INTEGRACIJA NESTRUKTURIRANIH PODATAKA U DATA WAREHOUSE

Matij Srzentić
Neos d.o.o.
Gundulićeva 63,
10000 Zagreb
Tel: +385 1 5555 600, GSM: +385 91 4838 604
E-mail: matij.srzentic@neos.hr
www.neos.hr

Mark Arbanas
Neos d.o.o.
Gundulićeva 63,
10000 Zagreb
Tel: +385 1 5555 600, GSM: +385 91 4838 606
E-mail: mark.arbanas@neos.hr
www.neos.hr

SAŽETAK

U suvremenom poslovnom okruženju proizvodi se velika količina podataka u nestrukturiranom obliku. Upravo zbog nedostatka strukture samo mali postotak tih podataka se koristi kao temelj i podrška poslovanju. Izvlačenje informacija iz nestrukturiranih podataka goruć je problem, a tržištu nedostaju pravi ETL alati specijalizirani za procesiranje sirovog teksta i integraciju procesiranog teksta u data warehouse sustav.

Predavanjem će se, na osnovu teorijske podloge, predstaviti praktično rješenje ovog problema temeljeno na Oracle tehnologijama (Oracle Text, Oracle Warehouse Builder), odnosno kako od naizgled nepovezanog skupa riječi nastaje korisna poslovna informacija koja daje bolji uvid u vlastito poslovanje.

In today's business environment large amount of data is produces in unstructured form. Because of that lack of structure, only a small percentage of this data is used as the basis and support for operations. Extraction from unstructured data is a burning issue, and the market lacks true ETL tools specialized for processing the raw text and integrating processed data into a data warehouse system.

The lecture will present a practical solution to this problem based on the theoretical basis and Oracle technology (Oracle Text, Oracle Warehouse Builder), and will show how the seemingly unrelated sets of words create a useful business information that gives better insight into business.

1 UVOD

U suvremenom poslovnom okruženju proizvodi se i konzumira velika količina podataka u nestrukturiranom obliku. Od internih izvora nestrukturiranih podataka najzastupljenije su razne vrste tekstualnih dokumenata različite namjene i sadržaja, tehnička dokumentacija, korespondencije putem e-maila ili dopisa, itd.

Osim unutrašnjih izvora, pravo bogatstvo informacija krije se i u eksternim izvorima, kao što su blogovi, forumi, web, a u zadnje vrijeme i društvene mreže (Facebook, LinkedIn, Twitter...).

Podaci skriveni u ovim izvorima mogu odgovoriti na pitanja na koja klasični BI sustavi ne mogu odgovoriti:

- Što tržište misli o našim proizvodima? Koji proizvodi dobivaju pohvale? Koji proizvodi trpe kritike?
- Koliko su zadovoljni naši klijenti? Što bi željeli od nas, a mi im to još ne pružamo?
- Koliko odjeka ima naša marketinška kampanja?
- Kakvo je mišljenje o našim proizvodima i uslugama u usporedbi s konkurencijom?

Očigledno je da postoji potreba za primjenom nestrukturiranih podataka u poslovnom odlučivanju i izvještavanju, ali upravo zbog nedostatka strukture samo mali postotak tih podataka se koristi kao temelj i podrška poslovnim procesima.

Izvlačenje informacija iz nestrukturiranih podataka gorući je problem, a tržištu nedostaju pravi ETL alati specijalizirani za procesiranje sirovog teksta i integraciju procesiranog teksta u data warehouse sustav. U očekivanju specijaliziranog sustava, rješenje se može postići primjenom postojećih Oracle alata i tehnologija.

2 TEORETSKA PODLOGA

Koji god alat, rješenje ili skup tehnologija koristili, uvijek se može prepoznati sličan proces dobivanja informacija iz sirovog teksta, čiji su osnovni koraci (kao na slici):

- ekstrakcija teksta iz izvora
- izgradnja baze domenskih pojmova
- izgradnja baze domenskog znanja
- parsiranje i tagiranje riječi
- povezivanje tagiranih pojmova s domenskim znanjem
- punjenje data warehouse-a

2.1 Ekstrakcija podataka s izvora

S obzirom na različite vrste izvora, koristit će se i različite metode ekstrakcije podataka – eksportiranje sadržaja u tekstualne datoteke, korištenje web crawlera za skupljanje postova i komentara s foruma, blogova i drugih web stranica, te aplikacije koje pomoći API-a socijalnih mreža traže tekstove i spremaju ih na dostupna mjesta.

