

OBIEE & Essbase – iskustva na temelju projekta izrade izvještajnog sustava



Davorin Cetto

Neos d.o.o., Gundulićeva 63 10000 Zagreb
++385 91 483 8610
davorin.cetto@neos.hr
www.neos.hr

Ivo Vrban

Neos d.o.o., Gundulićeva 63 10000 Zagreb
++385 91 483 8610
ivo.vrban@neos.hr
www.neos.hr

SAŽETAK

U sklopu opsežnog projekta izgradnje BI sustava izrađen je izvještajni sustav pomoću Oracle Business Intelligence izvještajnog alata. U predavanju ćemo pokušati objasniti probleme i načine na koje smo ih riješili prilikom izrade izvještaja. Fokus je na konkretne implementacijske metode kod BI sustava koji koristi Essbase kocke kao izvor podataka. Opisati ćemo značajke repozitorija, Answers dijela i integraciju BI Publisher alata.

Neke od tema koje ćemo opisivati su slijedeće: rad s više kocki unutar jednog izvještaja, rad s vremenskom dimenzijom i vremenskim funkcijama, korištenje custom napisanih funkcija (evaluate), odnos kolona i filtera u Answersima, razlozi korištenja BI Publisher alata.

UVOD

Ako ste upoznati s korištenjem OBIEE alata nad izvorima podataka kao što su baza podataka ili flat datoteke tada bi vam se izvještavanje na Essbase-u kao izvoru podataka moglo učiniti kao nešto vrlo slično. Doista, s obzirom da OBIEE konstruira multi-dimenzionalni model u logičkom sloju od fizičkih izvora podataka, može se učiniti da je izvještavanje nad izvorom podataka koji je već u multi-dimenzionalnom obliku, ako ništa drugo, jednostavnije.

Međutim, u praksi, izvještavanje nad Essbase-om kao izvorom podataka je dosta drugačije od izvještavanja nad relacijskim bazama i flat datotekama. Ključna činjenica je ta da je multi-dimenzionalni Essbase model puno sofisticiraniji od onoga kojega kreira OBIEE alat. Te razlike između modela imaju dvije posljedice:

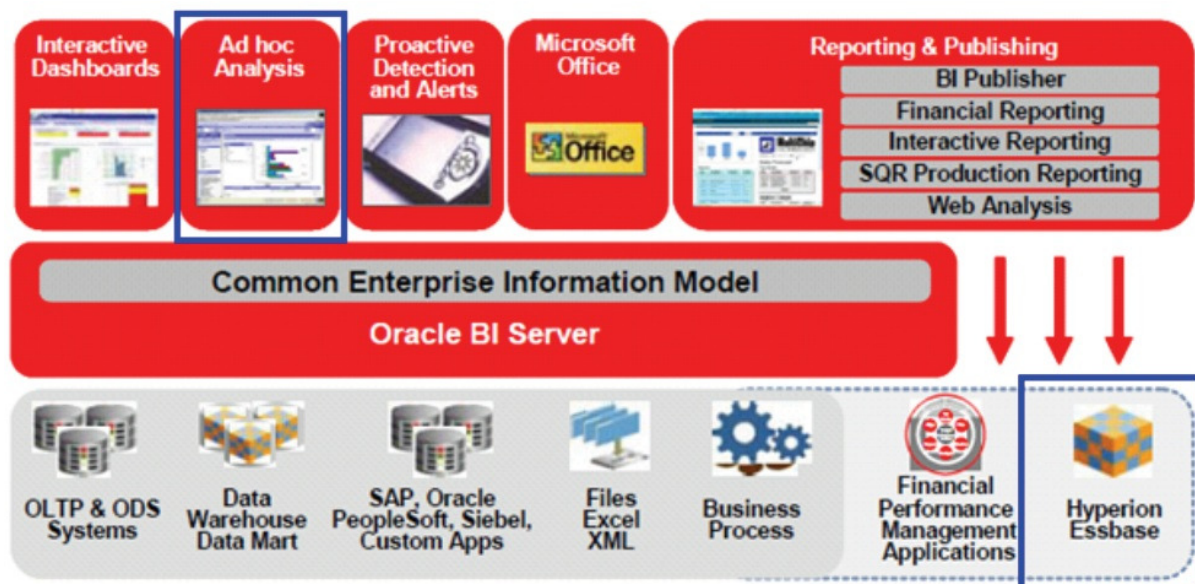
- Očekivanja krajnjih korisnika koji su navikli na sofisticirani Essbase model i
- Nedostatak ugrađenih OBIEE funkcija za potrebe kompleksnog izvještavanja.

Krajnji korisnici upoznati sa Essbase reportima, uglavnom preko Excel ugrađenog dodatka ili Hyperion reporting alata, očekivati će reporte koji će jednako dobro se moći nositi sa kompleksnim multi-dimenzionalnim Essbase modelom. Činjenica da se izvještaji sada prikazuju sa OBIEE izvještajnim alatom neće umanjiti njihova očekivanja.

