

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Simon Kraševc

**Nadgradnja informacijskega sistema z Oraclovim
podatkovnim skladiščem**

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE
STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

Mentor:
viš. pred. dr. Igor Rožanc

LJUBLJANA, 2012



Št. naloge: 00199/2012

Datum: 01.03.2012

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **SIMON KRAŠEVEC**

Naslov: **NADGRADNJA INFORMACIJSKEGA SISTEMA Z ORACLOVIM
PODATKOVNIM SKLADIŠČEM**
**UPGRADING AN INFORMATION SYSTEM USING ORACLE DATA
WAREHOUSE**

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija prve stopnje

Tematika naloge:

Za učinkovito uporabo podatkov informacijskega sistema pri odločanju je smiselno te zapisati v podatkovno skladišče, ki vsebuje potrebne mehanizme in orodja za analizo podatkov in predstavitev rezultatov. V diplomski nalogi najprej kratko opišite podatkovna skladišča, nato pa sistematično predstavite vse korake priprave, polnjenja in uporabe Oraclovega podatkovnega skladišča za obstoječ informacijski sistem. Nalogo zaključite z analizo opravljenega dela.

Mentor:

viš. pred. dr. Igor Rožanc

Dekan:

prof. dr. Nikolaj Zimic



IZJAVA O AVTORSTVU

diplomskega dela

Spodaj podpisani/-a Kraševac Simon,

z vpisno številko 63020090,

sem avtor/-ica diplomskega dela z naslovom:

Nadgradnja informacijskega sistema z Oraclovim podatkovnim skladiščem

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal/-a samostojno pod mentorstvom
viš. pred. dr. Igor Rožanc
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.)
ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki »Dela FRI«.

V Ljubljani, dne 15.06.2012

Podpis avtorja/-ice: _____

Zahvala

Zahvaljujem se vsem, ki so me podpirali, spodbujali in pomagali pri izdelavi diplomskega dela. Hvala mami, očetu, prijateljem in sodelavcem. Hvala mentorju Igorju Rožancu za strokovno vodstvo in pomoč. Hvala tudi Andreju Lenarčiču za pomoč in uvid pri izgradnji sistema za poslovno obveščanje. Še posebna zahvala Gospodu Jezusu, brez katerega ne bi bilo ničesar.

Simon Kraševc

*Mojemu nebeškemu Očetu,
ki me je naredil človeka, ki se ne vda in premaga vse ovire.*

Kazalo

1	UVOD.....	3
1.1	CILJI DIPLOMSKEGA DELA	4
2	POSTAVITEV OKOLJA ZA BI	5
2.1	ANALIZA TRENUTNEGA STANJA.....	5
2.2	PRIPRAVA NA NAMESTITEV	5
2.3	NAMESTITEV OBIEE 11G	6
2.4	PRENOS OBSTOJEČE REŠITVE IN NADGRADNJA	7
2.5	PRIPRAVA RAZVOJNEGA OKOLJA	9
3	RAZVOJ NOVIH FUNKCIONALNOSTI BI ZA ANALIZO IN PLANIRANJE	13
3.1	PRIPRAVA PODATKOVNIH VIROV	13
3.1.1	<i>Časovna dimenzija</i>	<i>13</i>
3.1.2	<i>Računovodski podatki.....</i>	<i>15</i>
3.1.3	<i>Konto</i>	<i>15</i>
3.1.4	<i>Stroškovno mesto</i>	<i>17</i>
3.1.5	<i>Poslovni partner, vrsta prometa in stroškovni nosilec</i>	<i>18</i>
3.1.6	<i>Tabela dejstev poslovnih dogodkov.....</i>	<i>18</i>
3.1.7	<i>Planski podatki.....</i>	<i>21</i>
3.2	MODEL PODATKOVNEGA SKLADIŠČA IN ADMINISTRATORSKO ORODJE	24
3.2.1	<i>Fizični nivo</i>	<i>24</i>
3.2.2	<i>Poslovni model in preslikave.....</i>	<i>25</i>
3.2.3	<i>Predstavitveni nivo</i>	<i>29</i>
3.2.4	<i>Namestitev modela skladišča</i>	<i>29</i>
4	POSTAVITEV PREDSTAVITVENEGA DELA SISTEMA BI	33
4.1	POROČILA.....	33
4.2	NADZORNA PLOŠČA	34
4.3	UVAJANJE UPORABNIKOV.....	36
5	SKLEPNE UGOTOVITVE.....	37
6	LITERATURA.....	39

Seznam kratic

BI (ang. Business Intelligence) – sistem za poslovno obveščanje

CRM (ang. Customer Relationship Management) – programska oprema za upravljanje odnosov s strankami

ERP (ang. Enterprise Resource Planning) – celovita programska rešitev za načrtovanje virov podjetja

ETL (ang. Extract, Transform, Load) – proces zajema, transformacije in prenosa podatkov med dvema podatkovnima bazama

FMW (ang. Fusion Middleware) – Oraclova programska oprema za vmesni nivo, ki nudi podporo in medsebojno povezuje več Oraclovih rešitev

LTS (ang. Logical Table Source) – logični podatkovni vir, ki se nanaša na podatkovni vir na fizičnem nivoju modela skladišča

OBIEE (ang. Oracle Business Intelligence Enterprise Edition) – Oraclov programski paket za poslovno obveščanje

ODBC (ang. Open DataBase Connectivity) – odprta podatkovna povezljivost, standardna SQL metoda pristopa iz poljubne aplikacije do podatkovnih baz

OLAP (ang. OnLine Analytical Processing) – sprotna analitična obdelava podatkov namenjena za takojšnjo analizo podatkov v podatkovni bazi oziroma podatkovnem skladišču

OLTP (ang. OnLine Transaction Processing) – sprotna obdelava transakcij, pri kateri se transakcije takoj zapišejo v podatkovno bazo in so vidne vsem uporabnikom

RCU (ang. Repository Creation Utility) – orodje za kreiranje shem, potrebnih v sistemu FMW

SQL (ang. Structured Query Language) – strukturiran povpraševalni jezik za delo s podatkovnimi bazami

WLS (ang. WebLogic Server) – strežnik WebLogic

WLS LDAP (ang. WebLogic Server Lightweight Directory Access Protocol) – protokol za dostop do imenikov strežnika WebLogic

Seznam slik

Slika 1: Magični kvadrant za BI [2]	4
Slika 2: Povzetek namestitve	6
Slika 3: Strategija nadgradnje BI.....	7
Slika 4: Določanje vira ODBC	10
Slika 5: Primer trinivojskega semantičnega modela z mnogimi viri podatkov.....	24
Slika 6: Shema poslovnega modela za tabelo dejstev "F01-Temeljnica"	26
Slika 7: Administratorsko orodje s fizičnim, PMP in predstavitvenim nivojem.....	27
Slika 8: Določanje agregacije za mero	28
Slika 9: Namestitev novega modela skladišča v poslovnem upravitelju.....	30
Slika 10: Stanje strežnika BI	30
Slika 11: Priprava poročila	33
Slika 12: Pripravljena poročila	34
Slika 13: Prikaz poročil na nadzorni plošči (1)	35
Slika 14: Prikaz poročil na nadzorni plošči (2)	35

Seznam tabel

Tabela 1: Izvedeni merljivi podatki.....	28
--	----

Povzetek

V diplomskem delu smo opisali nadgradnjo obstoječega direktorskega informacijskega sistema, ki vključuje postavitev novega sistema, nadgradnjo starega in razširitev obstoječih funkcionalnosti z novimi podatki. Sistem je zasnovan na orodju Oracle Business Intelligence Enterprise Edition (v nadaljevanju OBIEE) 11g. Že obstoječi sistem je bil razvit z orodjem OBIEE 10g, obsegal pa je podatke več poslovnih funkcij danega podjetja. Obstoječi sistem smo razširili s prikazom planskih in realiziranih podatkov poslovnih funkcij. Prenovljeni sistem tako nudi vpogled v podrobne podatke posameznih projektov ter informacije o njihovi uspešnosti, dolgoročno in kratkoročno pričakovane rezultate, pregled nad prihodki in odhodki ter njihovo odstopanje od načrtovanih vrednosti. Prikazi so dinamični in interaktivni. Uporabili smo razne gradnike prikazov in kombinacije le-teh: tabela, vrtilna tabela, graf, filter, pozivnik (ang. prompt), povezava nadrejeni zapis-podrobnost (ang. master-detail linking), izbirnik polj (ang. column selector), izbirnik pogledov (ang. view selector) in ostale. S temi gradniki smo sestavili poročila, le-te pa logično razporedili na nadzorne plošče.

Ključne besede

Podatkovno skladišče

Proces ETL

Poročila

Oracle Business Intelligence Enterprise Edition

Abstract

This thesis describes the upgrade of the existing management information system. It includes installation of a new system and upgrade and expansion of existing functionality with new data. The system is based on Oracle Business Intelligence Enterprise Edition (hereafter OBIEE) 11g. Existing system was developed using OBIEE 10g which encompassed information from more business functions of a given company. The existing system was expanded with reports of planned and actual data of business functions. The reformed system also provides insight into the details of individual projects and information on their performance, long-term and short-expected results, an overview of income and expenditure, and their deviation from the planned values. Reports are dynamic and interactive. Various building blocks were used and their combinations thereof: table, pivot table, graph, filter, prompt, master-detail linking, column selector, view selector and others. Reports were compiled with these building blocks and then they were logically allocated on the control panel.

Keywords

Data warehouse

ETL process

Reports

Oracle Business Intelligence Enterprise Edition

1 Uvod

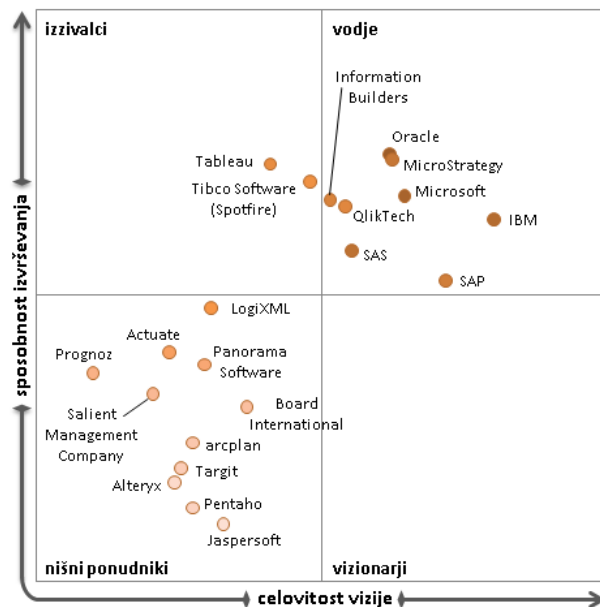
Začetki podatkovnih skladišč segajo v sam začetek razvoja računalništva. Podatki so se zbirali v transakcijskih zbirkah. Na teh so se izvajale tudi analitične funkcije, kar se je izkazalo za neučinkovito. Iz tega se je izoblikoval koncept podatkovnega skladišča (ang. data warehouse), ki je ločen od operativne baze. Že v letih po 1960 so se oblikovale ideje o dimenzijah in dejstvih, kar so pogloblitve lastnosti današnjih podatkovnih skladišč. Izoblikovali sta se dve smernici o zasnovi le-teh: ena zagovarja normalizacijo, druga pa dimenzijski pristop.

Večinoma so se s tem ukvarjala večja podjetja. Podatkovna skladišča so imela veliko različnih izvedenih oblik, postopkov zajema podatkov, načinov prikaza ter predelavo teh podatkov v informacije. Večje kot je bilo podjetje, večja je bila zahtevnost podatkovnega skladišča in s tem nastala tudi večja odstopanja med rezultati odgovora na isto vprašanje. V poslovnem svetu je to pomenilo več različnih "resnic" oziroma več različnih odgovorov na isto vprašanje. S tem se je pojavila potreba po poenotenju razvoja in uporabe podatkovnih skladišč.

Po letu 1996 je postajalo modeliranje podatkovnih skladišč z dimenzijami vse bolj splošno sprejeto. Ta način oblikovanja podatkovnih skladišč ob povečanju zahtevnosti in obsega podatkovnega skladišča ohrani attribute preprostosti in prijaznosti do uporabnika [1]. Lahko rečemo, da je v času dozorevanja teh tehnologij, metod in izdelkov večjo težo pri izbiri platforme poslovnega obveščanja (v nadaljevanju BI – ang. Business Intelligence) nosila prav izbrana tehnologija in izdelek ne glede na dejanske poslovne potrebe. Potrebna znanja za uresničitev projekta so zajemala od znanja za izbiro tehnologije do analitičnih zahtev. Do danes so koncept, tehnologija in izdelki poslovnega obveščanja prešli v zrelo stopnjo, stroški za vpeljavo so se zmanjšali, vse več je tudi kadrov s potrebnim znanjem. Najbolj odločilno dejstvo pa je prav to, da se je osnovni koncept premaknil tja, kamor spada, v poslovni svet. Ko je sistem BI pravilno postavljen, ga lahko uporablja velik del zaposlenih v podjetju in ne le vodstvo podjetja. Pogojev in načinov uporabe ne določajo v IT oddelku in vsak zaposleni ima lahko določen dostop do uporabe sistema, kjer dobi ustrezne informacije iz kateregakoli poslovnega procesa.

Podjetja, ki ne vpeljejo analitične obdelave podatkov, postajajo bolj in bolj nekonkurenčna in s tem že vnaprej izgubljajo svoj delež na trgu. Zato je tudi ponudba računalniških podjetij za integracijo poslovnega odločanja v obstoječi sistem proizvodnih in drugih podjetij kot storitev zelo smiselna. Tega se zavedajo tudi podjetniki. V primerjavi s predhodnimi leti je tudi ponudba takšnih storitev lani močno narasla. Ta ponudba večinoma zajema izdelavo rešitev na osnovi tehnologij vodilnih podjetij na tem področju, kot so IBM, Oracle, Microsoft, SAS, SAP, MicroStrategy; imamo pa tudi domača podjetja, ki ponujajo rešitve za poslovno inteligenco razvite povsem v lastni režiji. Na sliki 1 so prikazana 4 polja, v katere so

porazdeljeni določeni ponudniki rešitev [2]. Na abscisi imamo prikazano celovitost vizije, na ordinati pa sposobnost izvrševanja. Opazimo, da podjetja, ki imajo dobro zastavljeno vizijo, to tudi dobro uresničijo. Pojavljajo se predvsem vodilna podjetja, ki uvedejo celovito rešitev, ter nišni ponudniki, ki se osredotočajo na določene podrobnosti.