Bez obzira na korištenu metodu, cilj ekstrakcije je dovesti sirovi tekst s izvora u Data Staging Area u bazi podatka, kako bi se nad tim tekstom vršile transformacije.

2.2 Domenski pojmovi

Kad je sirovi tekst pospremljen u Data Staging Area, potrebno je utvrditi sadrži li pojmove koji su predmet interesa za organizaciju, odbaciti nezanimljivi tekst, te na zanimljivom tekstu vršiti daljnje transformacije.

Prije svega je ipak potrebno odrediti koji su pojmovi interesantni, te organizirati te pojmove u smislene kategorije. U jednoj organizaciji bit će više kategorija pojmova:

-kategorija vremena/datuma,
- kategorija imena (osoba, organizacijskih dijelova) unutar organizacije,
- kategorijaproizvoda i usluga,
- kategorijageografskih lokacija, itd.

Postoje gotove specijalizirane hijerarhijski organiziranekategorije, odnosno taksonimije(medicinske, biološke, strojarske, informacijske, marketinške, kemijske, itd.), ali u njihovom

korištenju postoji opasnost od preširoke baze nepotrebnih i općenitih pojmova koji na kraju mogu uzrokovati greške u podacima.

Pošto svaka organizacija najbolje zna na koje pojmove ona želi obratiti pažnju, preporučeno je da se specijalizirane kategorije izrađuju interno. Primjer hijerarhijske kategorije (taksonomije) proizvoda proizvođača vozila bi bila hijerarhija:

1. Vozilo
    1.1. Osobno vozilo
        1.1.1.Automobil
            1.1.1.1. Model Mali
            1.1.1.2. Model Mali 2
            1.1.1.3. Model CC
            1.1.1.4. Model Sport
        1.1.2.Motocikl
            1.1.2.1. Model Ef 50
            1.1.2.2. Model Ef 125
            1.1.2.3. Model Ef S250
    1.2. Teretno vozilo
        1.2.1.Kamion
            1.2.1.1. Model 3T
            1.2.1.2. Model 7T
            1.2.1.3. Model 10T
        1.2.2.Kombi vozilo
            1.2.2.1. Model DG 2
            1.2.2.2. Model DGi

2.3 Parsiranje i tagiranje

Na osnovi definiranih kategorija sirovi tekst se parsira, traže se željeni pojmovi, te ih se označava (eng. tagging) prikladnim oznakama, odnosno tagovima. To mogu biti jednostavni tagovi od jednog ili više specijalnih znakova (npr. #) ili tagovi kakvi se koriste u XML dokumentima, kojima bi se odmah pri tagiranju neki pojam mogao označiti tagom karakterističnim za neku pojmovnu kategoriju.

Tako bi npr. od sirovog teksta:

"U **kolovozu** ove godine počinje prodaja novog modela **iPhonea** u našim trgovinama"

tagiranjem nastao tekst:

"U **<DATUM> kolovozu </DATUM>** ove godine počinje prodaja novog modela **<PROIZVOD> iPhonea </PROIZVOD>** u našim trgovinama"

Od ovako tagiranih pojmova nastat će reference u budućim dimenzijskim i fact tablicama u data warehouse-u.

Već sa završetkom ovog koraka moguće je napraviti narativni data mart, ali bez mogućnosti detaljnog sintetiziranja podataka pomoću definiranih opisa znanja.

2.4 Domensko znanje

Tezaurus je vrsta terminološkog rječnika koji sadrži sustavno uređene nazive određenog znanstvenog područja i osnova je svake znanstvene discipline. Da bi bio uporabljiv, među nazivima moraju postojati određeni paradigmatski odnosi te odnosi ekvivalencije. Sličan koncept je i ontologija, odnosnoobrazac podatka koji predstavlja pojmove unutar neke domene, njihova svojstva i odnose između tih pojmova.

Ontologije su temelj semantičkog weba i baza znanja, što ih čini zanimljivima u promatranom procesu integracije teksta u data warehouse.

Postoji više jezika za ontologije, kao i modela i notacija za predstavljanje podataka, od kojih su vjerojatno najprihvaćeniji OWL (Web Ontology Language) i RDF (Resource Description Framework), nastali na temelju W3C specifikacija unutar koncepta semantičkog weba.