Kao posljedica jednostavnog multi-dimenzionalnog modela kojeg koristi OBIEE, nemoguće je definirati velik broj izvještaja koristeći ugrađene funkcije OBIEE-a te ih prosljediti BI Serveru koji bi

automatski poslao ekvivalentni MDX izraz prema Essbase-u. Umjesto toga OBIEE developeri su prisiljeni raditi direktno sa MDX jezikom i Essbase bazom preko „evaluate“ funkcije. MDX jezik kao takav nije zapravo previše kompliciran ali ima vrlo kompliciranu sintaksu koja postaje još kompliciranija kada se koristi u kombinaciji sa OBIEE evaluate funkcijom. U tom trenutku, kada koristimo evaluate funkciju, OBIEE postaje samo međuspremnik u kojeg se praktički samo dohvaćaju rezultati native Essbase kalkulacije, samo za potrebe prikaza. Implikacija tog nerazmjera između dva proizvoda je ta da ćete kao programer OBIEE-a morati dobro razumjeti MDX i Essbase bazu podataka kako bi se izradili izvješća koja će zadovoljiti očekivanja krajnjeg korisnika. Izrada izvještaja nad Essbase-om je potpuno drukčije iskustvo nego nad relacijskim bazama. Ako niste upoznati sa Essbase-om ili nekom drugom MDX-based multi-dimenzionalnom bazom tada trebate očekivati da će vam trebati nešto više vremena pri izradi izvještaja.

Na slijedećoj slici (Slika 1) je opisana veza između Oracle proizvoda. U našem slučaju od navedenih koristili smo Essbase kao multi-dimenzionalni izvor podataka na kojeg smo se spajali pomoću BI Servera i na kraju kao front end smo koristili Answers-e za izradu dinamičkih izvještaja i BI Publisher za izradu statičkih izvještaja. U slučaju BI Publishera kao izvor podataka su nam poslužili Answers izvještaji. Konačno su Answers i BI Publisher izvještaji objavljeni na Dashboard-ima kao jedinom alatu kojeg krajnji korisnik koristi za pregled izvještaja.



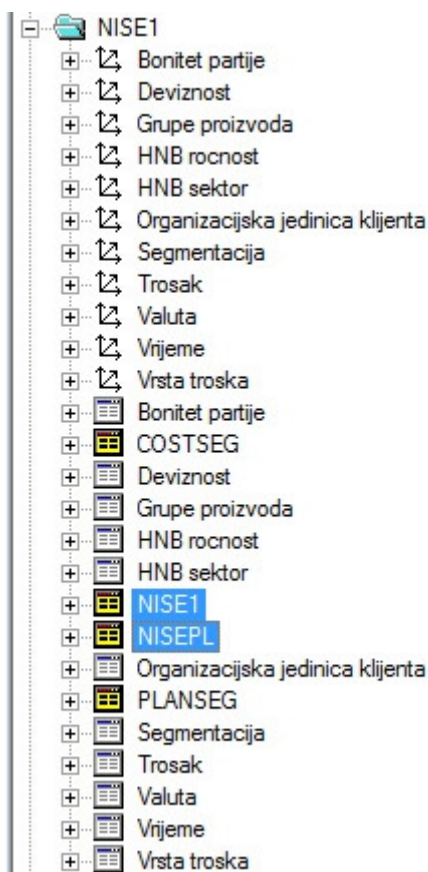
Slika 1 VEZA IZMEĐU ORACLE PRODUKATA

KORIŠTENJE VIŠE KOCKI U MODELU

Kada postoji potreba za korištenjem više kocki u istom business modelu tada se kao najbolja praksa preporučuje korištenje opcija particioniranja unutar samog Essbase servera. Najčešće, kao i u našem slučaju, to nije opcija iz razloga što nemamo pristup kockama, a tim koji se bavi time je odlučio iz performansnih razloga napraviti više kocki. Umjesto te opcije možemo takav slučaj izmodelirati unutar OBIEE alata. Ima nekoliko stvari na koje moramo pritom paziti. Moguća je pojava nekonzistentnih upita nad kockama u slučaju da se fizičke kocke jednostavno dovuku iz fizičkog modela repozitorija u business model sloj kako bi se kreirao jedinstveni business model. Izbjegavanje ovakvog problema se može postići na jedan od slijedećih načina:

- Kreirati poseban business model za svaku pojedinu kocku. To se ostvaruje na način da se svaka kocka posebno dovuče iz fizičkog modela u business model. Prednost ovog načina je to što imamo vrlo čistu situaciju oko upita i nemamo problema sa nekonzistentnim upitima. Mana je naravno ta što ne možemo ostvariti interakciju između kocki.
- Ako postoji potreba da se sve fizičke kocke iz fizičkog modela nalaze u jednom business modelu, tada je potrebno kocke dovući u isti business model. Kod ovog slučaja treba paziti na dimenzije sa istim imenom kod različitih kocki. Te dimenzije će se tretirati kao konformne i samo jedna logička dimenzija će biti kreirana za sve fizičke dimenzije istog naziva. To nadalje znači da će ta logička dimenzija kao izvor podataka imati sve fizičke dimenzije što može rezultirati sa potencijalnim problemima prilikom kreiranja upita. Prilikom korištenja te dimenzije u izvještajima, rezultat će biti unija svih dimenzija koje se koriste kao izvor za tu dimenziju. Prednost je što na ovaj način imamo samo jednu dimenziju za održavanje jer ju dijeli više fakt tablica. Preporuka bi bila da su te dimenzije identične u svim kockama ili kvalitetno mapirane kako ne bi došlo do pogrešnih upita kasnije.

Još jedna prednost posljednjeg opisanog načina rada je ta što možemo kombinirati mjere iz više iz više kocaka u kreiranju nove logičke mjere. Na primjer imamo kocku za plan i kocku za realizaciju. Želimo kreirati novu logičku mjeru koja će nam prikazivati razliku između realizacije i plana (slika 2).