Slika 1: Magični kvadrant za BI [2]

1.1 Cilji diplomskega dela

Naš cilj je nadgraditi obstoječo rešitev za direktorski informacijski sistem na verzijo OBIEE 11g (v. 11.1.1.5.0) ter dodati možnost prikaza planskih in realiziranih podatkov. Na koncu želimo končne uporabnike še seznaniti z novostmi in uspešno uvesti v uporabo novosti sistema.

Sistem BI na postavljenem strežniku mora vsebovati:

- obstoječa poročila, ki so bila razvita z orodjem OBIEE 10g,
- nova poročila za analiziranje bilance podjetja,
- nova poročila za analiziranje planiranih in realiziranih podatkov in
- omogočati mora preprosto izdelavo ad hoc poročil.

Možna mora biti tudi razširitev podatkovnega skladišča v bazi in modela skladišča (ang. repository) v administratorskem orodju z ostalimi poslovnimi funkcijami podjetja na enostaven način.

V majhnem podjetju z več kot 40 zaposlenimi, se obstoječi sistem BI uporablja za mesečno analizo stanja in strateško odločanje. Namen podjetja je dovršiti izdelek in ga uspešno tržiti.

2 Postavitev okolja za BI

BI ne temelji samo na tehnologiji in izdelkih, ki nudijo njegovo implementacijo [3]. Pravzaprav je tehnologija le manjši del poslovnega obveščanja in se skozi čas razvija ter zavzema razne oblike in rešitve. Kakor smo že navedli v uvodu, je več vodilnih podjetij in vsako nudi rahlo drugačno rešitev. Pri tem je osrednji koncept pri vseh enak.

Microsoft Business Intelligence [4] na primer nudi razširitev za programski paket Microsoft Office (Excel, PowerPivot), podporo za nadzorne plošče (SharePoint) ter strežnik SQL. Oracle podobno kot Microsoft nudi tudi celoviti paket, sestavljen iz lastnih rešitev, možno je tudi vključiti izdelke drugih izvajalcev [5]. Pri obeh zgoraj navedenih pa je končni prikaz sestavljen na enak način iz podobnih gradnikov, ki se večinoma razlikujejo le grafično. Ker gre pri našem projektu za nadgradnjo iz obstoječe rešitve na OBIEE 10g, smo izbrali OBIEE 11g. Tako nam ni treba ponovno od začetka graditi že obstoječe rešitve. Poleg tega pa imamo že nekaj znanja in izkušenj z Oraclevim poslovnim obveščanjem.

2.1 Analiza trenutnega stanja

Na začetku smo naredili kratek intervju s posameznimi uporabniki trenutnega sistema OBIEE 10g. Ta nam je pokazala, da zaposleni v podjetju večinoma vedo, za katera poročila lahko pričakujejo pravilne oziroma napačne rezultate. Odkrili so nam nekatere točke sistema, ki so bile slabo izvedene ali počasne. Opazili smo tudi, da za nekatera poročila še vedno zbirajo in obdelujejo podatke sami iz zunanjih virov. V nasprotju s tem pa odgovorni vodje nimajo časa sami preverjati pravilnosti rezultatov in pričakujejo, da so le-ti pravilni.

Preverili smo tudi tehnično izvedbo trenutne rešitve in si zabeležili, katere komponente delujejo pravilno, katere so potrebne prenove in katere so neuporabne. To bomo upoštevali pri nadgradnji opisani v poglavju 2.4.

2.2 Priprava na namestitve

Za namestitev OBIEE najprej potrebujemo ustrezno podatkovno bazo ter na njej pripravljene potrebne sheme. Podatkovno bazo smo imeli že nameščeno, in sicer Oracle database 11gR2 [6]. Sheme smo pripravili z orodjem RCU (ang. Repository Creation Utility), ki smo ga dobili na Oraclevi spletni strani za prenos datotek [7]. Na sistemu Microsoft Windows Server 2008 smo pognali orodje RCU. Navedli smo prijavo v podatkovno bazo z uporabnikom, ki ima pravice DBA in izbrali shemo DEV11G_BIPLATFORM za platformo poslovnega obveščanja ter shemo DEV11G_MDS za metapodatkovne storitve. Za obe smo določili isto geslo in dokončali postopek kreiranja shem. Potrebovali smo jih za postavitve sistema Oracle FMW (ang. Fusion Middleware) ter znotraj tega za OBIEE. Služile nam bodo za potrebe razvoja in

testa, ob prenosu končne rešitve na produkcijsko okolje pa bomo ustvarili nove sheme z istimi imeni, le z drugo predpono, seveda na produkcijski podatkovni bazi.

Za naslednji korak smo iz iste spletne strani kot prej prenesli Oracle Business Intelligence, v. 11.1.1.5.0 za operacijski sistem Microsoft Windows x86 (64-bit). Ta paket vsebuje:

- strežnik BI (ang. BI Server Enterprise Edition),
- administratorski strežnik (ang. Server Administrator),
- orodje Answers za pripravo in urejanje poročil in ad hoc povpraševanj,
- orodje za pripravo interaktivnih nadzornih plošč (ang. Interactive Dashboard),
- orodja za poročila in objavo (ang. Reporting and Publishing),
- orodje za pripravo opozoril (ang. Delivers),
- dodatek za Microsoft Office,
- orodje za pripravo predlog, poročil in dokumentov (ang. BI Publisher),
- razširitev za prikaz sistema kazalnikov (ang. Scorecard and Strategy Management) in
- orodje za pripravo pravil, napovedne analitike in optimizacijo poslovnih ciljev (ang. Real-Time Decisions)

2.3 Namestitev OBIEE 11g

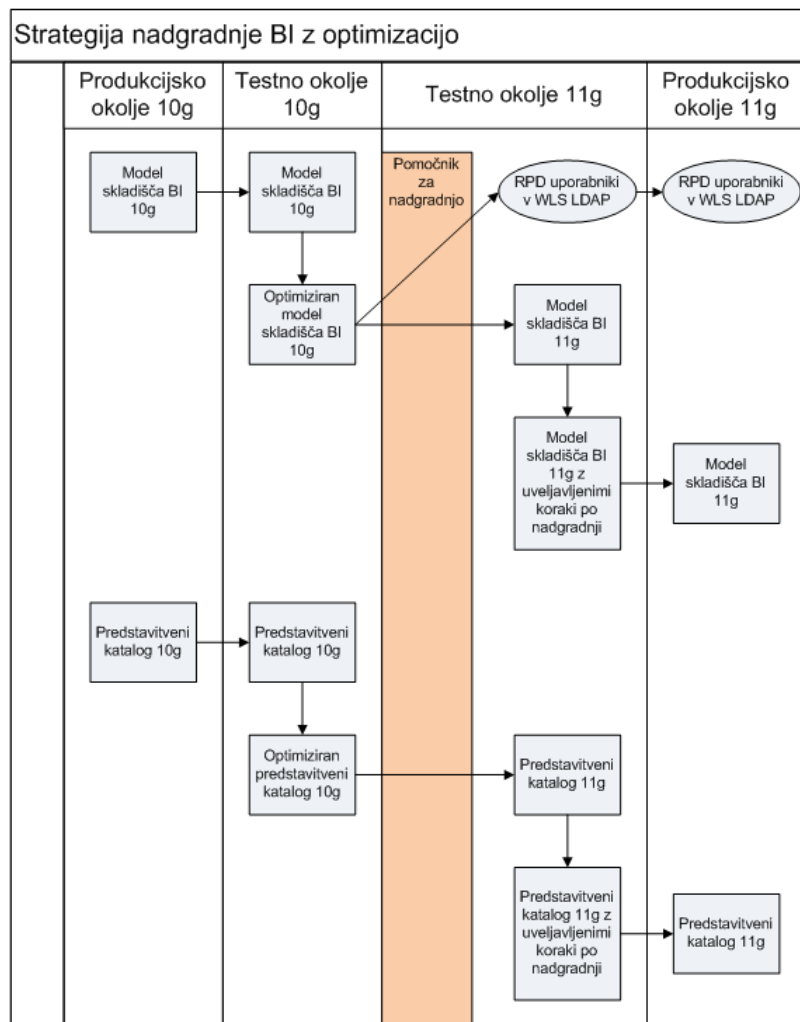
Zaradi določenih tehnoloških razlik v sami zasnovi med OBIEE 10g in 11g namestitev ne podpira nadgradnje neposredno čez obstoječi sistem. Še več, verzije 11g ne smemo namestiti na isti strežnik, kjer imamo nameščeno OBIEE 10g. Pri namestitvi OBIEE 11g smo izbrali novo postavitev, nato smo izbrali možnost splošna namestitev ter navedli izbrano podatkovno bazo ter sheme, ki smo jih pripravili. Po uspešni namestitvi se je vzpostavil celoten sistem vključujoč privzeti začetni model skladišča s splošnim primerom (SampleAppLite.rpd). Na sliki 2 je prikazan povzetek namestitve, kjer najdemo omrežne povezave do vseh orodij sistema od administracije WLS, administracije sistema BI, do orodij za končne uporabnike sistema.

```
Type: Simple Install
Installation Details
  Middleware Home: C:\OBIEE11g
  BI Oracle Home: C:\OBIEE11g\Oracle_BI1
  WebLogic Server Home: C:\OBIEE11g\wlserver_10.3
  BI Domain Home: C:\OBIEE11g\user_projects\domains\bifoundation_domain
  BI Domain Name: bifoundation_domain
  Instance Home: C:\OBIEE11g\instances\instance1
  Instance Name: instance1
Configure Components
  WebLogic Console
    http://mojip:7001/console
  Oracle Enterprise Manager
    http://mojip:7001/em
  Business Intelligence Enterprise Edition
    http://mojip:7001/analytics
  Business Intelligence Publisher
    http://mojip:7001/xmlpserver
  Real-Time Decisions
    http://mojip:7001/ui
```

Slika 2: Povzetek namestitve

2.4 Prenos obstoječe rešitve in nadgradnja

Proces nadgradnje je zahteven, saj smo se morali nanj dobro pripraviti tudi s podrobnim načrtom same izvedbe nadgradnje. Oracle v svoji dokumentaciji nudi več primerov, kako lahko sestavimo svojo strategijo nadgradnje [8]. Za naše potrebe je zadoščalo, da smo vanjo vključili le optimizacijo in nadgradnjo modela skladišča in predstavitvenega kataloga (ang. presentation catalog oziroma webcatalog). Ostale komponente niso bile uporabljene v obstoječi rešitvi, zato smo jih izpustili iz načrtovanega postopka nadgradnje (slika 3). Po vsakem izvedenem koraku nadgradnje ali prenosa smo izbrali testni vzorec in testirali končna poročila. S tem smo zagotovili, da so vse komponente ažurne in se ni podrla nobena funkcionalnost. Pred prenosom iz testnega okolja 11g na produkcijsko pa smo izvedli temeljiti test.



Slika 3: Strategija nadgradnje BI

Upravljanje varnosti je bilo v verziji 10g vključeno v okviru modela skladišča, pri verziji 11g pa je preneseno v podsistem upravljanja varnosti znotraj Oracle FMW. Zato smo pri procesu

nadgradnje ta del ločili iz datoteke modela skladišča BI 10g in ga s pomočnikom za nadgradnjo uvozili v WLS LDAP. V modelu skladišča BI 11g imamo še vedno prikazane uporabnike in aplikacijske vloge (v modelu 10g so bile to grupe), vendar so te le reference na dejanske objekte v sistemu WLS LDAP. Pred nadgradnjo smo zagotovili, da imena uporabnikov in skupin v modelu skladišča ne vsebujejo posebnih znakov (»«, «», »+«, »=«, »"«, »\«, »<«, »>«, »;«, in ».«)

Pomočnik za nadgradnjo ne bo prenesel objektov s takšnim imenom v WLS LDAP. Zato jih preimenujemo, sicer jih bomo morali sami ročno dodati. Kakršne koli uporabnike, ki so določeni v zunanjem imeniškem sistemu in se nahajajo tudi v našem modelu skladišča, odstranimo iz zunanjega sistema. Ti uporabniki se bodo identificirali prek WLS LDAP, ko se le-ti prenesejo iz modela skladišča.

Najprej smo optimizirali model skladišča BI 10g in predstavitveni katalog 10g. Preverili smo skladnost modela skladišča in odpravili napake in opozorila. Odstranili smo neuporabljene in nepotrebne elemente ter uredili uporabnike in skupine. Ostalo nam je še nekaj neskladnosti, nekatere funkcije so bile opredeljene ročno in so se sklicevale na podatke, katerih uporaba in namembnost se je v preteklosti nekoliko spremenila. Reševali jih bomo v okviru razširitve modela skladišča.

Model skladišča in predstavitveni katalog smo skupaj zajeli v istem procesu nadgradnje. Pomočnik za nadgradnjo je potreboval 4 minute, da je pripravil nov model skladišča in katalog, kar je dokaj hitro za ta postopek. V verziji 11g je izboljšana funkcija preverjanja skladnosti modela skladišča. Z administratorskim orodjem (opisano je v poglavju 4) preverimo, ali so vsi podatkovni viri dostopni in systemske spremenljivke pravilne. Preverimo lahko skladnost modela skladišča in odpravimo morebitne napake in opozorila.

Po nadgradnji preverimo ime strežnika, ki je določeno v nastavitvenih datotekah:

- [DOMAIN_HOME]/config/bipublisher/repository/Admin/Configuration/xmlp-server-config.xml
- [DOMAIN_HOME]/config/bipublisher/repository/Admin/DataSource/datasources.xml

Ime strežnika se mora nanašati na strežnik BI 11g.

Pomembno je, da poznamo imeniško strukturo sistema OBIEE 10g in tudi kakšne so spremembe v verziji 11g. Če smo dodatno vstavili kakšne datoteke, spreminjali vrednosti v nastavitvenih datotekah, urejali sloge ali prilagodili uporabniški vmesnik, jih moramo sami prenesti tudi v novi sistem 11g. Nastavitve v 11g večinoma ne moremo več določati neposredno prek nastavitvenih datotek, kot smo jih lahko v sistemu BI 10g. Uredili smo jih v

upravljalniku WebLogic na naslovu <http://mojip:7001/console> ali v poslovnem upravitelju Oracle Enterprise Manager na naslovu <http://mojip:7001/em>.

Pomembno je tudi, da poznamo strukturo predstavitvenega kataloga. V njem se hranijo vsi imeniki, bližnjice, poročila, filtri, pozivniki, nadzorne plošče, skratka vsi objekti, ki jih ustvarimo s predstavitveno storitvijo BI. Vsak objekt ima dve datoteki: vsebinsko (npr: Porocilol1) ter opisno (npr: Porocilol1.atr), kamor se zapišejo lastnosti objekta.