Prema RDF modelu, pojam se može predstaviti tripletima, odnosno trodijelnim kombinacijama subjekta, predikata i objekta, pa bi tako neki od tripleta za opisivanje osobe bili:

<IVO IVIĆ – godina - 20>
<IVO IVIĆ – zanimanje - STUDENT>
<IVO IVIĆ – otac – PERO IVIĆ>
<IVO IVIĆ – majka – KATA IVIĆ>

Kao i kod taksonomija, i ovdje se uočava potencijalni problem preopširno definirane ontologije, jer se samo za osobu mogu definirati tisuće tripleta, od kojih većina nije zanimljiva za buduću analizu, pa je potrebno uložiti veliki trud u definiranje pravog skupa tripleta, odnosno takvog skupa koji u najvećoj mjeri zadovoljava poslovne potrebe, a u isto vrijeme posjeduje najmanji mogući broj suvišnih tripleta.

2.5 Povezivanje pojmova

Slijedeći korak u procesu integracije teksta u data warehouse je povezivanje pojmova tagiranih u prethodnim koracima uz pomoć definiranog tezaurusa, a cilj je dobiti jedinstven i jednoznačan pogled na izvorne podatke eliminiranjem redundancije u značenju pojmova.

Primjer koji će se koristiti u ovom radu je analiza mišljenja internet zajednice o nekom proizvodu, u kojoj se traži samo informacija o postotku ljudi koji na internetu izražavaju pozitivno i negativno mišljenje o proizvodu. U tom slučaju tezaurus bi se sastojao uglavnom od pojmova koji su sinonimi za "dobro mišljenje" i "loše mišljenje", odnosno:

<DOBAR – **sinonim** - ODLIČAN>
<DOBAR – **sinonim** - SUPER>
<DOBAR – **sinonim** - ZADOVOLJAN>
<DOBAR – **sinonim** - KORISTAN>
<LOŠ – **sinonim** - KATASTROFA>
<LOŠ – **sinonim** - NEPRIKLADAN>
<LOŠ – **sinonim** - NEUPOTREBLJIV>
<LOŠ – **sinonim** - PRESKUP>

Na temelju ovih tripleta svi pojmovi koji znače "dobro mišljenje" bi se referencirali na pojam "DOBAR", a svi pojmovi koji znače "loše mišljenje" bi se referencirali na pojam "LOŠ", čime bi se eliminirala redundancija i postigao fokus u analizi tekstualnih podataka.

Naravno, moguće je i finije definirati razine sintetiziranja pojmova uvodeći u ontologije kategorije "NEUTRALAN", "JAKO DOBAR", "JAKO LOŠ" i njihove triplete, ali svako proširenje tezaurusa povlači za sobom kao posljedicu povećanu mogućnost neispravnog povezivanja i tumačenja pojmova.

2.6 Narativni i sintetizirani podatkovni model

Zadnji korak u integraciji nestrukturiranih podataka u Data Warehouse je punjenje obrađenog teksta u DW model posebno prilagođen podacima nastalima od sirovog teksta. Model se sastoji od dvije vrste dimenzija i fact tablica – narativnih i sintetiziranih.

Narativne dimenzije i fact tablice su po svojoj prirodi vrlo općenite, a grade se na temelju podataka iz taksonomija i tagiranog teksta. Svaka tablica u narativnom modelu mora imati kolonu s identifikatorom izvora, odnosno referencom na originalni izvorni tekst iz kojega je dobiven podatak u tablici, a koji se čuva u data stage-u.

Veze između dimenzija su vrlo labave (nisu obvezne, nerijetko niti ne postoje), osim po identifikatoru izvora, jer se u narativni model još ne spremaju sintetizirani i potpuno obrađeni podaci.

Narativni model može odgovoriti na veći raspon pitanja s raznovrsnijim temama i analizama, ali je podložniji greškama i krivim interpretacijama podataka.

U narativni model je moguće ugraditi i indikator "povjerenja", odnosno neki oblik označavanja pouzdanosti i točnosti podataka, kao metodu ublažavanja grešaka nastalih iz preopćenitih podataka.

Sintetizirane dimenzije i fact tablice su produkt potpuno obrađenih podataka, skupljenih s usko specijaliziranim ciljem i namjenom, pa su podaci u sintetiziranom modelu puno pouzdaniji od onih u narativnom modelu.

Tablice u sintetiziranom modelu su identične onima u običnom data warehouse modelu u smislu da nemaju posebnih pravila kao one u narativnom modelu.

Uz očitu prednost u točnosti podataka, nedostatak sintetiziranog modela je fokusiranost na samo jednu ili barem manji broj tema, kao posljedicu sintetiziranja podataka na temelju ontologijskog znanja.Obično se u data warehouse-u može naći jedan narativni data mart i mnoštvo sintetiziranih data martova.