Slika 2 DVIJE KOCKE U JEDNOM BUSINESS MODELU

Fizičku mjeru ne možemo direktno dovući iz fakta za plan u fakt za realizaciju. Prvo moramo kreirati novu logičku mjeru u faktu za plan koja će sadržavati fizičku mjeru, u našem slučaju neto kamatni prihod (slika 3).

General | Data Type | Aggregation | Levels

Name: 01P-07 Neto kamatni prihod - MPLAN TODAY

Belongs to Table: "NISE1"."NISEPL"

Sort order column: None [Set... Clear]

Use existing logical columns as the source

IFNULL(ROUND(NISE1.NISEPL."Neto kamatni prihod", 4), 0)

Description:

OK Cancel Help

Slika 3 KREIRANJE MJERE U FAKTU PLANSKE KOCKE

Nakon toga u faktu za realizaciju kreiramo logičku kolonu koja sadrži logičku kolonu iz fakta za plan (slika 4):

General | Data Type | Aggregation | Levels

Name: 01-07 Neto kamatni prihod - MPLAN TODAY

Belongs to Table "NISE1"."NISE1"

Sort order column

None Set... Clear

Use existing logical columns as the source

IFNULL(NISE1.NISEPL."01P-07 Neto kamatni prihod - MPLAN TODAY", 0)

Description:

OK Cancel Help

Slika 4 KREIRANJE LOGIČKE MJERE U FAKTU REALIZACIJSKE KOCKE

Na kraju u faktu za realizaciju kreiramo logičku kolonu koja sadrži odnos između jedne fizičke ili izvedene mjere iz fakta za realizaciju i jedne izvedene mjere iz fakta za plan (slika 5):

General | Data Type | Aggregation | Levels

Name: 01-09 Neto kamatni prihod - MPLAN DELTA

Belongs to Table: "NISE1"."NISE1"

Sort order column: None [Set... Clear]

Use existing logical columns as the source

IFNULL(NISE1.NISE1."01-04 Neto kamatni prihod - MOSTV TODAY" - NISE1.NISE1."01-07 Neto kamatni prihod - MPLAN TODAY", 0)

Description:

OK Cancel Help

Slika 5 ODNOS IZMEĐU PLANSKE I REALIZACIJSKE KOCKE

NEBALANSIRANE HIJERARHIJE

Jedna od stvari koje se zasada ne radi automatski u OBIEE-u je prepoznavanje vrste hijerarhije iz kocke. OBIEE će prihvatiti hijerarhiju iz Essbase-a ali vi ćete ručno morati dodatno opisati kakva je to hijerarhije inače će OBIEE automatski postaviti da se radi o balanced hijerarhiji. Postoje slijedeće vrste hijerarhija koje dolaze iz Essbase-a:

- FULLY BALANCED – standardna hijerarhija, sa jednakim brojem nivoa, bez preskakanja nivoa
- UNBALANCED – hijerarhija koja ima nejednaki broj nivoa, npr. Jedna grana hijerarhije završava na nivou 2, a neka druga grana ide do nivoa 3 (slika 6).

Level 1	Level 2	Level 3
USA	CA	San Francisco
USA	CA	Los Angeles
USA	Washington DC	<NULL>
Vatican City	Vatican City	<NULL>

Slika 6 UNBALANCED HIJERARHIJA

- RAGGED BALLANCED – hijerarhija koja ima rupe na mjestu nekih čvorova. Npr. U istoj grani hijerarhije postoje podaci za 1 i 3 level, ali je 2 level prazan (slika 7).

Country	State	City
USA	CA	San Francisco
USA	CA	Los Angeles
USA	<NULL>	Washington DC
Vatican City	<NULL>	Vatican City

Slika 7 RAGGED BALANCED HIJERARHIJA

Konkretni primjer u OBIEE alatu nebalansirane hijerarhije koja nije označena kao nebalansirana (slika 8):

Gen1, Segmentacija	Gen2, Segmentacija	Gen3, Segmentacija	Gen4, Segmentacija	Gen5, Segmentacija	EVA
Segmentacija	1.RETAIL	INDIVIDUAL	AFFLUENT	SG AFFLUENT	160,200,507
				SG AFFLUENT +	77,187,894
				SG AFFLUENT PO RM-U	82,945,364
				SG PRIVATE BEZ USLUŽNOG MODELA PB	-3,183,730
			MASS	SG A KLJENT BEZ USLUŽNOG MODELA	1,154,897
				SG IZNADPROSJEŠNI KLJENT	-11,999,516
		SMALL BUSINESS	MIKRO	SG STANDARDNI KLJENT	-1,431,520,009
				SG Mikro klijenti	65,503
				SG Mikro klijenti bez kredita u banci i bez potencijala	-103,765,356
			SB	SG Mikro klijenti s kreditnim i garantnim poslovanjem u banci	-30,412,888
				SG Mikro klijenti s potencijalom	8,994,106
				SG Nepoznati mikro klijenti po podacima o poslovanju i potencijalu	7,025,083
				SG Mali klijenti	-64,880,369
				SG Mali obrtnik	470,268

Slika 8 FULLY BALANCED HIJERARHIJA

Isti izvještaj ali je sada ispravno označena hijerarhija segmentacija kao unbalanced (slika 9):