Sedaj lahko preverimo, če se je vsa vsebina pravilno prenesla v novi katalog. V BI 11g na naslovu <http://mojip:7001/analytics> pa smo preverili, če vse deluje tako kot na sistemu BI 10g. Medtem ko je sam videz strani in gradnikov v nekaterih primerih drugačen, smo se osredotočili na funkcionalnost sistema. Izbrali smo en podsklop nadgrajene rešitve in preverili vse objekte, če delujejo enako, kot na sistemu BI 10g, ali so njihove akcije in rezultati enaki. Še posebej smo bili pozorni na gradnike in akcije, za katere vemo, da so v verziji Oracle BI 11g dopolnjeni ali spremenjeni.

V našem primeru ni preveč razlik. Te se večinoma odražajo samo v drugačnih privzetih vrednostih, kot so formati polj, število prikazanih vrstic v vrtilnih tabelah, format in poravnava teksta, način izpisa datumov. Imamo nekaj polj, ki so bila označena kot skrita, vendar kljub temu vključena v prikaze in na njih vidna, sedaj niso več vidna. Na nekaterih prikazih ni pravilne razporejenosti. Vse te in podobne spremembe na postavitvi BI 11g niso kritične, vendar so lahko zelo moteče končnim uporabnikom in je prav tako pomembno, da jih popravimo. Pri nadgrajenih elementih se potrudimo, da so videti in da delujejo čim bolj podobno, kot so v prejšnji verziji sistema, da zagotovimo enako uporabniško izkušnjo.

2.5 Priprava razvojnega okolja

Ob namestitvi OBIEE 11g smo na strežniku dobili tudi potrebna orodja za razvoj modela skladišča in urejanje predstavitvenega kataloga. Ker pa ne nameravamo razvijati rešitve na strežniškemu sistemu, moramo pripraviti novo razvojno okolje. Za postavitev tega imamo dve možnosti:

- bodisi uporabimo namestitveni paket za celoten OBIEE 11g z izbiro namestitve samo razvijalskih orodij (ang. software only install),
- ali pa namestimo poseben Oraclov programski paket razvijalskih orodij za odjemalca.

Pri obeh možnostih smo omejeni z zelenim ciljnim operacijskim sistemom. Pri prvi možnosti smo že izbrali sistem Microsoft Windows 64-bit, kar pa ni naš zeleni OS za razvojno okolje. Če bi izbrali drug ciljni OS, bi morali pridobiti nov celotni namestitveni paket, ta pa obsega več kot 5 GB. Glede na to, da potrebujemo le par orodij, ki zavzamejo manj kot 200 MB, je to pomanjkljivost Oraclove ponudbe. Poleg tega bi morali v tem primeru samo za potrebe orodja

za odjemalca kreirati ločeni shemi BIPLATFORM in MDS z orodjem RCU. Na srečo je druga možnost pravšnja za nas. Na Oraclovi spletni strani [9] dobimo programski paket razvijalskih orodij za odjemalca za operacijski sistem Windows 32-bit.

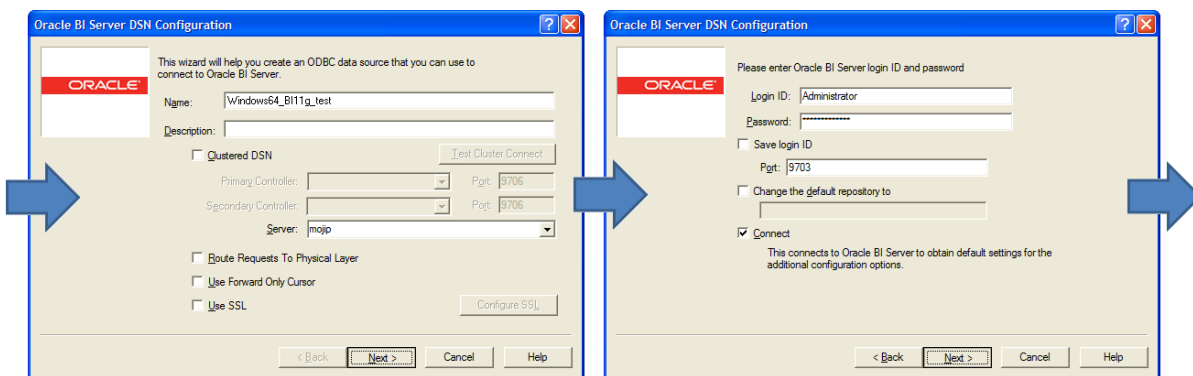
Na izbranem računalniku imamo nameščenih veliko ostalih programov, med drugim tudi starejše različice istega programskega paketa OBIEE 10g. Oracle sicer podpira sočasno namestitvev različnih verzij tega orodja, vendar smo na tem računalniku že imeli nameščenih tudi več verzij od 10g in tudi 11g za preizkusne in testne namene, ki smo jih pozneje odstranili. Da bi se izognili možnim zapletom, smo pripravili novo okolje na navidezni napravi (ang. virtual machine).

Preden smo dejansko namestili orodja za odjemalca, smo še preverili, če imamo dostop do novega sistema BI 11 g na strežniku Microsoft Windows Server 2008 (<http://mojip:7001/em>), potem pa nadaljevali z namestitvijo. Pridobili smo naslednja tri orodja:

- administratorsko orodje (ang. Oracle BI Administration Tool),
- upravljevec poslov (ang. Oracle BI Job Manager) ter
- upravljevec kataloga (ang. Oracle BI Catalog Manager),

vendar še niso uporabna.

V sistemu Windows moramo zagotoviti podatkovni vir ODBC, ki kaže na naš strežnik BI. V administrativnih orodjih sistema Windows odpremo "Data Sources (ODBC)", izberemo jeziček "System DSN" ter kliknemo gumb "Add...". Med možnimi gonilniki smo izbrali Oracle BI Server 1. Odpre se okno za vnos podrobnosti podatkovnega vira ODBC, nadaljujemo, kakor prikazuje slika 4 ter po zaključku dobimo nov sistemski vir Windows64_BI11g_test.



Slika 4: Določanje vira ODBC

Po uspešni nastavitvi ODBC vira smo pripravili delovni imenik na deljenem omrežnem mestu. Vanj bomo odložili model skladišča.

Delovno okolje je pripravljeno in strežnik BI je dostopen. Vidimo, da je postavljeni sistem porabil veliko količino pomnilnika. Pri končni rešitvi pričakujemo, da se bo ta poraba še povečala, večinoma zaradi medpomnjenja zajetih podatkov na poročilih. Za testno postavitev je to sprejemljivo. Za produkcijsko postavitev sistema BI pa je dobrodošlo in priporočeno, da izberemo namenski strežnik. Na njem bomo izvedli navpično razširitev (ang. vertical scaleout), to je pomnožitev postavljene gruče, v kateri je postavljen sistem BI. S tem bomo dosegli večjo zanesljivost v slučaju, da se na prvi gruči pojavi programska napaka.

3 Razvoj novih funkcionalnosti BI za analizo in planiranje

V naslednjih podpoglavjih so opisane tabele dimenzij in dejstev ter njihova struktura. Najprej smo načrtovali celoten model, ki smo ga ustvarili na bazi, na koncu pa smo prevzeli podatke iz operativne baze, jih preoblikovali in zapisali v tabele dimenzij. Ko smo imeli urejene tabele dimenzij, smo napolnili še tabele dejstev.

3.1 Priprava podatkovnih virov

Obstoječi viri so bili zasnovani na transakcijski podatkovni bazi, katere glavni namen je podpora aplikacijam podjetja in da jih je ob regularnem delovnem času mogoče uporabljati. To pomeni, da pričakujemo slabo odzivnost sistema pri povpraševanjih iz teh virov ob delovnem času. Edina omembe vredna prednost take rešitve je v tem, da ni treba zasnovati novega podatkovnega skladišča ter procesa zajema, transformacije in prenosa podatkov (v nadaljevanju proces ETL), kar skrajša čas načrtovanja in razvoja. Kljub temu da so pomanjkljivosti takšnega pristopa večje kot njegove prednosti, se nismo lotili preoblikovanja teh virov v podatkovna skladišča. Za to bi morali analizirati celotno obstoječo rešitev in podrobno spoznati tudi obliko in vsebino vseh virov, s tem pa bi prekoračili cilje tega diplomskega dela. Za naše potrebe je dovolj, da uspešno nadgradimo obstoječe vire. V modelu skladišča smo dopolnili le tiste funkcionalnosti, ki so vračale napačne rezultate.

Zagotoviti smo morali uporabo planskih podatkov ter njihovo primerjavo z dejanskimi. Za ta namen smo pripravili novo shemo na dani podatkovni bazi v obliki podatkovnega skladišča. Planske podatke smo prenesli iz datoteke. Zajeti smo morali tudi nekatere računovodske podatke iz transakcijske baze. Najprej smo pripravili dimenzije, nato pa na osnovi teh kreirali tabelo dejstev.

3.1.1 Časovna dimenzija

Najpomembnejša komponenta vsakega sistema BI je zagotovo časovna dimenzija, saj vse podatke merimo glede na določeno obdobje. Brez nje si ne moremo zamisliti realnega primera poslovnega obveščanja. Podatki bi bili prikazani brez informacije okvirnega obdobja. Pravzaprav je časovna dimenzija tako pomembna, da že samo na to temo obstaja veliko zapisanega, veliko navodil in skript o tem kako jo kreirati ter napolniti. V skladu s splošno sprejetimi načini kreiranja časovne dimenzije smo jo v našem podatkovnem skladišču pripravili z vsemi potrebnimi podatki in njihovimi kombinacijami. V njej hranimo ob splošnem datumu še posamezne podatke, kot so leto, mesec, kvartal, leto + mesec, leto + kvartal, mesec + leto in podobno. Ti podatki nam omogočajo hitrejše poizvedovanje ne glede na to, kako zastavimo naš pogoj v končni analizi. Če jih ne bi imeli vključene že v viru

podatkov, bi jih morali v našem modelu skladišča izvesti iz ostalih polj, to pa nekoliko poveča kompleksnost in časovno zahtevnost končne poizvedbe. Dimenzija DIM_DATUM je sestavljena, kakor prikazuje naslednja koda:

```
CREATE TABLE DIM_DATUM
(DATUM_KEY NUMBER(5) NOT NULL
, DATUM DATE NOT NULL
, MESEC_IME VARCHAR2(15) NOT NULL
, MESEC_STEV VARCHAR2(2) NOT NULL
, MESEC_KVARTAL VARCHAR2(5) NOT NULL
, MESEC_LETO VARCHAR2(7)
, KVARTAL VARCHAR2(2) NOT NULL
, KVARTAL_MESEC VARCHAR2(5) NOT NULL
, KVARTAL_LETO VARCHAR2(7)
, LETO VARCHAR2(4) NOT NULL
, LETO_KVARTAL VARCHAR2(7) NOT NULL
, LETO_MESEC VARCHAR2(7) NOT NULL
);

ALTER TABLE DIM_DATUM
ADD CONSTRAINT DIM_DATUM_PK PRIMARY KEY
(DATUM_KEY);

CREATE INDEX DDAT_IDX1 ON DIM_DATUM
(DATUM ASC);

CREATE INDEX DDAT_IDX2 ON DIM_DATUM
(LETO_MESEC ASC);
```

Dimenzijo smo napolnili s podatki za leta od 2008 do 2012 z naslednjo kodo:

```
...
r_dat.datum := l_datum_od;
r_dat.mesec_stev := '00';
LOOP
  IF r_dat.mesec_stev <> TO_CHAR (r_dat.datum, 'MM')
  THEN
    r_dat.mesec_stev := TO_CHAR (r_dat.datum, 'MM');
    r_dat.mesec_leto := TO_CHAR (r_dat.datum, 'MM-YYYY');
    r_dat.leto := TO_CHAR (r_dat.datum, 'YYYY');
    r_dat.leto_mesec := TO_CHAR (r_dat.datum, 'YYYY-MM');
    CASE r_dat.mesec_stev
      WHEN '01' THEN r_dat.mesec_ime := 'Januar';
      WHEN '02' THEN r_dat.mesec_ime := 'Februar';
      WHEN '03' THEN r_dat.mesec_ime := 'Marec';
      WHEN '04' THEN r_dat.mesec_ime := 'April';
      WHEN '05' THEN r_dat.mesec_ime := 'Maj';
      WHEN '06' THEN r_dat.mesec_ime := 'Junij';
      WHEN '07' THEN r_dat.mesec_ime := 'Julij';
      WHEN '08' THEN r_dat.mesec_ime := 'Avgust';
      WHEN '09' THEN r_dat.mesec_ime := 'September';
      WHEN '10' THEN r_dat.mesec_ime := 'Oktober';
      WHEN '11' THEN r_dat.mesec_ime := 'November';
      WHEN '12' THEN r_dat.mesec_ime := 'December';
    END CASE;
    r_dat.kvartal := 'K' || TO_CHAR (FLOOR ((
      TO_NUMBER (r_dat.mesec_stev)-1)/3)+1);
    r_dat.kvartal_mesec := r_dat.kvartal || '-' || r_dat.mesec_stev;
    r_dat.kvartal_leto := r_dat.kvartal || '-' || r_dat.leto;
    r_dat.mesec_kvartal := r_dat.mesec_stev || '-' || r_dat.kvartal;
    r_dat.leto_kvartal := r_dat.leto || '-' || r_dat.kvartal;
  END IF;
```

```

INSERT INTO dim_datum (datum_key, datum, mesec_ime
                      , mesec_stev, mesec_kvartal, mesec_letno
                      , kvartal, kvartal_mesec, kvartal_letno
                      , leto, leto_kvartal, leto_mesec)
VALUES (r_dat.datum_key, r_dat.datum, r_dat.mesec_ime
      , r_dat.mesec_stev, r_dat.mesec_kvartal, r_dat.mesec_letno
      , r_dat.kvartal, r_dat.kvartal_mesec, r_dat.kvartal_letno
      , r_dat.leto, r_dat.leto_kvartal, r_dat.leto_mesec
      );

g_rowcount := g_rowcount + 1;

EXIT WHEN r_dat.datum = l_datum_do;

r_dat.datum      := r_dat.datum + 1;
r_dat.datum_key := r_dat.datum_key + 1;
END LOOP;
...