3 PRIMJENA U PRAKSI

BI sustav punjen podacima nastalima obradom sirovog teksta može naći svoju primjenu u svakom poslovnom sustavu koji u nekom dijelu svog poslovnog procesa koristi nestrukturirane podatke. Primjeri ovakvih procesa su obrada help-desk tiketa, analiza odnosa s klijentima, obrada prijavljenih šteta u osiguravajućoj industriji, analiza dijagnoza u medicini, analiza interakcija lijekova u farmaciji, rezultati marketinških kampanja, itd.

Izvorni podaci mogu dolaziti s različitih internih i eksternih izvora, između ostalih i društvenih mreža kakve su Facebook, LinkedIn i Twitter. Cilj integracije nestrukturiranih tekstualnih podataka u data warehouse je dobiti općenite i/ili vrlo specifične informacije iz tih podataka.

Za obradu teksta u ovom kontekstu koristit će se Oracle Text, alat koji koristi standardni SQL za indeksiranje, pretraživanje i analizu teksta pohranjenog u bazi podataka, na file sistemu ili na webu, a služiza lingvističku analizu dokumenata, pretragu teksta po ključnim riječima, kontekstu, temama, dijelovima riječi, itd.

Kao primjer iz prakse obradit će se marketinška kampanja jedne izmišljene pivovare (nazovimo je Pivovara Zlatica) pri plasiranju nove vrste piva (nazovimo ga Zlatno pivo) , odnosno reakcije konzumenata s Twittera. Twitter je jednostavan izvor iz kojega je olakšano pretraživanje ključnih pojmova u potrazi za dijelovima koji su interesantni za poslovanje.

3.1 Dohvat komentara s Twittera

Pretraživanjem Twitter kanala Pivovare Zlatica (@Pivovara_Zlatica) pronađenoje nekoliko komentara:

Pivoljubac	27.07.2010 12:46:17:	Izašlo novo pivo iz Zlatice, je li itko probao? #zlatnopivo
Majstor	27.07.2010 13:50:50:	Kod nas još nije stiglo u trgovine, ali jedva čekam... #zlatnopivo
Oskosk	27.07.2010 14:00:10:	Meni je pregorko, ne valja :(#zlatnopivo
Pro777	27.07.2010 14:07:34:	Jedno od najkvalitetnijih piva na tržištu, ali preskupo #zlatnopivo
Veseli	27.07.2010 14:31:10:	Meni je cijena OK, okus super, ja odsad pijem samo #zlatnopivo
Mrgud	27.07.2010 14:59:59:	Ambalaža totalni #fail, ostalo može proći

Za temu nije previše bitan način izvlačenja podataka s izvora u bazu podataka, pa se u ovom tekstu neće posebno obrađivati, već ćemo pretpostaviti da su tweetovi već učitani u Data Staging Area (DSA), u tablicu DSA_TWITTER_KOMENTARI, koja ima slijedeću strukturu:

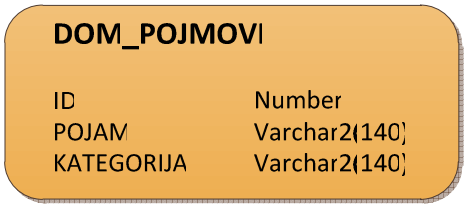
DSA_TWITTER_KOMENTARI

ID	Number
KORISNIK	Varchar2(140)
DATUM_TWEETA	Date
TWEET	Varchar2(140)

U polje ID puni se broj iz sekvence koji će u cjelokupnom procesu integracije služiti kao jedinstveni identifikator tweeta. U polje KORISNIK puni se ime korisnika koji je ostavio komentar, a u polje DATUM_TWEETA datum i vrijeme nastanka komentara. Ova dva polja čine prirodni ključ tweeta iz izvora. U polje TWEET puni se cijeli neobrađeni komentar s izvora.

