

UNIVERZITET U BEOGRADU
MATEMATIČKI FAKULTET

Nevena Joksić

**Primena OLAP tehnika u analizi
otplate duga klijenata Banke
Poštanske štedionice a. d.**

Master rad

Beograd, 2010. god.

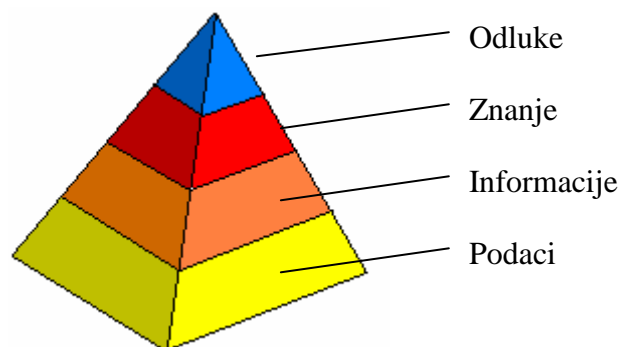
Sadržaj

1. INTELIGENTNO POSLOVANJE	2
2. ANALIZA PODATAKA	3
2.1 Izvori podataka.....	3
2.2 Napredne analize podataka	3
3. MULTIDIMENZIONNI PODACI	5
3.1 Ključni koncepti.....	6
3.2 Šema baze podataka	7
3.2.1 Šema zvezde.....	7
3.2.1.1 Jednostavna šema zvezde.....	8
3.2.1.2 Višestruka tabela činjenica	9
3.2.1.3 Višestruka šema zvezde	10
3.2.2 Šema pahulje.....	11
3.3 Hijerarhije	12
4. OLAP	14
4.1 OLTP.....	14
4.2 Definicija OLAP-a	15
4.3 Različita značenja OLAP-a.....	15
4.4 OLAP kocka.....	16
4.5 Operacije nad OLAP kockom.....	16
4.6 Tipovi OLAP arhitekture	19
4.6.1 MOLAP.....	19
4.6.2 ROLAP	20
4.6.3 Realizacija OLAP sistema	21
4.6.4 Poređenje ROLAP-a i MOLAP-a	23
4.7 OLAP alati	26
5. OPIS PROBLEMA I PREDLOG REŠENJA	28
6. REALIZACIJA REŠENJA	29
6.1 Objedinjeni sistemi (eng. <i>Federated Systems</i>).....	29
6.2 Organizacija podataka.....	30
6.3 Alati koji su korišćeni i način njihove primene	37
6.3.1 Kontrolni centar	37
6.3.2 OLAP Center	38
6.3.3 QMF/ Windows	42
7. REZULTATI	43
8. ZAKLJUČAK	48
9. Literatura	49

1. INTELIGENTNO POSLOVANJE

Inteligencija ili intelekt (lat. *Intellectus*) je mentalna osobina koja se sastoji od više sposobnosti: učenje iz iskustva, adaptiranje na nove situacije, shvatanje i razumevanje novih situacija i korišćenje stečenog znanja u interakciji sa okruženjem. Korišćenjem ovog tumačenja, pojam „poslovne inteligencije“ odnosno „inteligentnog poslovanja“ se može definisati kao proces prikupljanja najrazličitijih podataka, njihovo pretvaranje u korisne informacije, koje mogu da pomognu poslovnim korisnicima u donošenju odluka.

Inteligentno poslovanje je posebno počelo da se razvija kada su preduzeća počela da automatizuju svoje poslovne procese. Postojeći transakcioni sistemi su se pokazali kao savršeni generatori velikih količina podataka. Podaci su se vremenom sve više gomilali, ali se javio veliki problem - do tih podataka je bilo sve teže doći. Bilo je jasno da u tim podacima leži ogroman potencijal, pa se javila potreba da se oni nekako objedine, obrade i stave na raspolaganje rukovodstvu. Tako je nastalo inteligentno poslovanje, koje se, sa tehničke strane, najjednostavnije može opisati kao proces u kome se podaci pretvaraju u informacije. Informacije su rezultat analize i organizovanja podataka tako da se od se od njih dobija neko novo saznanje. Znanje je odgovarajući skup informacija. Na osnovu postojećeg znanja, mogu se donositi poslovne odluke.



Slika 1: Proces donošenja odluka od raspoloživih podataka

Prednosti koje donose sistemi inteligentnog poslovanja su: analiza poslovanja, analiza klijenata, bolja kontrola troškova poslovanja, nadzor poslovanja, brzo reagovanje, predviđanje budućih trendova, jednostavni grafički prikazi, jednostavno pravljenje i korišćenje izveštaja... Na tržištu donošenje pravovremenih odluka omogućava pružanje boljih usluga klijentima, izdizanje iznad konkurencije, a na osnovu toga i povećanje profita.

Sistem inteligentnog poslovanja omogućuje istorijske, sadašnje i buduće poglede na poslovne procese. Zajedničke funkcije sistema poslovne inteligencije su: izveštavanje, OLAP analitike, istraživanje podataka, upravljanje poslovnim procesom, testiranje karakteristika sistema itd. Cilj sistema inteligentnog poslovanja je da pomognu prilikom odlučivanja.

2. ANALIZA PODATAKA

Jezgro svake poslovne aktivnosti je obrada informacija. To uključuje skupljanje, čuvanje i obradu podataka. Značaj dobre informacije je u razlici između donošenja ispravne i pogrešne odluke. Dobra informacija mora da bude tačna, blagovremena i razumljiva.

2.1 Izvori podataka

Poslovni proces raspolaže velikim brojem podataka, koji su obično u različitim bazama, a često i na različitim fizičkim lokacijama. Podaci su ili dostupni ili arhivirani, ali se događa da su u različitim formatima. Prilikom donošenja odluka, korisniku je potrebno da ti podaci budu na jednom mestu (bar virtualno) i da pristup njima bude brz uprkos veličini podataka i koliko god da su oni stari. U te svrhe se koristi tzv. skladište podataka (eng. *data warehouse*) koje objedinjuje različite izvore podataka. Ažuriranje skladišta se vrši samo dodavanjem novih podataka, dok postojeći podaci najčešće ostaju nepromenjeni. Osnovni izvori podataka za koncept skladišta su: operativni podaci (transakcioni, tzv. OLTP (eng. *On-Line Transaction Processing*)), spoljne informacije nastale kao istorija poslovanja i razni drugi podaci uzeti iz javnih baza podataka. Najniži nivo obrade podataka je transakciono orijentisan i podržava obradu svakodnevnih operativnih poslova. Osnovna karakteristika transakcionog pristupa su normalizovani modeli podataka, kratko vreme obrade, veliki broj transakcija koje rade sa relativno malim brojem tabela i sa relativno malim brojem operacija nad njima. Dublja analiza podataka korišćenjem OLTP sistema je otežana i nepogodna i svodi se na analizu operativnih podataka i na korišćenje izveštaja koji se direktno generišu nad operativnim podacima.

2.2 Napredne analize podataka

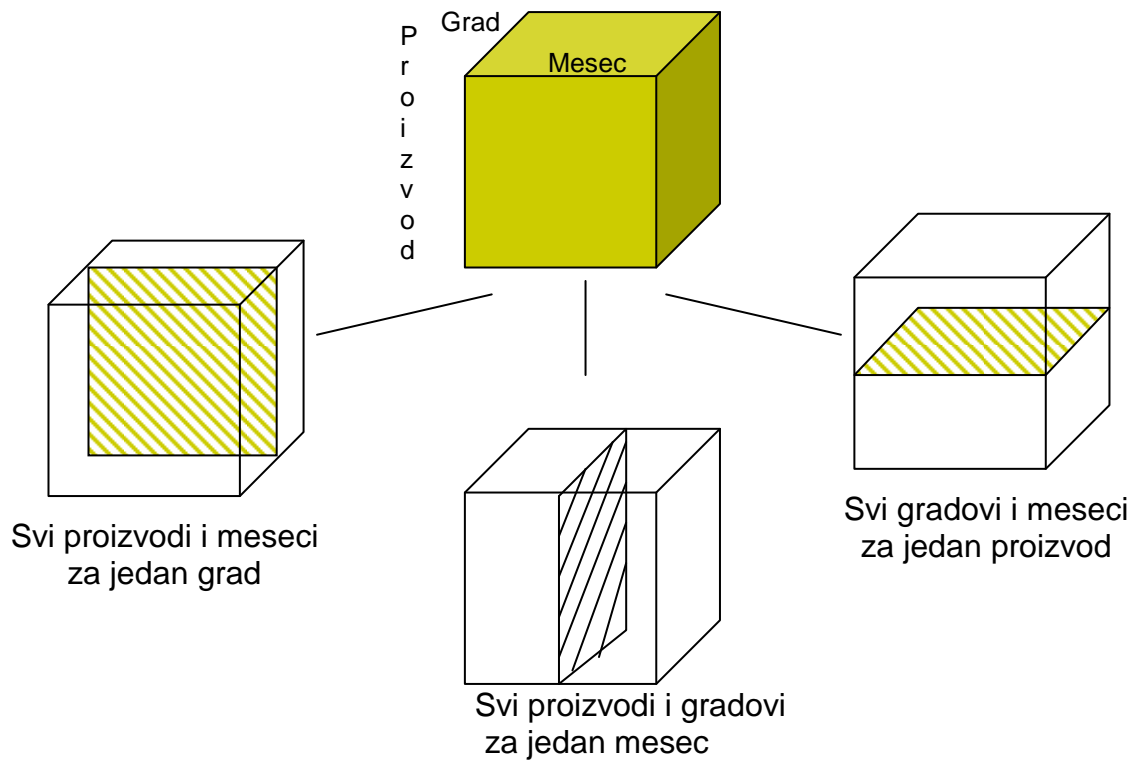
Analiza podataka je značajan deo inteligentnog poslovanja, jer korišćenjem odgovarajućih alata za analizu može da se dođe do relevantnih informacija potrebnih za donošenje novih odluka u poslovnom procesu. Različite tehnike analize podataka su: OLAP (eng. *On-Line Analytical Processing*), istraživanje podataka (eng. *Data mining* - DM), simulacije, korišćenje upitnih jezika, itd.

Istraživanje podataka je proces koji izvodi pravila i informacije iz velike količine podataka. Identifikuju se veze između naizgled nepovezanih podataka. Osnovna cilj je da se iz velikog broja operativnih podataka i veza koje ne mogu odmah da se sagledaju, definišu odgovarajuće relacije i obrasci ponašanja, na osnovu čega bi se dobile potrebne informacije. Simulacionim alatima se testiraju buduća moguća stanja na osnovu trendova u poslovanju i omogućuje se formiranje novih poslovnih pravila. Upitni jezici predstavljaju standardni upitno - izveštajni alat koji omogućuje korisnicima da pretražuju, analiziraju i uzimaju pojedinačne podatke iz postojećih baza radi formiranja različitih izveštaja. Pristup podacima putem upitnog jezika omogućuje detaljan pregled podataka i njihovih odnosa u transakcionoj bazi.

OLAP omogućuje brz, interaktivan pristup podacima, pružajući širok spektar različitih multidimenzionalnih pogleda na podatke. Povezani poslovni pojmovi se analiziraju preko dimenzija. Ideja ovog pristupa je u tome da poslovanje preduzeća, a i funkcionisanje ljudskog uma počiva na višedimenzionalnosti. Recimo, ako je potrebno da se analizira prodaja, onda želimo da je analiziramo u vremenu, po regijama, po prodajnim mestima, po kupcima, po artiklima... Navedeni pojmovi su, u stvari, različite dimenzije pojma prodaja. Osnovna razlika između OLAP-a i istraživanja podataka je u samom pristupu podacima. OLAP alati omogućuju višedimenzionalne analize, koje obuhvataju raščlanjavanje podataka na sitnije detalje, kao i njihovo sumiranje. Istraživanje podataka je, s druge strane, proučavanje odnosa među podacima i ponalaženje nekih obrazaca i međusobnog uticaja podataka. OLAP i istraživanje podataka mogu zajedno da se koriste kao veoma moćno sredstvo za analizu. U ovom radu najviše pažnje će biti posvećeno OLAP-u, a za to je potrebno prvo objasniti pojam multidimenzionalnosti podataka.

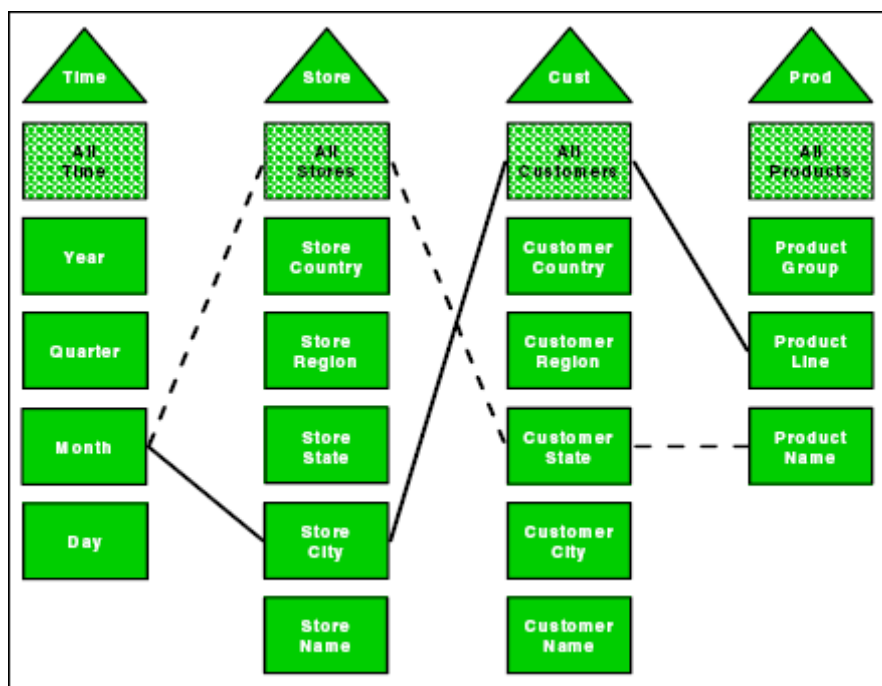
3. MULTIDIMENZIONIRANI PODACI

Organizovanjem podataka u višedimenzionalne strukture postiže se njihovo maksimalno iskorišćavanje u smislu dobijanja potrebnih informacija. Moguće je njihovo analiziranje iz različitih perspektiva, a cilj je dobijanje odgovora na različita poslovna pitanja. Složeni podaci se predstavljaju u obliku koji je jednostavan za razumevanje krajnjem korisniku. Pristup podacima preko odgovarajućih dimenzija omogućuje da korisnici prave svoje analize na brz i jednostavan način. Na primeru prodaje se mogu uočiti sledeće dimenzije: mesec, grad i proizvod, kao što je prikazano na Slici 2.



Slika 2: Različiti pogledi na iste podatke

Na istom primeru je može da se uoči još jedna dimenzija - kupac. Tako se dobija četvorodimenzionalna struktura, koju je teže grafički prikazati. Dodatno, sve dimenzije mogu da imaju više nivoa. Naredna slika predstavlja primer višedimenzionalne strukture podataka.



Slika 3: Primer analize višedimenzionane strukture podataka

Iz ovakve strukture podataka se jednostavno mogu dobiti odgovori na pitanja tipa: kakva je bila prodaja neke linije proizvoda tokom maja meseca za sve kupce po gradovima prodajnih mesta (označeno punom linijom na Slici 3), koje proizvode su kupci iz Meksika kupovali tokom juna (što je obeleženo isprekidanom linijom), itd.

3.1 Ključni koncepti

Osnova ovakve ideje je dimenzija. Na primeru kocke prodaje se mogu da se uoče dimenzije: vreme, proizvod, lokacija, kupac. Neka od pitanja na koja se pomoću ove strukture može dobiti odgovor su: poređenje prodaje ovog i prošlog meseca, ove i prošle godine, koji procenat ukupne zarade je doneo proizvod iz linije A, poređenje prodaje u Nemačkoj i Španiji, koji kupac donosi najviše profita, koji procenat kupaca kupuje neki proizvod ili neku liniju proizvoda, itd.

Drugi ključni koncept je kategorija koja se odnosi na pojedinačne podatke u okviru neke dimenzije. Npr. u vremenskoj dimenziji kategorija bi bila 2008. ili 2009. godina, u dimenziji koja se odnosi na lokaciju - Beograd, Barselona, Rim, ... Kategorije su hijerarhijski organizovane - neke kategorije su u okviru drugih, recimo Beograd pripada Srbiji.

Treći koncept je mera. Uz meru se pridružuje neka funkcija izračunavanja. Na primer, za prodaju, tipične mere su zarada, trošak, broj prodatih proizvoda, itd.

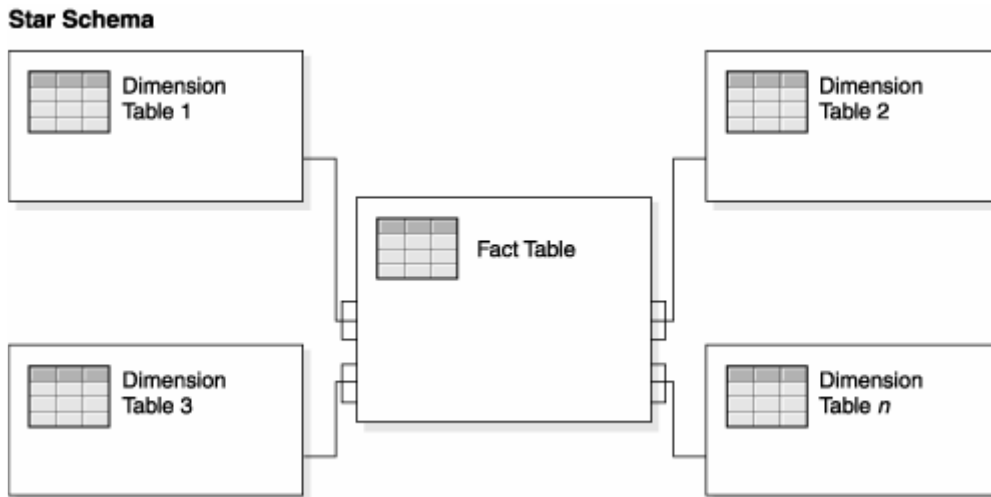
3.2 Šema baze podataka

Šema je kolekcija imenovanih objekata baze podataka koja obezbeđuje logičku klasifikaciju tih objekata [1]. Ukoliko se baza koristi za implementaciju skladišta podataka, ona najčešće ima tzv. dimenzionu šemu. Potreba za dimenzionom šemom se javlja iz razloga što ona daje mogućnost pravljenja upita koji treba da pruže odgovore na poslovna pitanja, i zbog neophodnosti da se ti upiti napisu SQL-om, koji koristi najveći broj korisnika RSUBP-a.