Gen1,Segmentacija	Gen2,Segmentacija	Gen3,Segmentacija	Gen4,Segmentacija	Gen5,Segmentacija	EVA	
Segmentacija	1.RETAIL	INDIVIDUAL	AFFLUENT	SG AFFLUENT	160,200,507	
				SG AFFLUENT +	77,187,894	
				SG AFFLUENT PO RM-U	82,945,364	
				SG PRIVATE BEZ USLUŽNOG MODELA PB	-3,183,730	
				SG A KLJENT BEZ USLUŽNOG MODELA	1,154,897	
		OTHER RETAIL	SG Other Retail	SG IZNADPROSJEŠNI KLJENT	-11,999,516	
				SG STANDARDNI KLJENT	-1,431,520,009	
				SG Mikro klijenti	158,624	
				SG Mikro klijenti bez kredita u banci i bez potencijala	65,503	
				SG Mikro klijenti s kreditnim i garantnim poslovanjem u banci	-103,765,356	
	2.PRIVATE	SMALL BUSINESS	MIKRO	SG Mikro klijenti s kreditnim i garantnim poslovanjem u banci	-30,412,888	
				SG Mikro klijenti s potencijalom	8,994,106	
				SG Nepoznati mikro klijenti po podacima o poslovanju i potencijalu	7,025,083	
				SG Mali klijenti	-64,880,369	
				SG Mali obrtnik	470,268	
		3.CORPORATE	SG PRIVATE	LARGE	SG Financijske institucije - društva za upravljanje fondovima	-5,948,380
					SG Financijske institucije - osiguravajuća društva	8,374,454
					SG Ostale nebankarske FI u Registru nerezidenata	35,115,689
					SG VELIKI KLJENTI	538,903
					SG Veliki obrtnik	-744,139,993
	4.MIB	MID	OTHER CORPORATE	SG Mid lower	644,439	
				SG Mid nonfinancijal	-284,601,284	
				SG Mid upper	9,997,212	
				SG Obrtnici lower	-497,635,255	
				SG Obrtnici upper	1,605,098	
		5.OTHER	MARKETS	ALM	SG Srednji klijenti	129,219
					SG Other Corporate	43,438,426
					SG Trading	7,313,812
					SG ALM	60,678,469
					SG NEPOZNATO	-448,018,526
	OTHER OTHER	SG_XL	649,280,995			
				-37		

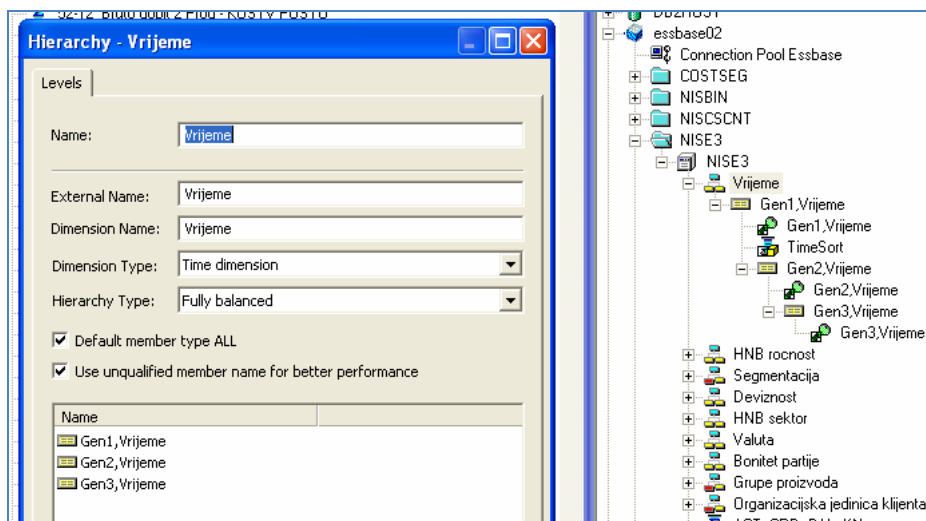
Slika 9 UNBALANCED HIJERARHIJA

Na ovom primjeru vidimo da se u prvom izvještaju prikazuje manje rezultata što je zapravo pogrešno.

KORIŠTENJE AGO I TODATE FUNKCIJA I VREMENSKE DIMENZIJE

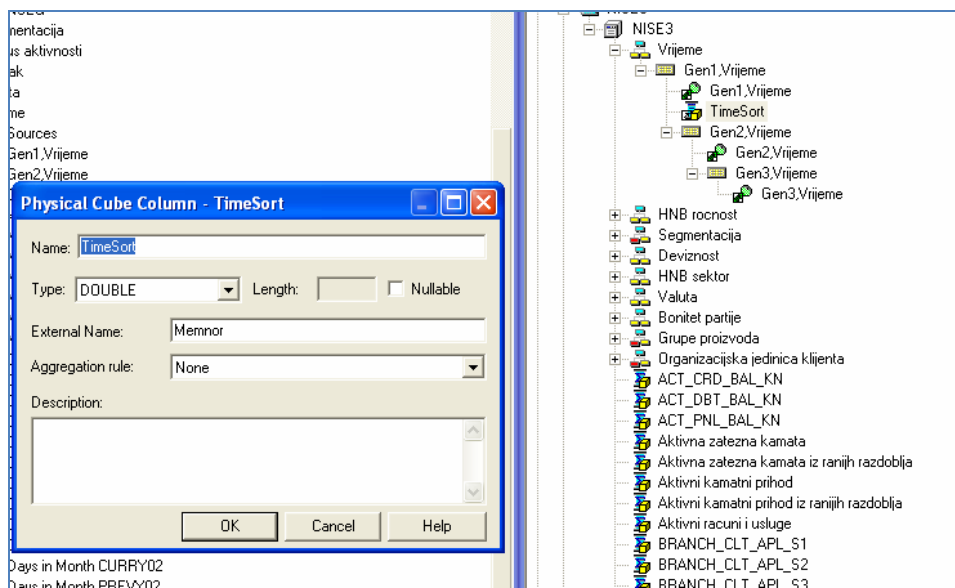
Essbase kocke imaju podržanu vremensku dimenziju kao standardnu mogućnost koja se često koristi u modelima podataka kojima se opisuje poslovni zahtjev, odnosno u pohrani podataka koja uključuje vrijeme kao jedan od pokazatelja.