3.2 Definiranje pojmova o proizvodu i dojmu

Jedan od osnovnih koraka u integraciji teksta je već spomenuto definiranje baze domenskih pojmova, odnosno domenskog znanja. Ovisno o temi na kojoj je fokus, ovaj korak može biti vrlo dugotrajan i kompliciran. Pojmove je potrebno spremiti u tablicu jednostavne strukture:



U slučaju naše pivovare, popis pojmova će biti:

pivo, pivovara, zlatica, zlatno, hmelj, okus, cijena, ambalaža, skupo, jeftino, prihvatljivo, OK, dobro, loše, ukusno, najbolje, najgore, gorko, blago, fail, win, kvalitetno, super, katastrofa, ...

kao i sve izvedenice iz istog korijena, ali i uobičajeni kolokvijalni nazivi (piva, bira, vopi, pivica, ...) i uobičajene pravopisne greške (cjena, supr, dorbo, ...). Definirani pojmovi bit će smješteni u jednu od dvije kategorije – PROIZVOD (svi pojmovi koji su vezani za promatrani proizvod u užem i širem smislu) i DOJAM (svi pojmovi kojima se izražava dojam ili mišljenje).

3.3 Parsiranje i tagiranje tweetova

Da bi se tekst tweeta mogao pretraživati koristeći Oracle Text, potrebno je na koloni DSA_TWITTER_KOMENTARI.TWEET podignuti kontekstni index:

```
CREATE INDEX dsa_twt_kom_twitter_idx
   ON DSA_TWITTER_KOMENTARI (tweet)
   INDEXTYPE IS CTXSYS.CONTEXT;
```

Kreiranjem kontekstnog indeksa Oracle Text automatski parsira i tokenizira tekst u koloni, te tokene sprema u indeksnu tablicu naziva DR$<ime_indexa>$I, odnosno u konkretnom slučaju DR$DSA_TWT_KOM_TWITTER_IDX$I.

Povezivanjem tokena iz te tablice s kategorijama moguće je jednostavno tagirati pojmove, odnosno pridružiti im definirane pojmovne kategorije, ali bolji način je koristiti operator CONTAINS za pretraživanje teksta:

```
SELECT t.id,
       t.tweet,
       p.pojam,
       p.kategorija
  FROM DSA_TWITTER_KOMENTARI t, DOM_POJMOVI p
 WHERE CONTAINS (t.tweet, p.pojam) > 0
```

ID	TWEET	POJAM	KATEGORIJA
2	Izašlo novo pivo iz Zlatice, je li itko probao? #zlatnopivo	pivo	PROIZVOD
4	Jedno od najkvalitetnijih piva na tržištu, ali preskupo #zlatnopivo	preskupo	DOJAM
4	Jedno od najkvalitetnijih piva na tržištu, ali preskupo #zlatnopivo	najkvalitetnijih	DOJAM
5	Meni je cijena OK, okus super, ja odsad pijem samo #zlatnopivo	okus	PROIZVOD
5	Meni je cijena OK, okus super, ja odsad pijem samo #zlatnopivo	cijena	PROIZVOD
5	Meni je cijena OK, okus super, ja odsad pijem samo #zlatnopivo	OK	DOJAM
5	Meni je cijena OK, okus super, ja odsad pijem samo #zlatnopivo	super	DOJAM
6	Ambalaža totalni #fail, ostalo može proći	ambalaža	PROIZVOD
6	Ambalaža totalni #fail, ostalo može proći	fail	DOJAM

3.4 Definiranje relacija među pojmovima

U ovoj fazi integracije potrebno je definirati domensko znanje, tj. odnose i značenja odnosa između domenskih pojmova.

Iako je moguće definirati mnogo vrsta odnosa, za ovaj slučaj bit će potrebno definirati samo sinonime pojmova iz kategorije DOJAM, i to na način da će svi pojmovi koji označavaju pozitivno mišljenje biti sinonimi pojma WIN, a svi pojmovi koji označavaju negativno mišljenje sinonimi pojma FAIL.