Dimenziona šema je struktura u kojoj se izdvajaju mere (podaci) koje predstavljaju vrednosti iz poslovnog procesa i opisni elementi (dimenzije) koji opisuju poslovni proces. Ako objekti u bazi podataka nisu formirani u skladu sa zahtevima dimenzione šeme, moguće je da se naprave pogledi tako da se napravi virtualna dimenziona šema. Obično, dimenziona šema ima jednu od sledećih formi: jedna tabela, šema zvezde ili šema pahulje. U slučaju da je dimenziona šema samo jedna tabela, tada su mere i dimenzije različiti atributi iste tabele. Šema zvezde i šema pahulje predstavljaju način da se mere i dimenzije odvoje u posebne tabele. Šema pahulje odvaja i različite nivoe hijerarhije u posebne tabele. U obe ove šeme su sve tabele međusobno povezane primarnim i stranim ključevima. Tabela činjenica je centralna tabela u šemama zvezde i pahulje u kojoj su sačuvani podaci o merama poslovnog procesa, kao što su zarada, gubitak, cena proizvoda. Ona sadrži strane ključeve kojima je povezana sa dimenzionim tabelama. Dimenziona tabela u šemi zvezde ili u šemi pahulje je tabela koja sadrži attribute koji opisuju aspekte dimenzije. Npr. u tabeli sa vremenskim podacima, aspekti bi bili godina, tromesečje, mesec i dan.

3.2.1 Šema zvezde

Najjednostavnije je prikazati dimenzioni model tzv. šemom zvezde. Ona se sastoji iz tabela koje predstavljaju dimenzije i jedne centralne tabele koja se naziva tabela činjenica. Kolone tabele činjenica sadrže različite numeričke vrednosti koje predstavljaju mere. Na te mere treba primeniti odgovarajuće funkcije i na taj način će se dobiti traženi rezultati po potrebnim dimenzijama. Recimo, ako je dimenzija vreme, a mera zarada, možemo prikazati količinu zarade u nekom vremenskom intervalu.

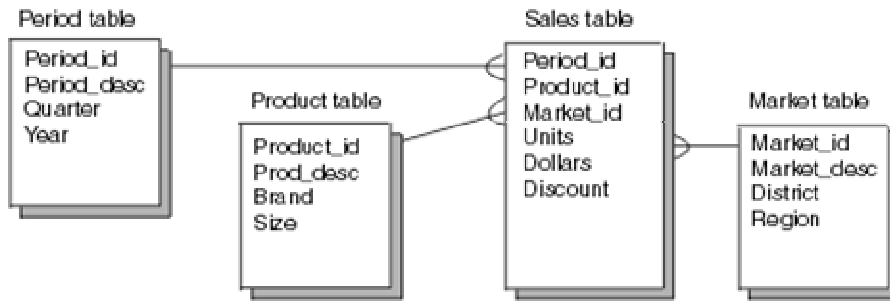


Slika 4: Šema zvezde

Tabele činjenica i dimenzija se razlikuju samo po načinu korišćenja u šemi zvezde. Njihova fizička struktura i SQL sintaksa koja se koristi za njihovo formiranje je identična. U složenijim šemama jedna ista tabela može da bude i tabela činjenica i dimenziona tabela, samo u zavisnosti toga na koji način joj se vrši obraćanje u upitu. Razlika između tabele činjenica i dimenzione tabele je u logičkom smislu. Da bi se razlika uočila, treba da se shvati kako analitičar posmatra poslovni proces: radnik u prodaji analizira prihod po kupcu, proizvodu, prodajnom mestu i vremenskom periodu; finansijski analitičar upoređuje ostvarene rezultate sa budžetom na nivou pojedinačnih stavki, po proizvodu i vremenskom periodu; radnik u marketingu kontroliše isporuku robe po proizvodu, tržištu i vremenskom periodu. Ono što će biti analizirano u ovim slučajevima je: profit, budžet, isporuka robe i nalaziće se u tabeli činjenica. Dimenzije ovih poslovnih procesa su: proizvod, prodajno mesto, vremenski period, linija proizvoda. Na primer, tabela činjenica za bazu podataka o prodaji, koja implementirana pomoću šeme zvezde, može da sadrži podatke o zaradi od prodaje svih proizvoda tog preduzeća za svakog kupca i sva prodajna mesta tokom nekog vremenskog perioda. Tabele dimenzija definišu kupce, proizvode, prodajna mesta i vremenski period. Dobro dizajnirana šema zvezde omogućuje korisniku da prikaže podatke iz baze i da radi upite nad njima na razumljiv i jednostavan način.

3.2.1.1 Jednostavna šema zvezde

Svaka tabela u šemi zvezde treba da ima primarni ključ. Kod jednostavne šeme zvezde primarni ključ tabele činjenica se sastoji iz jednog ili više stranih ključeva. Na sledećoj slici je dat primer jednostavne šeme zvezde:

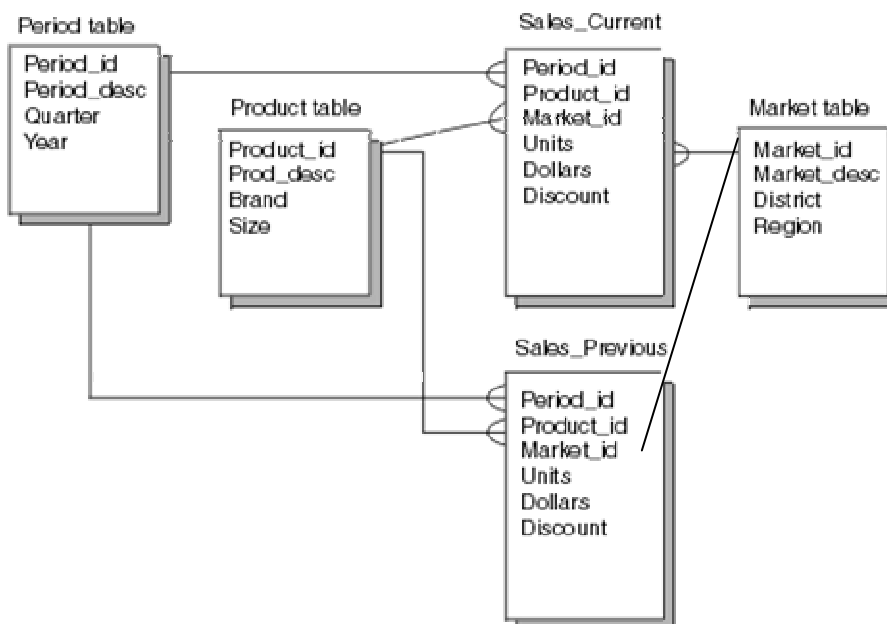


Slika 5: Jednostavna šema zvezde

U *Sales table* primarni ključ se sastoji iz tri kolone, odnosno tri strana ključa - *Period_id*, *Product_id*, *Market_id*, koji se odnose na primarne ključeve dimenzionih tabela: *Period table*, *Product table*, *Market table*.

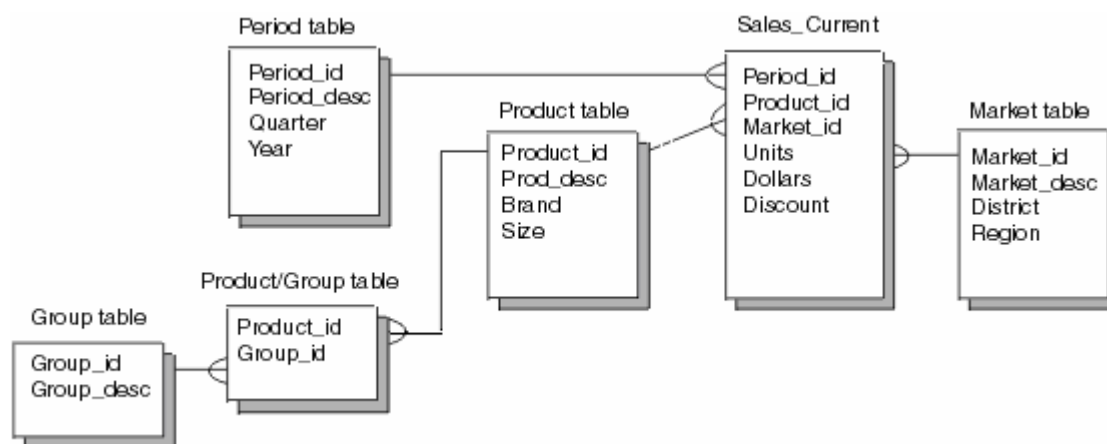
3.2.1.2 Višestruka tabela činjenica

Šema zvezde može da sadrži više tabela činjenica. Recimo, kada tabele činjenica nisu srodne, logično je da podaci stoje u odvojenim tabelama: npr. prodaja i računi. Više tabela činjenica se često javlja i u slučajevima kada je potrebno poboljšati performanse, ili kada se čuvaju različiti nivoi sumarnih podataka - dnevna prodaja, mesečna, godišnja. Sledeća slika prikazuje bazu prodaje koja uključuje dodatnu tabelu činjenica koja sadrži podatke od prethodne godine.



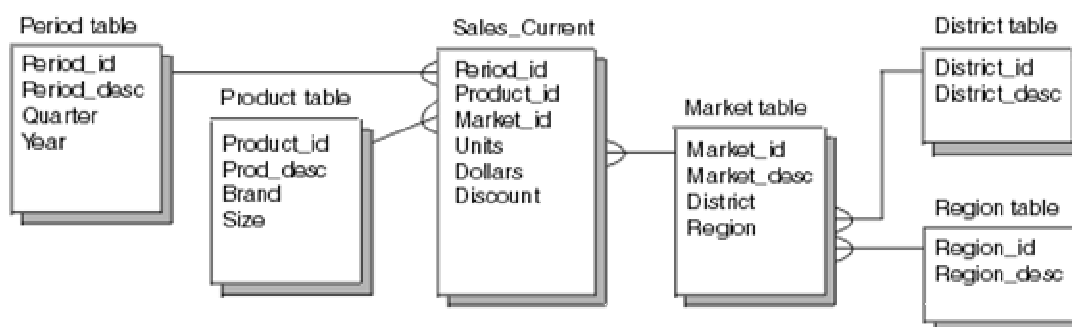
Slika 6: Šema zvezde sa višestrukom tabelom činjenica

Takođe, jedna dimenzija se može sastojati iz više tabela. Npr. svaki proizvod pripada jednoj ili više grupa proizvoda, a svaka grupa sadrži veći broj proizvoda (*Product/Group table* i *Group table*), kao što je prikazano na slici:



Slika 7: Šema zvezde sa odnosnim tabelama

Moguće je da dimenziona tabela sadrži jedan ili više stranih ključeva koji se odnose na primarne ključeve drugih dimenzionih tabela. Tada se „spoljna“ dimenziona tabela ne vezuje direktno sa tabelom činjenica, nego preko druge dimenzione table:

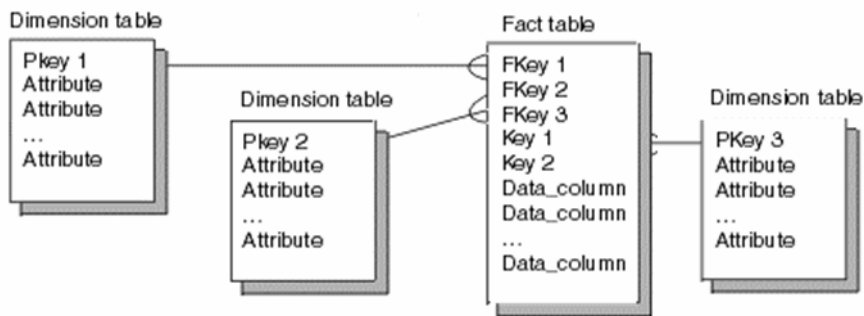


Slika 8: Šema zvezde sa „spoljnim“ tabelama

U ovom slučaju *Market table* može biti i tabela činjenica i dimenziona tabela, u zavisnosti od toga kako se koristi u upitu.

3.2.1.3 Višestruka šema zvezde

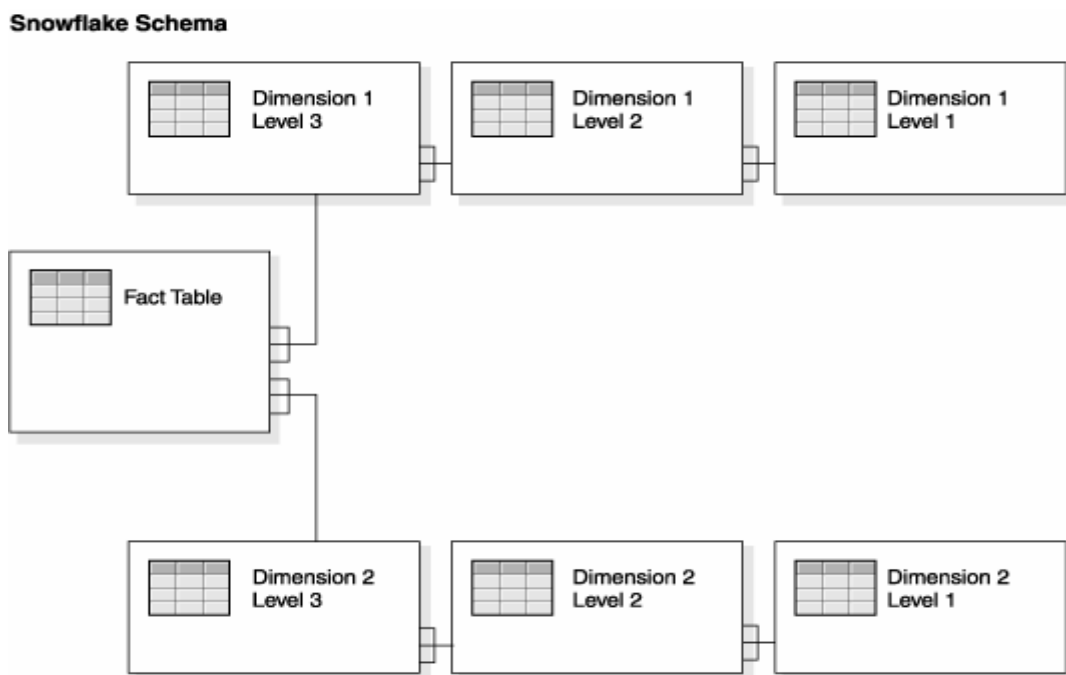
U jednostavnoj šemi zvezde primarni ključ table činjenica čini skup stranih ključeva stranih tabela. U nekim slučajevima skup stranih ključeva neće obezbediti jedinstvenost svakog reda table činjenica. Tada se javlja potreba za višestrukom šemom zvezde. U šemi na slici primarni ključ table činjenica čine *Key1*, *Key2* i *Fkey1*, a *Key1* i *Key2* se ne odnose ni na jedan primarni ključ dimenzione table.



Slika 9: Višestruka šema zvezde

3.2.2 Šema pahulje

Šema pahulje se dobija normalizovanjem dimenzionih tabela šeme zvezde. Primer šeme pahulje sa dve dimenzije od kojih svaka ima 3 nivoa je prikazan na sledećoj slici:



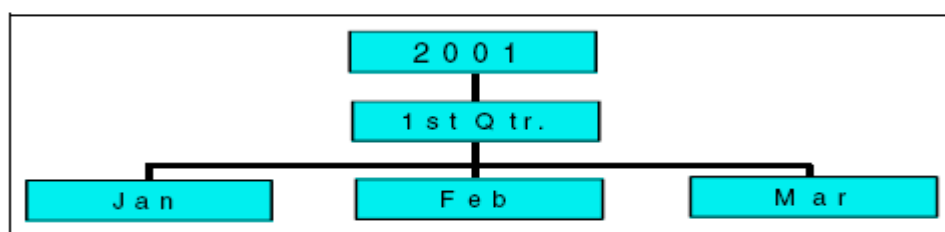
Slika 10: Šema pahulje

Postavlja se pitanje da li odabrati šemu pahulje ili šemu zvezde. Ukoliko je prioritet što manje zauzeće memorijskog prostora, onda treba izabrati šemu pahulje, zato što su u tom slučaju dimenzione tabele u 3NF, a ako su prioritet bolje performanse, treba izabrati šemu zvezde zato što se tada koristi manji broj stranih ključeva i ima manje spajanja tabela. Upiti kod šeme zvezde su jednostavniji za razumevanje i brži za izvršavanje, ali je šema pahulje jednostavnija za održavanje, jer nema dupliranja podataka.

3.3 Hijerarhije

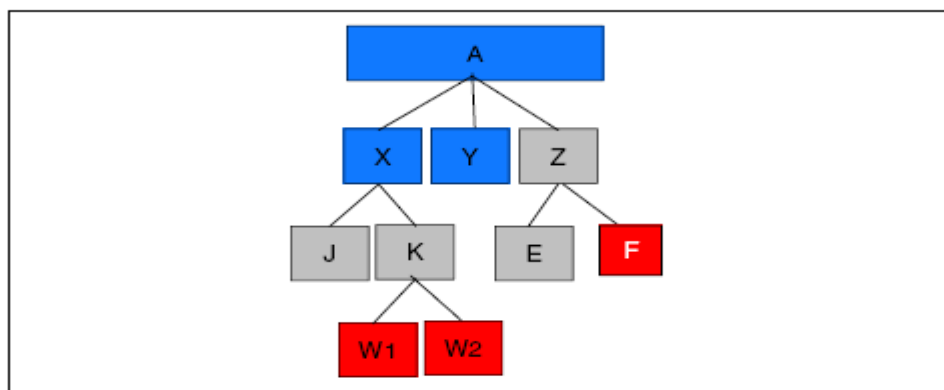
Različiti nivoi jedne dimenzije čine jednu njenu hijerarhiju. Kod šeme zvezde ti nivoi mogu biti u jednoj tabeli, a kod šeme pahulje su u različitim tabelama. Npr. hijerarhiju vremenske dimenzije činili bi nivoi : dan, mesec, godina. Neki od tipova hijerarhija su:

- Balansirana hijerarhija (eng. *balanced*) je ona kod koje nivoi imaju povezano značenje, i svako dete ima tačno jednog roditelja na nivou iznad. Ova hijerarhija može da se primeni na vremensku dimenziju, kao što je prikazano na Slici 11.



Slika 11: Balansirana hijerarhija

- Nebalansirana hijerarhija (eng. *unbalanced*) je ona kod koje svako dete ima jednog roditelja, ali nivoi nemaju povezano značenje. Na sledećoj slici je prikazan primer nebalansirane hijerarhije: Proizvod A se sastoji od proizvoda X i Y i komponente Z; komponenta Z se sastoji od komponente E i dela F; proizvod X se sastoji od komponentata J i K, a komponenta K od delova W1 i W2.



Slika 12: Nebalansirana hijerarhija

- Neusklađena hijerarhija (eng. *ragged*) je tip hijerarhije kod koje svako dete ima jednog roditelja, nivoi imaju povezano značenje, ali roditelj ne mora da se nađe na nivou neposredno iznad nivoa deteta. Ova hijerarhija može da se primeni na dimenziju lokacije, kao što je prikazano na sledećoj slici.