```

3.1.2 Računovodski podatki

Če hočemo v sistemu prikazovati rezultate, morajo biti merljivi. Naše področje zajema računovodske in knjigovodske podatke finančne funkcije, na osnovi katerih merimo uspešnost poslovnih procesov. Ti se zbirajo v centralni aplikaciji podjetja, ki združuje vse poslovne funkcije. Vsaka poslovna funkcija je en modul aplikacije. Predvsem nas zanimajo poslovne knjige podjetja. V njih so posamezni poslovni dogodki – vsebinsko, časovno, prostorsko, količinsko in vrednostno opredeljene enote poslovnega procesa [10]. Običajno izberemo najmanjši možni nivo za zrno (ang. grain) merljivosti podatkov, v našem primeru poslovni dogodek, posamezen zapis v poslovnih knjigah. Ti so sicer že prikazani v obstoječi rešitvi BI, lahko bi se osredotočili bolj na ostale potrebne nivoje logičnega združevanja teh po izbranih pravilih, vendar nam bo osnovni nivo zrna po poslovnih dogodkih prišel prav. Tako ne bo potrebe po poizvedovanju v operativni bazi, pripravljene bomo pa tudi za nadaljnji razvoj modela skladišča.

Vknjižbe na določenemu knjigovodskemu računu same ne določajo narave poslovnega dogodka, zato se beležijo na različne konte, poslovne partnerje, stroškovna mesta, stroškovne nosilce, vrste prometa in seveda na določeni datum. To so glavna določila poslovnega dogodka, ki nas zanimajo. Iz tega vidimo, da ni dovolj samo združevati podatke skozi čas. Poznati moramo naravo posameznega poslovnega dogodka, s tem smo tudi podrobno določili zrno.

3.1.3 Konto

Vsak konto ima svoj naziv in namen. Konti so razvrščeni v skupine, ki določajo prihodke, odhodke, kupce, dobavitelje, glavno knjigo in podobno. Urejeni so tudi hierarhično, poznamo razrede kontov ter njihove podkote. Vse te lastnosti so določene v šifrantu kontni plan. Splošno pravilo je, da v razred konta (npr. 2) spadajo vsi konti, ki se začnejo z isto oznako (npr. 220010), v sintetiko razreda 2 vsi konti, ki se začnejo z istima dvema znakoma, enako

velja za sintetiko razreda 3. Iz danega konta lahko določimo, v katero sintetiko in razred spada.

Primer klasifikacije konta 471000 [11]:

konto	: 471000	Plače zaposlencev
sintetika 3:	471	Plače zaposlencev
sintetika 2:	47	Stroški dela
razred	: 4	Stroški storitev

Z naslednjo kodo smo napolnili dimenzijo DIM_KONTO. Iz nje je razvidna tudi struktura dimenzije.

```

...
INSERT INTO dim_konto (konto_key, konto, konto_naziv, tip_konta
, sintetika3, sintetika3_naziv
, sintetika2, sintetika2_naziv
, razred, razred_naziv
, vrsta_kontnega_plana, stran_knjizenja)
SELECT g_key_maksi + ROWNUM, konto, naziv, tip
, s3, NVL ((SELECT naziv FROM mcr_kontni_plan
WHERE vrsta_kopl = a.vrsta_kopl
AND konto = a.s3), 'Ni podatka')
, s2, NVL ((SELECT naziv FROM mcr_kontni_plan
WHERE vrsta_kopl = a.vrsta_kopl
AND konto = a.s2), 'Ni podatka')
, s1, NVL ((SELECT naziv FROM mcr_kontni_plan
WHERE vrsta_kopl = a.vrsta_kopl
AND konto = a.s1), 'Ni podatka')
, vrsta_kopl, stran_knj
FROM (SELECT konto, naziv, tip
, SUBSTR (konto,1,3) s3
, SUBSTR (konto,1,2) s2
, SUBSTR(konto,1,1) s1
, vrsta_kopl, stran_knj
FROM mcr_kontni_plan
WHERE LENGTH (konto) > 3
AND vrsta_kopl = NVL(p_kopl, vrsta_kopl)
AND konto = NVL(p_konto, konto)
ORDER BY vrsta_kopl, konto
) a
WHERE NOT EXISTS (SELECT 1 FROM dim_konto k
WHERE k.vrsta_kontnega_plana=a.vrsta_kopl
AND k.konto=a.konto);
g_rowcount := SQL%ROWCOUNT;
...

```

Obstajajo podatki, ki imajo v operativni bazi za nas neveljavne vrednosti. Ena takšnih je prazna vrednost NULL. Za te primere smo morali poskrbeti z dodatno vrstico v dimenziji, pod katero beležimo neopredeljene podatke. Naslednja koda prikazuje vstavitvev takšne vrstice za vsako različno vrsto kontnega plana.

```

INSERT INTO dim_konto (konto_key, konto, konto_naziv, tip_konta

```

```

, sintetika3, sintetika3_naziv
, sintetika2, sintetika2_naziv
, razred, razred_naziv
, vrsta_kontnega_plana, stran_knjizenja)
SELECT g_key_mini - ROWNUM, '599999', 'Neopredeljeno', NULL
, '599', 'Neopredeljeno'
, '59', 'Neopredeljeno'
, '5', 'Neopredeljeno'
, vrsta_kontnega_plana, ''
FROM (SELECT DISTINCT vrsta_kontnega_plana
      FROM dim_konto)
WHERE NOT EXISTS
      (SELECT 1 FROM dim_konto k
       WHERE k.vrsta_kontnega_plana=a.vrsta_kontnega_plana
         AND k.konto='599999'
         AND konto_key < 0);

```

Neopredeljene podatke kontov smo razvrstili na konto 599999. Po uradnem listu Republike Slovenije je razred 5 prost za uporabo [11]. Vrednost za glavni ključ konto_key smo za te zapise izbrali z negativnim predznakom, da se ne mešajo med ostale podatke in so dobro opazni.

3.1.4 Stroškovno mesto

Organizacija podatka stroškovno mesto sloni na širšem pojmu organizacijske enote. Vanjo so vključeni še ostali načini razvrščanja enot, tudi po hierarhiji, ki pa nima predpisane strukture. Nivo gnezdenja organizacijskih enot med seboj je teoretično neskončen. Še bolj se pa zaplete, ko se tip nadrejene enote razlikuje od tipa podrejene. Osnovni koncept naj bi bil takšen, da je glavni nivo organizacijska enota tipa OEZR, ki lahko ima več podrejenih stroškovnih mest. Le-te so tudi organizacijske enote, vendar tipa STM. Ta preprost koncept vzdrži logiko operativne aplikacije in nam olajša delo. Še vedno pa imamo problem s kombinacijami nadrejenih in podrejenih enot. Da se izognemo tej kompleksnosti, smo zajeli samo tiste organizacijske enote in stroškovna mesta, ki se dejansko pojavijo v zajetih podatkih poslovnih dogodkov. Dimenzijo DIM_STROSKOVNO_MESTO smo ustvarili, napolnili pa smo jo kasneje z naslednjo proceduro:

```

...
INSERT INTO dim_stroskovno_mesto (stm_key, stm, stm_sifra, stm_naziv
                                , oezr, oezr_sifra, oezr_naziv)
SELECT g_key_maksi + ROWNUM, stm, stm_sifra, stm_naziv
, nalog_oezr, nalog_oezr_sifra_pk, nalog_oezr_naziv
FROM (SELECT DISTINCT stm, stm_sifra, stm_naziv, nalog_oezr
      , nalog_oezr_sifra_pk, nalog_oezr_naziv
      FROM stage_temeljnica
      WHERE t.nalog_oezr = NVL (p_oezr, t.nalog_oezr)
        AND t.stm = NVL (p_stm, t.stm)
        AND NOT EXISTS (SELECT 1 FROM dim_stroskovno_mesto stm
                       WHERE stm.oezr=t.nalog_oezr AND stm.stm=t.stm)
      ORDER BY stm, stm_sifra
) a;
g_rowcount := SQL%ROWCOUNT;

INSERT INTO dim_stroskovno_mesto (stm_key, stm, stm_sifra, stm_naziv, oezr

```

```

, oezr_sifra, oezr_naziv)
SELECT g_key_mini - ROWNUM, -1, '-1', 'Neopredeljeno', oezr
, oezr_sifra, oezr_naziv
FROM ( SELECT -1 oezr, '-1' oezr_sifra, 'Neopredeljeno' oezr_naziv
FROM dual
UNION ALL
SELECT DISTINCT oezr, oezr_sifra, oezr_naziv
FROM dim_stroskovno_mesto
) t
WHERE NOT EXISTS (SELECT 1 FROM dim_stroskovno_mesto stm
WHERE stm.oezr=t.oezr AND stm.stm=-1);
...

```

V kodi je razvidno, da smo uporabili tabelo STAGE_TEMELJNICA. Kako smo jo pripravili, bomo opisali v poglavju 3.1.6.

3.1.5 Poslovni partner, vrsta prometa in stroškovni nosilec

Ostale dimenzije so bolj ali manj splošne in enostavne. Podatke zanje smo zajeli iz šifrantov operativne baze brez dodatnih preoblikovanj. Nato smo vstavili še splošne zapise z negativno vrednostjo glavnega ključa, podobno kot pri dimenziji konto.

Le pri vrstah prometa imamo posebnost, da je sestavljena iz štirih nivojev. Glavni nivo je vrsta analitike, nato so grupa dokumenta, vrsta dokumenta ter še končna vrsta prometa. V operativni bazi so ti podatki predstavljeni s štirimi različnimi tabelami, ki so normalizirane; dimenzijo DIM_VRSTA_PROMETA pa smo sestavili tako, da so vsi štirje nivoji prisotni v istem zapisu.

3.1.6 Tabela dejstev poslovnih dogodkov

Tabelo dejstev smo ustvarili s povezavami na dimenzije, ki smo jih že pripravili. Pomembni podatki so še zneski oziroma debet in kredit, ki smo jih določili za mere. Poleg tega smo dodali še nekaj opisnih podatkov, ki so prav tako pomembne lastnosti poslovnega dogodka, kot je opredeljen v operativni aplikaciji, ne spadajo pa v nobeno izmed dimenzij. Ob tem smo pripravili tudi prehodno tabelo, ki vsebuje zajete podatke iz operativne baze, ki smo jih nato preoblikovali in prenesli v tabelo dejstev. Koda za pripravo tabele:

```

CREATE TABLE STAGE_TEMELJNICA
(NALOG_PK VARCHAR2(8)
, NALOG_POZICIJA_PK VARCHAR2(5)
, NALOG_OEZR NUMBER(5)
, NALOG_OEZR_SIFRA_PK VARCHAR2(10)
, NALOG_OEZR_NAZIV VARCHAR2(60)
, NALOG_OPIS VARCHAR2(60)
, DNEVNIK NUMBER(5)
, DEBET NUMBER(18,2)
, KREDIT NUMBER(18,2)
, DOKUMENT VARCHAR2(15)
, DOKUMENT_ORIG VARCHAR2(21)
, DATUM_KEY NUMBER(5)
, DATUM_DOKUMENTA DATE
, KONTO_KEY NUMBER(5)

```

```

,VRSTA_KONTNEGA_PLANA VARCHAR2(2)
,KONTO VARCHAR2(12)
,VRSTA_PRM_KEY NUMBER(5)
,VRSTA_PRM NUMBER(3)
,VRSTA_DOK NUMBER(2)
,GRUPA_DOK NUMBER
,VRSTA_ANALITIKE NUMBER
,STM_KEY NUMBER(5)
,STM NUMBER
,STM_SIFRA VARCHAR2(10)
,STM_NAZIV VARCHAR2(60)
,STN_KEY NUMBER(5)
,STN VARCHAR2(12)
,PPARTNER_KEY NUMBER(5)
,PPARTNER VARCHAR2(10)
);

```

```

CREATE TABLE FACT_TEMELJNICA
(DATUM_KEY NUMBER(5) NOT NULL
,KONTO_KEY NUMBER(5) NOT NULL
,VRSTA_PRM_KEY NUMBER(5) NOT NULL
,STM_KEY NUMBER(5) NOT NULL
,STN_KEY NUMBER(5) NOT NULL
,PPARTNER_KEY NUMBER(5) NOT NULL
,NALOG_PK VARCHAR2(8) NOT NULL
,NALOG_POZICIJA_PK VARCHAR2(5) NOT NULL
,NALOG_OEZR_SIFRA_PK VARCHAR2(10) NOT NULL
,NALOG_OEZR_NAZIV VARCHAR2(60)
,NALOG_OPIS VARCHAR2(40)
,DNEVNIK NUMBER(10)
,DEBET NUMBER(18,2)
,KREDIT NUMBER(18,2)
,DOKUMENT VARCHAR2(15)
,DOKUMENT_ORIG VARCHAR2(21)
);

```

```

ALTER TABLE FACT_TEMELJNICA
ADD CONSTRAINT FACT_TEMELJNICA_PK PRIMARY KEY
(NALOG_PK
,NALOG_POZICIJA_PK
,NALOG_OEZR_SIFRA_PK);

```

```

ALTER TABLE FACT_TEMELJNICA
ADD CONSTRAINT FTEM_DSTM_FK FOREIGN KEY (STM_KEY)
REFERENCES DIM_STROSKOVNO_MESTO (STM_KEY);

```

```

ALTER TABLE FACT_TEMELJNICA
ADD CONSTRAINT FTEM_DDAT_FK FOREIGN KEY (DATUM_KEY)
REFERENCES DIM_DATUM (DATUM_KEY);

```

```

ALTER TABLE FACT_TEMELJNICA
ADD CONSTRAINT FTEM_DSTN_FK FOREIGN KEY (STN_KEY)
REFERENCES DIM_STROSKOVNI_NOSILEC (STN_KEY);

```

```

ALTER TABLE FACT_TEMELJNICA
ADD CONSTRAINT FTEM_DVP_FK FOREIGN KEY (VRSTA_PRM_KEY)
REFERENCES DIM_VRSTA_PROMETA (VRSTA_PRM_KEY);

```

```

ALTER TABLE FACT_TEMELJNICA ADD CONSTRAINT FTEM_DKON_FK FOREIGN KEY (KONTO_KEY)
REFERENCES DIM_KONTO (KONTO_KEY);

```

```

ALTER TABLE FACT_TEMELJNICA
ADD CONSTRAINT FTEM_DPP_FK FOREIGN KEY (PPARTNER_KEY)
REFERENCES DIM_POSLOVNI_PARTNER (PPARTNER_KEY);

