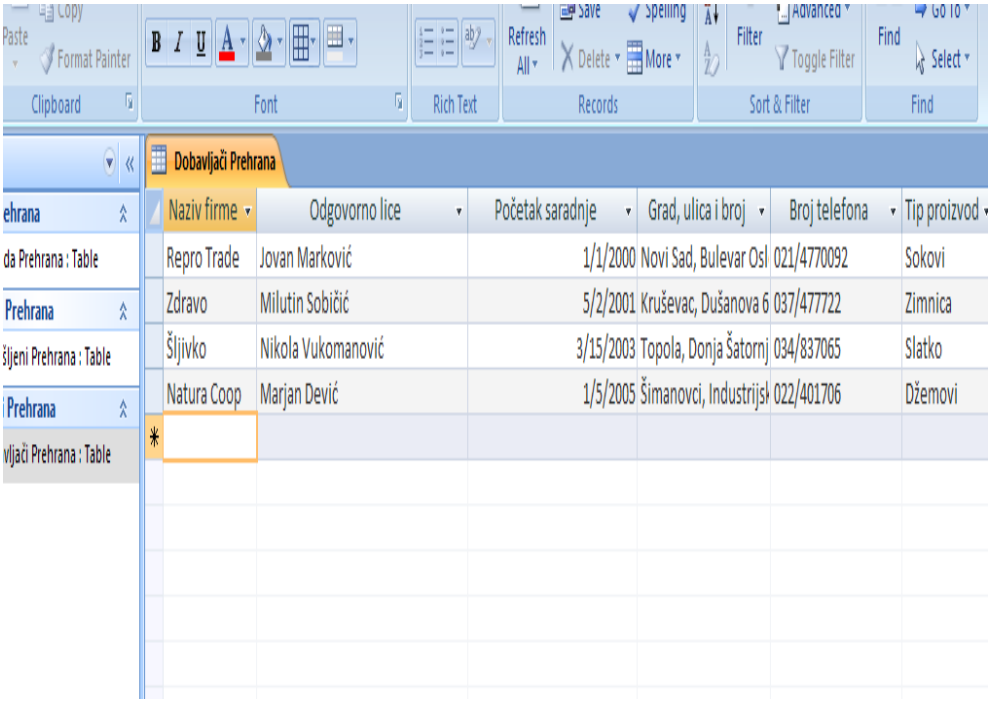
Након броја комада и категорије производа уношење цене у табелу „Понуде“, остварује се путем бирања *„Currency“* из падајућег менија *„Data type“* и том приликом се бира валидна валута, односно формат цене у коме је потребно изразити цене производа (слика 29). Након уношења основних података и самих производа, поступак се, затим, темељи на уношењу табеле запослених (слика 31) и табеле добављача (слика 32), поступак је апсолутно исти као и при уношењу „Понуда“.



Слика 31. Формирање табеле запослених у прехрани



Слика 32а. Формирање табеле добављача у прехрани



Слика 32б. Формирање табеле добављача у прехрани

Приликом израде извештаја (слика 33), приступа се форми тако што из менија *„Create“*, кликом на табелу понуда, бира опција *„Form“*, па из листа (прозора), десним кликом иде на *„Design view“* и кроз опцију *„Button“* се формирају командне излазне форме (*Form Operations/Close form*).

Тиме се формирају „Понуде Прехране “, ради израде самог *“Report”*-а, исти поступак се примењује при изради извештаја „Запослени Прехрана“ и „Добављачи Прехрана“.

Ponuda Prehrana

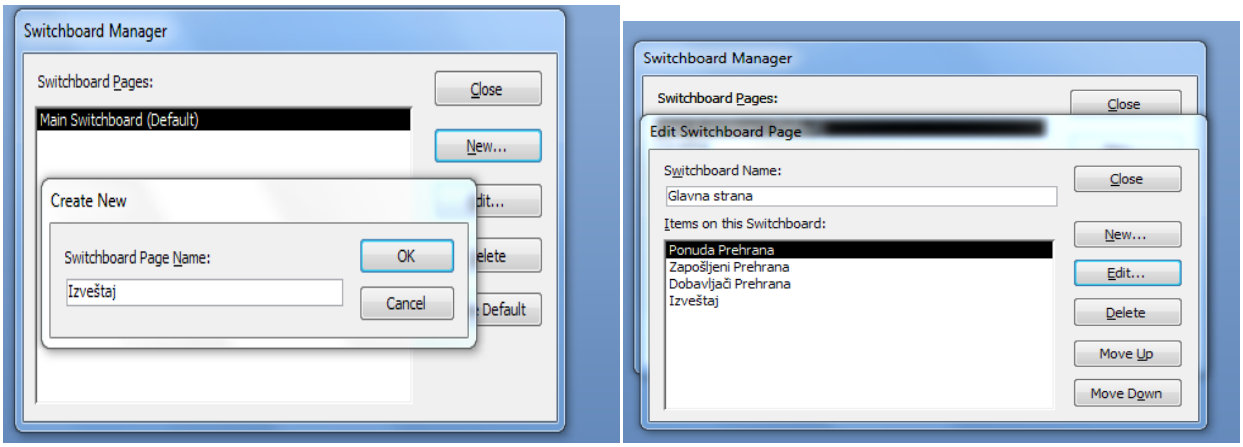
Thursday, July 03, 2014
11:24:18 AM

Naziv proizvoda	Neto težina	Cena	Broj komada	Kategorija proizvoda
Domaća ljutenica	350 gr	€17.00	10	Zimnica
Domaća marmelada od drenjina	700 gr	€24.00	5	Džemovi
Domaća marmelada od šipuraka	700 gr	€28.00	4	Džemovi
Domaća mešana marmelada	700 gr	€22.00	83	Slatko
Domaća šarena salata	700 gr	€15.00	17	Zimnica
Domaće slatko od borovnica	480 gr	€30.00	15	Slatko
Domaće slatko od šljiva	480 gr	€24.00	27	Slatko
Domaće slatko od šumskih jagoda	480 gr	€32.00	34	Slatko
Domaće slatko od višanja	480 gr	€26.00	47	Slatko
Domaći ajvar o pečene paprike-blagi	350 gr	€21.00	5	Zimnica
Domaći ajvar od pečene paprike-blagi	700 gr	€41.00	31	Zimnica
Domaći džem od aronije	700 gr	€26.00	7	Džemovi
Domaći džem od jagoda	700 gr	€28.00	13	Džemovi
Domaći džem od kajsija	700 gr	€24.00	6	Džemovi
Domaći džem od kupina	700 gr	€25.00	25	Džemovi
Domaći džem od malina	700 gr	€32.00	5	Džemovi
Domaći džem od šljiva	700 gr	€23.00	16	Džemovi
Domaći džem od višanja	700 gr	€24.00	14	Džemovi
Domaći kompot od višanja	700 gr	€19.00	10	Sokovi
Domaći pekmez od šljiva sa crnom čokoladom	700 gr	€26.00	3	Džemovi
Domaći sirup od aronije sa plodovima aronije	1 Litar	€41.00	15	Sokovi
Domaći sirup od borovnica sa plodovima borovnice	1 Litar	€40.00	51	Sokovi
Domaći sok od paradajza (nova boca)	1 Litar	€16.00	50	Zimnica