Region	Zemlja	Država	Grad
Severna Amerika	SAD	Kalifornija	San Jose
Evropa	Grčka	-	Atina
-	Island	-	Rejkjavik



Slika 13: Neusklađena hijerarhija

- Hijerarhija mreže (eng. *network*) je tip hijerarhije kod koje dete može da ima više od jednog roditelja. U ovom slučaju primer može da bude porodično stablo.

4. OLAP

4.1 OLTP

OLTP (eng. *On – Line Transaction Processing*) označava obrade pri kojima sistem u kratkom vremenskom periodu odgovara na zahtev korisnika. U procesu poslovanja, javlja se potreba za izradom izveštaja koji bi pomogli rukovodstvu u donošenju odluka. Problemi kod OLTP sistema prilikom izveštavanja su sledeći: pristup podacima je komplikovan, obrada podataka usporava poslovne transakcije, podaci su različiti i složeni. Izveštaji se izrađuju pomoću upitnog jezika. Za to je potrebno odlično poznavanje baze podataka i potpuno vladanje upitnim jezikom. Za realizaciju vrlo složenih upita je potrebno dugo vreme odziva, a obim podataka koji se mogu uzeti u obzir je ograničen.

Edgar Kod, autor relacionog modela baza podataka, je definisao pojam OLAP-a 1993. godine. Svrha OLAP alata je analiza i obrada podataka i izrada izveštaja koji će biti od koristi za strateško upravljanje. Dolazi se do novog pogleda na postojeće podatke. Razlike između dotadašnjeg i novog pristupa su prikazane u narednoj tabeli:

	OLTP	OLAP
korisnici	službenici	„pametan“ korisnik
funkcija	svakodnevene operacije	podrška odlučivanju
dizajn baze podataka	aplikativno orijentisana, visoko normalizovana baza sa više tabela	orijentisana prema oblasti korišćenja, denormalizovana baza sa manje tabela, sa korišćenjem šeme zvezde ili pahulje
podaci	operativni, originalni izvor podataka	istorijski, sumarni, skupljeni iz raznih OLTP baza podataka
korišćenje	često, ponavlja se	ad hoc
pristup	čitanje/pisanje	mного pregleda
jedinica posla	kratka, jednostavna transakcija	složeni upiti
broj slogova kojima se pristupa	desetine	milioni
broj korisnika	hiljade	stotine
veličina baze	100 MB- GB	100 GB- TB

Slika 14: Poređenje OLTP i OLAP sistema

4.2 Definicija OLAP-a

OLAP (eng. *On - Line Analytical Processing*) predstavlja skup alata koji omogućuju analizu podataka sačuvanih u bazi podataka. Pomoću OLAP-a, rukovodioci i analitičari mogu da imaju brz, interaktivan pristup podacima kroz širok spektar različitih uglova posmatranja. OLAP pripada kategoriji aplikacija i tehnologija za sakupljanje, upravljanje, obradu i predstavljanje multidimenzionih podataka za potrebe analize. Najšire usvojenu definiciju OLAP-a, koja se danas koristi, dao je Nigel Pendse [9] i opisana je pomoću sledećih 5 reči: Brza Analiza Deljenih Multidimenzionih Informacija:

- **Brza** - odnosi se na brzinu kojom OLAP može da donese što veći broj odgovora svojim krajnjim korisnicima.
- **Analiza** - odnosi se na sposobnost OLAP sistema da savlada poslovnu logiku i statističke analize relevantne za aplikaciju i korisnika i da ih dovoljno pojednostavi za razumevanje krajnjem korisniku.
- **Deljenih** - omogućeno je da se podaci dele između više korisnika.
- **Multidimenzionih** - odnosi se na koncept koji je primaran zahtev OLAP-a. OLAP sistem mora da omogući multidimenzioni pogled na podatke i da uključi hijerarhije i višestruke hijerarhije dimenzija.
- **Informacija** - predstavlja sve podatke i izvedene podatke koji su relevantni za aplikaciju.

4.3 Različita značenja OLAP-a

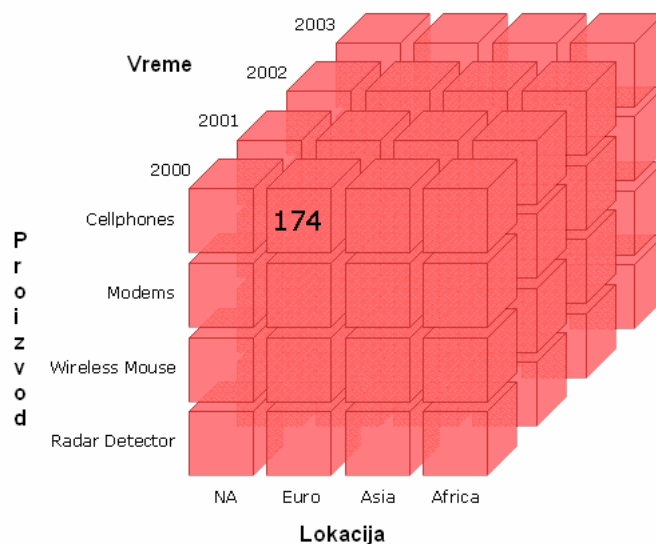
Pojam OLAP obuhvata više celina između kojih ne postoji jasna granica. Neke od njih su:

- OLAP koncepti koji obuhvataju ideju o višestrukim dimenzijama sa hijerarhijama.
- OLAP formalni jezici koji obuhvataju: jezik za definisanje podataka (eng. *Data Definition Language* - DDL), jezik za obradu podataka (eng. *Data Manipulation Language* - DML), jezik za prikazivanje podataka (eng. *Data Representation Language* - DRL) i povezane analizatore (opciono i kompajlere). OLAP jezici mogu da se koriste za opisno modelovanje, ili za transakcionu podršku ili podršku u odlučivanju.
- OLAP proizvodi koji se oslanjaju na relacionu bazu podataka. Svaki upit koji korisnik postavi OLAP sistemu se transformiše u niz SQL naredbi. Čuvanje podataka i pristup njima su definisani na nivou baze.
- OLAP proizvodi koji sadrže kompajler, metode čuvanja i pristupa podacima. Ovi proizvodi su optimizovani za brz pristup podacima i brza izračunavanja i koriste se u sistemima za podršku odlučivanju.

4.4 OLAP kocka

Prema samoj definiciji OLAP-a, ključni zahtev je višedimenzionalnost. OLAP postiže svoju višedimenzionu funkcionalnost korišćenjem strukture koja je nazvana kocka. Kocka omogućuje multidimenzioni pogled na podatke, a, sa druge strane, može da se poredi sa tabelom u bazi podataka. Specifičan dizajn OLAP kocke garantuje optimizaciju izveštaja.

Višedimenzionalne strukture podataka (opisane u prethodnom poglavlju) se najbolje vizuelizuju kao kocke podataka koje se sastoje iz manjih, „jediničnih“ kocki. Svaka strana kocke je jedna njena dimenzija. Dimenzija predstavlja skup kategorija podataka iste vrste. Npr. u kocki prodaje jedna dimenzija je lokacija, njene različite kategorije mogu biti gradovi, okruzi, države... Svaka ćelija te kocke sadrži agregirane podatke (mere o kojima je ranije bilo govora) koji su u vezi sa dimenzijama. Npr. jedna ćelija može da sadrži podatke o ukupnoj prodaji za dati proizvod i region u toku jedne godine:

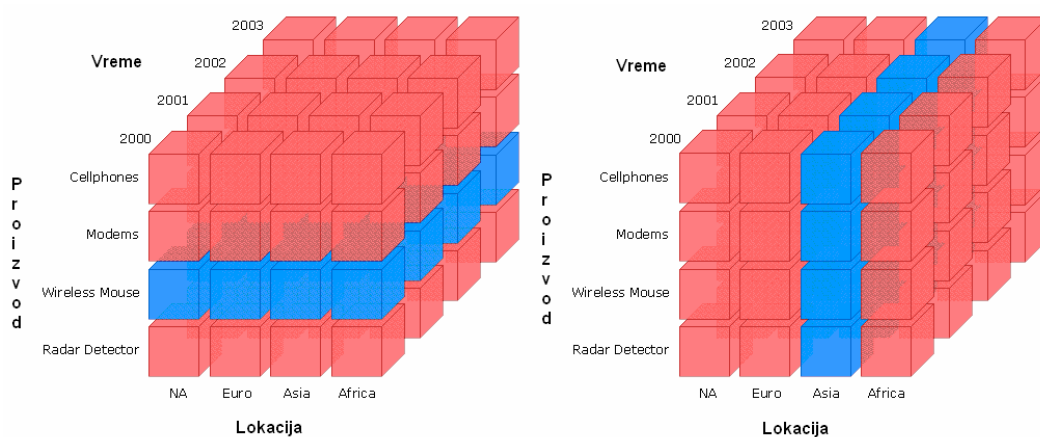


Slika 15: OLAP kocka sa podacima o prodaji

4.5 Operacije nad OLAP kockom

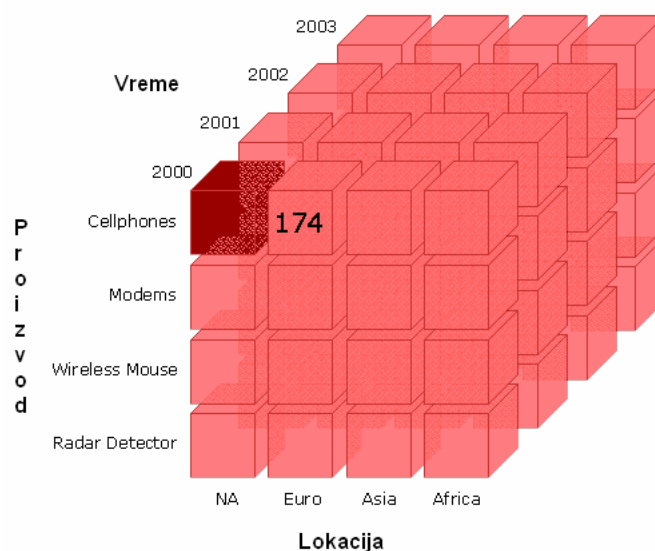
Osnovne operacije koje su bitne za analizu podataka pomoću OLAP kocke su: sečenje na pojaseve (eng. *slice*), komadanje (eng. *dice*), bušenje (eng. *drill down*), zavijanje (eng. *roll up*), rotacija.

Sečenje na pojaseve predstavlja izdvajanje podataka za dati uslov po jednoj dimenziji. Na sledecoj slici je prikazana prodaja za konkretan proizvod - bežični miš, odnosno lokaciju - Azija.



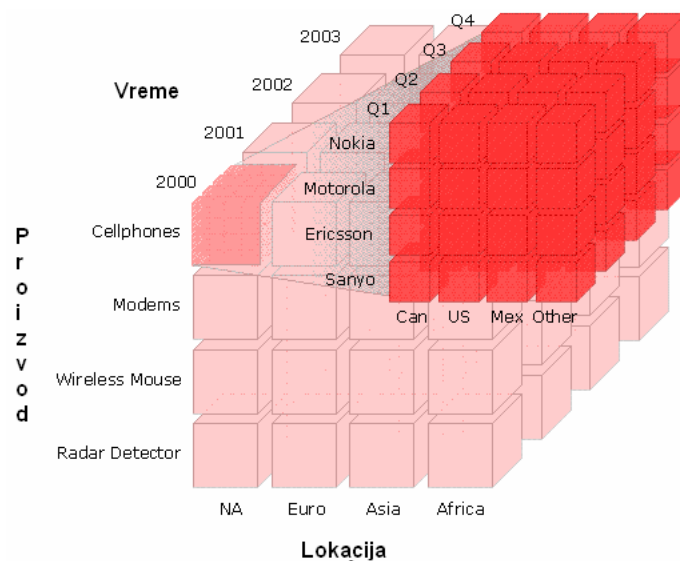
Slika 16: Operacija sečenja OLAP kocke na pojaseve

Komadanje je izdvajanje podataka za uslove po dve ili više dimenzija. Na sledećoj slici je prikazana prodaja telefona za 2000. godinu u Severnoj Americi.



Slika 17: Operacija komadanja OLAP kocke

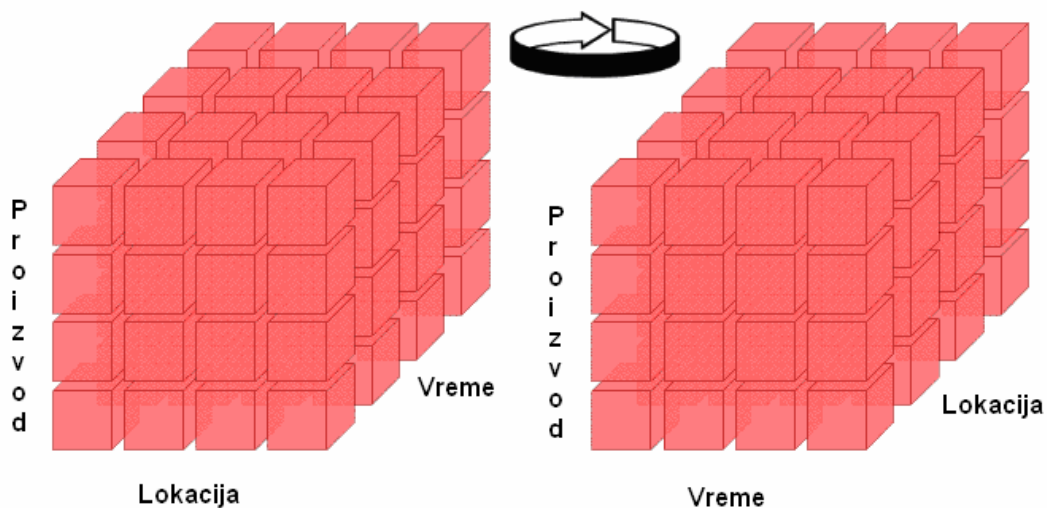
Bušenje predstavlja detaljizaciju kocke, spuštanjem po hijerarhiji dimenzije. Na sledećoj slici je prikazana kombinacija operacija komadanja (prikaz prodaje telefona u Severnoj Americi 2000.godine) i bušenja (vremenska dimenzija je prikazana po kategoriji kvartala, lokacija po državama, a grupe proizvoda po pojedinačnim vrstama).



Slika 18: Kombinacija operacija komadanja i bušenja OLAP kočke

Zavijanje je suprotna operacija od bušenja, tj. predstavlja izdvajanje podataka penjanjem po hijerarhiji neke od dimenzija.

Rotacija se u literaturi naziva i pivotiranje. Predstavlja vizuelizaciju kočke okretanjem dimenzionih osa radi alternativnog prikaza podataka (Slika 19).



Slika 19: Operacija rotacije OLAP kočke

4.6 Tipovi OLAP arhitekture

Sistemi koji pomažu prilikom odlučivanja se često nazivaju OLAP sistemima. Oni omogućuju „pametnim“ korisnicima da intuitivno i brzo obrađuju operativne podatke. Postoje dve osnovne arhitekture OLAP sistema: multidimenzioni OLAP (MOLAP) i relacioni OLAP (ROLAP). Da bi se obezbedile različite analize, MOLAP arhitektura koristi multidimenzione baze, dok ROLAP arhitektura pristupa direktno podacima koji su smešteni u relacionoj bazi podataka.

Sposobnost brzog reagovanja i donošenja odluka na današnjem tržištu predstavlja put ka uspehu. Obim podataka kojima kompanije raspolažu raste velikom brzinom. Kompanije imaju za cilj da efikasno upravljaju tim podacima i na taj način da koriste informacije koje su potrebne prilikom donošenja odluka da bi ostvarile prednost na tržištu. Skladište podataka (eng. *Data Warehouse* - DW) može da bude prvi korak ka upravljanju velikim količinama podataka. DW je postao integrisani deo velikog broja sistema koji pomažu prilikom odlučivanja, zato što se podaci iz različitih izvora smeštaju na jedinstvenu centralnu lokaciju. Ali, samo posedovanje DW-a ne omogućuje zaokruživanje procesa od posedovanja transakcionih podataka do donošenja poslovne odluke. Potrebno je da se korisnicima omogući način da „pročitaju“ informacije koje su skrivene u DW- u. Za to je zadužen OLAP. OLAP omogućuje „pametnim“ korisnicima da upravljaju operativnim podacima, a da pri tom koriste sebi bliske poslovne termine. Npr. koristeći OLAP, korisnik može da „seče“ i „komada“ informacije duž klijentske dimenzije, kao i da posmatra rezultate poslovnog procesa po proizvodima ili tokom nekog vremenskog perioda. Izveštaji mogu da budu urađeni tako da se poslovni proces posmatra iz različitih perspektiva, što omogućuje kako globalni, tako i detaljni pregled bilo kog aspekta poslovanja. Da bi obezbedio potpuno funkcionalne poslovne analize, OLAP sistem treba da podrži:

- najsloženije analize
- veliki broj dimenzija
- velike skupove atomičnih podataka.

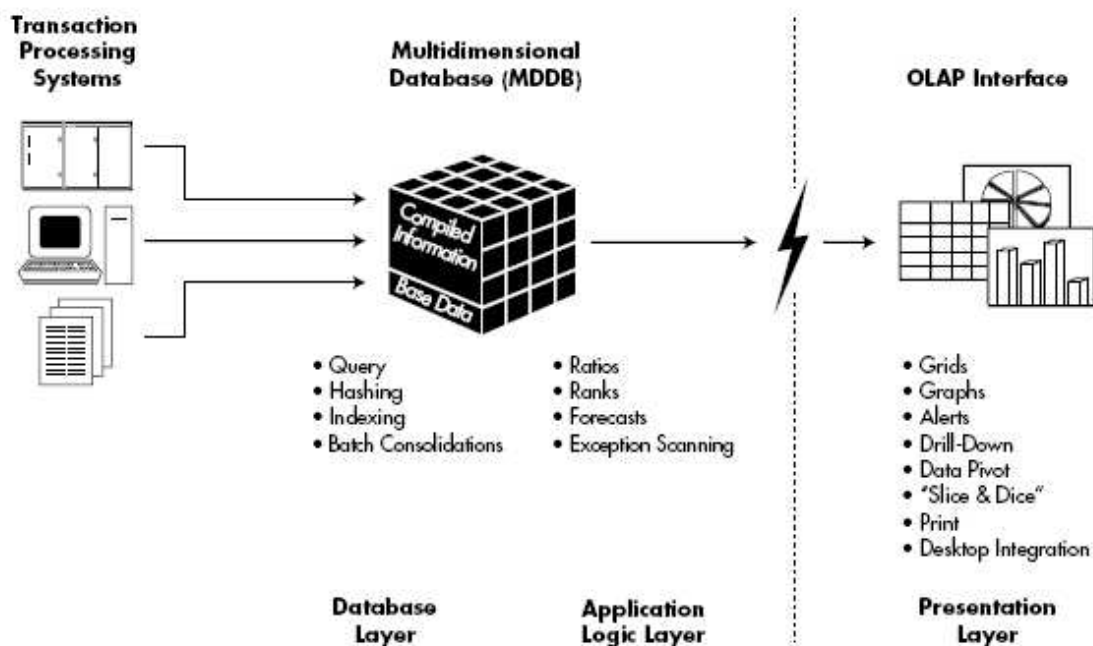
Klasifikacija OLAP proizvoda na MOLAP i ROLAP je zasnovana na arhitekturi OLAP sistema.