```

Napolnili smo prehodno tabelo STAGE_TEMELJNICA:

```

INSERT INTO stage_temeljnica (nalog_pk, nalog_pozicija_pk, nalog_oezr
, nalog_oezr_sifra_pk, nalog_oezr_naziv, nalog_opis, dnevnik
, debet, kredit, dokument, dokument_orig
, datum_key, datum_dokumenta, konto_key
, vrsta_kontnega_plana, konto, vrsta_prm_key, vrsta_prm
, vrsta_dok, grupa_dok, vrsta_analitike, stm_key, stm
, stm_sifra, stm_naziv, stn_key, stn
, ppartner_key, ppartner)
SELECT fpt.nalog, fpt.pozicija, fpt.s_oe_zr
, o1.orig_oe, o1.n_oe, fpt.opis, fpt.jurnal
, fpt.debet, fpt.kredit, fpt.dokument, fpt.orig_dokument
, NULL datum_key, fpt.datum_dokumenta, NULL konto_key
, fpt.vrsta_kopl, fpt.konto, NULL vrsta_prm_key, fpt.vp
, fpt.vd, fpt.gr_dok, fpt.vr_anal, NULL stm_key, fpt.stm
, o2.orig_oe, o2.n_oe, NULL stn_key, fpt.sif_nos
, NULL ppartner_key, fpt.id_pp
FROM mcr_fin_postavke_temeljnic fpt
, mcr_org_enote o1
, mcr_org_enote o2
WHERE fpt.s_oe_zr = o1.s_oe
AND fpt.stm = o2.s_oe (+)
AND fpt.datum_obdobja >= mcrbi01.datum
AND (fpt.vp<>0 OR TO_CHAR(fpt.datum_obdobja,'DD.MM.') <> '31.12.');
```

S tem postopkom smo pripravili tudi podatke o uporabljenih stroškovnih mestih, o katerih smo pisali v poglavju 3.1.4. Nato smo dejansko napolnili tudi dimenzijo stroškovnih mest. S tem smo pripravili vse dimenzije, ki so vezane na tabelo dejstev FACT_TEMELJNICA. Tako smo pridobili podatke, ki se nanašajo na pripravljene dimenzije, nato smo jih lahko povezali s prehodno tabelo in jo napolnili s potrebnimi tujimi ključi, kar prikazuje naslednja koda:

```

UPDATE stage_temeljnica t
SET datum_key = (SELECT d.datum_key
FROM dim_datum d
WHERE d.datum = t.datum_dokumenta)
, konto_key = (SELECT k.konto_key
FROM dim_konto k
WHERE k.vrsta_kontnega_plana = t.vrsta_kontnega_plana
AND NVL(k.konto, '599999') = NVL(t.konto, '599999'))
, vrsta_prm_key = NVL((SELECT v.vrsta_prm_key
FROM dim_vrsta_prometa v
WHERE v.vrsta_analitike = t.vrsta_analitike
AND v.grupa_dok = t.grupa_dok
AND v.vrsta_dok = t.vrsta_dok
AND v.vrsta_prm = t.vrsta_prm)
, -1)
, stm_key = NVL((SELECT s.stm_key
FROM dim_stroskovno_mesto s
WHERE NVL(s.stm, -1) = t.stm
AND NVL(s.stm_sifra, '-1') = t.stm_sifra
AND NVL(s.stm_naziv, 'Neopredeljeno') = t.stm_naziv
AND s.oezr = t.nalog_oezr
AND s.oezr_sifra = t.nalog_oezr_sifra_pk
AND s.oezr_naziv = t.nalog_oezr_naziv)
, -1)
, stn_key = NVL((SELECT sn.stn_key
FROM dim_stroskovni_nosilec sn
WHERE sn.stn = NVL(t.stn, 'N/A'))
, -1)
, ppartner_key = NVL((SELECT p.ppartner_key
```

```

FROM dim_poslovni_partner p
WHERE p.ppartner = NVL(t.ppartner, 'N/A')
, -1);

```

Zagotovili smo še, da so se napolnili vsi ključni, ter prenesli dejansko potrebne pripravljene podatke v tabelo dejstev.

3.1.7 Planski podatki

Od vodstva podjetja smo prejeli plansko preglednico Excel, na osnovi katere moramo izdelati nekatere kazalce in kazalnike. Ne glede na to, da sistem podpira uporabo tudi takšnih virov podatkov, smo se odločili, da bomo uporabili drugi pristop. Podatke iz preglednice smo uvozili v podatkovno bazo ter nato naredili proces ETL, ki preoblikuje te podatke in jih uvozi v naše podatkovno skladišče. Takšni podatki so prečiščeni in manj obremenjujoči za sistem. Planski podatki so razmeroma preproste oblike, pripravljene so enkrat letno in se jih tudi samo enkrat na leto uvozi v sistem. Na ta način ohranimo tudi zgodovino plana, kar bi sicer izgubili, če bi izbrali za vir datoteko Excel. Naslednje leto bi le-to morali nadomestiti z novo ali pa dodatno razširiti in dopolniti model skladišča. S tem smo omogočili tudi primerjavo realizacije proti planu tudi med več preteklimi leti. Z vidika administratorja sistema pa smo se s tem izognili marsikateremu problemu, med drugim tudi na primer temu, da bi nekdo datoteko pokvaril, jo odstranil ali premaknil iz predvidenega mesta.

Seznam v planski preglednici Excel je narejen na podlagi pravil, ki so določena v operativni aplikaciji. Narejen je načrt, kateri konti se združujejo, seštevajo ali odštevajo, kateri so izločeni in podobno. Obstajajo tudi kompleksnejša pravila in formule za prikaz določenih kontov, sestavljena iz bolj preprostih pravil. Izbrali smo samo pravila, ki jih potrebujemo za obravnavo planskih podatkov in jih prenesli v novo dimenzijo DIM_ANALIZA.

Oblikovali smo fiksno hierarhijo kontov s petimi nivoji in zagotovili, da je vseh pet nivojev določenih v vsakem primeru. Kjer je končni konto na višjem nivoju, se isti prepíše tudi vse do petega.

Podatki so razdeljeni po posameznih računovodskih postavkah, zneski so vpisani na ravni celega leta. Razbili smo jih na posamezne mesece preprosto tako, da smo jih razdelili na dvanajstine. Poleg tega smo prejeli podatke le za tekoče leto, za potrebe razvoja in testa pa smo jih izpolnili še za 3 pretekla leta. Te smo generirali na osnovi prejetih podatkov z izbrano formulo (1).

$$x_i \text{ leto, mesec} = \text{ROUND } x_i \text{ leto} + 1, \text{ mesec} \times \text{RANDOM } 1 + 0.65, 6 \quad (1)$$

Funkcija RANDOM je deterministična funkcija in vrne naključno decimalno število večje ali enako 0 ter manjše od 1. S tem smo pridobili podatke z odstopanjem [0.65, 1.65) od prvotne vrednosti, zaokroženo na 6 decimalnih mest, kot so podatki v podatkovni bazi.

Nekatera pravila prikazujejo posamezne konte, druga pa združujejo več kontov, ki spadajo v skupno kategorijo. Ta pravila slonijo na dimenziji kontov in jih združujejo v skupine in podskupine. Glavni skupini sta odhodki in prihodki. Koda za pripravo agregiranih podatkov iz tabele dejstev FACT_TEMELJNICA na osnovi pravil združevanja kontov:

```

PROCEDURE extract_agg_tem
IS
  CURSOR c1
  IS
    SELECT *
      FROM dim_analiza;

  CURSOR c2 (cp_sifra NUMBER, cp_sestava NUMBER)
  IS
    SELECT analiza_sifra, analiza_sestava, analiza_razdelitev
           , analiza_konto_od, analiza_konto_do, analiza_stran
      FROM meta_anal_sest_razdelitve
     WHERE analiza_sifra = cp_sifra
           AND analiza_sestava = cp_sestava;

  CURSOR c3 (cp_konto_od VARCHAR2, cp_konto_do VARCHAR2)
  IS
    SELECT konto_key, konto, vrsta_kontnega_plana, stran_knjizenja
      FROM dim_konto
     WHERE konto BETWEEN cp_konto_od AND cp_konto_do;

  CURSOR c4 (cp_konto_key NUMBER, cp_leto VARCHAR2)
  IS
    SELECT d.last_day.datum_key, d.last_day.leto_mesec, k.konto_key
           , stm.stm_key, stm.stm_sifra, stm.oezr_sifra, sn.stn_key, sn.stn
           , NVL (SUM (ft.debet), 0) AS debet
           , NVL (SUM (ft.kredit), 0) AS kredit
      FROM dim_konto k
           , dim_stroskovni_nosilec sn
           , dim_stroskovno_mesto stm
           , dim_datum d
           , dim_datum d_last_day
           , fact_temeljnica ft
     WHERE ft.datum_key = d.datum_key
           AND d.leto_mesec = d_last_day.leto_mesec
           AND d_last_day.datum = LAST_DAY (d.datum)
           AND d.leto >= cp_leto
           AND ft.konto_key = k.konto_key
           AND k.konto_key = cp_konto_key
           AND ft.stm_key = stm.stm_key
           AND ft.stn_key = sn.stn_key
     GROUP BY d_last_day.datum_key, d_last_day.leto_mesec, k.konto_key, stm.stm_key
           , stm.stm_sifra, stm.oezr_sifra, sn.stn_key, sn.stn;

  l_debet          NUMBER (18,2);
  l_kredit         NUMBER (18,2);
  l_saldo          NUMBER (18,2);
BEGIN
  g_rowcount := 0;
  FOR r1 IN c1
  LOOP
    FOR r2 IN c2 (r1.analiza_sifra, r1.analiza5_sestava)
    LOOP

```

```

FOR r3 IN c3 (r2.analiza_konto_od, r2.analiza_konto_do)
LOOP
  FOR r4 IN c4 (r3.konto_key, mcrbi01.leto)
  LOOP
    IF r2.analiza_stran = 'D'
    THEN
      l_saldo := r4.debet - r4.kredit;
    ELSIF r2.analiza_stran = 'K'
    THEN
      l_saldo := r4.kredit - r4.debet;
    ELSE
      l_saldo := 0;
    END IF;

    INSERT INTO stage_tem_mes_kon_stm_stn_anal
      (datum_key, leto_mesec, konto_key, konto
      , vrsta_kontnega_plana, stm_key, stm_sifra, oezr_sifra
      , stn_key, stn, analiza_key, analiza_sifra
      , analiza_naziv, analiza5_oznaka, analiza5_opis
      , analiza5_konto_od, analiza5_konto_do, analiza5_stran
      , debet, kredit, saldo, plan)
    VALUES (r4.datum_key, r4.leto_mesec, r3.konto_key, r3.konto
      , r3.vrsta_kontnega_plana, r4.stm_key, r4.stm_sifra, r4.oezr_sifra
      , r4.stn_key, r4.stn, r1.analiza_key, r1.analiza_sifra
      , r1.analiza_naziv, r1.analiza5_oznaka, r1.analiza5_opis
      , r2.analiza_konto_od, r2.analiza_konto_do, r2.analiza_stran
      , r4.debet, r4.kredit, l_saldo, 0);
    g_rowcount := g_rowcount +1;
    IF MOD(g_rowcount,5000)=0
    THEN
      COMMIT;
    END IF;
  END LOOP;
END LOOP;
END LOOP;
END LOOP;

COMMIT;
END extract_agg_tem;

```

Z naslednjim insert stavkom pa smo pripravljene podatke prenesli v tabelo vmesnih seštevkov iz dejstev z dodano porazdelitvijo glede na pravila združevanja kontov:

```

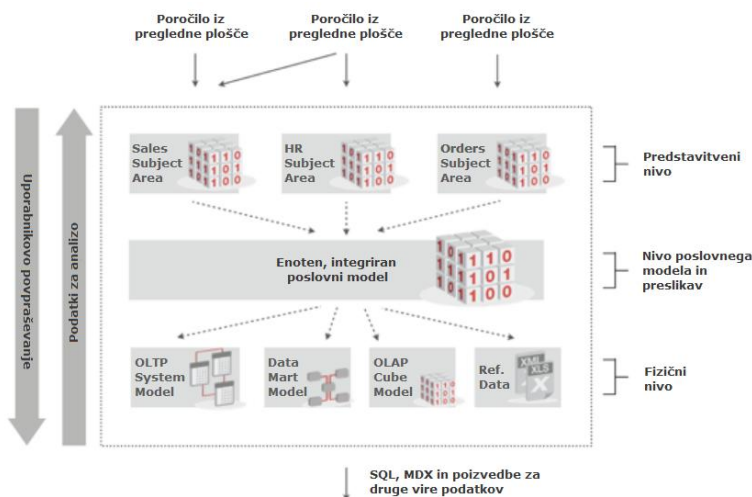
INSERT INTO agg_tem_mes_kon_stm_stn_anal
  (datum_key, konto_key, stm_key, stn_key, analiza_key
  , debet, kredit, saldo, plan)
SELECT datum_key, konto_key, stm_key, stn_key, analiza_key
  , debet, kredit, saldo, plan
FROM stage_tem_mes_kon_stm_stn_anal s
WHERE NOT EXISTS (SELECT 1 FROM agg_tem_mes_kon_stm_stn_anal a
  WHERE a.datum_key=s.datum_key
  AND a.konto_key=s.konto_key
  AND a.stm_key=s.stm_key
  AND a.stn_key=s.stn_key
  AND a.analiza_key=s.analiza_key);

```

3.2 Model podatkovnega skladišča in administratorsko orodje

Osrednji del sistema je model skladišča. Narejen je z administratorskim orodjem, ki je del programskega paketa OBIEE 11g. Orodje je razdeljeno na tri nivoje:

- fizični nivo (Physical Layer),
- poslovni model in preslikave (Business Model and Mapping Layer) in
- predstavitveni nivo (Presentation Layer).



Slika 5: Primer trinivojskega semantičnega modela z mnogimi viri podatkov

Slika 5 predstavlja strukturo modela skladišča ter prikazuje potek povpraševanja iz končne analize do virov podatkov. Na najnižjem nivoju imamo heterogene vire, ki jih nato preoblikujemo v en sam poslovni model. Ko povpraševanje prispe do fizičnega nivoja, se strežnik Oracle BI na osnovi strukture in povezav odloči, katere vire bo uporabil in nato izvorno povpraševanje razčleni na več posameznih, ki jih pošlje vsako svojemu viru. Ko prejme vse rezultate, jih logično združi in posreduje navzgor po nivojih.