Oracle Text nudi mogućnost korištenja postojećeg tezaurusa, ali kako je već navedeno da je preporučljivo graditi specijalizirane baze domenskih pojmova i znanja, kreirat ćemo novi tezaurus u kojem ćemo definirati spomenute sinonime.

```
begin
CTXSYS.CTX_THES.CREATE_THESAURUS('thes_dojam');
CTXSYS.CTX_THES.CREATE_RELATION('thes_dojam', 'win', 'SYN',
'najkvalitetnijih');
CTXSYS.CTX_THES.CREATE_RELATION('thes_dojam', 'win', 'SYN', 'super');
CTXSYS.CTX_THES.CREATE_RELATION('thes_dojam', 'win', 'SYN',
'kvalitetno');
CTXSYS.CTX_THES.CREATE_RELATION('thes_dojam', 'win', 'SYN', 'blago');
CTXSYS.CTX_THES.CREATE_RELATION('thes_dojam', 'win', 'SYN', 'jeftino');
CTXSYS.CTX_THES.CREATE_RELATION('thes_dojam', 'win', 'SYN',
'prihvatljivo');
CTXSYS.CTX_THES.CREATE_RELATION('thes_dojam', 'win', 'SYN', 'OK');
CTXSYS.CTX_THES.CREATE_RELATION('thes_dojam', 'win', 'SYN', 'dobro');
CTXSYS.CTX_THES.CREATE_RELATION('thes_dojam', 'win', 'SYN', 'ukusno');
CTXSYS.CTX_THES.CREATE_RELATION('thes_dojam', 'win', 'SYN', 'najbolje');
CTXSYS.CTX_THES.CREATE_RELATION('thes_dojam', 'fail', 'SYN',
'preskupo');
CTXSYS.CTX_THES.CREATE_RELATION('thes_dojam', 'fail', 'SYN',
'katastrofa');
CTXSYS.CTX_THES.CREATE_RELATION('thes_dojam', 'fail', 'SYN', 'gorko');
CTXSYS.CTX_THES.CREATE_RELATION('thes_dojam', 'fail', 'SYN', 'najgore');
CTXSYS.CTX_THES.CREATE_RELATION('thes_dojam', 'fail', 'SYN', 'skupo');
CTXSYS.CTX_THES.CREATE_RELATION('thes_dojam', 'fail', 'SYN', 'loše');
end;
```

Na ovaj način moguće je napraviti i pravu hijerarhiju koristeći relacije NT (narrower term – uži pojam, podpojam) i BT (broader term – širi pojam, nadpojam), ali za ovaj primjer dovoljna je dvorazinska hijerarhija dobivena relacijom SYN (synonym – sinonim).

3.5 Stvaranje dojma sintezom pojmova

Kreiranjem tezaurusa i sinonimskih relacija između domenskih pojmova iz kategorije DOJAM stvorena je pretpostavka za sintezu svih pojmova sa sličnim značenjem u jednu grupu pojmova, odnosno stvoren je temelj odgovora na pitanje "Sviđa li se naše novo pivo tržištu?".

Novi tezaurus se koristi u okviru operatora CONTAINS, na slijedeći način:

```
SELECT t.id, t.tweet, 'win' dojam
  FROM DSA_TWITTER_KOMENTARI t
 WHERE CONTAINS (tweet, 'syn(win, thes_dojam)') > 0
UNION ALL
SELECT t.id, t.tweet, 'fail' dojam
  FROM DSA_TWITTER_KOMENTARI t
 WHERE CONTAINS (tweet, 'syn(fail, thes_dojam)') > 0;
```

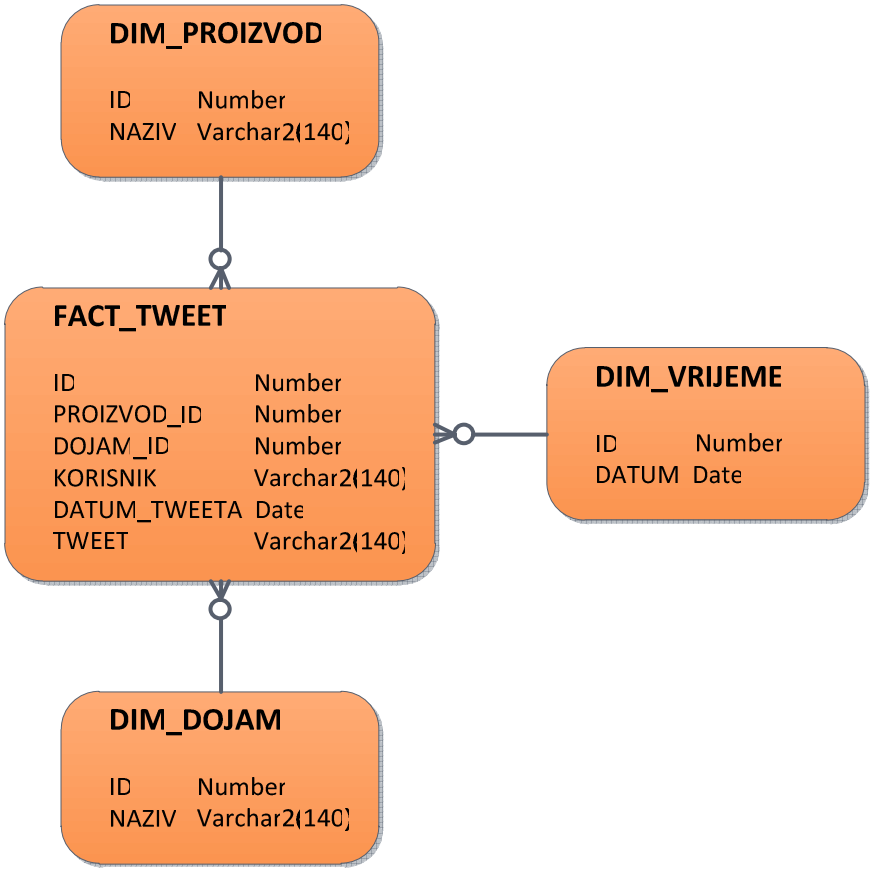
ID	TWEET	DOJAM
4	Jedno od najkvalitetnijih piva na tržištu, ali preskupo #zlatnopivo	win
5	Meni je cijena OK, okus super, ja odsad pijem samo #zlatnopivo	win
4	Jedno od najkvalitetnijih piva na tržištu, ali preskupo #zlatnopivo	fail
6	Ambalaža totalni #fail, ostalo može proći	fail