4.6.1 MOLAP

MOLAP kao osnovu ima multidimenzionu bazu podataka (eng. *Multidimensional Data Base* - MDDDB). Osnovni stav pristalica ove arhitekture je da podaci moraju da budu čuvani u multidimenzionim strukturama da bi pogled na njih bio multidimenzionalan. Multidimenziona baza se puni podacima iz različitih izvora pomoću serije grupnih, pozadinskih (eng. *batch*) obrada. Kada se jednom MDDDB napuni atomičnim podacima, potrebno je da se uradi niz pozadinskih izračunavanja duž dimenzija i da se tim vrednostima dopune nizovi u MDDDB. Nakon popunjavanja nizova, primenjuju se različiti algoritmi koji poboljšavaju performanse upita. Kada se jednom uradi proces kompilacije, MDDDB je spremna za korišćenje. Korisnik zahteva OLAP izveštaj preko interfejsa, dok logički aplikativni nivo MDDDB pronalazi

sačuvane podatke. Prilikom pravljenja izveštaja se čitaju podaci dobijeni prekompilacijom, a mogućnosti dinamičkih izračunavanja vrednosti koje nisu dobijene prekompilacijom su ograničene.

MOLAP predstavlja klijent – server arhitekturu. U ovom tipu arhitekture MDDB obuhvata nivo baze podataka, kao i aplikativni, odnosno logički nivo. MDDB sistem je odgovoran za čuvanje podataka, pristup podacima i njihovo pronalaženje. Na aplikativnom nivou, MDDB je zadužen za izvršavanje svih OLAP zahteva. Nivo prikaza integrisan sa ovim nivoom formira interfejs preko koga krajnji korisnik daje zahteve i gleda rezultate OLAP analize. Klijent - server arhitektura omogućuje da više korisnika pristupa istoj multidimenzionoj bazi. Prikaz MOLAP arhitekture je dat na sledećoj slici:

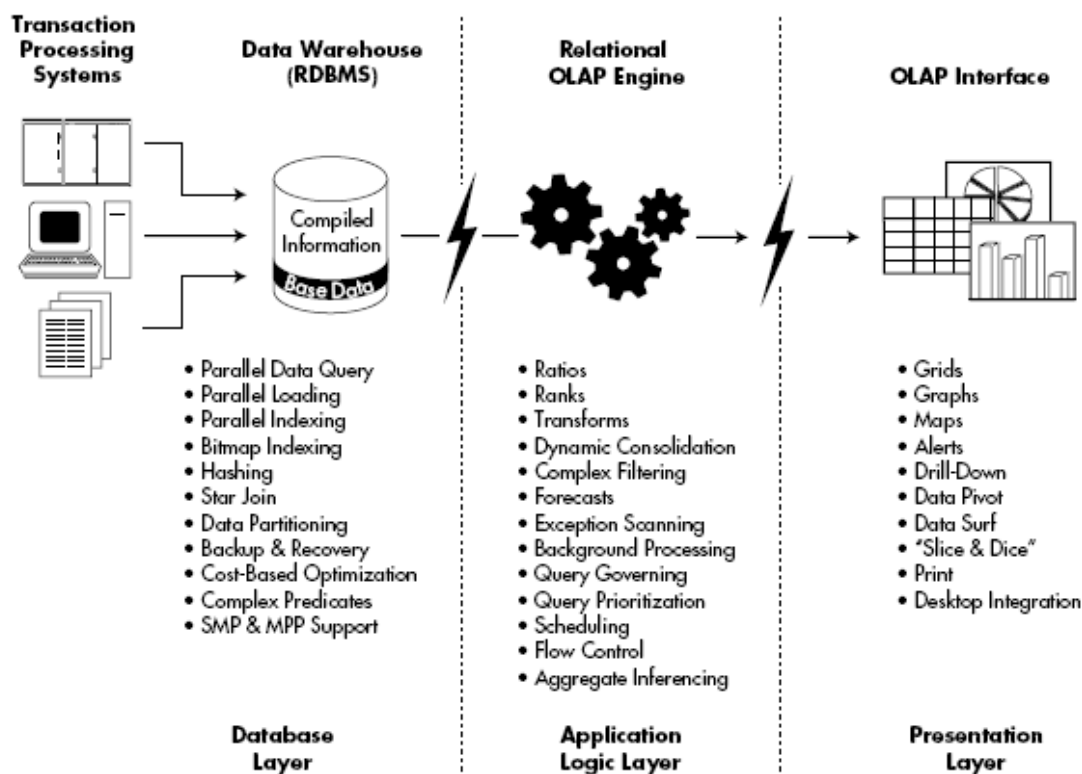


Slika 20: MOLAP arhitektura

4.6.2 ROLAP

ROLAP radi direktno sa relacionim bazama podataka. Osnovni stav pristalica ove arhitekture je da se korišćenjem relacione baze podataka maksimalno iskorišćavaju sve mogućnosti OLAP-a. Pomoću odgovarajućih rutina nad bazom se dobijaju agregirani podaci, ukoliko model podataka to zahteva. Nakon toga se formiraju odgovarajući indeksi nad tabelama koji će poboljšati performanse upita. Kada krajnji korisnik uputi zahtev za analizu, on se dinamički transformiše u niz SQL naredbi. SQL naredbe se prosleđuju relacionoj bazi na izvršenje, a korisniku se prosleđuje rezultujući multidimenzioni skup. ROLAP omogućuje prethodno izračunavanje, kao i dinamičko generisanje rezultata od atomičnih podataka kada je to potrebno. Sistem dizajneru se prepušta optimizacija u smislu pronalaženja odgovarajućeg odnosa između količine prethodnih pozadinskih obrada i skraćivanja vremena odziva prilikom postavljanja upita za dobijanje izveštaja.

ROLAP je je takođe realizovan u obliku klijent – server arhitekture. Na nivou baze se vrši čuvanje podataka, pristup podacima, kao i njihovo pronalaženje. ROLAP uključuje relacionu mašinu koja predstavlja aplikativni logički nivo OLAP sistema i omogućuje formiranje multidimenzionih izveštaja od strane velikog broja korisnika. ROLAP je povezan sa nivoom na kome se vrši prikaz podataka, preko koga krajnji korisnici vrše OLAP analize. Prikaz ROLAP arhitekture je dat na sledećoj slici:



Slika 21: ROLAP arhitektura

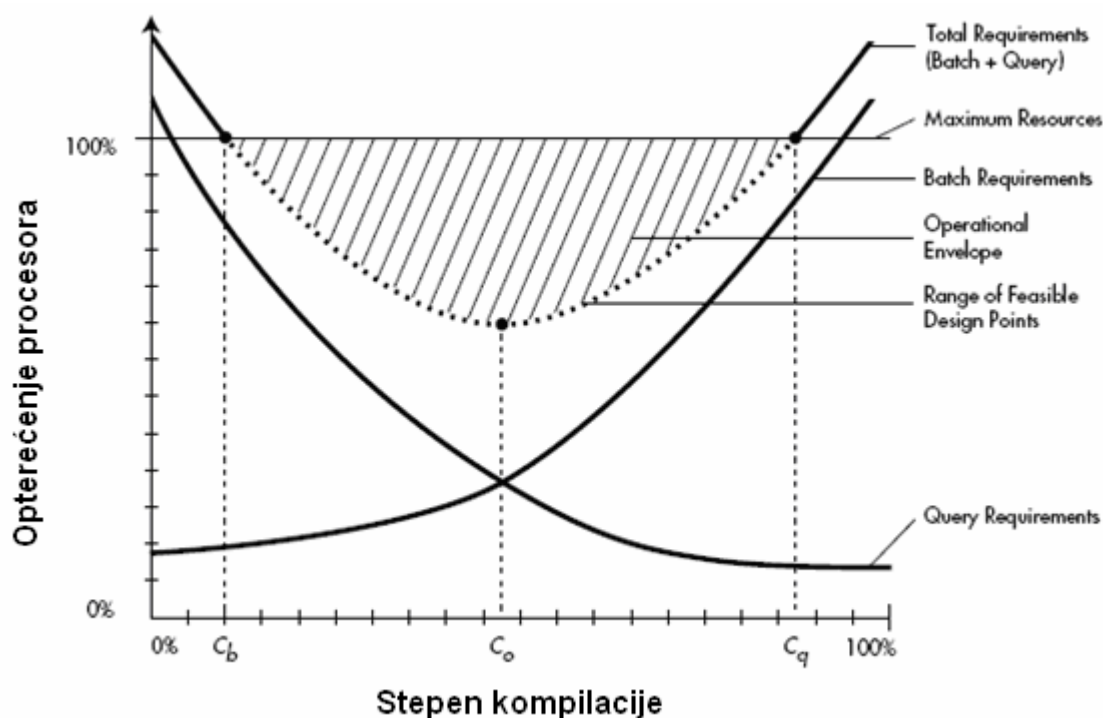
4.6.3 Realizacija OLAP sistema

I ROLAP i MOLAP arhitektura imaju sličan nivo analitičke funkcionalnosti, tj. mogu da odgovore na ista pitanja korisnika. MOLAP to postiže prvenstveno pristupom prethodno izračunatim vrednostima, dok ROLAP koristi izračunate vrednosti iz memorije, ukoliko one postoje, ili se vrše naknadna izračunavanja, ako je to potrebno. OLAP izveštaji prikazuju informacije koje mogu da budu na atomičnom nivou (nivo transakcionih podataka u bazi), kao i na višem, agregiranom nivou. Kada je potrebno da se podaci agregiraju, OLAP sistem može da ih izračuna dinamički ili da uzme vrednosti koje su prethodno izračunate i sačuvane u memoriji. Da bi obezbedio zahtevane performanse, OLAP sistem vrši predizračunavanja nekih, moguće i svih tih vrednosti. Npr. dnevne informacije koje postoje u bazi mogu da budu sumirane prilikom pozadinskih obrada i rezultujuće mesečne vrednosti se mogu sačuvati u bazi. Prilikom zahtevanja izveštaja na mesečnom nivou, ove vrednosti će samo biti pročitane iz baze. Očigledno je da je stepen predizračunavanja (kompilacije) podataka srazmeran broju pozadinskih obrada. Što je više kompilacija zahtevano, biće

više zahteva za pozadinskim obradama. Proces kompilacije podataka obuhvata sledeće:

- Izdvajanje atomičnih podataka iz transakcionog sistema.
- Agregiranje atomičnih podataka
- Indeksiranje atomičnog i agregiranog nivoa podataka, da bi se skratilo vreme odziva prilikom postavljanja upita

Veza između stepena kompilacije i zahtevanih resursa predstavlja nelinearnu funkciju zahteva za pozadinskim obradama (kriva *Batch Requirements* na narednoj slici). Što je veći stepen kompilacije, to je potrebno više pozadinskih obrada, tj. potrebno je više procesorskog vremena. Funkcija performansi upita je takođe nelinearna funkcija zavisnosti potrebnih resursa od stepena kompilacije (kriva *Query Requirements* na Slici 22), ali u ovom slučaju, što je veći stepen kompilacije, to su bolje performanse upita, a samim tim je potrebno manje procesorskog vremena za njegovo izvršavanje. Sabiranjem ove dve funkcije dobija se rezultujuća kriva (*Total Requirements* na narednoj slici) koja predstavlja ukupno opterećenje procesora u zavisnosti od stepena kompilacije.



Slika 22: Analiza mogućnosti implementacije OLAP sistema

Horizontalna linija (*Maximum Resources*) na prethodnoj slici pokazuje maksimalni raspoloživi nivo procesorskih resursa na serveru. Treba da se primeti da će porast, odnosno smanjenje snage serverskog hardvera uticati na pomeranje ove granice. OLAP sistem može da se realizuje u granicama između ukupnih zahteva za procesorom i raspoloživih procesorskih resursa, što je prikazano šrafiranom oblasti (*Operational Envelope*) na prethodnoj slici. Karakteristične tačke te oblasti su: C_b – predstavlja stepen kompilacije na kom su minimizovani zahtevi za pozadinskim obradama, C_q – predstavlja stepen kompilacije gde se postižu najbolje performanse

upita, C_o – optimalna tačka između performansi upita i količine zahteva za pozadinskim obradama. Treba uočiti da se u tačkama C_b i C_q postiže 100% iskorišćenost procesora, dok je u tački C_o njegovo minimalno opterećenje.

Promenljive koje utiču na dizajn OLAP sistema su:

- Hardver servera: Povećanjem procesorske snage dolazi do povećanje nivoa *Maximum Resources* krive, a samim tim omogućuje se veća fleksibilnost prilikom dizajniranja OLAP sistema. Smanjenje procesorske snage će izazvati suprotan efekat.
- Performanse upita: Složeniji upiti, kao i četo postavljanje upita utiču na rast *Query Requirements* krive, što smanjuje površinu oblasti *Operational Envelope* za realizovanje OLAP sistema. Jednostavniji upiti, kao i smanjenje frekvencije njihovog postavljanja dovešće do suprotnog efekta.
- Broj dimenzija: Povećanje broja dimenzija u sistemu prouzrokuje rast *Total Requirements* krive zbog potrebe za većim brojem pozadinskih obrada, kao i zbog komplikovanja upita, a samim tim će doći do smanjenja fleksibilnosti prilikom dizajniranja OLAP sistema. Smanjenjem broja dimenzija se prouzrokuje suprotan efekat.
- Obim atomičnih podataka: Rast količine atomičnih podataka će povećati potrebno procesorsko vreme za sve operacije, a samim tim će doći do rasta *Total Requirements* krive, odnosno do smanjenja fleksibilnosti prilikom dizajniranja sistema. Smanjenje obima podataka će dati suprotan efekat.
- Promenljivost podataka: Ukoliko se pravila agregacije atomičnih podataka često menjaju ili ako korisnici mogu da definišu svoja sopstvena pravila, *Batch Requirements* kriva će biti znatno strmija. To će dovesti do smanjenja mogućnosti prilikom dizajniranja OLAP sistema.

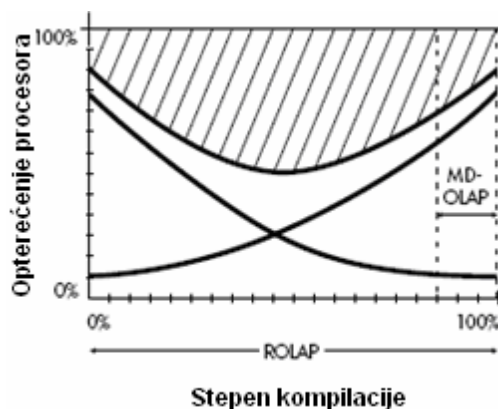
Od pet navedenih promenljivih, menjanje obima atomičnih podataka, hardverska sposobnost, promenljivost podataka i zahtevi krajnjih korisnika linearno utiču na krive prikazane na prethodnoj slici. Dimenzionalnost ima eksponencijalni uticaj na zauzeće procesora kod povećanja broja kompilacija.

4.6.4 Poređenje ROLAP-a i MOLAP-a

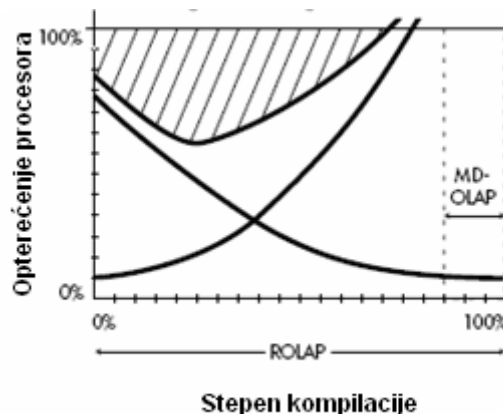
Performanse upita se poboljšavaju povećanjem stepena kompilacije podataka, ali to povećava broj zahteva za pozadinskim obradama. ROLAP arhitektura ne utiče, niti zavisi od količine kompilacija podataka, odluka o tome se prepušta sistem dizajneru. Za MOLAP arhitekturu se vezuje visok stepen kompilacije podataka.

Pojam promenljivosti podataka predstavlja stepen do kog se podaci i strukture podataka menjaju tokom vremena. Podaci sa niskim stepenom promenljivosti se označavaju kao relativno konstantni. Npr. vremenski podaci imaju mali stepen promenljivosti: određeni dani se uvek grupisu u mesece, a meseci u godine. Suprotno od toga, proizvodi, zaposleni i operativni podaci su veoma promenljive prirode.

Naime, zaposleni menjaju radna mesta, a proizvodi menjaju svoje kategorije. ROLAP sistem može da bude implementiran za bilo koji stepen kompilacije i zbog toga može da podrži visok stepen promenljivosti podataka. MOLAP zahteva visok stepen kompilacije i zato se ne može primeniti na sisteme sa vrlo promenljivim podacima. Na sledećim slikama je prikazana mogućnost implementacije ROLAP-a i MOLAP-a u zavisnosti od promenljivosti podataka:



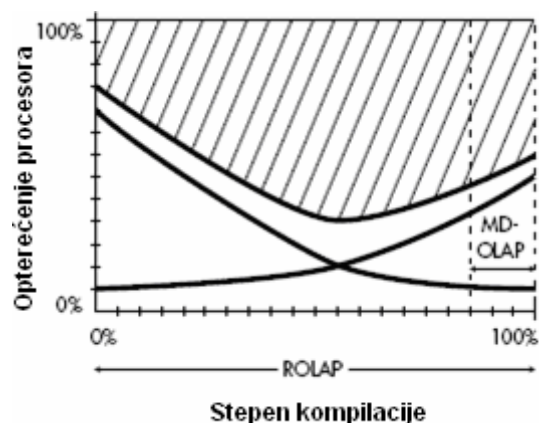
Slika 23: Mogućnost imlementacije ROLAP-a i MOLAP-a kod male promenljivosti podataka



Slika 24: Mogućnost imlementacije ROLAP-a i MOLAP-a kod velike promenljivosti podataka

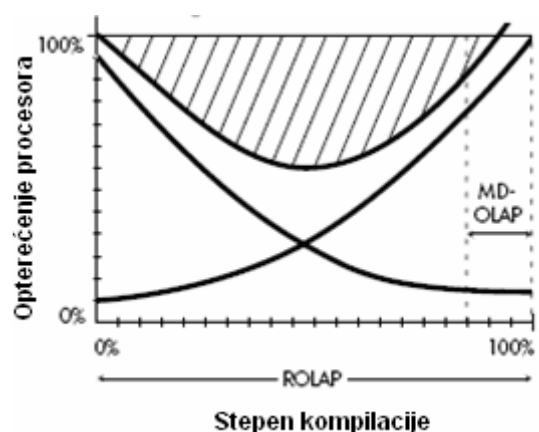
Što se tiče dimenzionalnosti OLAP sistema, naredna tri primera će ilustrovati efekat povećanja broja dimenzija za uzet model podataka i sposobnosti ROLAP i MOLAP sistema da to podrže.