Слика 33. Формиран извештај понуда прехране

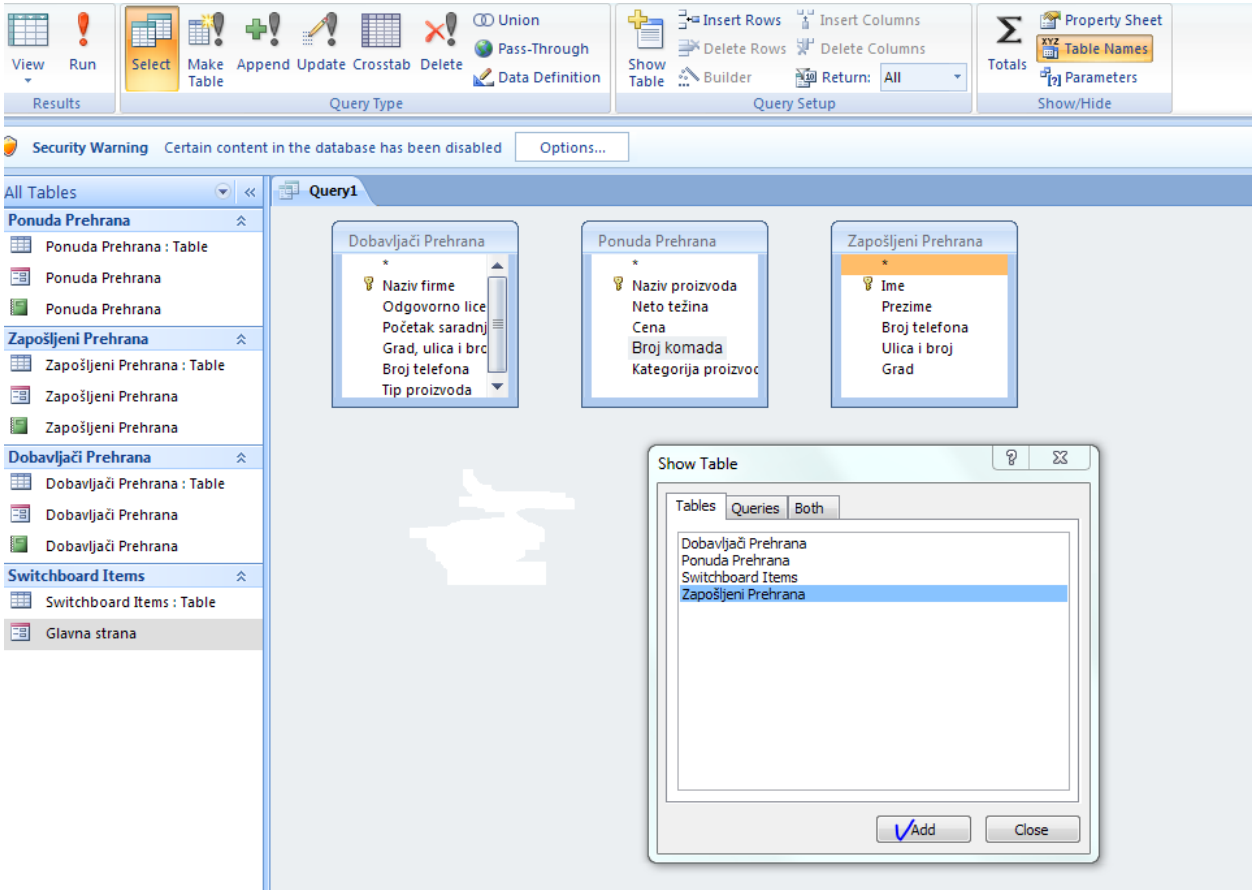
Након успешног формирања извештаја, из менија алата базе података (*Database tools*), бира се *„Switchboard manager“*, у коме се формирају-постављају почетна (главна) страна, називи тих страна, као и подстрана и њени називи (слика 34 и слика 35).

Слика 34. Приступ алатима базе података

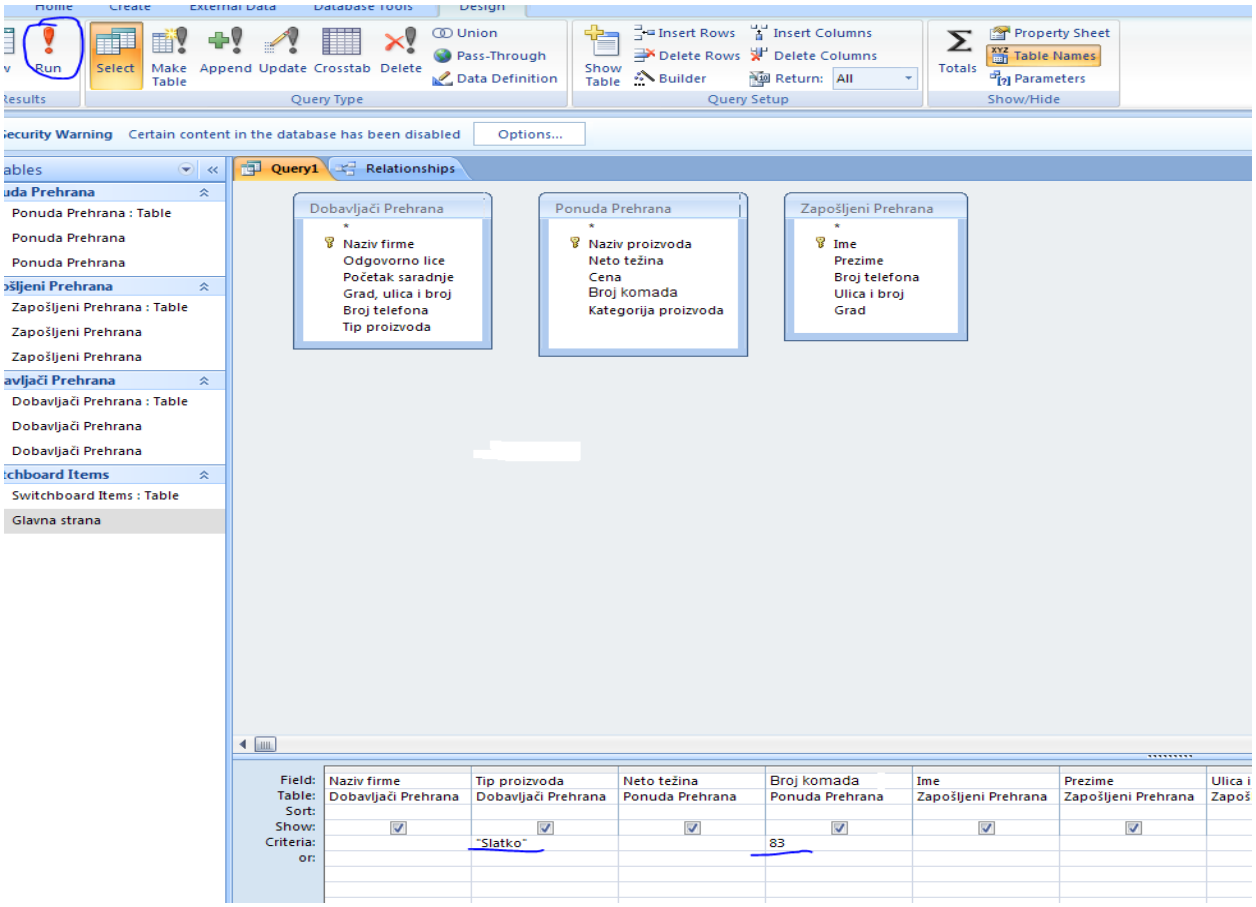


Слика 35.Формирање главне стране и подстране базе података

Из сређене почетне стране се приступи *„Query setup“*-у и из падајућег менија изабере *„Query design“* у ком су смештене све табеле и упити, који се уносе простим додавањем њих самих (слика 36).