Rezultat ukazuje na dojam koji tweet ostavlja, a to je upravo informacija koja je u fokusu cijelog procesa integracije. Ovim korakom završava transformacija teksta i može se početi s oblikovanjem informacija u prihvatljive izvještajne strukture.

3.6 Kreiranje data martova

Kao što je već spomenuto, u integraciji teksta udata warehouse mogu se razlikovati dvije vrste data martova – narativni i sintetizirani.

Unutar narativnog datamarta definirane su tri dimenzije (DIM_VRIJEME, DIM_PROIZVOD i DIM_DOJAM), te jedna fact tablica FACT_TWEET.



Veze između dimenzija i fact tablice su vrlo neobavezne, bez posebnih pravila. Informacije koje se mogu dobiti iz ovog modela su vrlo općenite, raznovrsne i ne nužno točne, pa je potrebna dublja angažiranost korisnika u interpretaciji podataka.

Dimenzija DIM_VRIJEME se puni jednom od generiranih podataka. Ostale dvije dimenzije se pune na temelju tablice domenskih pojmova, na način da se svaka definirana pojmovna kategorija pretvara u posebnu dimenziju. Da je definirana prava hijerarhija, u dimenzijskim tablicama bi bili definirani i dodatni hijerarhijski atributi, no u ovom slučaju to nije potrebno.

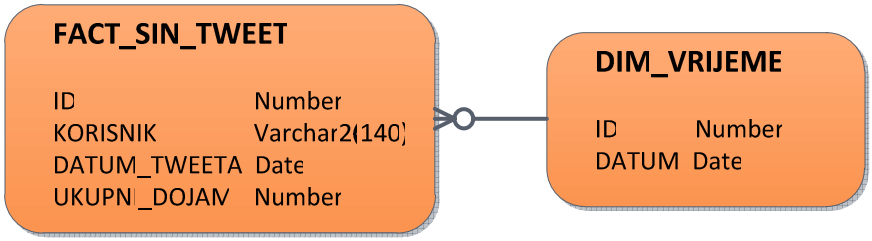
```
INSERT INTO DIM_DOJAM (ID, NAZIV)
     SELECT id, pojam
       FROM dom_pojmovi
      WHERE kategorija = 'DOJAM';

INSERT INTO DIM_PROIZVOD (ID, NAZIV)
     SELECT id, pojam
       FROM dom_pojmovi
      WHERE kategorija = 'PROIZVOD';
```

Fact tablica se puni spajanjem tweetova s pojmovima koji se odnose na proizvode i s pojmovima koji se odnose na dojmove. Da bi se izbjeglo višestruko umnožavanje i povezivanje pojmova u slučaju kada u jednom tweetu postoje dojmovi za više proizvoda ("cijena OK, okus super"), koristi se operator NEAR koji ograničava povezivanje samo na pojmove koji su u tekstu blizu jedan drugome (pritom je "blizu" parametar koji označava broj susjednih riječi koje se razmatraju kod povezivanja pojmova, a poprima vrijednosti od 1 do 100).

```
INSERT INTO FACT_TWEET (ID, DOJAM_ID, PROIZVOD_ID, KORISNIK,
            DATUM_TWEETA, TWEET)
(SELECT id, dojam_id, proizvod_id, korisnik, datum_tweeta, tweet
  FROM (SELECT d.*, p.proizvod_id, proizvod_pojam
          FROM    (SELECT t.id,
                          p.id dojam_id,
                          p.pojam dojam_pojam,
                          korisnik,
                          datum_tweeta,
                          tweet
                     FROM DSA_TWITTER_KOMENTARI t, DOM_POJMOVI p
                    WHERE CONTAINS (t.tweet, p.pojam) > 0
                          AND kategorija = 'DOJAM') d
                JOIN
                   (SELECT t.id, p.id proizvod_id, p.pojam proizvod_pojam
                      FROM DSA_TWITTER_KOMENTARI t, DOM_POJMOVI p
                     WHERE CONTAINS (t.tweet, p.pojam) > 0
                           AND kategorija = 'PROIZVOD') p
                ON (d.id = p.id))
 WHERE CONTAINS (tweet,
       'near(('||proizvod_pojam||','||dojam_pojam||'),1,FALSE)') > 0)
```