Sa tri dimenzije u modelu podataka, zahtevi za pozadinskim obradama su relativno mali. Sa adekvatnim resursima, moguće je postići visok stepen kompilacije i na taj način omogućiti kratko vreme odziva prilikom postavljanja upita. U ovom slučaju, obe arhitekture će efikasno raditi nad trodimenzionalnim modelom podataka, što je prikazano na sledećoj slici:



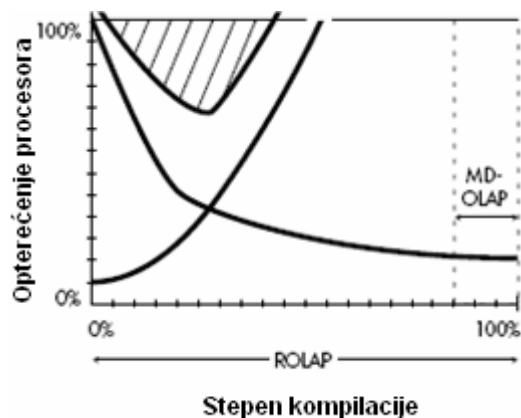
Slika 25 : Implementacija ROLAP i MOLAP arhitekture u slučaju sistema sa tri dimenzije

Model podataka sa deset dimenzija porastom stepena kompilacije zahteva veće zauzeće procesora zbog povećanih zahteva za pozadinskim obradama. Potpuna kompilacija (100%) u ovom slučaju nije dostižna, pa MOLAP sistem nije adekvatan izbor. ROLAP sistem sa 50% do 60% kompilacija je dobar izbor. Grafički prikaz je dat na sledećoj slici:



Slika 26: Implementacija ROLAP i MOLAP arhitekture u slučaju sistema sa deset dimenzija

U slučaju da model podataka ima trideset dimenzija, MOLAP arhitektura nije moguće rešenje zato što je vreme potrebno za kompilaciju multidimenzionalne baze mnogo veće od raspoloživog procesorskog vremena. Jedini mogući izbor u ovom slučaju je ROLAP arhitektura sa malim stepenom kompilacije. Grafički prikaz je dat na sledećoj slici:



Slika 27: Implementacija ROLAP i MOLAP arhitekture u slučaju stema sa trideset dimenzija

Za datu količinu atomičnih podataka, stepen kompilacije je ključna stvar za veličinu baze. Što je veći broj kompilacija atomičnih podataka u više agregatne nivoe, veličina baze se povećava. ROLAP arhitektura može da podrži veći obim atomičnih podataka od MOLAP arhitekture.

Na kraju može da se zaključi da je ROLAP dobar izbor arhitekture koji može da pruži osnovu najširem skupu sistema koji daju podršku odlučivanju i da podrži najveći broj OLAP zahteva. MOLAP može biti dobro rešenje za sisteme sa malim obimom podataka i sa ograničenom dimenzionalnošću.

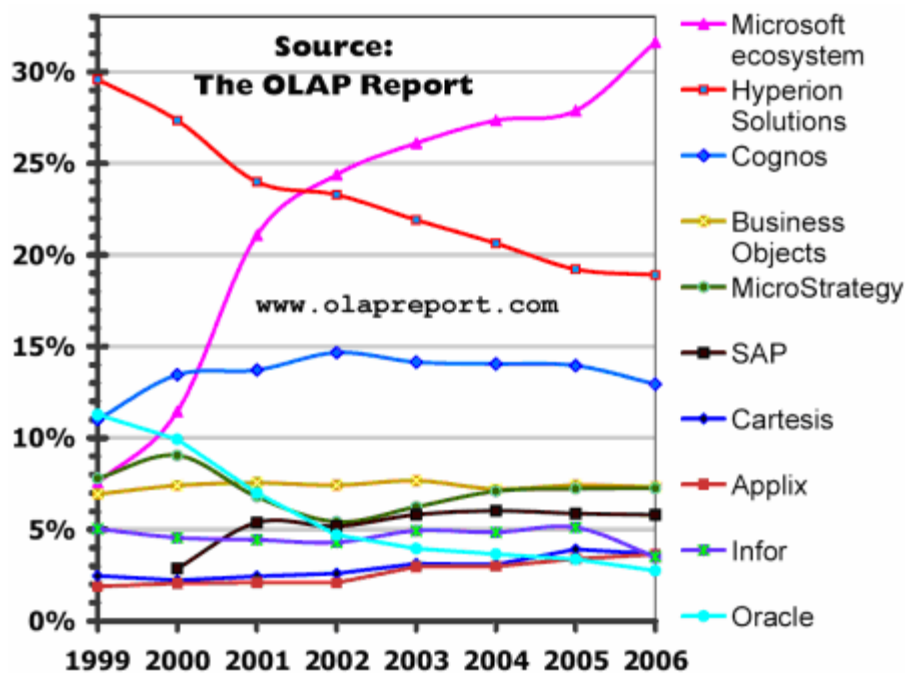
4.7 OLAP alati

Danas na tržištu postoji veliki broj OLAP alata. Nameće se pitanje kako da se izabere odgovarajući. Alati su ocenjivani na osnovu sledećih kriterijuma [10]:

- Sposobnost da obezbede paralelizam u korišćenju RSUBP i hardvera – ovo znatno povećava performanse alata i pomaže da popunjavanje kocki podataka bude što brže.
- Performanse – prilikom obezbeđenja paralelizma u radu, alat treba da podjednako brzo popunjava kocke podacima, kao i da čita podatke iz kocki.
- Mogućnost prilagođavanja – sve više se OLAP alati koriste kao napredni alati za izveštavanje. Razlog je, naročito kod ROLAP implementacije, to što je u najvećem broju slučajeva moguće koristiti OLAP kao alat za izveštavanje. U tim slučajevima jednostavnost uklapanja u postojeći sistem postaje bitan faktor prilikom izbora alata.
- Zaštita podataka – pošto je korišćenje OLAP alata namenjeno većem broju korisnika, potrebno je da se obezbede različiti nivoi zaštite.

- Podržavanje meta podataka – informacije o relacionim podacima se čuvaju u meta objektima koji obezbeđuju novi pogled na te podatke. Neki meta objekti predstavljaju osnov za direktan pristup relacionim podacima, dok drugi opisuju veze između osnovnih meta objekata. Svi meta objekti mogu da se grupišu preko svojih veza u nov meta objekat – dimenzioni model (model kocke).

Na osnovu navedenih kriterijuma, primat na tržištu su do 2006. godine imali OLAP alati sledećih proizvođača [14]: Microsoft, Hyperion, Cognos, MicroStrategy i Business Objects, kao što je prikazano na sledećoj slici:



Slika 28: Stanje na tržištu OLAP alata u periodu od 1999. do 2006. godine

2007. godine je došlo do značajnih promena na tržištu OLAP alata. Naime, Oracle je kupio Hyperion, SAP je kupio Business Objects, a IBM je kupio Cognos. Dodatno, tendencija je da OLAP alati više ne postoje samostalno, već da predstavljaju sastavni deo BI proizvoda, tako da više nije moguće izračunati prisutnost pojedinih proizvođača na tržištu sa tačnošću kao prethodnih godina.

5. OPIS PROBLEMA I PREDLOG REŠENJA

Autor ovog teksta radi na mestu samostalnog projektanta u Banci Poštanskoj štedionici a. d. Jedno od zaduženja mu je održavanje Projekta pravne službe, koji je vezan za naplatu duga od vlasnika tekućih računa koji su u nedozvoljenom minusu. Najčešći zahtevi u vezi sa ovim projektom su izrada izveštaja koji predstavljaju preglede podataka o tim računima. Da bi korisnici izveštaja imali što potpuniju sliku o vlasnicima tih računa, o stanjima na računima, kao i o samom toku dolaska u nedozvoljeni minus i procesu otplate duga, potrebno je da imaju uvid u što veću količinu podataka. S druge strane, podaci treba da budu grupisani i prikazani na takav način, da izveštaji budu pregledni i jednostavni za korišćenje. Za svaki taj skup podataka, potrebno je da se napiše pojedinačni program. Takođe, trebalo bi da podaci budu sortirani po različitim stavkama, radi njihove bolje preglednosti, u cilju potpunije analize. Svaka potreba za novom perspektivom prema istim podacima traži izradu novog izveštaja, tj. novog programa. Za to je potrebno pisanje velikog broja programa. Samim tim se dolazi do povećanog broja obrada, tj. povećava se vremenski interval za koji korisnici imaju podatke na raspolaganju, a njihovom analizom se dolazi do informacija koje su bitne za poslovanje.

Primena OLAP-a može da bude rešenje za ovakav tip problema. Potrebno je da se podaci dobro organizuju, da bi mogli da budu na raspolaganju korisnicima koji onda mogu sami da vrše njihovu vizualizaciju na način koji im odgovara.

U ovom radu biće obrađena izrada skupa izveštaja koji se odnosi na praćenje toka naplate duga od vlasnika tekućih računa. Prvenstveno bi trebalo da se objasni proces nakon kog tekući račun postaje predmet Službe pravnih poslova. Procedura je sledeća: kada račun dospe u nedozvoljeni minus, on se blokira. Nakon toga se šalju dve opomene korisniku da vrati dug. Ako vlasnik računa ne uplati novac na ime duga u roku od nekoliko meseci, postavlja mu se status koji označava da račun prati Služba pravnih poslova. Ponovo se klijentu šalju opomene. Broj poslatih opomena zavisi od procene referenta koji vodi predmet. Ako se ni tada dug ne naplati, pokreće se sudski postupak. Kada vlasnik tekućeg računa vrati dug Banci, postavlja mu se status „otplaćen dug“ i račun se gasi.

Prilikom pokušaja da se neki od ovih izveštaja urade *batch* obradama, obrade su trajale i do 40 minuta, u zavisnosti od trenutnog opterećenja sistema. Problem je u tome što postoji velika količina podataka u raznim tabelama. Takođe, ti podaci su u različitim formatima, pa je potrebno i vreme za njihovu konverziju. Rešenje je bilo da se izdvoje, prečiste i objedine potrebni podaci iz tih tabela. Izdvajanje podataka znači uzimanje samo onih kolona iz tabela koje će se koristiti u izveštajima. Prečišćavanje je uzimanje samo korektnih vrednosti, jer se dešava da prilikom upisa u tabele nije vršena kontrola podataka, tako da se u njima nalaze vrednosti koje se ne mogu uzeti za tačne. Objedinjavanje podataka predstavlja njihovo grupisanje, tako da krajnjeg korisnika ne mora da zanima gde, odnosno u kojim tabelama, se ti podaci nalaze.

6. REALIZACIJA REŠENJA

6.1 Objedinjeni sistemi (eng. *Federated Systems*)

Podaci koji su korišćeni u radu su smešteni na IBM z/OS platformi, na kojoj nivo instaliranog softvera nije dopuštao mogućnost korišćenja OLAP-a. Ideja je bila da se koristi alat koji radi u Windows okruženju. Trebalo je naći način da se pristupi podacima na mainframe-u sa lokalne mašine. Rešenje su tzv. objedinjeni sistemi, odnosno posebni sistemi za upravljanje bazama podataka koji omogućuju izvršavanje heterogenih distribuiranih upita. Pojam distribuiranosti označava da su delovi baze na različitim lokacijama, dok se heterogenost odnosi na različite sisteme za upravljanje bazama podataka. Sa DB2 UDB verzija 8 i odvojenim proizvodom DB2 Relational Connect (IBM-ov proizvod koji se koristi sa DB2 za Unix i Windows) moguće je jednim SQL upitom pristupiti podacima koji se nalaze na različitim platformama. Distribuirani upiti mogu da rade nad podacima koji pripadaju različitim izvorima, kao što su familija DB2 baza podataka, OLE DB izvor, Oracle, Sybase i Microsoft SQL Server baza podataka. Ovakve vrste upita se nazivaju distribuirani zahtevi i ograničeni su samo na čitanje podataka. Krajnji korisnici i klijentske aplikacije imaju predstavu kao da se zahtevani podaci nalaze u jednoj kolektivnoj bazi podataka. DB2 Relational Connect nije potreban ukoliko se pristupa podacima iz familije DB2 baza podataka.

Komponente objedinjenih sistema su DB2 server, koji se naziva objedinjeni server i više različitih izvora podataka. Objedinjeni server uključuje objedinjenu bazu podataka, dok je izvor podataka određen sistemom za upravljanje bazama podataka i samom bazom podataka koja se nalazi na tom sistemu. Sistemski katalog (koji se naziva i globalni katalog) objedinjene baze podataka sadrži informacije o izvorima podataka. Klijentska aplikacija šalje upit objedinjenoj bazi podataka, koji se onda usmerava ka odgovarajućem izvoru podataka, dobijaju se traženi podaci i rezultat se vraća aplikaciji. Distribuirani zahtev može da bude u bilo kojoj od sledećih formi: podupit, skup operatora (unija, presek, razlika) ili spajanje tabela. Preko DB2 SQL- a moguće je referencirati kolone tabela bilo kog tipa podataka koje DB2 podržava, sem LOB (eng. *Large Object*) tipova podataka.

Objekti koji su potrebni da bi se uspostavio i koristio objedinjeni sistem su: omotači, serveri i nadimci. Dodatni objekti uključuju preslikavanje korisničkih imena i lozinki koji su vezani za izvor podataka u nova korisnička imena i lozinke, preko kojih se vrši povezivanje sa objedinjenom bazom podataka, Takođe, obuhvataju i preslikavanje tipova podataka koji pripadaju izvornoj bazi u DB2 tipove podataka, preslikavanje lokalnih funkcija na funkcije u DB2 sistemu i specifikaciju indeksa (radi poboljšanja performansi). Upotreba objedinjenog sistema podrazumeva sledeće korake:

- Povezivanje sa objedinjenom bazom podataka
- Formiranje omotača za svaki tip baze podataka koji će biti uključen u objedinjeni sistem. Funkcija omotača je da omogući komunikaciju između objedinjenog servera i izvora podataka, kao i preuzimanje podataka. Omotači identifikuju module (dll ili biblioteke) koje koriste da pristupe određenom tipu izvora podataka.

- Formiranje servera da bi se definisao izvor podataka za objedinjeni sistem. Podaci o serveru obuhvataju ime servera, tip servera, verziju, ime omotača, informacije o ovlašćenjima i opcijama servera.
- Preslikavanje korisničkih imena.
- Formiranje nadimka za svaku tabelu i pogled u izvoru podataka. Nadimci identifikuju izvorne tabele i poglede na podatke. Aplikacije mogu da ih referenciraju u upitima samo pomoću dodeljenih naziva, a ne po njihovom pravom imenu. Krajnjim korisnicima ovi dodeljeni nazivi izgledaju kao aliasi.
- Pravljenje upita nad tabelama korišćenjem nadimaka.

6.2 Organizacija podataka

Skup podataka koji će biti obrađen u radu je ograničen na račune koji su dospeli u Pravnu službu tokom 2007. i 2008. godine. Pratiće se promene po tim računima tokom te dve godine, a u periodu tokom kog je račun bio predmet Pravne službe. Podaci koji su korišćeni se nalaze u sledećim tabelama:

- Tabela istorije rada po računima, za 2007. godinu, u smislu finansijskih transakcija, i ona sadrži proknjižene promene
- Tabela istorije rada po računima za 2008. godinu
- Tabela predmeta Pravne službe (aktivni predmeti)
- Arhivska tabela predmeta Pravne službe (završeni predmeti)
- Tabela sa podacima o poštama
- Tabela sa podacima o opštinama
- Tabela sa podacima o regionima
- Tabela šifara ekspozitura (sa njihovom pripadnošću poštanskom broju)
- Šifarnik tipova transakcija za tekuće račune (sadrži vrste promena i vrste dokumenata za te promene)

Da bi se dobio dimenzioni model, ove podatke treba izdvojiti, prečistiti i organizovati tako da se dobije šema zvezde ili šema pahulje. U ovakvoj šemi će podaci biti grupisani u tabele koje će biti u međusobnoj zavisnosti preko primarnih i stranih ključeva.

Na početku postupka se izdvajaju podaci iz tabela o predmetima iz Pravne službe kojima je datum ustupanja (datum predaje predmeta Pravnoj službi) u navedenom periodu (u daljem tekstu, tabela T3). Ukoliko dug nije bio otplaćen u istom periodu, datum otplate se postavlja na '01.01.0001'. Za račune koji se nalaze u predmetima su izdvojene sve transakcije iz tabela istorije koje su se desile u periodu između datuma ustupanja i datuma otplate (ako je datum otplate '01.01.0001', onda se posmatraju transakcije do kraja 2008. godine). Od tih podataka će biti formirana tabela činjenica označena kao tabela T1. Kolone koje čine ovu tabelu su: *Broj računa*, *Datum*, *Mesto*, *Vrsta promene*, *Vrsta dokumenta*, *Iznos promene* i *Broj transakcija*. Primarni ključ je *Broj računa*, *Datum*, *Mesto*, *Vrsta promene* i *Vrsta dokumenta*. Iznosi promena svih transakcija koje su istog tipa (ista vrsta promene i vrsta dokumenta) za isti račun, datum i mesto će se posmatrati u zbiru. Taj zbir će biti upisan u kolonu *Iznos promene*, a broj transakcija koje ga čine će biti upisan u kolonu

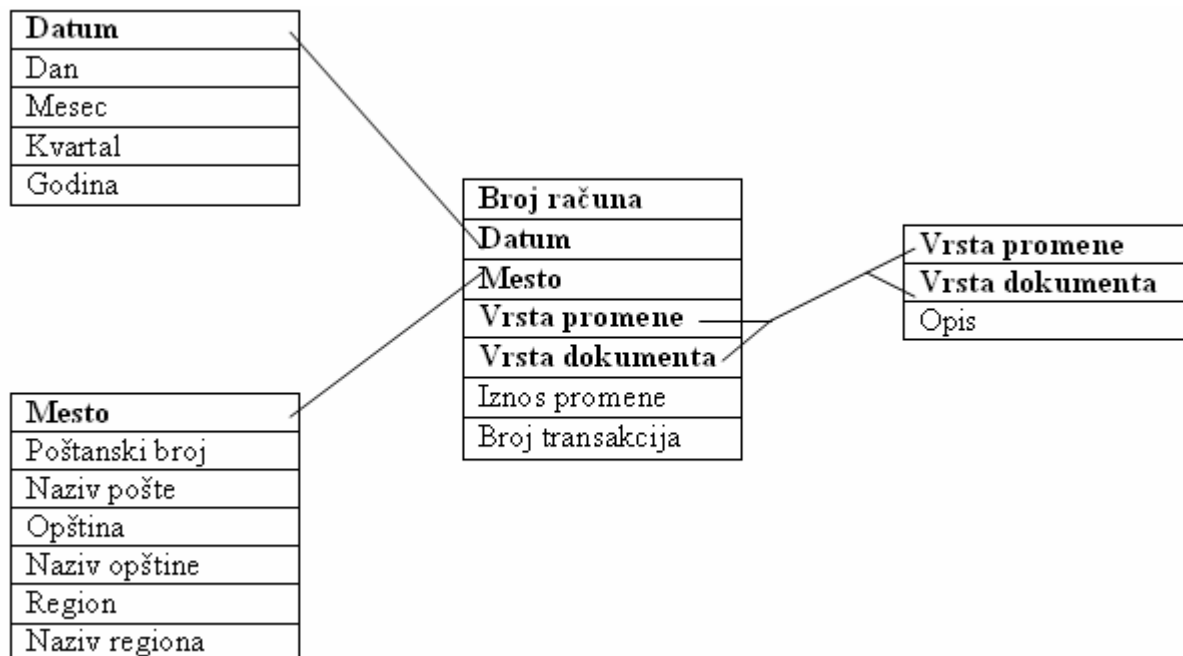
Broj transakcija. *Iznos promene* i *Broj transakcija* predstavljaju mere ove tabele činjenica, na koje će se primenjivati agregatne funkcije sumiranja. Kolona *Datum* se odnosi na vremensku dimenziju i zbog toga treba da se formira sledeća vremenska tabela, čiji je primarni ključ *Date_time_value*, a kolone koje je čine su: *Date_time_value*, *Day_of_month*, *Day_of_week*, *Day_of_year*, *Hour_of_day*, *Julian_day*, *Month_of_year*, *Name_of_day*, *Name_of_month*, *Period_num*, *Quarter_of_year*, *Sequence_num*, *Week_of_month*, *Week_of_year* i *Year*. Ova tabela je formirana u *Warehouse Center*-u koji se nalazi u lokalnoj mreži i obuhvata vremenski period od 01.01.2007. do 31.12.2008. godine. Njoj se pristupa pomoću tzv. Objedinjenih sistema koji su opisani u prethodnom odeljku. Iz nje su izdvojene sledeće kolone: *Date_time_value*, *Day_of_month*, *Name_of_month*, *Quarter_of_year* i *Year* i njihove vrednosti su upisane dimenzionu vremensku tabelu koju čine sledeći atributi: *Datum*, *Dan*, *Mesec*, *Kvartal* i *Godina*.