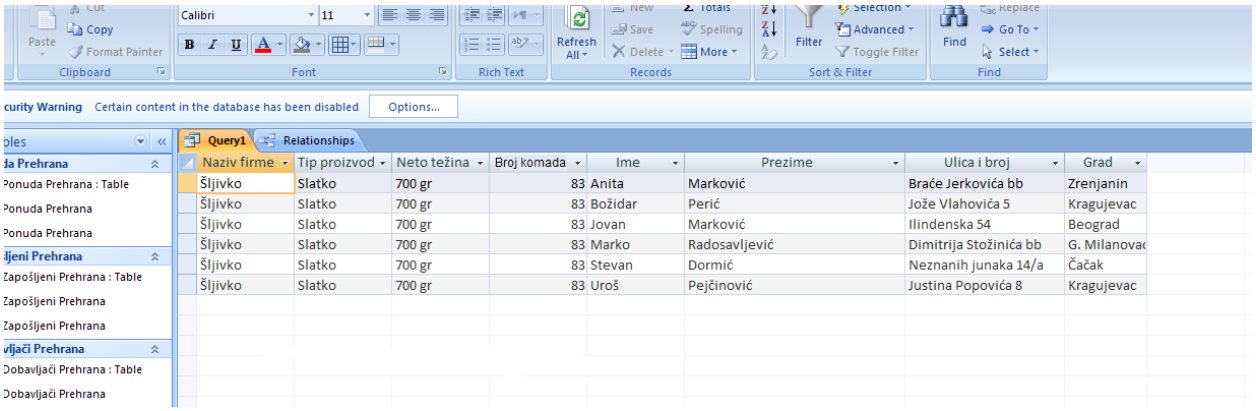


Слика 36. Додавање упита

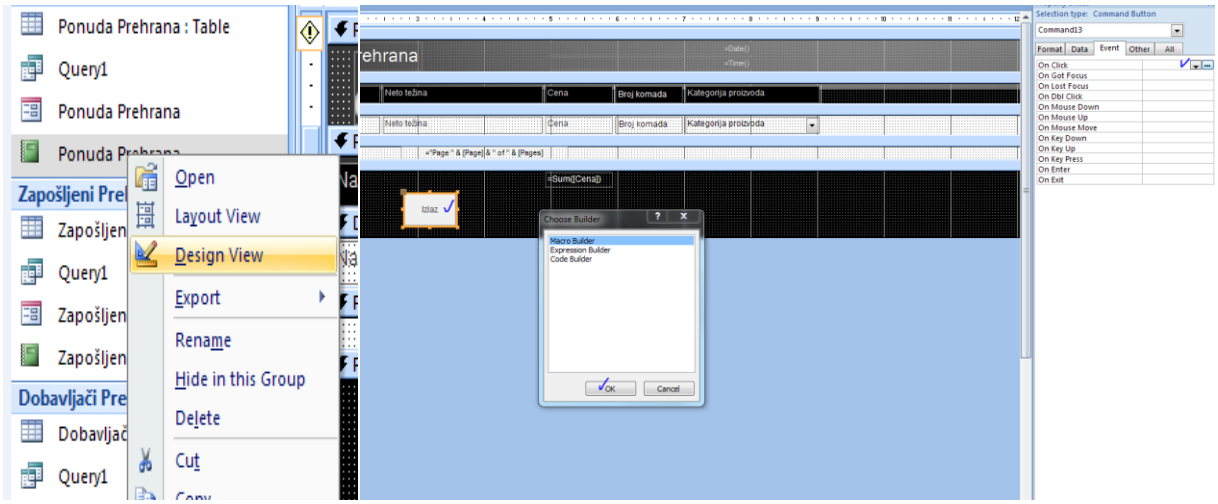


Слика 37. Пример уношења типа производа и броја комада, покушај проналажења датотеке

Након уношења табеле, као што је дато на слици 37., сортирају се назив фирме, тип производа, нето тежина, број комада, име и презиме.., након чега се, приликом бирања критеријума типа производа и броја комада, жели наћи одређени производ, кликом на опцију *„Run“*.