Prilikom uvlačenja određene Essbase kocke u OBIEE repozitorij, dimenzija koja se u Essbase-u definirala kao vremenska se i u repozitoriju automatski prepoznaje kao vremenska (time dimension) zajedno sa pripadajućom hijerarhijom i levelima (npr. *godina-kvartal-mjesec*).



Slika 10 – HIJERARHIJE VREMENSKE DIMENZIJE, PRIKAZ U FIZIČKOM SLOJU

Kod vremenske dimenzije moguće je, ako je potrebno, koristiti fizičku kolonu za sortiranje koja se definira tzv. memnor (*member number*) funkcionalnošću, a kojom se postiže sortiranje prema nekom ključu neovisno o sortiranju prema npr. nazivu člana u dimenziji.



Slika 11 – MEMNOR KOLONA U VREMENSKOJ DIMENZIJI, CUSTOM SORT

U BMM dijelu repozitorija definira se koja je dimenzija vremenska (označavanjem „time dimension“ property-a) te ključ vremenske dimenzije, odn. svaki od levela hijerarhije označava se kao kronološki ključ što čini identifikator vremenskog zapisa. Definiranjem vremenske kolone na ovaj način omogućen je i rad s vremenskim funkcijama.

Kod korištenja vremenskih funkcija na raspolaganju nam stoje dvije konkretne funkcije koje nudi OBI, a to su AGO i TODATE. Ove dvije vremenske funkcije od velikog su značaja za izvještavanje, budući da u velikoj većini izvještaja vrijeme predstavlja vrlo važnu komponentu (odn. u

terminima BI-a dimenziju). Također, činjenica da se kod ovih funkcija radi o funkcionalnosti dohvaćanja podataka u prošlosti, dodatno pokazuje važnost ovih funkcija za izvještavanje, pogotovo kod financijskih izvještaja kod kojih je ovo standardni zahtjev i potreba kod korisnika.

Konkretno, u našem projektu izvještajnog sustava imali smo potrebu za velikim brojem izvještaja koji su uključivali: prikaz niza podataka (mjera filtriranih preko dimenzijskih podataka) za trenutno izabrani mjesec, za mjesec prije, 2 mjeseca prije, godinu dana prije, te za kumulativne vrijednosti – i tekuće godine i prethodne godine.

Ovi vremenski pokazatelji bili su ključan dio velikog broja izvještaja. Za izradu tih izvještaja intenzivno su se koristile vremenske funkcije AGO i TODATE te je naš zaključak kako one ispravno funkcioniraju nad Essbase kockama kao izvorom podataka.

AGO vremenska funkcija služi dohvaćanju podatka za neku mjeru u određenom vremenskom periodu u prošlosti (vrijednost na prethodni mjesec, prethodne godine, 3 mjeseca prije i sl.), u odnosu na neku referentnu točku (odabrani mjesec npr.).

TODATE vremenska funkcija služi dohvaćanju vrijednosti neke mjere u obliku kumulativa, odn. sume vrijednosti za određeni broj vremenskih perioda (npr. suma 3 mjeseca). Također se odnosi prema nekoj referentnoj točki (odabrani mjesec) tako da vraća zbroj **prvih** n mjeseci pri čemu je n broj mjeseca koji je odabran. Pod referentnom točkom podrazumijeva se početak nekog vremenskog perioda, ovisno o levelu kojeg definiramo pozivom TODATE funkcije. Npr. ako odabiremo mjesec, referentna točka može biti početak godine u kojoj je taj mjesec, odn. prvi siječanj te godine budući da nam je za izvještaj bitan kumulativ nekog broja mjeseci.

Općenito, AGO funkcija koristi se na način:

```
AGO(business_model.ime_tablice.ime_kolone, business_model.vremenska_dimenzija.level, broj_vremenskih_perioda).
```

Kao općeniti primjer može se uzeti mjera „Neto kamatni prihod“ – ako želimo dobiti vrijednost te mjere za prošli mjesec, koristio bi se izraz (business model i ime tablice su „NISE1“):

```
AGO(NISE1.NISE1."Neto kamatni prihod", NISE1.Vrijeme."Gen3,Vrijeme", 1).
```

Za prošlu godinu koristio bi se sličan izraz s razlikom levela vremenske dimenzije – umjesto „Gen3,Vrijeme“ koristio bi se level „Gen2,Vrijeme“.

Za vrijednost mjere od prije 3 mjeseca početni izraz bio bi drugačiji samo u zadnjem parametru koji bi umjesto 1 bio 3.