Dimenziona tabela koja sadrži tipove transakcija je označena kao tabela T2. Primarni ključ ove tabele čine dva atributa: *Vrsta promene* i *Vrsta dokumenta*. Ona se koristi u praćenju uplata (*vrsta promene*=1) i isplata (*vrsta promene*=2). U principu, ne bi smelo da dođe do isplate po tekucem računu koji je predmet Pravne službe, ali to može da se dogodi ako čekovi, rate kredita ili zaduženja po kreditnim karticama dospeju na naplatu. Postoji jedna specifična situacija: ako se dogodi da je korisnik imao uplatu koja je veća od njegovog duga, račun mu gubi status „blokiran“ i može da se izvrši isplata ostatka novca. Takav račun dobija status „otplaćen dug“ nakon izvršene isplate. Isplate sa takvog računa biće obuhvaćene u analizi kao transakcija koja se izvršila u periodu od ustupanja računa pravnoj službi i do postavljanja statusa „otplaćen dug“. Naknade i kamate takođe kao vrstu promene imaju vrednost 2 (isplata), i novac se skida sa računa, ali u korist Banke. Atributi tabele T2 su: *Vrsta promene*, *Vrsta dokumenta* i *Opis*.

Što se tiče mesta vršenja transakcije, potrebno je da se razmotre vrste promena i vrste dokumenata, pa da se na osnovu toga vidi da li je to polje adekvatno popunjeno. Ukoliko je transakcija gotovinska i ako je urađena preko druge banke, prilikom primanja datoteka sa podacima o transakciji, ne dobija se precizan podatak o njenoj lokaciji. U tom slučaju, za vrednost atributa *Mesto* u tabeli T1 se uzima 0. U slučaju bezgotovinskih transakcija nije merodavno uzimanje šifre lokacije, pošto ona ne predstavlja mesto na kom se transakcija desila (recimo, u slučaju uplate zarada, u šifri lokacije je šifra preduzeća koje je uplatilac). I u ovom slučaju je vrednost atributa *Mesto* jednaka 0. Smatraće se da su poštanski broj, opština i region za *Mesto* čija je vrednost 0 takođe jednaki 0. Izuzetak kod bezgotovinskih transakcija predstavljaju razne naknade, kamate, rate kredita i zaduženja po kreditnoj kartici, koja se obračunavaju i skidaju sa računa pozadinskim obradama. Kod njih šifre mesta imaju različite vrednosti, ali će sve biti objedinjene i posmatrane kao šifra Centrale Banke. U slučaju da se transakcija izvrši u Banci Poštanskoj štedionici, u polju *Mesto* će se nalaziti odgovarajuća šifra lokacije. Na osnovu te šifre, a pomoću prethodno navedene tabele šifara ekspozitura, određuje se pripadnost šifre lokacije poštanskom broju. Ako je transakcija izvršena u Pošti, šifra (*Mesto*) će imati vrednost poštanskog broja. Iz tabela pošta i regiona mogu da se dobiju podaci o opštini i regionu kom ti poštanski brojevi pripadaju. Na taj način se dobija tabela koja predstavlja dimenziju lokacije, čiji je primarni ključ *Mesto*. Njene kolone su: *Mesto*, *Poštanski broj*, *Naziv pošte*, *Opština*, *Naziv opštine*, *Region* i *Naziv regiona*.

Tek nakon ove analize koja se odnosi na vrednosti kolone *Mesto* u tabeli T1, moguće je da se ona napuni podacima, pošto se sumiranje iznosa i brojanje transakcija vrši i po toj koloni.

Prethodno opisane tabele čine šemu zvezde koja je prikazana na slici:



Slika 29: Prva šema podataka

Iz ove šeme zvezde mogu da se dobiju odgovori na sledeća pitanja, na navedene načine:

1. Kolika je ukupna suma uplata za predmete Pravne službe?

Uzima se suma promena sa uslovom ograničenja da je vrsta promene uplata. U pitanju je operacija sečenja po dimenziji tipa transakcije.

2. Kolika je suma uplata na nekoj lokaciji pojedinačno posmatrano po mesecima?

Uzima se suma promena, sa uslovom ograničenja za tu lokaciju i vrstu promene uplatu, a grupisanje podataka se vrši po vremenskoj dimenziji. Ovo predstavlja kombinaciju operacija komadanja (izdvajanje podataka za uslove po dve i više dimenzija) i bušenja (vremenska dimenzija se posmatra po kategoriji meseci).

3. Kolika je suma isplata za neki vremenski period, ali samo za određene vrste dokumenata?

Uzima se suma promena, uslovi ograničenja su traženi vremenski period, za vrstu promene isplatu i za navedene vrste dokumenata.

4. Koliko je uplata bilo u martu 2008. godine na području Niša čija je vrsta dokumenta 22?

Sabira se broj transakcija, uslovi ograničenja su: godina 2008, mesec mart, naziv opštine je Niš, vrsta promene je uplata i vrsta dokumenta je 22.

5. Kolika je suma uplata bila u januaru 2007. u Somboru, ali sa odvojenim prikazom po poštanskim brojevima u opštini Sombor?

Sumiraju se promene sa uslovom ograničenja da su transakcije izvršene u opštini Sombor, u toku januara 2007. godine, a zbrovi se grupišu po poštanskim brojevima.

6. Koliki je broj uplata, a koliki broj isplata koje su se desile na području Vojvodine tokom 2008. godine?

Sumira se broj transakcija sa ograničenjem na regione koji čine Vojvodinu, za 2008. godinu, a grupisanje se vrši po vrsti promene.

7. Kojim tipom transakcije klijenti najčešće otplaćuju dug?

Sumira se broj transakcija, ograničenje je da je tip transakcije uplata, a grupisanje se vrši po vrsti dokumenta.

8. Koji su najčešći tipovi transakcija kojima se vrše isplate sa računa koji pripadaju Pravnoj službi?

Sumira se broj transakcija, ograničenje je da je tip transakcije isplata, a grupisanje se vrši po vrsti dokumenta.

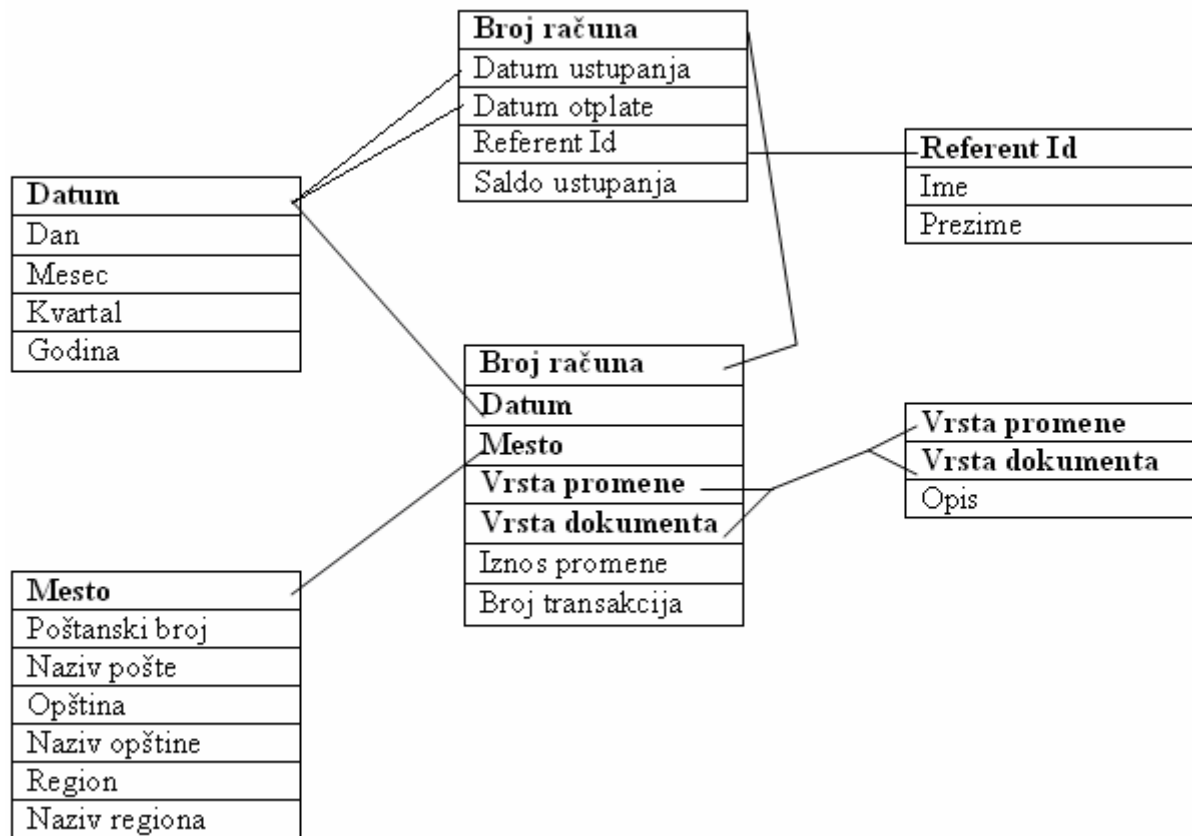
9. Na koji način se kretao priliv sredstava Banci od strane klijenata čiji su računi predmet Pravne službe?

Posmatra se suma iznosa transakcija, ograničenje je da je vrsta promene uplata, a grupisanje se vrši po vremenskoj dimenziji. Grupisanje može da se izvrši prvo po godini, pa po mesecu, a zatim po danu. Prikaz će predstavljati tzv. „bušenje kocke“, odnosno, prvo će biti prikazani sumarni podaci za godinu, a zatim se može izvršiti detaljniji prikaz po mesecima, pa po danima u okviru svakog meseca.

Može da se doda još jedna dimenziona tabela (tabela T3), koja sadrži aktivne i arhivirane predmete Pravne službe, i koja bi se povezala sa tabelom činjenica preko kolone *Broj računa*, koja je njen primarni ključ. Kolone ove tabele su: *Broj računa*, *Datum ustupanja*, *Datum otplate*, *Referent Id*, *Saldo ustupanja*.

Ovoj dimenziji može da se pridruži i tabela podataka o referentima koji vode postupke protiv vlasnika računa koji su u nedozvoljenom minusu. Njen primarni ključ je *Referent Id*, a čine je kolone : *Referent Id*, *Ime* i *Prezime*.

Takođe, može da se izvrši povezivanje *Datuma ustupanja* i *Datuma otplate* sa vremenskom tabelom. Pošto je u *Datum otplate* upisana vrednost '01.01.0001' (u slučaju kada dug nije otplaćen), potrebno je upisati još jedan red u vremensku tabelu, koji će da odgovara tom datumu. Na ovaj način se dobija tzv. šema zvezde sa „spoljnim“ tabelama, koja je prikazana na sledećoj slici:



Slika 30: Druga šema podataka

Tabela T3, u zavisnosti od toga kako se koristi u upitu, može da bude i dimenziona tabela i tabela činjenica. Naime, ako T3 posmatramo kao tabelu činjenica, njena mera će biti *Saldo ustupanja*, a dimenzije *Referent*, *Datum ustupanja* i *Datum otplate*. Moguće je da i *Broj računa* bude mera za tabelu činjenica (T3), a u tom slučaju će na nju biti primenjena agregatna funkcija brojanja. Tada je moguće da se dobiju odgovori na sledeća pitanja:

1. Kolika je ukupna suma koju je u nekom periodu određeni referent dobio da naplati?

Sumiraju se iznosi ustupanja sa ograničenjem da je datum ustupanja u navedenom vremenskom periodu, za traženog referenta.

2. Koliki broj predmeta je dobio svaki referent u toku 2008. godine?

Broje se računati, ograničenje je da je godina ustupanja 2008. a grupisanje se vrši po dimenziji Referent.

3. Koliko predmeta je uspešno rešeno u toku 2. kvartala 2007. godine, sa pojedinačnim prikazom po referentima?

Broje se računati za koje datum otplate pripada 2. kvartalu 2007. godine, a grupisanje se vrši po dimenziji Referent.

4. Koliki je ukupan broj računa koji su dospeli u Pravnu službu u toku maja 2008. godine?

Broje se računi za koje je mesec datuma ustupanja bio maj 2008.

Ako se u tabeli T3 *Broj računa* posmatra ne kao mera, već kao nova dimenzija (dimenziona tabela od jednog reda se dobija iz tabele činjenica, to je tzv. degenerativna dimenzija), tada mogu da se dobiju odgovori na sledeća pitanja:

1. Koji su računi i sa kojim iznosima ustupanja postali predmet pravne službe tokom prvog kvartala 2007. godine?

Posmatraju se iznosi ustupanja, uslov ograničenja je da je datum ustupanja bio u prvom kvartalu 2007. godine, a grupisanje se vrši po brojevima računa.

2. Koji predmeti su uspešno rešeni u toku maja 2008. godine (potrebno je prikazati iznos otplaćenog duga)?

Posmatraju se iznosi ustupanja, uslov ograničenja je da je datum otplate maj 2008. godine, a grupisanje se vrši po brojevima računa.

U slučaju da se tabela T3 posmatra kao dimenziona tabela (odnosna tabela) za tabelu T1, moguće je doći do odgovora na sledeća pitanja:

1. Koliku je sumu novca u toku 2007. godine određeni referent uspeo da naplati? Sumiraju se iznosi promena sa ograničenjem su se desile 2007. godine, da je vrsta promene uplata, za navedenog referenta.

2. Kakav je tok naplate duga po referentima tokom 2008. godine?

Posmatra se suma transakcija, sa ograničenjem da se transakcija desila 2008. godine i da je u pitanju uplata, a grupisanje se vrši po vremenskoj dimenziji i dimenziji referenta. U zavisnosti koje se grupisanje prvo izvrši, dobiće se različiti pregledi podataka. Moguće je da se za svakog referenta pojedinačno prati naplata u toku godine (po mesecima, na primer), a moguće je posmatrati naplatu (npr. po kvartalima u 2008. godini), pa u toku svakog kvartala, naplaćenu sumu po referentima.

3. Koji računi su imali isplate tokom prvog kvartala 2007. godine?

Sumira se broj transakcija sa ograničenjem da je vrsta promene isplata, da je vreme vršenja transakcije prvi kvartal 2007. godine, a grupisanje se vrši po brojevima računa.

4. Po kojim računima je bilo uplata tokom januara 2008. godine?

Sumira se broj transakcija sa ograničenjem da je vrsta promene uplata, vreme vršenja transakcije januar 2008. godine, a grupisanje se vrši po brojevima računa.

5. Koji referent je imao najveću naplatu na teritoriji opštine Novi Sad?

Sumiraju se iznosi promena, sa ograničenjem da je vrsta promene uplata, naziv opštine u kojoj je izvršena transakcija je Novi Sad, a grupisanje se vrši po referentima.

Moguće je da se tabele T3 i T1 posmatraju kao višestruka tabela činjenica (ali bez njihovog spajanja) i tada se može odgovoriti na sledeća pitanja:

1. Kakav je odnos sume koja je dospela na naplatu u toku februara 2007. godine i naplaćene sume za taj mesec?

Sabira se iznos uplata gde je datum vršenja promene bio u februaru 2007. godine, a paralelno s tim se posmatra suma salda ustupanja, tako da je datum ustupanja bio u navedenom mesecu.

2. Koji su referenti „u plusu“ u toku nekog perioda (više su naplatili nego što su dobili za naplatu)?

Posmatra se razlika između sume transakcija za navdeni mesec i sume salda ustupanja koja je dobijena tog meseca za naplatu, a grupisanje se vrši po referentima.

Moguće je uraditi dekompoziciju tabele mesta, tako da se dobiju sledeće tabele: T11 (*Mesto, Pošta*), T12 (*Pošta, Naziv pošte, Opština*), T13 (*Opština, Naziv opštine, Region*), T14 (*Region, Naziv regiona*). Ukoliko bi se uradila i dekompozicija vremenske tabele, dobila bi se šema pahulje. Tada ne bi bilo dupliranja podataka, ali se usporilo izvršavanje upita zbog povećanog broja spajanja tabela, kao što je opisano ranije u radu.

Takođe, moglo bi da se u tabelu T3 doda kolona *JMBG* koja predstavlja matični broj vlasnika računa. Ta tabela bi se onda povezala sa novom dimenzionom tabelom čiji je ključ *JMBG*, a koja sadrži matične podatke o klijentu Banke (pol, godište, prebivalište, mesto rođenja, rezidentnost, itd.). Na taj način bi se znatno proširio spektar potencijalnih pitanja i odgovora koji se mogu dobiti iz raspoloživih podataka.