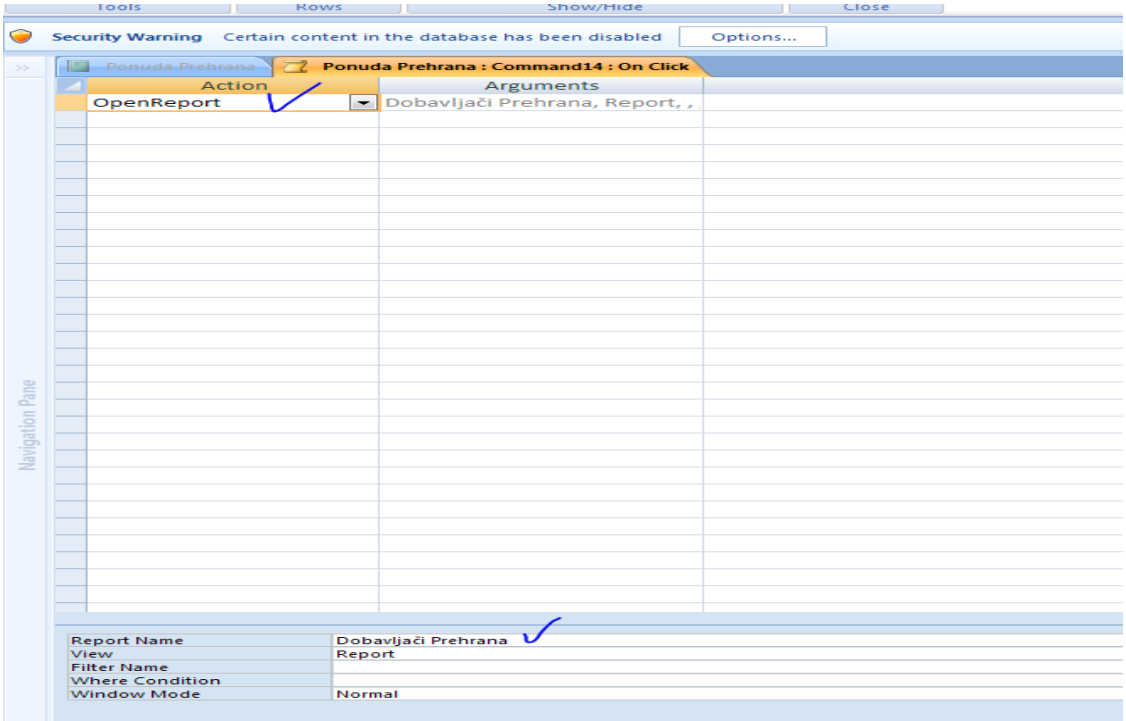


Слика 38. Критеријум задовољен

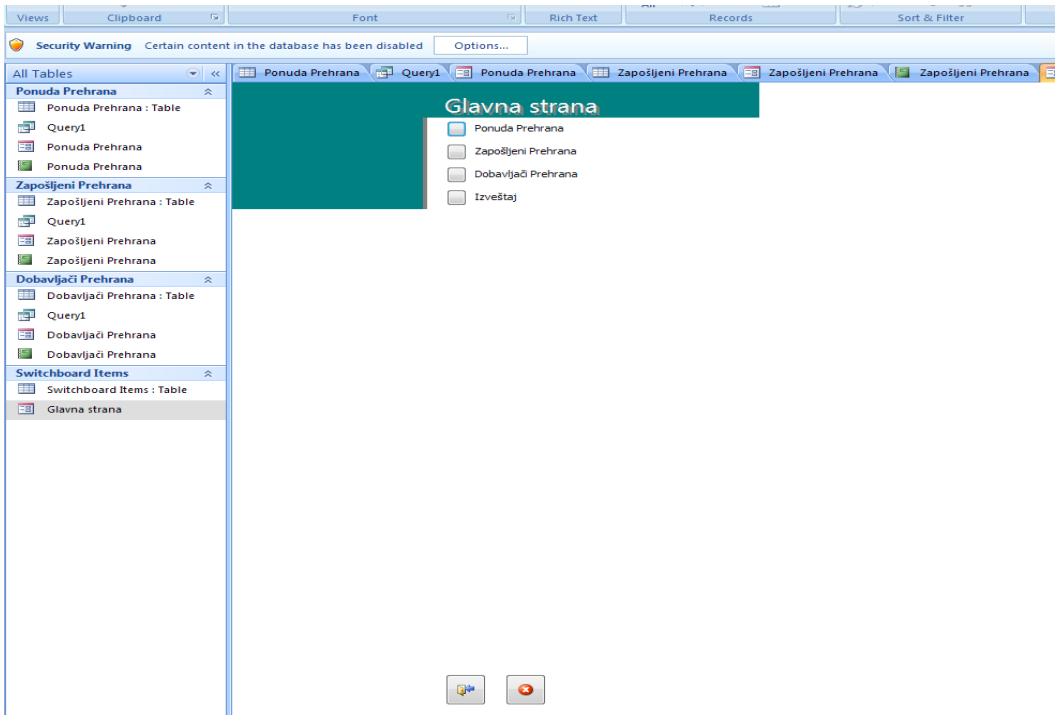


Слика 39. Формирање узајамне везе уз помоћ „Macro Builder“

Из стабла формираних табела, извештаја и упита, се десним кликом на извештај понуда прехране бира *„Desing view“*, па се уз помоћ опције *„Button“* формира *„Comand Button“* и као на слици, са десне стране, изабере прозор *„Event“* и из падајућег менија *„On Click“*-ом се бира *„Builder“ (Micro Builder)*, (слика 39) притиском на ОК, отвориће се прозор команди чланова у ком је потребно изабрати акцију отварања извештаја *(Open report)* и именовати извештај у „*Понуда Прехрана*“ (слика 40). Исти поступак важи и за извештај запослених и извештај добављача. Тиме је постигнуто да понуда, запослени и добављачи буду у међусобној повезаности приликом отварања извештаја.



Слика 40. Именовање и акција отварања извештаја „Понуда Прехрана“



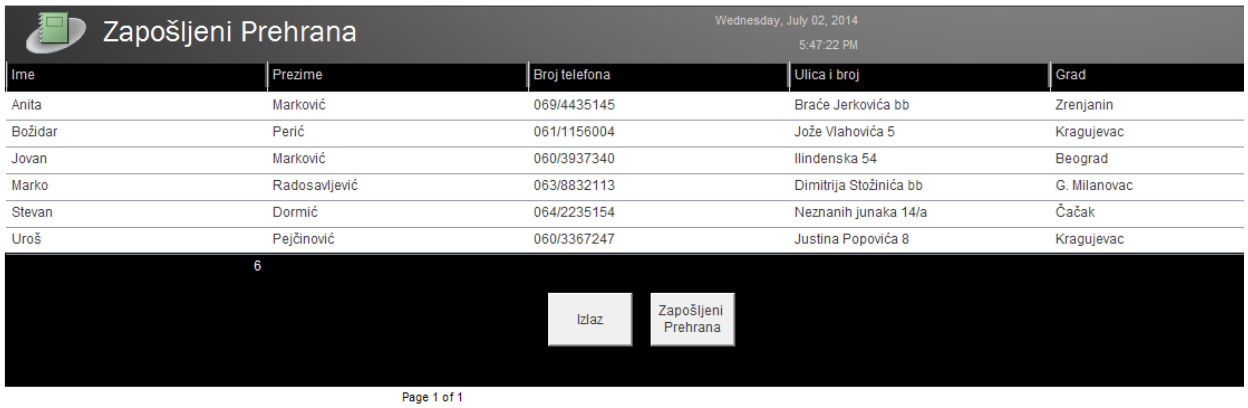
Слика 41. Формирана почетна страна и пречице (подстране)

Ponuda Prehrana

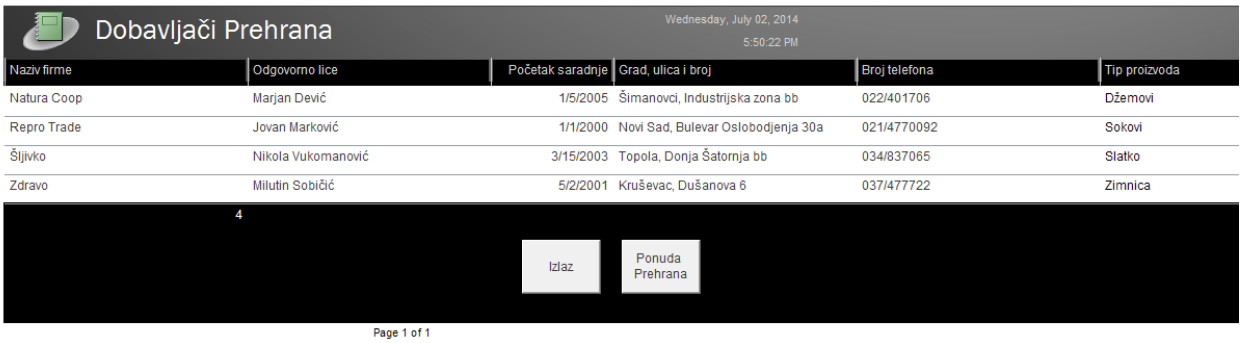
Thursday, July 03, 2014
11:24:18 AM

Naziv proizvoda	Neto težina	Cena	Broj komada	Kategorija proizvoda
Domaća ljutenica	350 gr	€17.00	10	Zimnica
Domaća marmelada od drenjina	700 gr	€24.00	5	Džemovi
Domaća marmelada od šipuraka	700 gr	€28.00	4	Džemovi
Domaća mešana marmelada	700 gr	€22.00	83	Slatko
Domaća šarena salata	700 gr	€15.00	17	Zimnica
Domaće slatko od borovnica	480 gr	€30.00	15	Slatko
Domaće slatko od šljiva	480 gr	€24.00	27	Slatko
Domaće slatko od šumskih jagoda	480 gr	€32.00	34	Slatko
Domaće slatko od višanja	480 gr	€26.00	47	Slatko
Domaći ajvar o pečene paprike-blagi	350 gr	€21.00	5	Zimnica
Domaći ajvar od pečene paprike-blagi	700 gr	€41.00	31	Zimnica
Domaći džem od aronije	700 gr	€26.00	7	Džemovi
Domaći džem od jagoda	700 gr	€28.00	13	Džemovi
Domaći džem od kajsija	700 gr	€24.00	6	Džemovi
Domaći džem od kupina	700 gr	€25.00	25	Džemovi
Domaći džem od malina	700 gr	€32.00	5	Džemovi
Domaći džem od šljiva	700 gr	€23.00	16	Džemovi
Domaći džem od višanja	700 gr	€24.00	14	Džemovi
Domaći kompot od višanja	700 gr	€19.00	10	Sokovi
Domaći pekmez od šljiva sa crnom čokoladom	700 gr	€26.00	3	Džemovi
Domaći sirup od aronije sa plodovima aronije	1 Litar	€41.00	15	Sokovi
Domaći sirup od borovnica sa plodovima borovnice	1 Litar	€40.00	51	Sokovi
Domaći sok od paradajza (nova boca)	1 Litar	€16.00	50	Zimnica