Općenito, TODATE funkcija koristi se na način:

```
TODATE(business_model.ime_tablice.ime_kolone, Business_model.vremenska_dimenzija.level).
```

Za mjeru „Neto kamatni prihod“, ako želimo dobiti kumulativnu vrijednost, koristio bi se izraz:

```
TODATE(NISE1.NISE1."Neto kamatni prihod", NISE1.Vrijeme."Gen2,Vrijeme").
```

Ranije spomenuta specifičnost TODATE funkcije je ta da ona funkcionira na način da sumira vrijednosti vremenskih perioda od početnog vremenskog perioda odabranog levela pa do

vremenskog perioda koji se gleda u izvještaju. U gornjem izrazu, u slučaju da u izvještaju gledamo 3. mjesec neke godine, rezultat bi bio suma 1., 2. i 3. mjeseca te godine.

Još jedna mogućnost korištenja vremenskih funkcija je u tzv. *nested* obliku – kada imamo potrebu dohvatiti kumulativ vrijednosti, ali u nekom vremenskom periodu različitom od onog koji je u izvještaju odabran. Primjer toga je kumulativ za 3 mjeseca prethodne godine.

Korištenje vremenskih funkcija na taj način, za dobivanje kumulativa „Neto kamatnog prihoda“ prethodne godine, izgleda ovako:

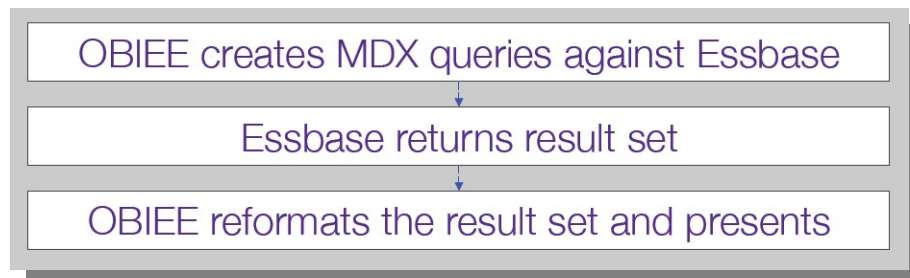
```
AGO(TODATE(NISE1.NISE1."Neto kamatni prihod", NISE1.Vrijeme."Gen2,Vrijeme"),  
NISE1.Vrijeme."Gen2,Vrijeme", 1).
```

Ograničenje ovog *nested* oblika vremenskih funkcija je nemogućnost korištenja različitih levela vremenske dimenzije u AGO i TODATE komponentama.

Ovime se pokazuje kako se vremenskim funkcijama uspješno riješeni zahtjevi za prikazom podataka u vremenskom 'kontekstu' opisani na početku ovog poglavlja.

KORIŠTENJE EVALUATE FUNKCIJE

Prvo treba pojasniti kako se komunicira sa essbase-om iz OBIEE alata. Kada korisnik izvrši upit, Prezentacijski server šalje BI serveru select upit sličan SQL upitu. BI Server taj upit pretvara prekalkulira na način da uzme u obzir sve formule koje se koriste za logičke i izvedene kolone u business modelu i na temelju toga generira MDX upit kojeg prosljeđuje Essbase-u. Essbase zaprima upit i vraća set rezultata koje dalje BI Server primi, obradi ih i proslijedi na Prezentacijski server (slika 12).



Slika 12 KOMUNIKACIJA OBIEE-A I ESSBASE-A

EVALUATE funkcija omogućava korištenje mogućnosti BI servera kojom se u repozitoriju mogu koristiti i funkcije baze podataka koja nam služi kao izvor. Ovime se postiže proširena funkcionalnost alata jer nismo ograničeni samo na funkcije koje su dostupne u BI serveru (npr. AGO). U osnovi, EVALUATE se funkcijom na izvor podataka izravno mogu poslati upiti specifični za taj izvor (ovu ili onu bazu, OLAP server itd.), odn. mogu se iskoristiti funkcionalnosti baze podataka koje alat nema standardno implementirane – npr. kod rada s relacijskom bazom to mogu biti neke analitičke funkcije, kod Essbase-a npr. *lead i lag* za rad s vremenskim 'pomacima' i unaprijed ili MDX-specifične funkcije itd. Drugim riječima, ne prepuštamo generiranje upita (bilo SQL-a, bilo MDX-a) u potpunosti BI serveru već 'šaljemo' svoj upit (ili dio upita) na izvor podataka.

Drugi je razlog korištenja ove funkcije taj što nam omogućuje premošćivanje određenih problema ili specifičnosti alata koje nam u nekom trenutku ne odgovaraju, što je i najčešći slučaj kod projekta Niše. To mogu npr. biti bug-ovi, spori upiti, sintaksne greške i slično.

U našem slučaju, EVALUATE funkcija koristila se kod određenih mjera zbog specifičnosti koje alat doživljava u radu s Essbase kockama – neispravan prikaz mjera za neke kombinacije filtera, odn. kolona u Answers izvještaju. Budući da nismo uspjeli sa preciznošću utvrditi uzroke niti pravilnosti u neispravnom prikazu rezultata, možemo sugerirati jedino korištenje EVALUATE funkcije kao moguće alternative, odn. rješenja koje će ispravno dohvaćati podatke iz Essabase kocke.