3.2.1 Fizični nivo

Na fizičnem nivoju določimo izvor podatkov. Lahko jih zajemamo iz raznovrstnih virov, kot so različne podatkovne baze: OLTP (ang. OnLine Transaction Processing), OLAP (ang. OnLine Analytical Processing), področno podatkovno skladišče (ang. Data Mart); poslovni proces, ERP (ang. Enterprise Resource Planning) in CRM (ang. Customer Relationship Management) sistemi ter podatkovne datoteke (preglednice Excel, XML datoteke in podobno). Najbolj optimalna oblika virov podatkov je podatkovno skladišče OLAP, saj je že tako organizirano, da je mogoče hitro poizvedovanje in ni potrebno veliko dela, da bi v našem modelu skladišča prilagodili podatke za uporabo v končnih analizah. Pri OLTP relacijski bazi

že izgubimo hitrost zaradi mnogih povezav med normaliziranimi tabelami, razne ostale podatkovne datoteke pa niso niti relacijsko organizirane.

V administratorskem orodju smo odprli model skladišča, ki smo ga nadgradili iz verzije 10g na verzijo 11g. Z orodjem za uvoz metapodatkov smo ustvarili novi vir in zanj izbrali tabele ter zanje glavne in tuje ključe, ki smo jih predhodno ustvarili. Vir smo določili s povezavo na bazo `rb4` in shemo `MCRBI`. Z uvozom izbranih tabel smo na fizičnem nivoju ustvarili tabele, ki temeljijo na dejanskih tabelah v viru. V skladu z dobro prakso smo za vsako logično tabelo opredelili vzdevek (ang. alias). Pomen vzdevka na fizičnem nivoju modela skladišča je prav takšen, kot je v običajni notaciji SQL. Z njim smo ustvarili povezavo na fizično tabelo, ki nam služi kot nadomestna tabela s povsem enako sestavo kot osnovna tabela. Za vsako dimenzijo smo naredili toliko vzdevkov, kot je tabel z dejstvi, ki se nanjo povezujejo. Nato smo določili glavne in tuje ključe in poskrbeli za vse povezave med vzdevki. Vsak vzdevek tabele z dejstvi smo povezali z njo ustrezajoče vzdevke dimenzij v obliko, ki jo imenujemo zvezdasta shema (ang. star schema).

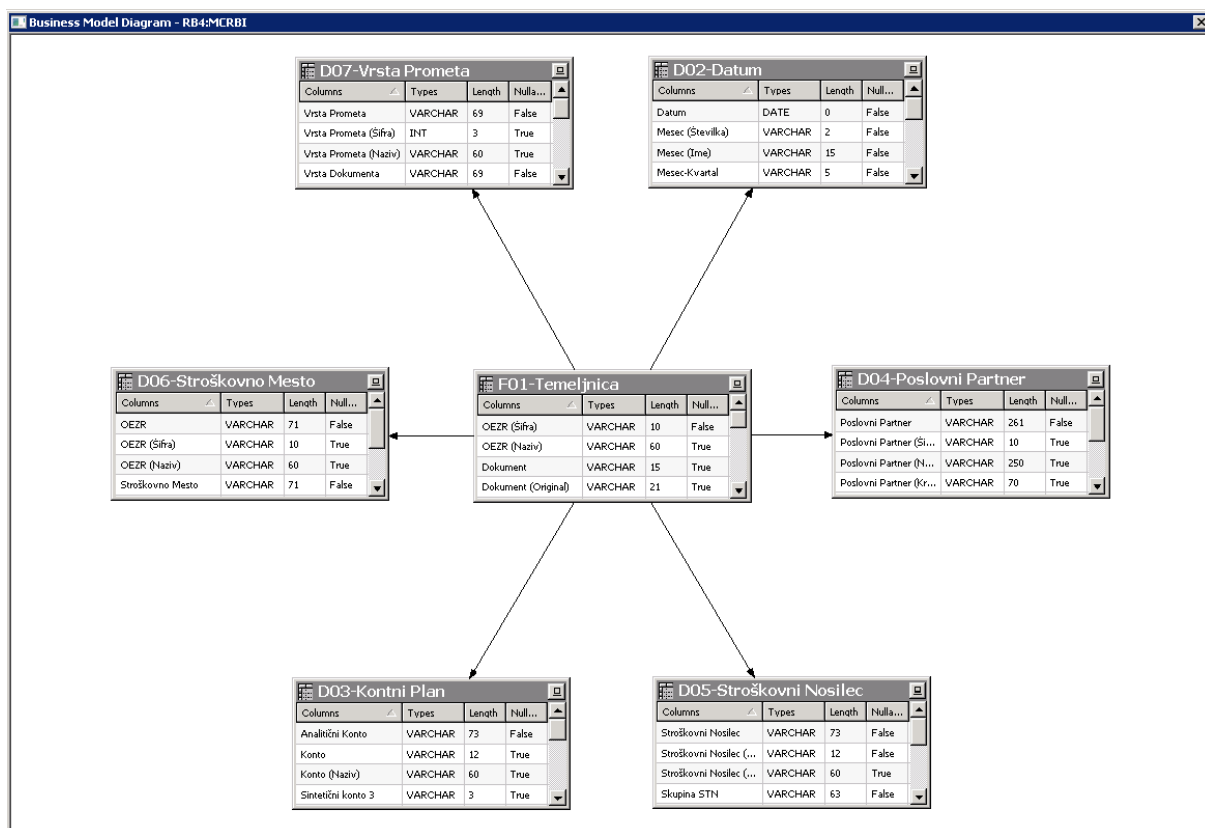
Ko določimo povezave, se lahko zgodi, da se pojavi tako imenovana krožna povezava. To pomeni, da lahko dostopamo do določene tabele na dva različna načina, kar podre strukturo podatkovnega skladišča. Ker moramo zagotoviti doslednost modela skladišča, razločimo takšne krožne povezave z uporabo vzdevkov. Naredimo dva vzdevka za isto ciljno tabelo in pravilno postavimo povezave. Pomembno torej je, da za vsako uporabo tabele naredimo lastni vzdevek.

Z upoštevanjem dobre prakse smo se izognili krožni referenci. V meniju z orodji se nahaja možnost, s katero smo iz modela skladišča preizkusili povezavo na definirani vir `MCRBI` na `rb4` (ang. Tools > Update All Row Counts). S tem smo dobili informacijo o številu zapisov za vsako tabelo.

3.2.2 Poslovni model in preslikave

Na plasti poslovnega modela in preslikav (v nadaljevanju PMP) smo ustvarili nov poslovni model in vanj združili vse področne vire v enovit sistem. Označili smo zelene tabele v fizični plasti in jih prenesli v poslovni model. S tem smo ustvarili logične tabele in za vsako tudi logični podatkovni vir LTS (ang. Logical Table Source), kjer so definirane preslikave med fizičnimi in poslovnimi tabelami in njihovimi podatki. Preverili smo, če so se povezave iz fizičnega nivoja pravilno preslikale v logične povezave med tabelami v poslovnem modelu in jih po potrebi dopolnili. Shema na nivoju PMP mora biti zvezdaste oblike (slika 6). Če ne bi imeli podatkovnega vira v obliki podatkovnega skladišča, bi ga predstavili kot takšnega v plasti poslovnega modela in preslikav. Podobno velja, če ne bi imeli urejenih povezav na

fizičnem nivoju, bi jih morali urediti na nivoju PMP. Logične povezave definirajo kardinalnost med tabelami in s tem določajo, katere tabele so dejstva.



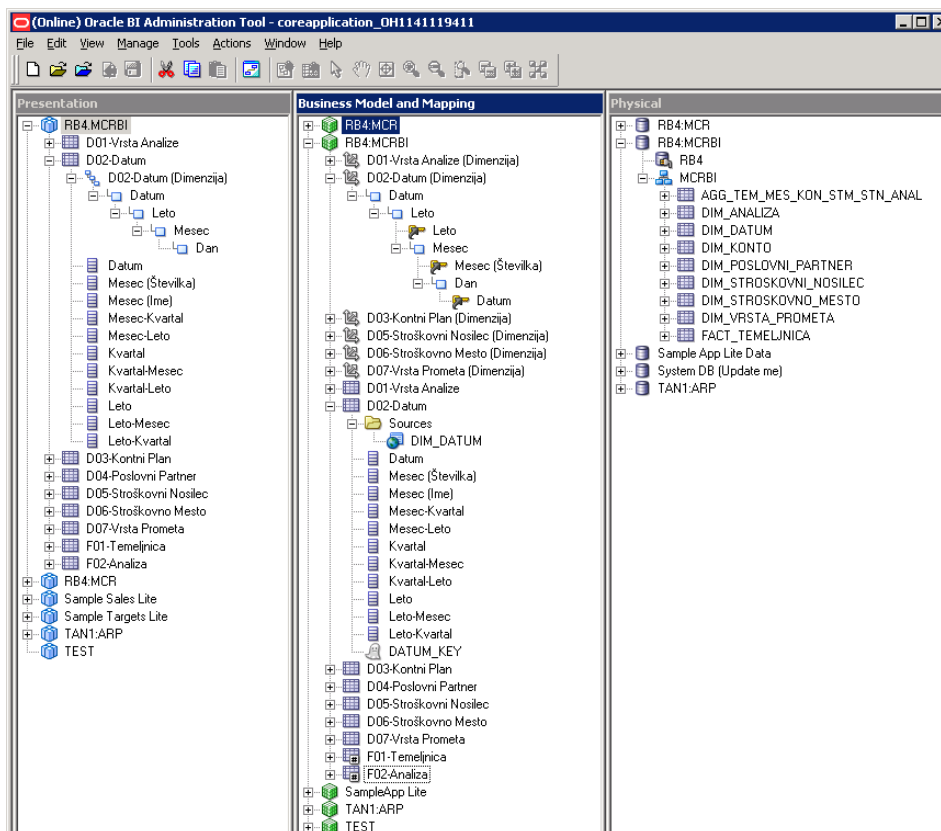
Slika 6: Shema poslovnega modela za tabelo dejstev "F01-Temeljnica"

Na tabelah v poslovnem modelu potrebujemo samo podatke, ki jih bomo uporabili za prikaz in poizvedovanje. Podatkov, ki so del logičnih povezav, ne prikazujemo, zato smo odstranili vsa polja, ki niso poslovno zanimiva in jih ne potrebujemo za izgradnjo poslovnega modela. Nadalje smo preimenovali imena tabel in polj z uporabniku bolj prijaznimi imeni. Za lažje delo pri sestavljanju analiz smo na tem mestu po potrebi izvedli tudi nova polja, sestavljena iz kombinacij šifre in naziva. Na primer: na logični tabeli za poslovne partnerje smo ustvarili novo polje "Poslovni Partner", ki sloni na poljih šifra in Naziv. Definicija novega polja je:

```
"RB4:MCRBI"."D04-Poslovni Partner"."Poslovni Partner (Šifra)" ||
' ' || "RB4:MCRBI"."D04-Poslovni Partner"."Poslovni Partner
(Naziv)"
```

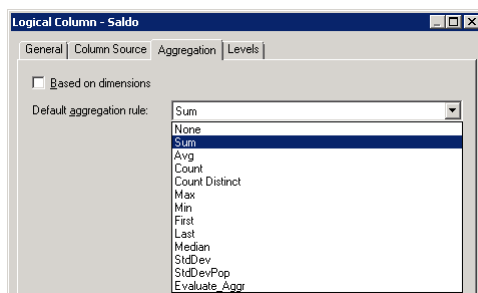
Dimenzije "D01-Vrsta Analize", "D02-Datum", "D03-Kontni Plan", "D05-Stroškovni Nosilec", "D06-Stroškovno Mesto" in "D07-Vrsta Prometa" so hierarhične oblike, zato smo iz njih izvedli hierarhije. Dimenzija "D06-Stroškovno

Mesto" teoretično predstavlja hierarhijo staršev in otrok, saj ima podrejene tipe podatkov istega tipa, kot je nadrejeni, vendar smo jo v poglavju 3.1.4 poenostavili in je zato tudi ta hierarhija nivojska kot ostale. Glavni nivo vsake hierarhije je skupni nivo, ostale nivoje smo določili z logičnimi podatki iz dimenzije, ki predstavljajo ključne posameznega nivoja. Slika 7 prikazuje razmerje med datumsko dimenzijo in njej pripadajočo hierarhijo "D02-Datum (Dimenzija)" in tudi razmerje med poslovnim in predstavitvenim modelom.



Slika 7: Administratorsko orodje s fizičnim, PMP in predstavitvenim nivojem

Vsaka tabela dejstev ima dve vrsti podatkov. Polja, ki predstavljajo povezave na dimenzije so ključni, druga pa predstavljajo podatke, ki jih lahko merimo. Vsebuje lahko tudi opisna polja, vendar ta niso potrebna in ne tako pogosta. Mere so polja, nad katerimi lahko izvajamo agregacijske funkcije (vsota, povprečje, minimum, maksimum,...). V tabelah dejstev smo vsa vrednostna polja (zneski, količine, izvedeni vrednostni podatki) označili kot mere tako, da smo na polju nastavili agregacijsko pravilo (slika 8). Ostala običajna logična polja nimajo nastavljenega agregacijskega pravila. Na tabeli "F01-Temeljnica" smo za mere izbrali podatke "Debet" in "Kredit", na tabeli "F02-Analiza" pa podatke "Debet", "Kredit", "Saldo" in "Plan".



Slika 8: Določanje agregacije za mero

Na lastnostih datumske hierarhije smo označili možnost, da je to časovna hierarhija. Ta lastnost omogoča uporabo časovnih funkcij na merljivih podatkih. S tem smo lahko na tabeli dejstev "F02-Analiza" izvedli nove merljive podatke. Nekaj jih je naštetih v tabeli 1.

Logično polje	Definicija polja
Saldo (PL)	<code>Ago ("RB4:MCRBI"."F02-Analiza"."Saldo", "RB4:MCRBI"."D02-Datum (Dimenzija)". "Leto", 1)</code>
Saldo (LDD)	<code>ToDate ("RB4:MCRBI"."F02-Analiza"."Saldo", "RB4:MCRBI"."D02-Datum (Dimenzija)". "Leto")</code>
Plan (PL)	<code>Ago ("RB4:MCRBI"."F02-Analiza"."Plan", "RB4:MCRBI"."D02-Datum (Dimenzija)". "Leto", 1)</code>
% Saldo/Saldo (PL)	<code>CASE WHEN "RB4:MCRBI"."F02-Analiza"."Saldo (PL)" <> 0 THEN IFNULL ("RB4:MCRBI"."F02-Analiza"."Saldo",0) * 100 / "RB4:MCRBI"."F02-Analiza"."Saldo (PL)" ELSE NULL END</code>
% Saldo/Plan	<code>CASE WHEN "RB4:MCRBI"."F02-Analiza"."Plan" <> 0 THEN IFNULL ("RB4:MCRBI"."F02-Analiza"."Saldo",0) * 100 / "RB4:MCRBI"."F02-Analiza"."Plan" ELSE NULL END</code>

Tabela 1: Izvedeni merljivi podatki

Funkcija "Ago" zna izračunati vrednost podatka glede na določen nivo časovne hierarhije. Mera "Saldo (PL)" je saldo prejšnjega leta tako, da smo za časovni nivo seštevanja izbrali "Leto" in število 1 za eno leto nazaj. Lahko bi izvedli vrednosti tudi za prejšnje mesece po nivoju "Mesec" ali celo četrletje po nivoju "Kvartal", če bi ga vstavili med nivoja "Leto" in "Mesec". Podobno smo pri uporabi funkcije "ToDate" izbrali časovni nivo "Leto", kar izračuna vrednost od začetka leta do tekočega dne. Tako smo dobili mero "Saldo (LDD)". Iz teh podatkov smo izvedli indekse primerjave podatka "Saldo" med tekočim in prejšnjim letom. Izvedli smo tudi indeks primerjave med dejanskimi in

pričakovanimi vrednostmi – "% Saldo/Plan" ter indeks primerjave lanskih vrednosti proti letošnjemu planu "Index Saldo (PL)/Plan". Ti novi podatki nam omogočajo, da lahko na poročilih enostavno, točno in časovno ažurno prikažemo potrebne kazalce in kazalnike.