Слика 42. Извештај понуде прехране

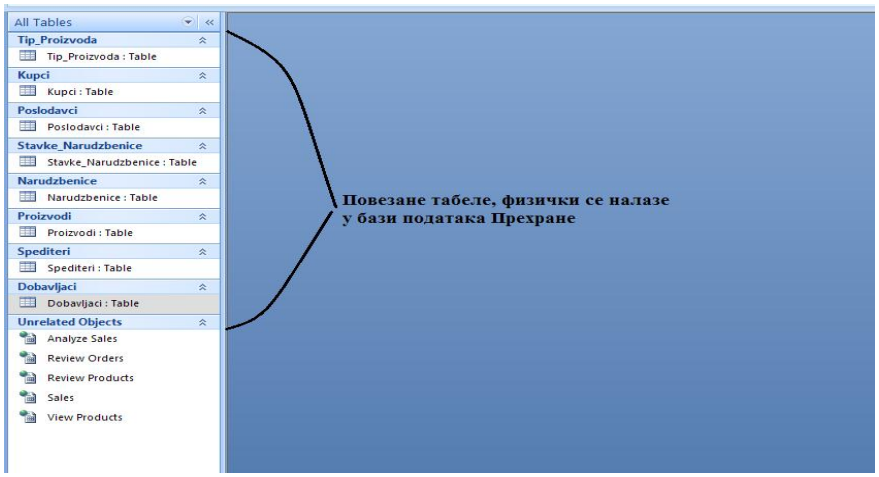


Слика 43. Извештај запослених у прехрани

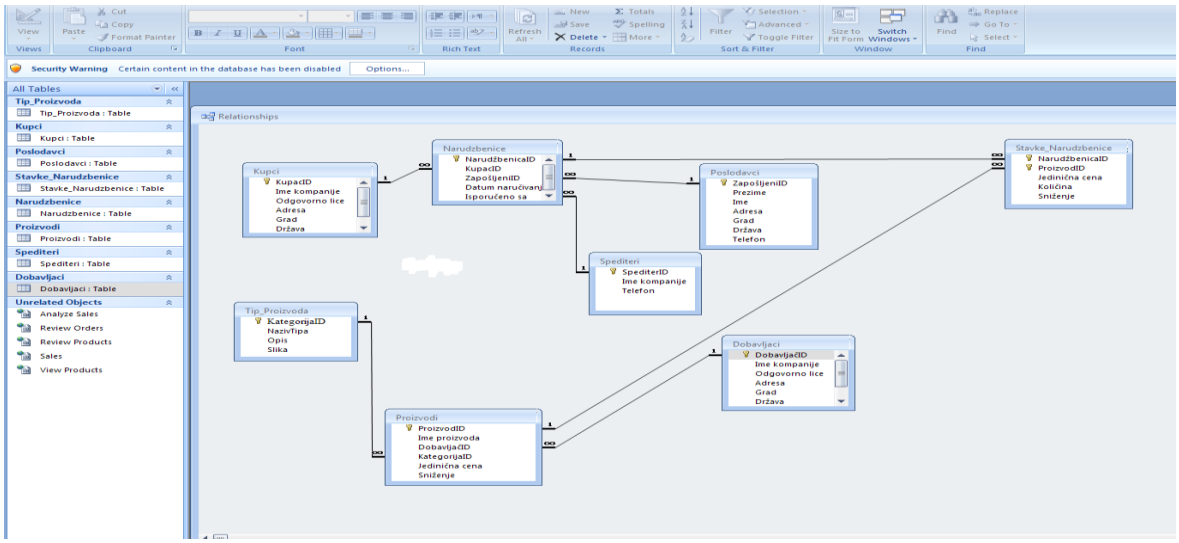


Слика 44. Извештај добављача у прехрани

Из менија „*External data*“, линкујемо табеле како би се физички налазиле у бази података компаније (слика 45), тако груписане табеле представљају нормализовану и пречишћену базу података (слика 46), приказану кроз „*Relationships Report*“.



Слика 45. Повезана табела из базе података прехране



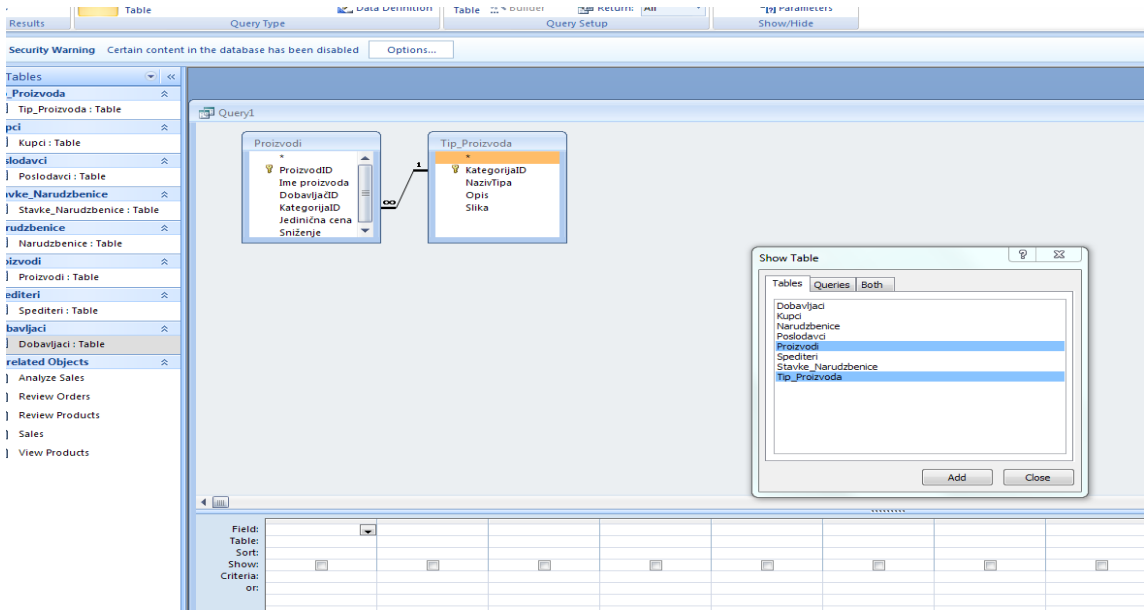
Слика 46. Нормализована оригинална база података

8.2.Креирање табела димензија

- Креирање табеле (димензије) користећи извршне упите *(„action queries“)*:
  - ✓ Производи
  - ✓ Купци
  - ✓ Временски период (квартал)