EVALUATE funkcija može se implementirati na dva načina i to kao standardni izraz u nekoj od kolona BMM dijela, te kao mapirana kolona. Kao mapirana kolona koristi se u slučaju da se mjera nalazi u dvije različite kocke (odn. fact tablice u BMM dijelu repozitorija) pa se time izbjegava istovremeno povlačenje podataka iz obje kocke u mjeru koja treba imati podatke samo iz jedne. Ovo se događa jer logička tablica koja sadrži fact tablice ima definirane sve izvorne tablice navedene kao *sources* – u našem su slučaju su to bile više fact tablica (odn. kocke), a koje su imale zajedničke dimenzije. Korištenjem EVALUATE funkcije u standardnom obliku kao izraz u nekoj koloni povlačili su se podaci iz svih kocaka koje su bile definirane kao izvor (*sources*), dok smo mi trebali potrebu povlačenja podataka samo iz jedne određene kocke.

Konkretan primjer korištenja EVALUATE funkcije za mjeru neto kamatni prihod – razlog je što su se rezultati za određene izvještaje pokazali neprecizni kada se nije koristila EVALUATE funkcije.

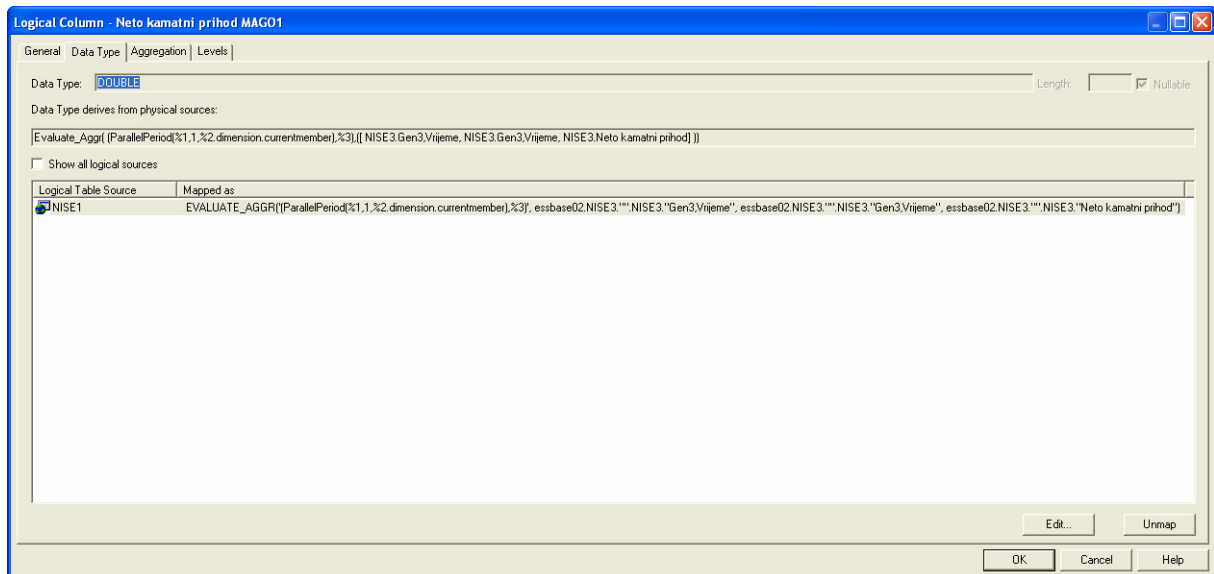
Izraz za tako korištenu EVALUATE funkciju koja vraća kumulativ prošle godine je oblika (pomak na prošlu godinu postiže se MDX funkcijom *Lag*, a kumulativ MDX funkcijom *PeriodsToDate*):

```
CAST ( EVALUATE('AGGREGATE(PeriodsToDate(%1,%2.dimension.currentmember.lag(12)),  
Accounts.[Neto kamatni prihod]', NISE1.Vrijeme."Gen2,Vrijeme", NISE1.Vrijeme."Gen3,Vrijeme")  
AS DOUBLE PRECISION ).
```

Formalnim parametrima %1 i %2 te stvarnim parametrima *Neto kamatni prihod, Vrijeme."Gen2,Vrijeme"* i *Vrijeme."Gen3,Vrijeme"* postiže se sintaksna ispravnost EVALUATE funkcije koja je potrebna kako bi se izraz kojeg smo sami definirali uspješno ukomponirao u ostatak MDX upita te ispravno izvršio nad Essbase kockom.

Donja slika prikazuje korištenje EVALUATE funkcije kao mapirane kolone pri čemu je točno definirana kocka iz koje se dohvaćaju podaci – dio upita s „essbase02.NISE3“ točno definira konkretnu kocku s koje trebamo povlačiti podatke.

Primjer na slici prikazuje dohvaćanje vrijednosti na prethodni mjesec (slika 13).



Slika 13 – FUNKCIJA EVALUATE KAO MAPIRANA KOLONA

Primjer korištenja EVALUATE funkcije, odn. kako se ono očituje u samom MDX upitu slijedi u nastavku:

With

set [Organizacijska jedinica klijenta2] as '{[Organizacijska jedinica klijenta].[DOM 019999 ZAGREBAČKA BANKA D.D.]}'

set [Vrijeme3] as '{[Vrijeme].[MJESEC_3_2010]}'

member [Accounts].[MS1] as
'(ParallelPeriod([Vrijeme].Generations(3),1,[Vrijeme].Generations(3).dimension.currentmember),[Accounts].[Neto kamatni prihod])', SOLVE_ORDER = 101

member [Accounts].[MS2] as
'(ParallelPeriod([Vrijeme].Generations(3),12,[Vrijeme].Generations(3).dimension.currentmember),[Accounts].[Neto kamatni prihod])', SOLVE_ORDER = 101

member [Accounts].[MS3] as
'AGGREGATE(PeriodsToDate([Vrijeme].Generations(2),[Vrijeme].Generations(3).dimension.currentmember), [Accounts].[Neto kamatni prihod])', SOLVE_ORDER = 101

select

{ [Accounts].[Neto kamatni prihod],

[Accounts].[MS1],

[Accounts].[MS2],

[Accounts].[MS3],

[Accounts].[EVA]