Ubuduće bi mogla bi da vrši kontinuirana dopuna ovih podataka. Naime, u vremensku tabelu bi se dodali datumi posle 31.12.2008. godine (a svaki sledeći put od datuma poslednjeg punjenja tabele), a u tabelu predmeta računima kojima je datum ustupanja posle 31.12.2008, uz ažuriranje vrednosti datuma otplate za postojeće predmete kojima je datum otplate '01.01.0001'. U tabelu mesta bi se dodale nove lokacije (u slučaju otvaranja nove ekspoziture, npr.), a u tabelu činjenica (istorije finansijskih promena) bi se na ranije opisan način dodali novi redovi.

6.3 Alati koji su korišćeni i način njihove primene

6.3.1 Kontrolni centar

Kontrolni centar (eng. *Control Center*) predstavlja jedan od pratećih paketa koji je deo IBM UDB DB2 paketa, verzija 8. To je grafički korisnički interfejs koji se koristi se za upravljanje i administriranje DB2 instanci i njihovih objekata na različitim operativnim sistemima (z/OS, OS/390, Windows, Linux, ...). Iz Kontrolnog centra mogu da se otvore drugi alati pomoću kojih može da se izvrši optimizacija upita, programa i skriptova, koji omogućuju izvršavanje zadataka vezanih za skladište podataka, pravljenje uskladištenih procedura, i rad sa DB2 i IMS komandama.

Funkcije Kontrolnog centra su:

- Dodavanje DB2 UDB sistema, objedinjenih sistema, DB2 UDB za z/OS i OS/390 sistema, baza podataka, objekata baza podataka.
- Upravljanje objektima baza podataka. Omogućeno je formiranje, menjanje i brisanje baza podataka, prostora tabela, tabela, pogleda, indeksa, trigera i šema. Takođe je omogućeno upravljanje korisnicima.
- Upravljanje podacima. Moguće je punjenje tabela podacima, uvoženje i izvoženje podataka, kao i njihova reorganizacija. Takođe je moguće da se prati statistika o podacima.
- Održavanje putem pravljenja rezervnih kopija i vraćanja sadržaja baza podataka i prostora tabela.
- Konfigurisanje i podešavanje instanci i baza podataka.
- Upravljanje povezivanjem na baze podataka.
- Upravljanje IMS sistemima
- Upravljanje aplikacijama.
- Analiza upita korišćenjem *Visual Explain*-a za praćenje plana na osnovu koga se pristupa tabelama.
- Pristupanje drugim alatima, kao što je npr. OLAP Center.

Pomoću Kontrolnog centra je izvršena realizacija koraka koji su neophodni za uspostavljanje i korišćenje objedinjenog sistema. Tabele kojima su na ovaj način napravljeni nadimci su prepoznatljive (po tim nadimcima) u alatima koji su korišćeni da bi se formirale kocke podataka i višedimenzioni izveštaji.

Kontrolnom centru se pristupa na sledeći način: **Start ->Programs ->IBM DB2 -> General Administration Tools -> Control Center**. Odabirom odgovarajuće baze i povezivanjem sa njom omogućuje se obrada nad njenim objektima.

6.3.2 OLAP Center

Jedan od alata koji su korišćeni za realizaciju rešenja je *OLAP Center* koji je u sklopu *Business Intelligence Tools*, jedne od opcija IBM UDB DB2 paketa, verzija 8. OLAP Centar je grafički korisnički interfejs koji omogućuje pregled, formiranje i menjanje kocki, modela kocki, kao i OLAP meta podataka koji su sačuvani u DB2 katalogu. Nakon formiranja meta podataka, moguće je optimizovati performanse upita koji će se praviti pomoću drugih OLAP alata.

Osnovni zadaci OLAP Centra su:

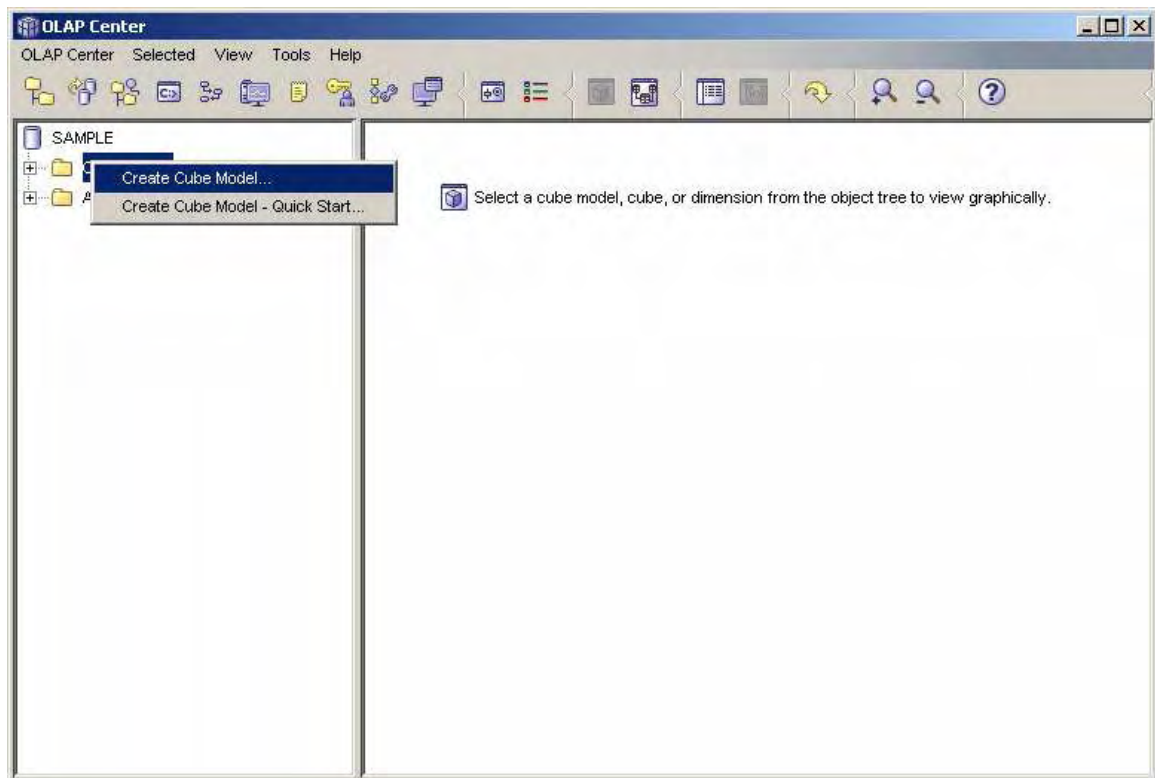
- Formiranje modela kocke
- Formiranje kocke
- Menjanje meta objekata iz OLAP Centra
- Razmena meta podataka između OLAP Centra i drugih OLAP alata
- Optimizacija performansi upita koji se vrše nad kockom

OLAP Centar služi za formiranje potpunog modela kocke, nakon čega se može napraviti sama kocka kao objekat koji sadrži sve ili deo skupa karakteristika modela kocke. DB2 model kocke predstavlja dimenzionu šemu zvezde ili šemu pahulje. Model kocke je grupa određenih dimenzionih objekata grupisanih oko centralnog objekta činjenica. Svaka dimenzija ima višestruki sistem hijerarhija, što povećava fleksibilnost rada sa modelom kocke. Informacije o tome kako su tabele činjenica i dimenzija povezane se takođe čuvaju u modelu kocke. Alati koji „razumeju“ model kocke služe za preglede podataka iz različitih perspektiva (po dimenzijama). Minimalni zahtevi DB2 sistema za upravljanje meta podacima prilikom pravljenja modela kocke su postojanje: objekta činjenica, bar jedne dimenzije, bar jedne odgovarajuće hijerarhije za svaku dimenziju, a zatim je potrebno da su sve dimenzije na odgovarajući način spojene sa objektom činjenica i da su svi iskorišćeni atributi izvedeni iz kolona postojećih tabela.

Prilikom startovanja OLAP Centra (**Start -> Programs -> IBM DB2 -> Business Intelligence Tools -> OLAP Center**), dobija se pomoćni prozor, da bi se izvršila konekcija na željenu bazu.

Naredni koraci predstavljaju deo većeg zadatka, a to je formiranje modela kocke. Desnim klikom na *Cube Model*, a zatim izborom koji je prikazan na Slici 31, dobija se *wizard* čije opcije treba popuniti.

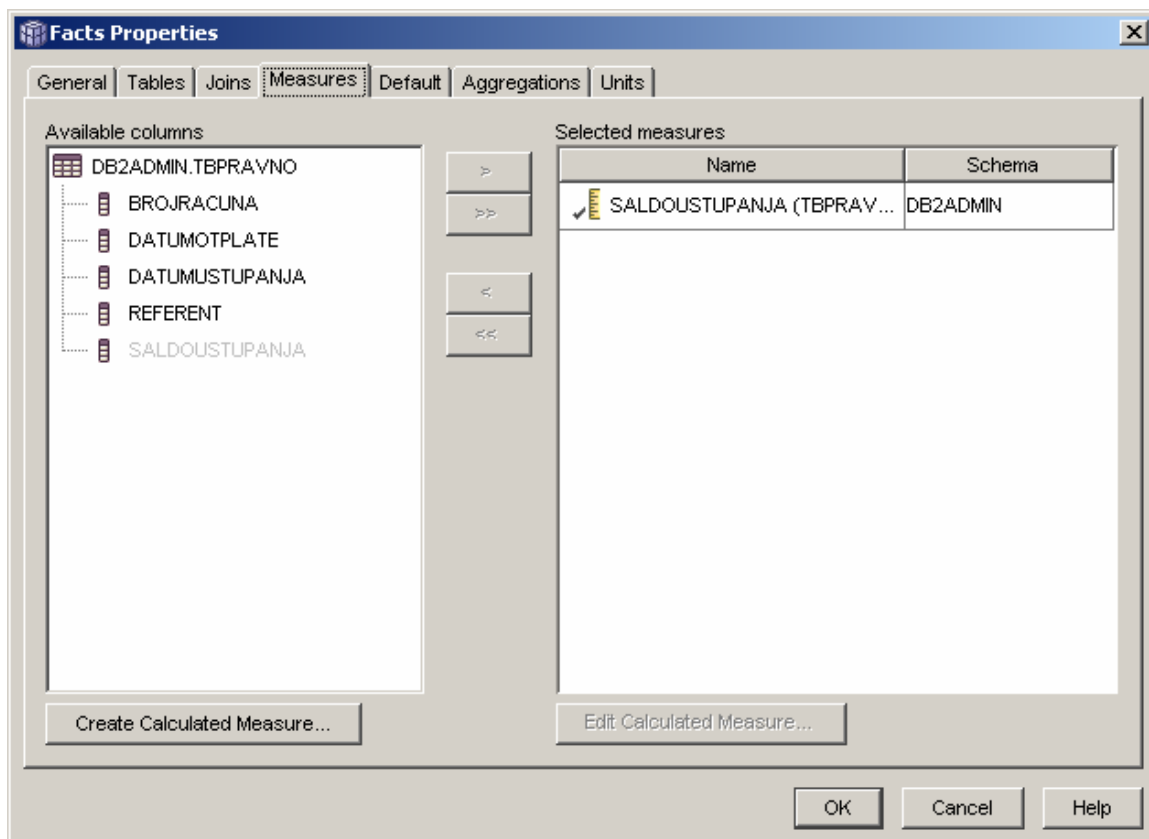
Pomoću *Create Cube Model* može da se jednostavno formira potpun model kocke iz postojećih relacionih tabela. Specificira se ime modela kocke, kao i shema kojoj će pripadati (odnosno kojoj pripadaju tabele koje će se koristiti). Moguće je da se kreira model kocke zasnovan na tabelama iz šeme zvezde ili šeme pahulje. Treba da se definišu ograničenja referencijalnog integriteta za tabelu činjenica i tabele dimenzija, i to pre korišćenja *wizard*- a, jer u suprotnom nemoguće je da se formiraju dimenzije, odgovarajući atributi i spajanja u modelu kocke. Tabela činjenica se formira kao deo već napravljenog modela kocke i onda treba da obuhvata povezane mere koje su relevantne za datu aplikaciju.



Slika 31: Olap Centar

Desnim klikom na model kocke, dobija se padajući meni iz koga treba izabrati opciju *Create Facts* u kojoj se definiše ime tabele činjenica i bira šema kojoj ona pripada. Zatim treba izvršiti izbor tabela koje će činiti tabelu činjenica. U slučaju da se izabere više tabela, mora da se definiše na koji način će se uraditi njihova spajanja. Ovo je mana Olap centra, zato što tabela činjenica u tom slučaju može da sadrži tabele između čijih redova može da se uspostavi bijekcija (inače bi npr. funkcije sumiranja nad kolonama koje predstavljaju mere davale pogrešne rezultate, jer bi se isti redovi jedne tabele sumirali više puta).

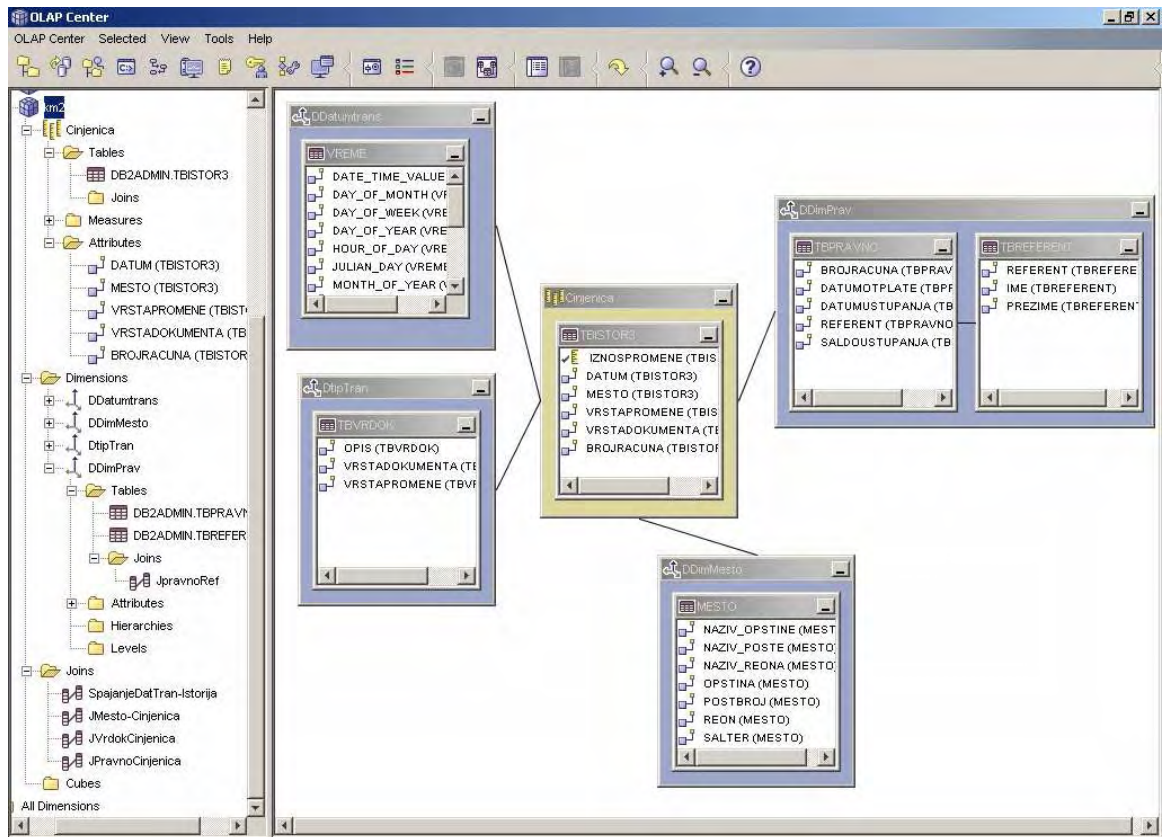
Nakon toga treba izabrati kolone koje će predstavljati mere tabele činjenica (moguće je uzeti postojeće kolone, kao i izračunati „novu kolonu“ iz postojećih), što je prikazano na Slici 32. U slučaju da je izabrano više kolona koje će biti mere, u narednom koraku treba obeležiti kolonu koja će biti podrazumevana vrednost. Za svaku meru klikom na opciju *Aggregations* može da se izabere agregatna funkcija po kojoj će se vršiti izračunavanje za datu meru po dimenzijama. Takođe, može da se definiše i merna jedinica u kojoj će biti izražen rezultat primenjene agregatne funkcije. Na svaki sledeći korak se prelazi pomocu dugmeta *Next*, a moguće je da se vrati na neki od prethodnih odabirom odgovarajuće opcije u levom delu prozora. Nakon formiranja tabele činjenica, njene karakteristike se mogu izmeniti desnim klikom na nju i odabirom opcije *Properties*.