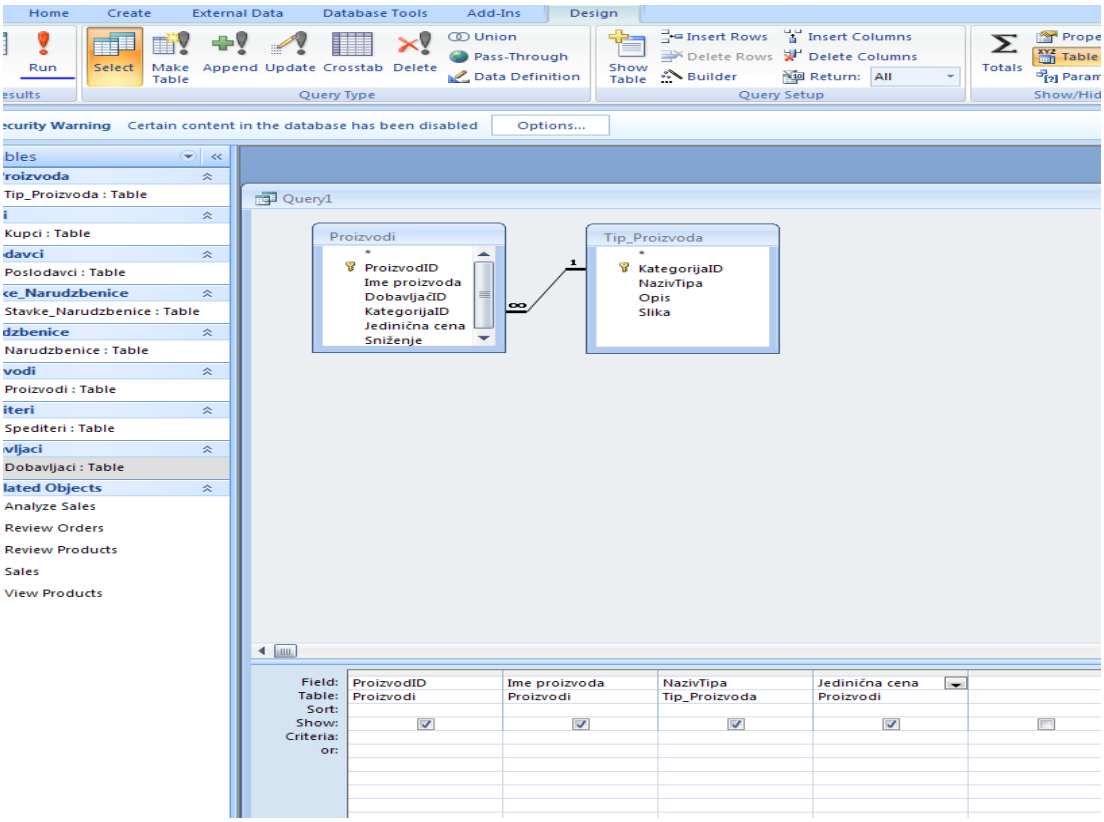
8.2.1. Димензија производа

Креирање и тестирање *„select query qDimenzijaProizvodi“*, (*ID, име, име категорије, јединична цена*).

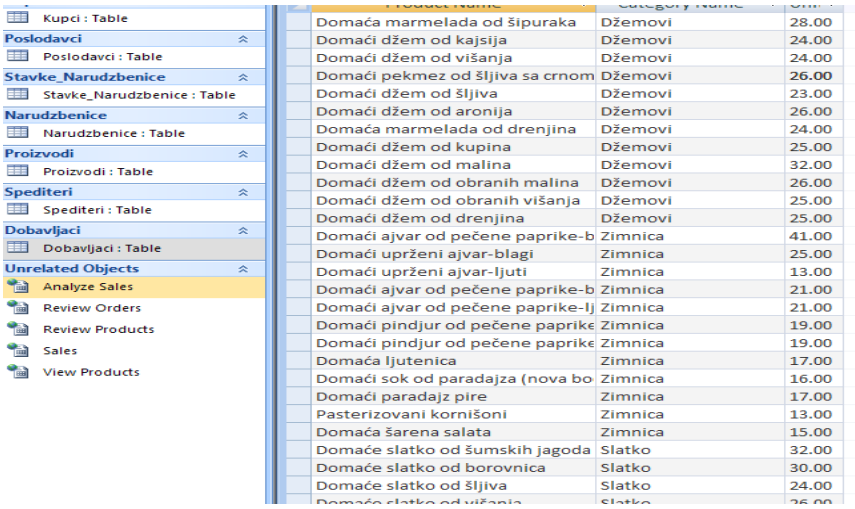


Слика 47. Креирање select query за табелу DimenzijaProizvodi

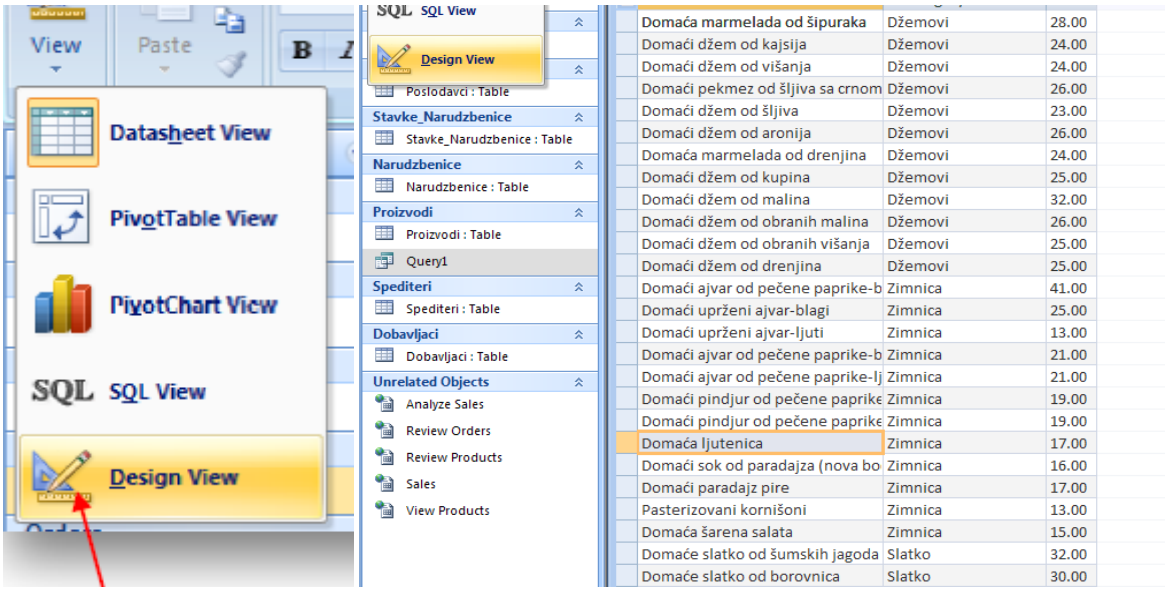
При тестирању ове димензије, из *Query* менија се покреће *make table query*, за табелу димензија производа (слика 47). Додавањем одређене категорије (Производи и тип_производа), као на слици, добијамо њихову међусобну зависност и формира се упит. Онда када су категорије у међусобној зависности, добија се низ испуњених поља и свако поље је из одговарајуће категорије (слика 48). При покретању опције *„Run“*, након испуњавања табеле, приказано је тражење одређеног производа уз назив производа, категорију (тип производа) и јединичне цене (слика 49).



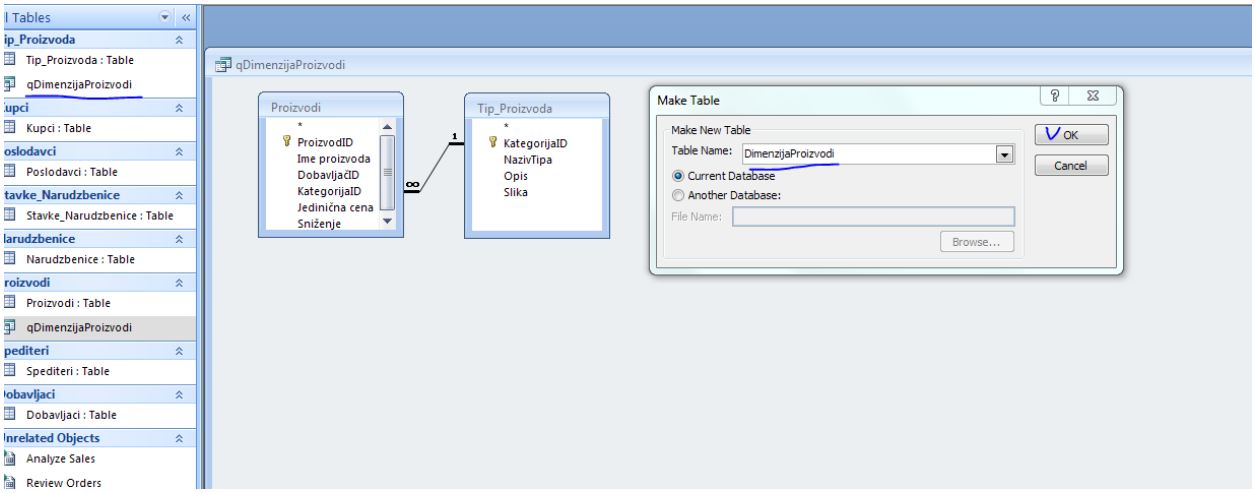
Слика 48. Испуњавање поља табеле



Слика 49.Производи упита испуњене табеле



Слика 50. Повратак из „Datasheet view“ у „Design view“, тестирање упита за табелу димензија „DimenzijaProizvodi“