Sintetizirani data mart se sastoji od već postojeće vremenske dimenzije DIM_VRIJEME, te fact tablice FACT_SIN_TWEET koja sadrži sintetizirane podatke o dojmu, s tim da je uglavnom pozitivan tweet označen brojem 1, uglavnom negativan brojem -1, a ni pozitivan ni negativan nulom.




```
INSERT INTO FACT_SIN_TWITTER (ID, KORISNIK, DATUM_TWEETA, UKUPNI_DOJAM)
  (SELECT id, korisnik, datum_tweeta, SUM (dojam) ukupni_dojam
    FROM (SELECT t.*, 1 dojam
            FROM DSA_TWITTER_KOMENTARI t
           WHERE CONTAINS (tweet, 'syn(win, dojam_o_pivu)') > 0
          UNION ALL
          SELECT t.*, -1 dojam
            FROM DSA_TWITTER_KOMENTARI t
           WHERE CONTAINS (tweet, 'syn(fail, dojam_o_pivu)') > 0)
GROUP BY id, korisnik, datum_tweeta)
```

Na ovaj način moguće je sumiranjem dobiti dojam na razini dana, gdje bi pozitivna suma značila dobar dojam, a negativna suma loš dojam.

Kad bi se u ovaj model dodala još i dimenzija s podacima o promotivnim akcijama i marketinškim kampanjama, mogla bi se pratiti uspješnost pojedine akcije. Moguće je napraviti i kompleksnije sinteze podataka, ali ovo je dovoljno dobar primjer za dokaz koncepta.

4 ZAKLJUČAK

"Gdje postoji potreba, postoji i način", kaže poznata izreka. Neosporno postoji potreba za korištenjem nestrukturiranih podataka u modenom poslovanju. Poslovne organizacije se sve više oslanjaju na internet kanale za prodaju, marketing i ostale vrste komunikacije, a starije od njih

posjeduju i zalihe internih nestrukturiranih podataka, što u kombinaciji s on-line podacima predstavlja neiskorišteni rudnik zlata.

Trenutno je najveći problem nedostatak prikladnih alata specijaliziranih za integraciju ovakvih podataka, koji bi podržavao cjelokupni proces od ekstrakcije s izvora preko upravljanja domenskim pojmovima i znanjem, do dizajna data martova, pa čak i dalje, sve do prezentacijskog sloja. Ovakvi alati su još u začecima, mada razni proizvođači nude profesionalna rješenja koja najčešće pokrivaju samo jedan dio integracijskog procesa, što znači da za podržati cijeli proces najčešće treba koristiti proizvode od više proizvođača, a to opet povlači za sobom neke neželjene posljedice.

Srećom, postojeći Oracle alati i tehnologije su vrlo dobri za postavljenu zadaću. Prije svega to se odnosi na Oracle Text, koji je jedan od najmoćnijih alata za tekstualne operacije. Oracle Warehouse Builder je moguće koristiti u gotovo svim koracima integracije umjesto ručnog pisanja SQLa i PL/SQLa, što ima očite i poznate prednosti. Čak i u prezentacijskom dijelu Oracle ima neke od najjačih aduta (OBIEE, Discoverer, Reports, BI Publisher), ali najveća prednost je to što sve navedene komponente počivaju na jednoj platformi – Oracle bazi podataka, što olakšava integraciju inputa i outputa spomenutih alata.

U ovom dokazu izvedivosti procesa integracije nestrukturiranih podataka u Data Warehouse treba riješiti još neke probleme, ali jasno je pokazano da se može itekako kvalitetno postojećim tehnologijama zadovoljiti poslovna potreba za informacijama iz nestrukturiranih podataka.