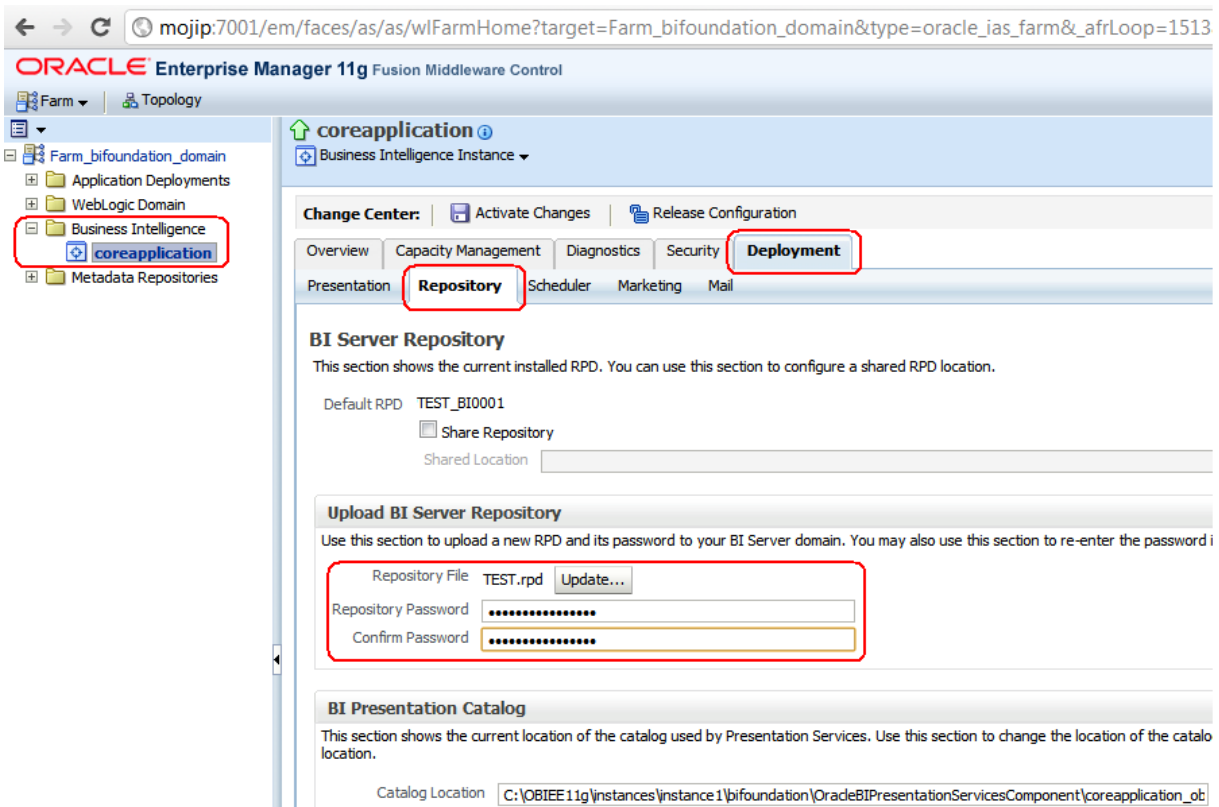
3.2.3 Predstavitveni nivo

Predstavitveni nivo zajema podatke, ki so dostopni za uporabo v končni rešitvi. Za končnega uporabnika je to vstopna točka v celoten sistem poslovnega obveščanja. Tukaj so elementi s katerimi uporabniki lahko upravljajo, podatki, iz katerih so sestavljene analize in nadzorne plošče. Predstavitveni model smo ustvarili iz poslovnega modela tako, da smo poslovni model prenesli na predstavitveni nivo. Odstranili smo še polja, ki so potrebna na nivoju PMP, niso pa potrebna za prikaz in delo. Tabele v predstavitvenem modelu in ostala polja v tabelah smo še preuredili v smiselni vrstni red.

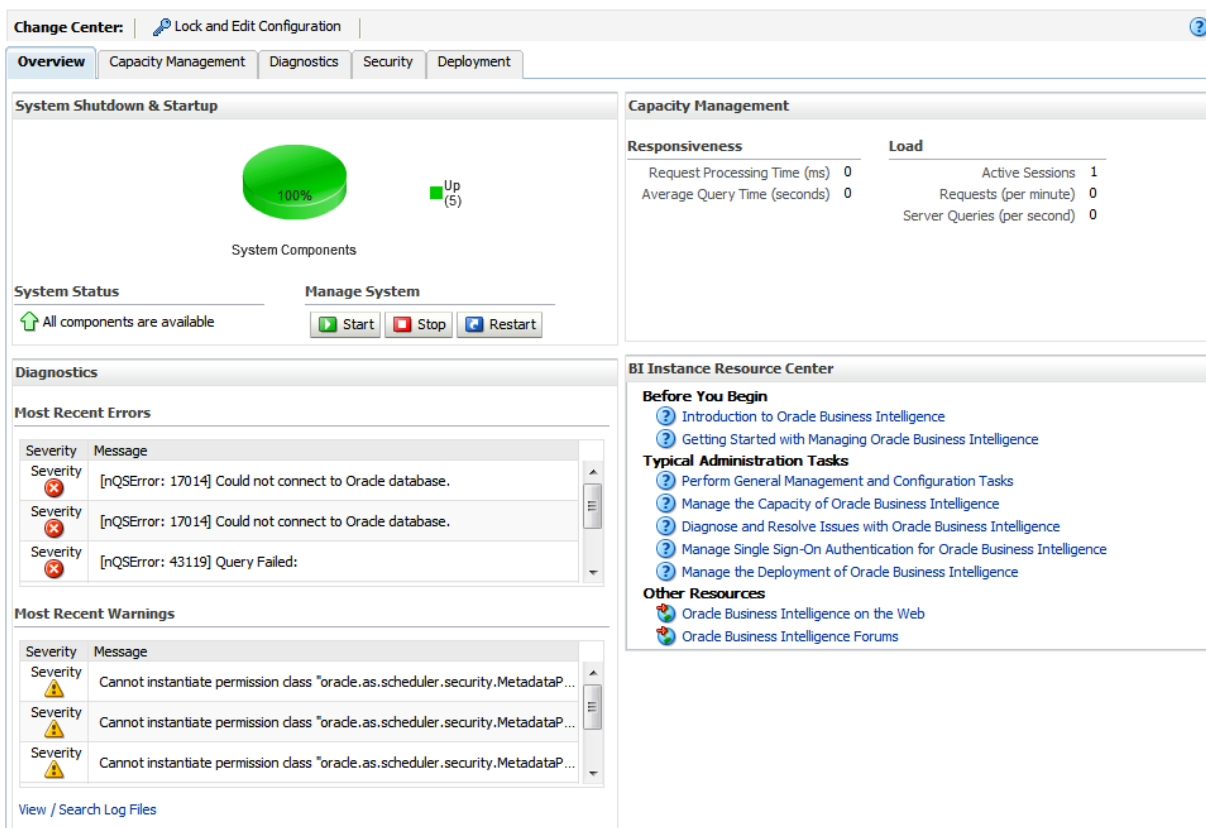
Pred zaključkom z delom na modelu skladišča smo shranili vse spremembe in preverili doslednost celotnega modela. V meniju "File" smo izbrali element menija "Check Global Consistency".

3.2.4 Namestitev modela skladišča

Namestitev modela skladišča je preprosta. V brskalniku smo odprli poslovnega upravitelja na naslovu <http://mojip:7001/em>. V levem delu okna smo razširili direktorij "Business Intelligence" in v njem izbrali "coreapplication". Na strani, ki se je odprla, smo izbrali jeziček "Deployment" in podjeziček "Repository". Tukaj smo najprej zaklenili nastavitve za urejanje s klikom na tekst "Lock and Edit configuration", nato smo izbrali naš model skladišča, vnesli geslo in potrdili izbiro z gumbom "Update", kakor je prikazano na sliki 9. S klikom na tekst "Apply changes" smo uveljavili spremembe. Potem smo na jezičku "Overview" sprožili ponoven zagon sistema za poslovno obveščanje. Ko se je sistem ponovno zagnal, smo preverili, če so vsi elementi sistema uspešno pognani (slika 10). S tem je podatkovno skladišče dokončano in aktivirano.



Slika 9: Namestitev novega modela skladišča v poslovnem upravitelju



Slika 10: Stanje strežnika BI

Pripravljeno podatkovno skladišče je preprosto in lahko služi kot osnova za razširitev poslovnega obveščanja z drugimi poslovnimi funkcijami. Iste dimenzije so ponovno uporabne, razen dimenzije stroškovnih mest, ki zahteva bolj obsežno interpretacijo, kot smo jo uporabili v temu diplomskemu delu.

Sedaj, ko je model skladišča v uporabi na sistemu BI, ga lahko z administratorskim orodjem dopolnjujemo v sprotnem (ang. online) načinu. V tem primeru ni potreben ponoven zagon sistema BI, le osvežitev metapodatkov.

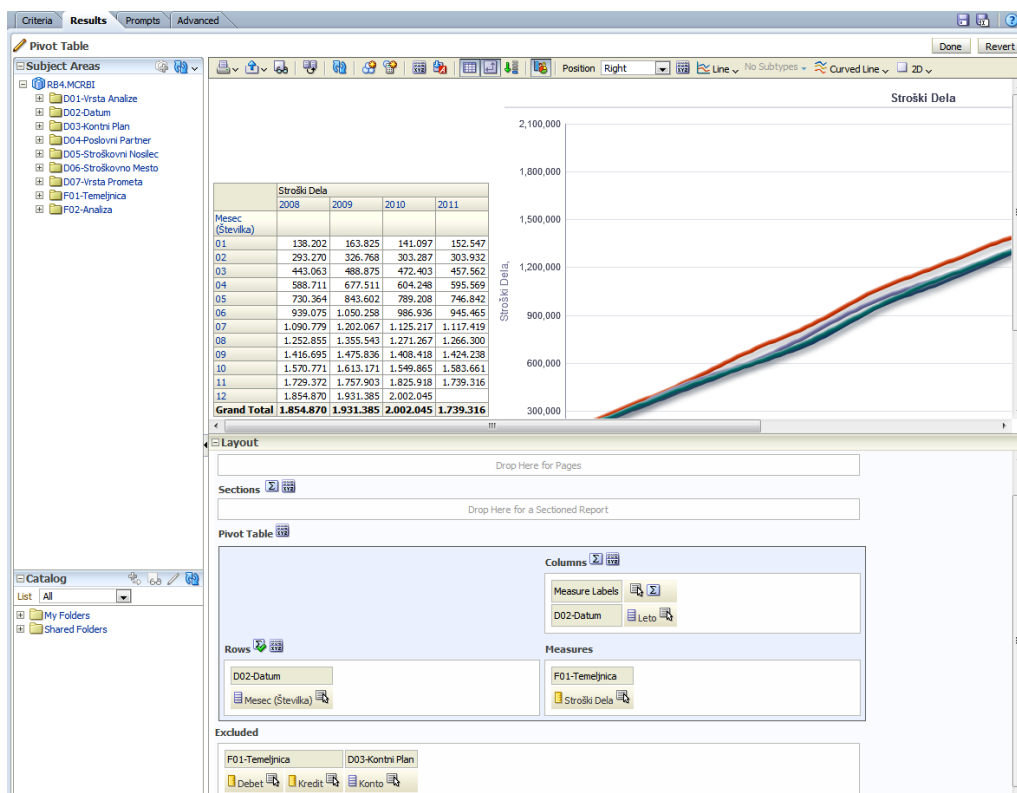
Poslovni upravitelj ob vsaki naložitvi novega ali dopoljenega modela skladišča vodi verzije le-teh. Ob morebitnih napakah lahko povrnemo prejšnjo verzijo skladišča.

4 Postavitev predstavitvenega dela sistema BI

Prijavili smo se v BI na naslovu <http://mojip:7001/analytics>. V administraciji smo spraznili predpomnilnik in osvežili metapodatke novo naloženega modela skladišča. To je treba storiti, ko spreminjamo model skladišča. Če večkrat popravljamo obstoječe objekte v modelu skladišča, lahko podremo delovanje že ustvarjenih poročil, zato se takšnemu delu izogibamo. Po vsaki dopolnitvi je treba preizkusiti novosti, zato smo pripravili nekaj novih poročil in preizkusili delovanje vseh že obstoječih. Ko smo se prepričali o ustreznosti dopolnitev, smo lahko začeli s pripravo predstavitvenega dela sistema poslovnega obveščanja.