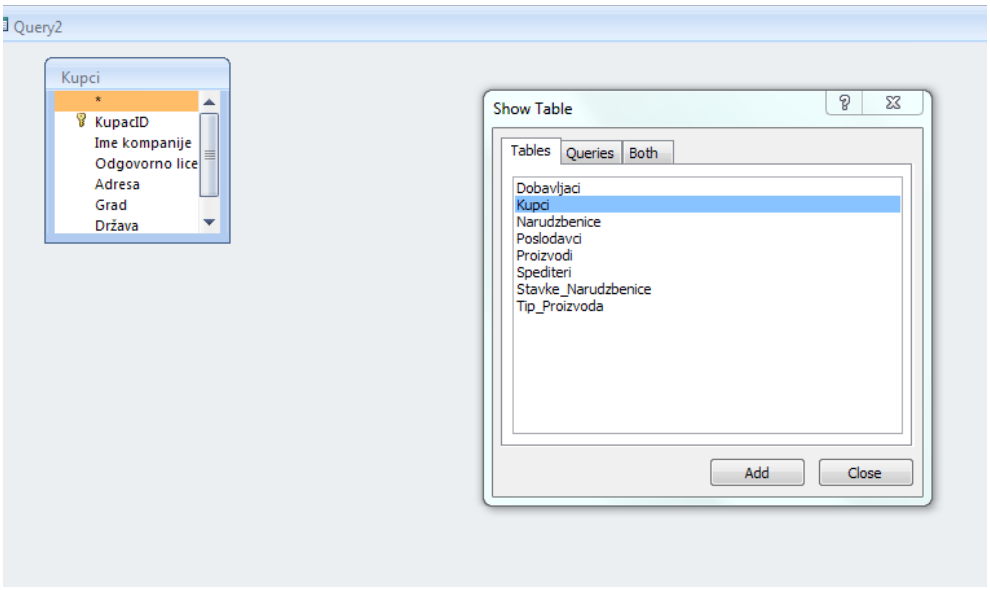


Слика 51. Креирање и снимање „make-table query“ за табелу димензија „DimenzijaProizvodi“

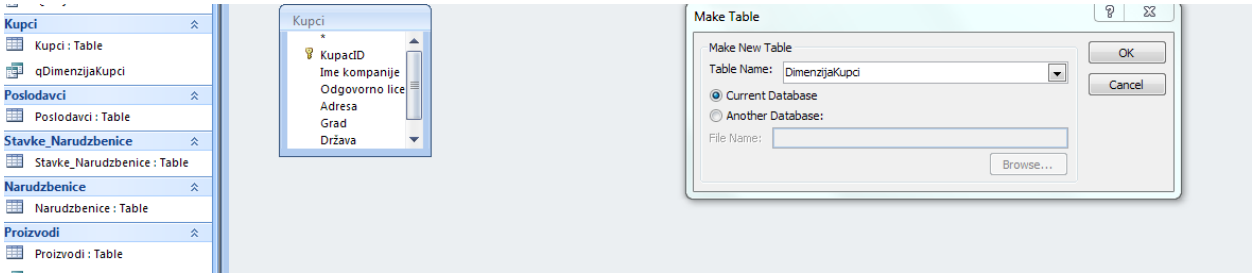
8.2.2. Димензија купци

Креирање и тестирање *„select query qDimenzijaKupci“*, (*ID, име компаније, град, држава*).

При изради димензије купаца, поступак је индентичан као и код димензије производа, али је мањи број параметара. Тачније, из *Query* менија се покреће *make table query* и бира се само једна категорија, категорија купаца (слика 52). Иста категорија се селектује по пољима табеле и кроз опцију *„Run“*, се долази до жељених анализа.Само креирање и чување за табелу димензија *„DimenzijaKupci“*, је већ познато из предходне димензије. Додаје се нова табела и њен назив је „*qDimenzijaKupci*“ (слика 53).



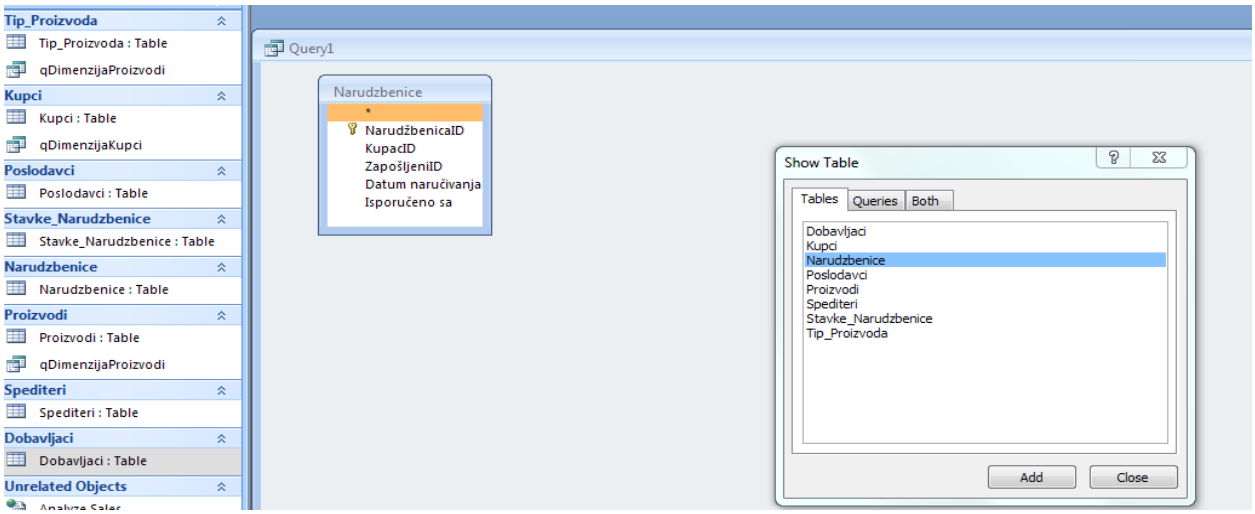
Слика 52. Креирање select query за табелу DimenzijaKupci