} on columns,

```
NON EMPTY {crossjoin ({[Organizacijska jedinica klijenta2],[Vrijeme3]})} properties  
ANCESTOR_NAMES, GEN_NUMBER on rows
```

```
from [NISE3.NISE3].
```

Crvenom bojom naznačeni su dijelovi MDX upita koji se odnose se na EVALUATE funkcije (u primjeru 3 mjere, „Neto kamatni prihod“ u prethodnim razdobljima), a zelenom standardni poziv mjera „EVA“ i „Neto kamatni prihod“ kroz alatom generirani dio upita.

OBIEE ANSWERS SPECIFIČNOSTI (KOLONE, FILTERI, BRZINA)

U OBIEE Answers komponenti alata smo se, prilikom izrade izvještajnog sustava Niše, susreli i sa nekim dodatnim specifičnostima korištenja Essbase kocki kao izvora podataka za izvještaje:

- Kada se koriste određene kolone (odn. leveli dimenzija) u Answers izvještaju za filtriranje podataka, izvještaj će se daleko brže izvršavati ako se ta kolona koristi i u samom izvještaju (odn. ako se ispisuje njezin sadržaj); ovo ponekad nije potrebno prema specifikaciji izvještaja, a može i proširiti veličinu izvještaja do neželjenih razmjera pa je na developeru odluka je li dobro koristiti kolone po kojima se filtrira i u samom izvještaju
- Filtriranje po nekoj dimenziji uvijek treba izvesti na način da se u filteru za ograničavanje koristi *member* te dimenzije, a ne neki izvedeni oblik istoga; primjer: ako su mjeseci u vremenskoj dimenziji u Essbase kocki pohranjeni u obliku „MJESEC_1_2010“, a u izvještaju se prikazuje izvedena kolona koja sadrži mjesece ispisane u obliku „1. mjesec 2010.“, onda se sugerira filter za mjesec napisati u obliku „Vrijeme.Gen3,Vrijeme“ = MJESEC_1_2010, a ne naziv_izvedene_kolone = 1. mjesec 2010.; razlog za ovo je vrlo velika razlika u brzini izvođenja upita, odn. izvještaja; razlog za razliku u brzini je taj što u slučaju filtriranja po izvedenoj koloni, BI server nije u mogućnosti u MDX upit ugraditi odgovarajući filter pod WHERE klauzulu te se sa Essbase servera moraju dohvatiti svi podaci koji postoje za tu dimenziju (odn. za level s kojim radimo) – filtriranje se zatim izvršava na strani BI servera što sve skupa znači rad s većom količinom podataka u prijenosu i obradi na klijentskoj strani nego što je to potrebno
- Prilikom rada s *unbalanced* hijerarhijama koje nisu u fizičkom dijelu repozitorija označene kao takve, događa se da se dio podataka ispusti, odn. ne prikazuje; podaci koji se ne prikazuju su oni za koje na nekom od levela hijerarhije nedostaje member, najčešće list (znači na krajnjem levelu hijerarhije)
- Ako radimo na izvještaju koji se temelji na mjerama iz više fact tablica koje se nalaze u različitim kockama, a koje imaju zajedničke dimenzijske tablice i neka od dimenzijskih tablica koja nije zajednička (nalazi samo u jednoj od kocaka), onda je potrebno pripaziti na filtriranje podataka; u slučaju da se u izvještaj stave mjere iz različitih kocki, no želi se filtrirati po dimenziji koja nije zajednička, javlja se greška i takav izvještaj nije moguće napraviti; greška se javlja zbog filtriranja po nečemu (ne-zajedničkoj dimenziji) što u jednoj od kocaka ne postoji.

ZAKLJUČAK

OBIEE se dokazao kao vrlo moćan izvještajni alat i dugo godina prije Oracle-ove akvizicije je bio među najboljima na tržištu. Sa druge strane imamo Essbase, OLAP engine koji je također već duže vrijeme broj jedan kada pričamo o multi-dimenzionalnim kockama. Oracle je napravio iskorak u smjeru međusobne integracije ova dva alata. Trenutna varijanta ima podosta problema. Neke stvari nisu

uopće moguće, a neke stvari su navedene kao moguće no u praksi su također neizvedive i potrebno je podosta mašte kako bi se riješili neki naoko jednostavni problemi. Ali također moramo imati u vidu da je OBIEE alat koji izvrsno funkcionira već godinama sa relacijskim bazama, flat datotekama i Excel datotekama. Tek odnedavno se pojavila mogućnost izrade izvještaja iz Essbase baze. Dakle tehnologija je u povojima.

U budućnosti vjerujemo da će ta integracija biti znatno bolja i prihvaćenija. Izlaskom nove verzije OBIEE 11g vide se brojni pomaci u kvaliteti, integraciji i inovacijama. Verzija 11 nudi nešto što trenutno nitko na tržištu nema, a to bi bila skoro pa potpuna primjena izvještavačkog alata nad kockama, nešto slično kao vrlo moćni Essbase excel add-in.