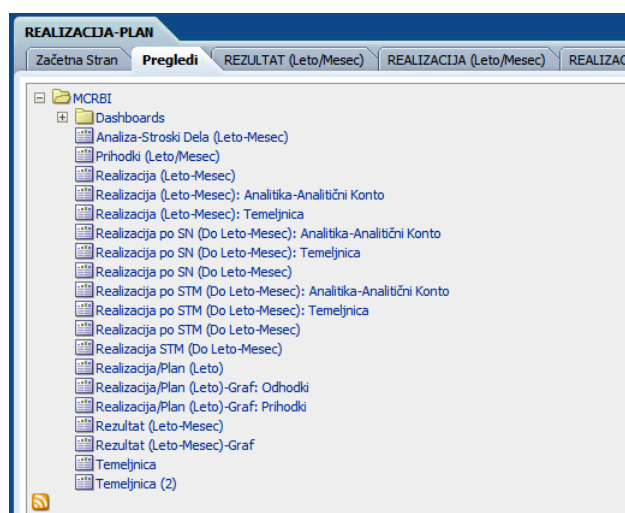
4.1 Poročila

Poročila sestavimo preprosto tako, da izberemo podatke iz dimenzij in dejstev. Na izbranih podatkih uredimo sortiranje in nastavimo pogoje za filtriranje podatkov. Filtriramo lahko tudi po podatkih, ki jih nismo vključili na poročilo, če pa hočemo, da se pogoj glede na izbrane podatke prenaša iz enega poročila v drugo, moramo nastaviti, da so ti podatki na poročilu filtrirani iz pozivnika. Nato izberemo vrsto prikaza za rezultate poročila, uredimo postavitev posameznih podatkov in shranimo poročilo. Na sliki 11 je prikazano, kako smo sestavili poročilo za analizo stroškov dela, oblikovali smo ga kot vrtilno tabelo, na koncu dodali vrstico seštevka ter izbrali opcijo za avtomatski prikaz grafa iz podatkov vrtilne tabele.



Slika 11: Priprava poročila

Za prikaz najosnovnejših vrstic z vsemi možnimi podatki smo naredili poročila v obliki preprostih tabel. Pripravili smo poročila za pregled realizacije po posameznih dimenzijah, poročila za primerjavo realizacije in plana, poročila za primerjavo realizacije med različnimi leti, poročila za prikaz posameznih podatkov in podobno (slika 12). Na osnovi izvedenih indeksov smo postavili pravila, ki določajo oblikovanje podatkov v različnih razponih vrednosti glede na postavljene omejitve (sliki 13 in 14). Na določenih podatkih smo vključili tudi povezave na druga, bolj podrobna poročila.

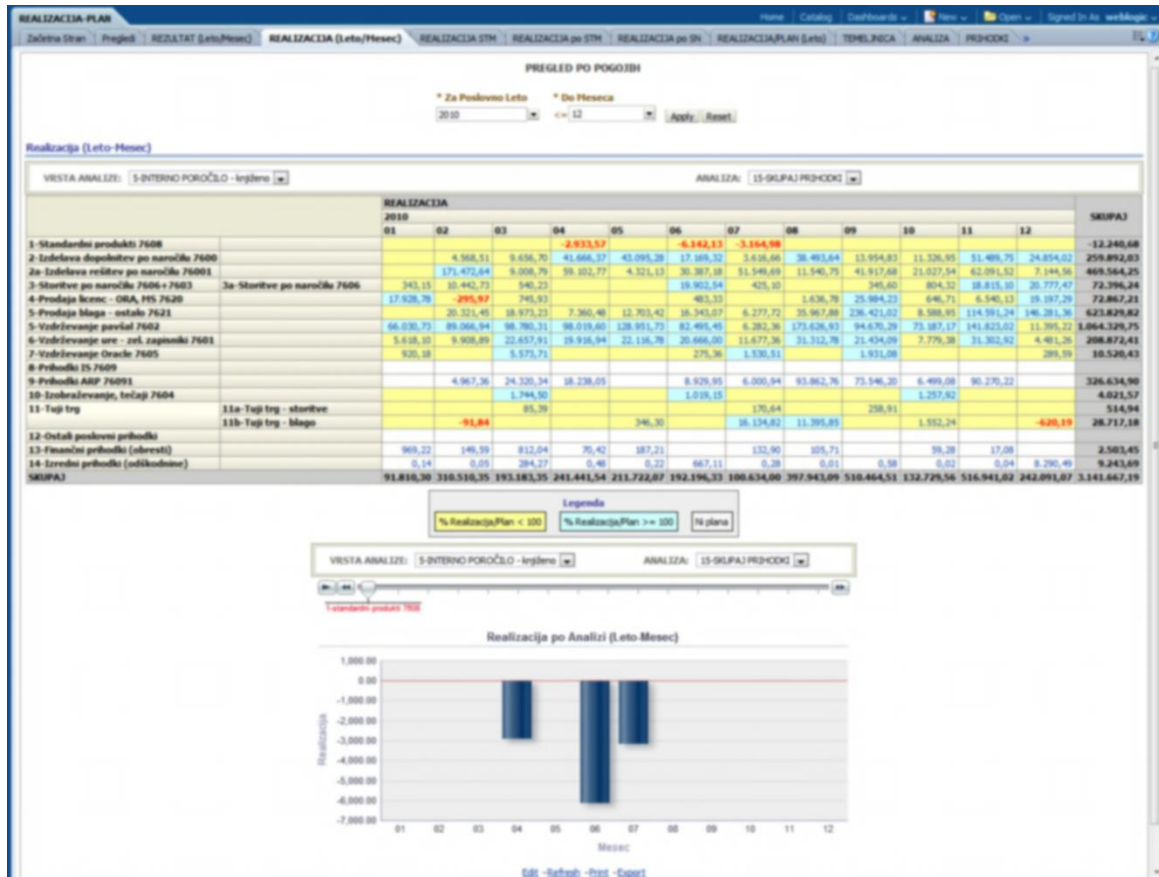


Slika 12: Pripravljena poročila

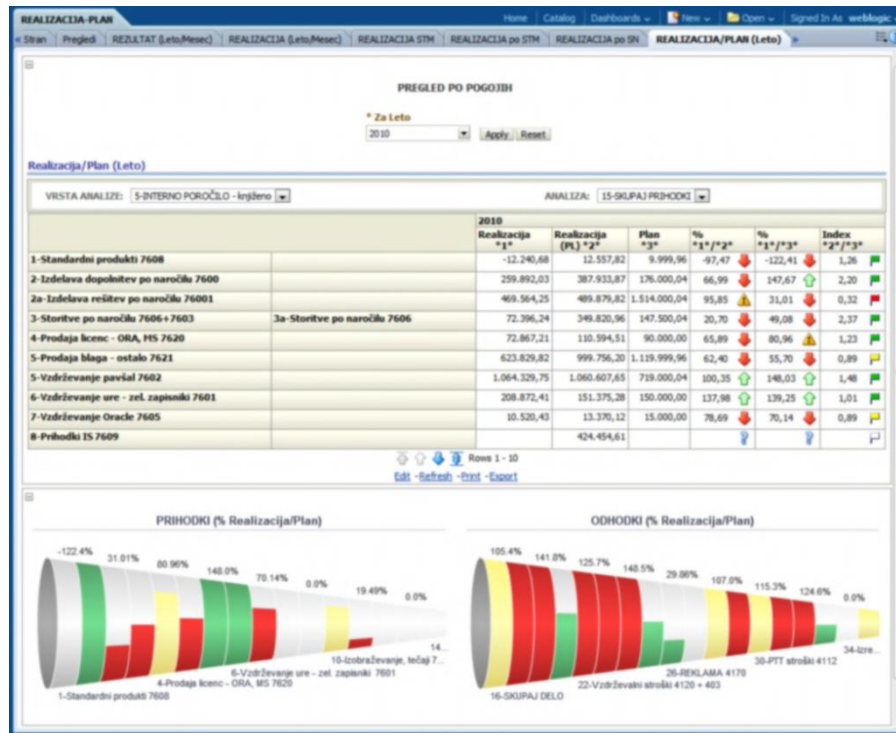
Posebej smo pripravili še pozivnike za uporabo na nadzorni plošči.

4.2 Nadzorna plošča

Za nove preglede o realizaciji in planiranih poslovnih procesih smo naredili novo nadzorno ploščo "REALIZACIJA-PLAN". Nanjo smo smiselno na več strani razdelili pripravljena poročila in ključne indikatorje uspeha. Na sliki 13 vidimo nadzorno ploščo z več stranmi, ki so dostopne kot jezički. Odprta stran "REALIZACIJA (Leto/Mesec)" je sestavljena iz pozivnika, poročila "Realizacija (Leto-Mesec)" in grafa. Oba, poročilo in graf, sta nastavljena, da se odzivata na izbrane vrednosti v pozivniku.



Slika 13: Prikaz poročil na nadzorni plošči (1)



Slika 14: Prikaz poročil na nadzorni plošči (2)

4.3 Uvajanje uporabnikov

Na koncu smo morali seznaniti uporabnike s spremembami in z novostmi. Le tako lahko zagotovimo, da bo sistem dejansko uporaben. Tudi če vse deluje brezhibno, uporabniki pa nimajo zaupanja v pravilnost njegovega delovanja, je vse zaman. Naš namen, ob tem da uspešno postavimo nov sistem, je torej dobro predstaviti končno rešitev vsem uporabnikom. Vključili smo jih tudi v sam postopek nadgradnje in razširitve sistema tako, da smo jih obveščali o spremembah in napredku. Prosili smo jih, naj preizkušajo rešitve na testnem okolju in s tem pridobili tudi koristne informacije o uporabnosti posameznih elementov. Ker smo si na takšen način vzeli več časa tudi za komunikacijo z njimi, nam sedaj bolj zaupajo in uporabljajo sistem poslovnega obveščanja, saj je navsezadnje namenjen prav njim.

Predstavitve so se udeležili vodstvo podjetja, vodje razvoja in podpore, ter zaposleni v računovodstvu podjetja; skupaj 12 ljudi. Ob prikazu novih poročil, njihovi uporabi in rezultatov so bili vidno zadovoljni. Vodstvo je bilo zadovoljno, da je projekt uspel. Zaposleni v računovodstvu sedaj bolj zaupajo sistemu BI, vendar še vedno sami preverjajo nekatere rezultate. Pričakujemo, da se bo potreba po dvojnem preverjanju, z uporabo zmanjšala.

5 Sklepne ugotovitve

V diplomskem delu smo prikazali uspešno postavitve sistema za poslovno obveščanje z Oraclevim orodjem OBIEE 11g. Dobra zasnova podatkovnega skladišča in procesa ETL zagotavlja ažurnost in dostopnost tekočih informacij. Olajša tudi izgradnjo poročil in ostalih prikazov, ti postanejo bolj dinamični in uporabniško nastavljivi.

Da bi izboljšali sistem na nivoju fizičnega podatkovnega skladišča, bi lahko preoblikovali obstoječe rešitve, ki so doslej slonele na pogledih in materializiranih pogledih operativne podatkovne baze. Na predstavitvenem nivoju bi lahko dodali tudi upravljanje z uporabniki in določili, kdo ima pravice do katerih poslovnih področij in nadzornih plošč. Določili bi lahko razvijalce, ki imajo pravice ustvarjati nova in popravljati obstoječa poročila, ter uporabnike, ki imajo samo pravico do uporabe poročil. Nekatera poročila in nadzorne plošče iz starega sistema bi bilo dobro prenoviti. Zaradi tehničnih razlik med verzijama OBIEE 10g in 11g stara poročila ne odražajo dobro nove grafične podobe, čeprav so bila del postopka nadgradnje. Tako dopolnjen sistem bi bil tudi bolj tržno zanimiv.

Pri izbiri programske opreme Oracle je potrebna previdnost. Velikokrat različne verzije istega orodja niso medsebojno skladne. V našem primeru se je pokazalo, da ne smeta biti nameščeni na istem sistemu verziji OBIEE 10g in 11g. Za nadgradnjo skladišča iz stare verzije je potreben poseben načrt, pri tem pa še vedno ostaja tveganje, da bo rezultat v celoti ali delno neuporaben.

Postavitve sistema BI je razmeroma zahtevna. Podrobno smo morali preveriti vse specifikacije in zahteve OBIEE 11g. Namestitveni paket, ki smo ga dobili na internetni strani [7], je razdeljen v več datotek za prenos v skupni velikosti več kot 5.4 GB in namenjen točno določeni vrsti strežniškega sistema. Pri prenosu namestitvenih datotek smo večkrat naleteli na težave. Nekatere datoteke, ki smo jih prenesli so bile pokvarjene in smo jih morali prenesti večkrat. Ločeno od namestitvenega postopka smo morali uporabiti orodje RCU. Ločeno smo tudi namestili razvijalska orodja.

Pri teh nevšečnostih pa Oracle poskrbi za vrhunsko dokumentacijo in izobraževanja. Za vsak element predpriprave okolja, namestitve in uporabe programske opreme najdemo podrobna navodila [5, 8]. Oracle z izobraževanji in certifikati nudi strokovno usposobljenost uporabnikom svojih rešitev.

6 Literatura

- [1] Ralph Kimball, *The data warehouse toolkit : the complete guide to dimensional modeling* / Ralph Kimball, Margy Ross, New York: Wiley Computer Publishing. 2002.
- [2] (2012) Gartner - Magic Quadrant for Business Intelligence Platforms. Dostopno na: <http://www.gartner.com/DisplayDocument?id=1915014>
- [3] Boštjan Kožuh, "*BI ali ne BI*", Monitor PRO, jesen 2010, str. 30–32, December 2010.
- [4] (2012) Microsoft Business Intelligence. Dostopno na: <http://www.microsoft.com/en-us/bi/default.aspx>
- [5] (2012) Oracle® Fusion Middleware Integrator's Guide for Oracle Business Intelligence Enterprise Edition 11g Release 1 (11.1.1). Dostopno na: http://docs.oracle.com/cd/E21764_01/bi.1111/e16364/toc.htm
- [6] (2012) Oracle Database 11g Release 2. Dostopno na: <http://www.oracle.com/technetwork/database/enterprise-edition/overview/index.html>
- [7] (2011) Oracle Business Intelligence 11.1.1.x Downloads – OBIEE 11g za Microsoft Windows 64-bit in RCU. Dostopno na: <http://www.oracle.com/technetwork/middleware/bi-enterprise-edition/downloads/biee-111150-393613.html>
- [8] (2011) Oracle Fusion Middleware Upgrade Guide for Oracle Business Intelligence 11g Release 1 (11.1.1). Dostopno na: http://download.oracle.com/docs/cd/E21764_01/bi.1111/e16452/toc.htm
- [9] (2011) Oracle Business Intelligence 11.1.1.x Downloads – Oracle Business Intelligence Developer Client Tools Installer (11.1.1.5.0) for Microsoft Windows x86 (32-bit). Dostopno na: <http://www.oracle.com/technetwork/middleware/bi-enterprise-edition/downloads/bus-intelligence-11g-165436.html>
- [10] Miran Mihelčič, *Ekonomika poslovanja za inženirje*, Ljubljana: Fakulteta za računalništvo in informatiko, 2003, str. 227–229.
- [11] (2011) Uradni list Republike Slovenije – Kontni okvir za gospodarske združbe. Dostopno na: http://www.uradni-list.si/files/RS_-2001-107-05172-OB~P001-0000.PDF