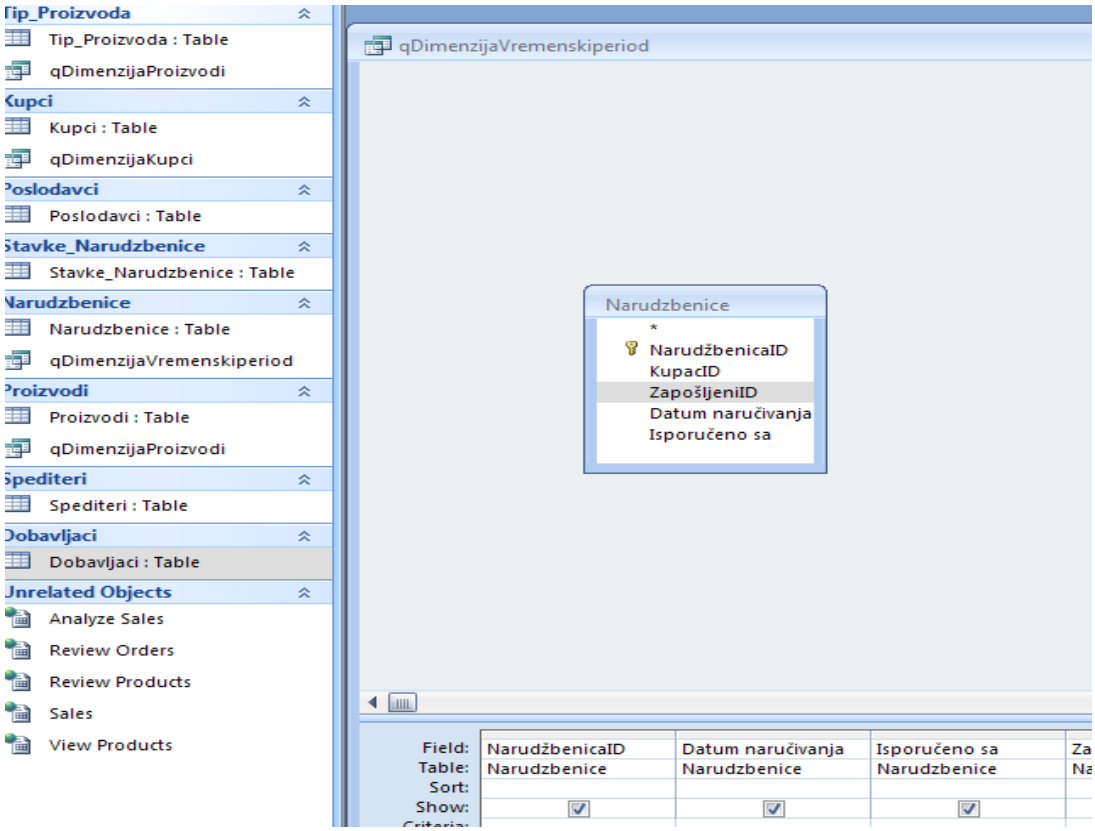
Слика 53. Креирање и снимање „make-table query“ за табелу димензија „DimenzijaKupci“

8.2.3. Димензија Временски период (Квартали)

Креирање и тестирање *„select query qDimenzijaVremenskiPeriod“*, (*ID, НаруџбеницаID, КупацID, ЗапослениID, датум наручивања, Испоручено са..*).



Слика 54. Креирање select query за табелу DimenzijaVremenskiPeriod



Слика 55. Креирање и снимање „make-table query“ за табелу димензија „DimenzijaVremenskiPeriod“

Креирањем ових димензија показани су зависност и међусобни однос категорија овог виртуелног предузећа „Прехрана“, дакле њихова расподела производа, зависност и утицај односа између датог производа и нпр. цене, количине, датума наручивања, као и знатно велики број осталих могућности при формирању категорије. Увођењем ових димензија, пружа се корисницима прави увид информација о производу.

9. Закључак

Складиште података намењено је менаџерима, али и свима који у свом послу обављају различите аналитичке задатке, као што су послови праћења и извештавања, који се темеље на примени различитих пословних правила, а обављају се постављањем усмерених питања и анализом добијених резултата, послови анализе и дијагнозе, који се темеље на умешности, а обављају се итеративним проналажењем и анализом добијених информација, и послови планирања и симулације, који се темеље на знању, а обављају се моделирањем и извршењем израђеног модела.

Најефикаснији вид приступа садржаја на *"web"*-у, јесте коришћење претраживања. На овај начин, избегава се потреба за систематском организацијом целокупног "web"а, већ се део проблема преноси на самог корисника, који својим упитима врши селекцију сегмената којима жели да приступи. Задатак "web" претраживача (*web search engine*) је да пружи механизме који кориснику омогућавају ефикасно претраживање и помоћ при селекцији релевантног садржаја. У последњој деценији, развијен је значајан број изузетно ефикасних претраживача, од којих неки данас предтстављају најпосећеније локације на *"web"*-у. *Google* карактерише једноставност коришћења, брзина одговора и константан развој понуде, производа и услуга. Наведена комбинација показала се као невероватан кориснички магнет. Провођењем споменутог начина рада у стварност Google постаје све популарнији међу искусним, као и неискусним; тј. новим корисницима.

Оптимизација интернет претраге је дуг процес који постепено побољшава ранкинг у резултатима претраге, број посетилаца и наравно вредност (односно зараду) неког *web* сајта. Побољшање ранкинга код водећих интернет претраживача се постиже споро, по пар месеци до преко годину дана., тако да је за то време потребно фокусирати се на локалне претраживаче где су могући резултати већ после неколико недеља. Најједноставнија провера релевантности претразивача јесте да откуцате кључне речи за које знате најрелевантније сајтове, било на енглеском или неком другом језику. У зависности од резултата претраживања видећете и колико су посматрани претраживачи релевантни.

Интернет претраживачи и поред поспешивања, напредка и популарности *„WWW"*-а, доприносе и његовој категоризацији и организацији. Кад би сви интернет претраживачи престали у тренутку са радом, наступио бих општи хаос на Интернету. Једноставно речено, без интернет претраживача не би било ни претраживања, већ само пуког тражења игле у пласту сена. Данас, на почетку 21. века, због велике количине информација које нас окружују, неопходно су нам потребни интелигентни агенти који омогућују што боље сналажење током претраживања "web"-а. Интелигентни агенти за претраживање јавили су се у другој половини осамдесетих година, а њихово изучавање је интезивно почело у деведесетим јер су препознати као обећавајуће решење за проблем прикупљања, обраде и чувања велике количине информација. Очекивања корисника су много нарасла од доба када су се претраживачки механизми тек појавили на Интернету и када је било право чудо када би нешто пронашли. Данас, ако корисник не добије што тражи унутар прва три линка, често одустаје од даљег претраживања. Зато се свакодневно, у великим тимовима, ради на исправљању алгоритама претраживања и побољшању самог претраживања.

Литература

[1].Kimball, B. R. (2002). The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling. Wiley.

[2].Suryakant B. Patil (2011). "Optimization of Data Warehousing System: Simplification in Reporting and Analysis"

[3].Bonifati, A. e. (2001). Designing Data Marts for Data Warehouse. Milano: ACM Transactions on Software Engineering and Methodology.

[4].Inmon, Bill (1992). *Building the Data Warehouse*. Wiley

[5].Rainardi, V. (2008). Building a Data Warehouse with Example in SQL Server. Apress.

[6]Claude S.: Data Mining with Macrosoft SQL Server 2000-Tehnical Reference, Microsoft Press, 2001.

[7].др А.Његуш, Пословни информациони системи, Београд, 2009. Година

[8].Б. Ничић, ЕТЛ процес у развоју система пословне интелигенције, Београд, 2009. Година

[9].Chris Todman: **Designing a Data Werehouse**, Hewlett-Packard Company, Prentice-Hall International (UK), London 2007.

[10].Паниан, Ж.: Богатство Интернета, Стрелац, Загреб, 2000.

[11].*"Professor and Head, Department of Computer Science and Engineering, Apollo Engineering College, „AN IMPLEMENTATION OF WEB PERSONALIZATION USING WEB MINING TECHNIQUES". Journal of Theoretical and Applied Information Technology."* Бр. 67-73

[12].*"Raymond Kosala, Hendrik Blockeel „Web Mining Research: A Survey" SIGKDD Explorations. Vol.2 Issue 1,"* бр. 1-15

[13]. *Тежак, Ђ.* (2002). *Претраживање информација на интернету*: Приручник са вежбама

[14].CONSIDERO.HR: *Најоптималнија "web" страница*,

[15].*"Youngblood, G. M., Web Hunting:Design of a Simple Intelligent Web Search Agent"*

[16].*"Nguyen N.T., L.C. Jain, (2009): „Intelligent Agents in the Evolution of Web and Applications"*

[17].*"SITE REFERENCE: SEO for Google,,*

[18]. *"BEANSTALK: SEO For Yahoo!"*

[19]. *"BEANSTALK: SEO For MSN"*