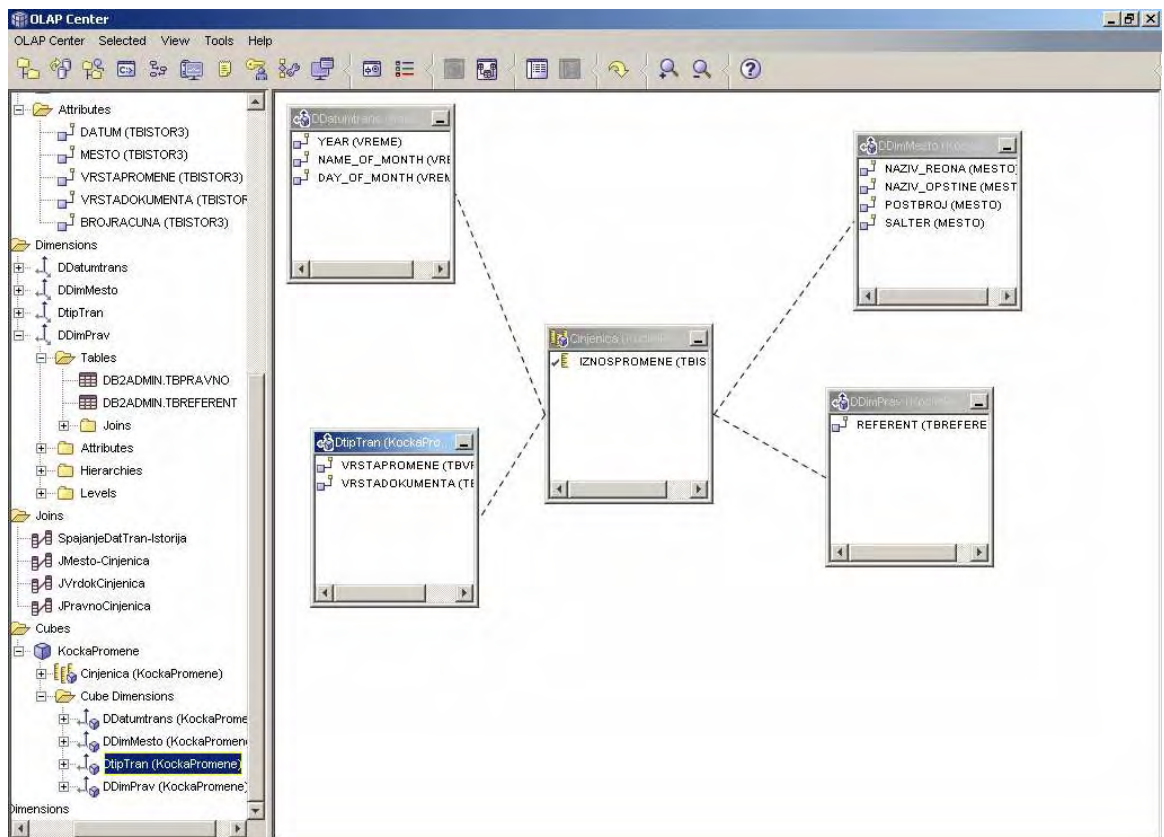
Slika 32: Formiranje mere za tabelu činjenica

Sledeći skup koraka je formiranje dimenzija modela kocke. Desnim klikom na *Dimensions*, a zatim odabirom opcije *Create Dimension* dobija se sličan niz opcija kao i pri formiranju tabele činjenica. Naime, definiše se ime dimenzije, shema kojoj pripada, izvrši se odabir tabela koje će činiti željenu dimenziju (ukoliko je u pitanju više tabela treba izvršiti spajanje po odgovarajućim atributima), a zatim se izaberu kolone iz izdvojenih tabela. Zatim treba da se izabere tip dimenzije koja se kreira-*Time* (ukoliko je u pitanju vremenska dimenzija) ili *Regular* (u ostalim slučajevima). Potom je potrebno izvršiti spajanje tabele činjenica i dimenzione tabele po odgovarajućem atributu. Kao i slučaju izmene tabele činjenica, moguće je izvršiti i izmenu formirane dimenzije desnim klikom na nju i odabirom opcije *Properties*. Kao što je analizirano ranije u radu, svaka dimenzija mora imati bar jednu hijerarhiju. Hijerarhija se formira desnim klikom na formiranu i odabirom opcije *Create Hierarchy*. Definišu se ime hijerarhije, njena shema i nivoi koji je čine. Treba napomenuti da jedna dimenzija može imati više hijerarhija. Na Slici 33 je dat primer modela kocke dobijenog iz šeme podataka koja je opisana ranije u radu.

Iz postojećeg modela kocke moguće je kreirati jednu ili više kocki. Desnim klikom na folder *Cube* i biranjem opcije *Create Cube* dobija se *wizard* za formiranje kocke. Definišu se ime kocke, shema, zatim se biraju mere iz skupa mera modela kocke. Nakon toga, treba izabrati dimenzije koje predstavljaju podskup skupa dimenzija iz modela kocke i za svaku dimenziju treba izabrati tačno jednu hijerarhiju. Na Slici 34 je kocka koja je izvedena iz prethodno prikazanog modela.



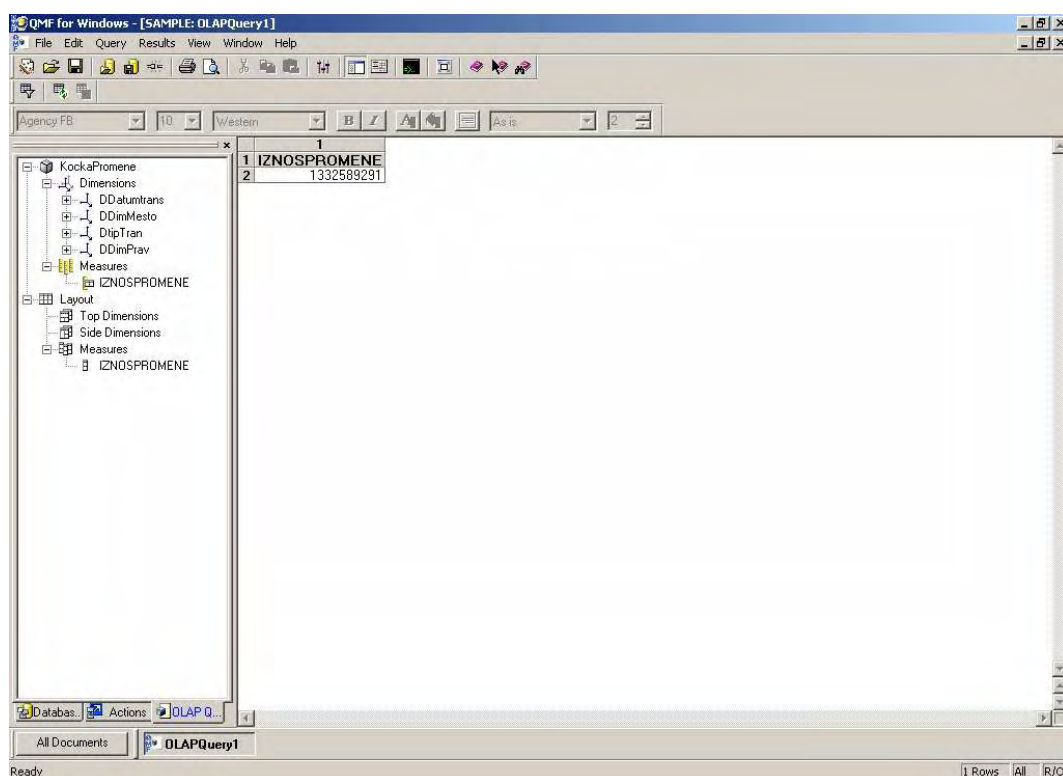
Slika 33: Model kočke



Slika 34: Kocka vraćanja duga

6.3.3 QMF/ Windows

Upiti nad formiranim kockama su rađeni pomoću alata *QMF for Windows* verzija 8.1. Prvo je potrebno je da se odabere odgovarajući server i da se izvrši konekcija na njega. Nakon toga se bira opcija *OLAP Cubes*, a potom model kocke, kao i kocka u okviru tog modela. Desnim klikom na željenu kocku i odabirom opcije *Open* dobija se ukupna vrednost podrazumevane mere na koju je primenjena izabrana agregatna funkcija i to po svim dimenzijama zajedno (kao što je prikazano na sledećoj slici).



Slika 35: QMF for Windows

Jednostavnim prevlačenjem dimenzija u odgovarajući *Layout* (prozor u levom delu ekrana) mogu se dobiti različiti izveštaji. U zavisnosti od željene forme izveštaja, dimenzije mogu da se postave u opciju *Side Dimensions*, koja odgovara y osi, ili *Top Dimensions* koja odgovara x osi koordinantnog sistema. Moguće je i da se više dimenzija postavi na istu osu. Takođe, postoji mogućnost spuštanja po hijerarhiji dimenzija, odnosno prikaza sumarnih podataka po nivoima hijerarhije. Dodatno, desnim klikom na odgovarajuću dimenziju i odabirom opcije *Filter*, može da se uradi prikaz samo po nekim vrednostima izabrane dimenzije. Različitim načinima prikaza se vrše operacije nad kockom, kao što je opisano ranije u radu.

7. REZULTATI

U ovom poglavlju će biti prikazani neki od izveštaja koje korisnik može da napravi iz opisanih struktura podataka, a koji mogu da se koriste u analizi otplate duga klijenata Banke. Akcentat je stavljen na pregled naplate duga po referentima koji vode postupke, iz razloga što su najčešći zahtevi koje autor teksta dobija upravo za izradu takvih izveštaja. Treba napomenuti da su podaci koji su prikazani u radu isključivo testni.

Jedan od najjednostavnijih izveštaja bi bio sumarni pregled naplaćenih iznosa po referentima. Dodatno, u izveštaj može da se postavi i vremenska dimenzija, tako da se dobija pregled naplate po godinama za referente. Forma tog izveštaja je prikazana na sledećoj slici i predstavlja operaciju komadanja kocke:

1	2	3
YEAR	REFERENT	IZNOSPROMENE
	??	-10000
	AC	24194264
	DJ	64249491
	DL	30157164
	DR	40244007
	LP	2854341
	NS	8165626
2007	SB	38472625
	SM	-3308478
	SP	52803233
	VD	50488786
	VB	5482569
	ZI	42356266
	ZP	8895032
	All values for 2...	365044926
	??	23764877
	AC	26199666
	BS	2498466
	DJ	6947933
	DL	140936516
	DR	65175157
	DV	-20000
	LP	104030214
	MD	91628916
2008	NS	36577188
	SB	58395221
	SM	36566600
	SP	88989307
	VD	76302129
	VB	45696905
	ZI	98314504
	ZP	65540766
	All values for 2...	967544365
	All values	1332589291

Slika 36: Tabelarni prikaz naplate duga po godinama i referentima

Na ekranu su prikazane 2007. i 2008. godina, jer su samo ti podaci obuhvaćeni. Klikom na '+' vrši se operacija bušenja kocke. Dobijaju se vrednosti za mesece, a ako se ide dalje u dubinu i za dane u mesecu, kao što je prikazano na sledećoj slici:

1	2	3	4	5
YEAR	NAME_OF_MONTH	DAY_OF_MONTH	REFERENT	IZNOSPROMENE
11			DL	-170000
11			DR	-149930
11			LP	-150000
11			MB	-150000
11			NS	-550000
11			SB	-40000
11			SM	-650000
11			SP	-150000
11			VD	-150000
11			VD	-150000
11			ZI	-180000
11			ZP	-1150000
11			All values for 11	2332070
11			DR	-234781
11			VD	-600000
11			ZI	400000
11			ZP	26674
11			All values for 12	-408107
11			ZI	-151563
11			ZP	-112020
11			All values for 13	-263583
11			MB	-303128
11			ZP	300000
11			All values for 14	-3128
11			DR	3309829
11			All values for 16	3309829
11			DJ	-5981966
11			NS	-106252
11			SM	1904562
11			ZP	-300000
11			All values for 17	-4483656
11			DL	-800000
11			SP	-193073
11			All values for 18	-993073
11			MB	500000

Slika 37: Tabelarni prikaz naplate duga po datumima za referente

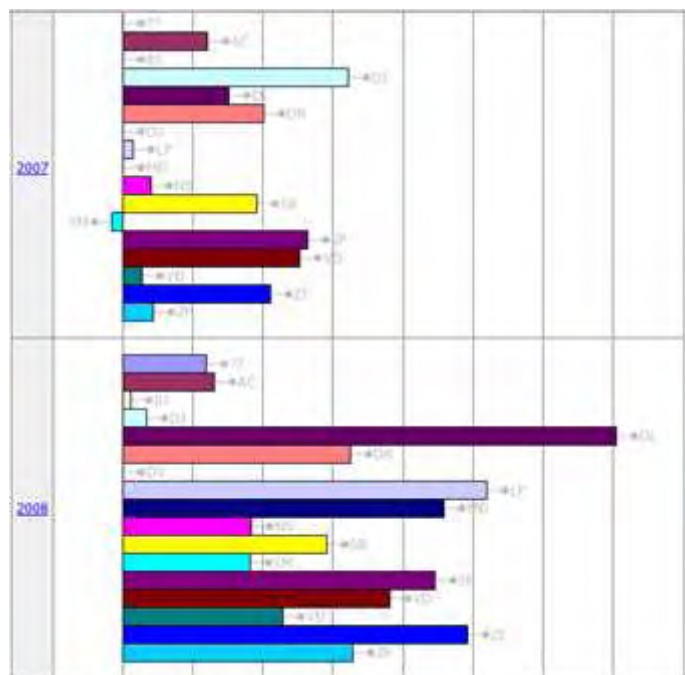
Takođe, može da se izvrši filtriranje po željenim vrednostima za dimenzije. Na primer, ako se izvrši filtriranje za referenta LP i mesec jun 2008. godine, dobiće se prikaz koliko je sumu novca naplatio referent LP u junu 2008. godine. Analogno prethodnom, u prikaz mogu da se dodaju i druge dimenzije, pa da se na taj način menjaju varijante prikaza željenih rezultata.

Naplata duga može da se predstavi i grafički. Na sledećoj slici je prikazana naplata po referentima:



Slika 38: Grafički prikaz naplate po referentima

Dodavanjem vremenske dimenzije u prikaz, dobijaju se naplate po godinama za referente, kao što se vidi na Slici 39.



Slika 39: Grafički prikaz naplate po godinama za referente

Operacijom bušenja po vremenskoj dimenziji mogu da se dobiju podaci o naplati duga po mesecima i danima.

Moguć je i obrnut prikaz, odnosno, uporedni pregled naplate duga po referentima za posmatrani vremenski interval (tj. za 2007. i 2008. godinu), što je prikazano na sledećoj slici:



Slika 40: Grafički prikaz naplate po referentima za 2007. i 2008. godinu

Operacijama bušenja i sečenja po vremenskoj dimenziji, može da se dođe do prikaza naplate po danima za zadati mesec i godinu. Na sledećoj slici je prikazana naplata referenta ZP u toku novembra 2008. godine.



Slika 41: Naplata duga za referenta ZP tokom novembra 2008. godine

Ako se za prikaz uzme samo mesto vršenja transakcije, dobiće se prikaz naplate duga po regionima, kao što je prikazano na sledećoj slici:



Slika 42: Grafički prikaz naplate po regionima gde su vršene transakcije

Ukoliko se izvrši spuštanje po hijerarhiji, za region „Beogradski venac“ će se dobiti prikaz kao na Slici 43. Ako se izvrši dalje spuštanje po hijerarhiji, npr. za opštinu Rakovica koja pripada navedenom regionu, dobija se prikaz kao na Slici 44:

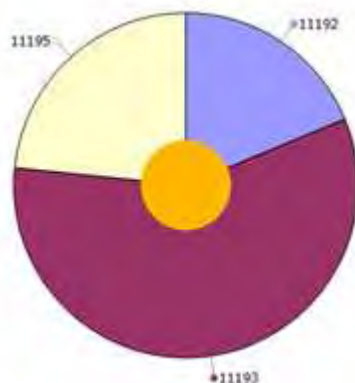


Slika 43: Grafički prikaz naplate za region „Beogradski venac“



Slika 44: Grafički prikaz naplate za opštinu Rakovica

Moguće je da se dodatno izvrši filtriranje za nekog referenta, pa se tako dobija prikaz naplate za navedenog referenta u opštini Rakovica, po poštanskim brojevima, kao što je prikazano na sledećoj slici:



Slika 45: Grafički prikaz naplate za referenta LP u opštini Rakovica

8. ZAKLJUČAK

Mali broj OLAP kocki može da da odgovor na veliki broj pitanja. Korisnik bez programerskog znanja može da vrši upite nad bazom podataka, jer se pitanja postavljaju na intuitivan način, bez poznavanja upitnih jezika. Posmatranjem podataka iz različitih perspektiva uočavaju su različiti trendovi ponašanja, ali, često, i neke neočekivane zakonitosti, koje mogu da pomognu prilikom donošenja budućih poslovnih odluka. Takođe, bitna karakteristika OLAP sistema je da su transakcioni podaci odvojeni od podataka koji se koriste za analizu, tako da se prilikom vršenja operacija na OLAP kockama ne usporava rad svakodnevnih transakcija.

OLAP tehnike su iskorišćene u reinžinjeringu aplikacije za izradu izveštaja o naplati duga od vlasnika tekućih računa koji su u nedozvoljenom minusu u Banci Poštanskoj Štedionici, a.d. Koliko je autoru poznato, ovo je prva (i to uspešna) primena OLAP tehnika u Banci. Realizacija rešenja postavljenog problema je ostvarena pomoću tri modela OLAP kocke. Prvi model sadrži tabelu istorije promena u ulozi tabele činjenica i dimenzije koje odgovaraju kolonama te tabele. Drugi model je ostvaren na takav način da je tabela činjenica tabela sa podacima iz Pravne službe, sa odgovarajućim dimenzijama i brojem računa kao merom. Treći model podataka sa istom tabelom činjenica kao i drugi, ali je, u ovom slučaju, broj računa posmatran kao dimenzija. Samo iz ova tri modela su dobijeni odgovori na sva pitanja koja su postavljena ranije u radu.

Najviše vremena je utrošeno u samu pripremu podataka, odnosno u njihovo prečišćavanje i izdvajanje u odgovarajuće tabele. Buduća dopuna ovih podataka se radi po prethodno opisanoj proceduri. Na ovaj način organizovani podaci su konstantno na raspolaganju korisnicima. Posao programera se svodi na eventualnu dopunu modela kocke u smislu proširenja skupa podataka koji potencijalno mogu da se prikažu. Pozadinske obrade postaju nepotrebne, a izveštaji nisu više glomazni i teško razumljivi. Svaki korisnik za sebe može da napravi prikaze koji su mu najjednostavniji i najlakši za razumevanje. Samo formiranje izveštaja je interaktivno. Najvažnije od svega je da su ti izveštaji odmah dostupni.

9. Literatura

[1] Pavlović-Lažetić Gordana, Uvod u relacione baze podataka, drugo izdanje, Matematički fakultet, 1999

[2] DB2 Cube Views A Primer, IBM Redbooks, SG24-7582-00 December 2008, <http://www.redbooks.ibm.com/abstracts/sg247002.html>

[3] DB2 Cube Views: Getting Started with Business Objects, REDP-3711-00 September 2003, <http://www.redbooks.ibm.com/abstracts/redp3711.html>

[4] Thomsen Erik, OLAP Solutions: Building Multidimensional Information Systems, Second Edition, John Wiley & Sons, 2002

[5] MicroStrategy, The Case for Relational OLAP, 1995, http://www.cs.bgu.ac.il/~dbm031/dw032/Papers/microstrategy_211.pdf

[6] Star schemas, <http://publib.boulder.ibm.com/infocenter/rbhelp/v6r3/index.jsp?topic=/com.ibm.redbrick.doc6.3/wag/wag32.htm>

[7] Star & snowflake schemas, <http://publib.boulder.ibm.com/infocenter/ablxhelp/8.3/index.jsp?topic=/com.ibm.db2.ax.cub.doc/sii-cubeschema-32488.htm>

[8] OLTP vs. OLAP, <http://datawarehouse4u.info/OLTP-vs-OLAP.html>

[9] The FASMI test, <http://www.olapreport.com/fasmi.htm>

[10] Business Intelligence: OLAP Tool Selection, <http://www.1keydata.com/datawarehousing/toololap.html>

[11] DB2 for OS/390 V5 and Federated Systems, <http://nlazovic.bravehost.com/Xephon/docs/DB2%20for%20OS390%20V5%20and%20Federated%20Systems.html>

[12] DB2 Cubes overview, http://publib.boulder.ibm.com/infocenter/db2luw/v8/index.jsp?topic=/com.ibm.db2.udb.db2_olap.doc/cwhat.htm

[13] Control Center overview, http://publib.boulder.ibm.com/infocenter/db2luw/v8/index.jsp?topic=/com.ibm.db2.udb.cc.doc/db2_ udb/intro.htm

[14] OLAP Market Share, <http://www.1keydata.com/datawarehousing/olap-market-share.html>