EKSTREMNO NORMALIZIRANI DATA WAREHOUSE MODELI PODATAKA

EXTREMLY NORMALIZED DATA WAREHOUSE DATA MODELS

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

Tradicionalne tehnike modeliranja DWa (Kimballova i Inmonova „škola“) su u većini slučajeva dokazano uspješne, a praćene su i tradicionalnim razvojnim metodologijama.

U novije vrijeme sve više probijaju agilne razvojne metodologije koje prate i odgovarajući podatkovni modeli, karakteristični po velikom stupnju normaliziranosti i neosjetljivosti na promjene u poslovnoj okolini, bez naknadne potrebe za redizajnom. Najbolji primjeri takvih ekstremno normaliziranih modela su Data Vault i Anchor.

Osim osnovnih karakteristika, bit će opisani potrebni preduvjeti i načini implementacije tih modela u bazi podataka, njihove prednosti i nedostaci, te usporedba s tradicionalnim modelima.

.

Traditional DW modeling techniques (Kimball and Inmon "schools") are in most cases proven to be successful, and are accompanied by traditional development methodologies.

In recent years, we see an advancement of agile development methodologies, accompanied by their own data models, which are characterized by a high degree of normalization and insensitivity to changes in the business environment, without having to redesign them later. The best examples of these extremely normalized models are Data Vault and Anchor.

In addition to the basic characteristics, necessary prerequisites and ways of implementation of these models in the database will be described, as well as their strengths and weaknesses when compared to the traditional models.

.

1. UVOD

Skladišta podataka se uglavnom modeliraju na temelju pravila i principa jedne od dviju glavnih „škola“ modeliranja – Kimballove i Inmonove. Ove tradicionalne tehnike modeliranja su provjerene, u većini slučajeva dokazano uspješne, imaju veliki broj pristaša s velikim iskustvom u takvom modeliranju, a praćene su i tradicionalnim razvojnim metodologijama, koje ih uspješno nadopunjuju.

Kombinacija tradicionalnih razvojnih metodologija i tradicionalnih tehnika modeliranja često znači dulje vrijeme izgradnje upotrebljivog skladišta podataka (ili dijela skladišta podataka), pa se iz tog razloga sve više probijaju „agilne“ razvojne metodologije koje prate i odgovarajući podatkovni modeli. Spomenuti modeli karakteristični su po velikom stupnju normaliziranosti i neosjetljivosti na promjene u poslovnoj okolini, odnosno prilagođeni čestim promjenama, kao i modeliranju malih dijelova podatkovnog modela, bez naknadne potrebe za redizajnom. Najbolji primjeri ovih ekstremno normaliziranih modela su Data Vault model i Anchor model.

Osim osnovnih karakteristika, bit će opisani potrebni preduvjeti i načini implementacije ekstremno normaliziranih podatkovnih modela u bazi podataka, njihove prednosti i nedostaci, te pregled i usporedba s tradicionalnim, normaliziranim modelima.

2. EKSTREMNO NORMALIZIRANI DATA WAREHOUSE MODELI

Primarne funkcije data warehouse sustava su olakšati izvještavanje, analizu i rudarenje podataka. Da bi DW sustavi ispunili ove funkcije, moraju ispunjavati određene preduvjete, od kojih su najvažniji:

- integracija podataka iz različitih izvora
- čuvanje povijesnih podataka
- agregiranje podataka
- dostava podataka BI alatima

Osim ovih osnovnih preduvjeta, postoje i druge važne karakteristike na koje treba obratiti pažnju prilikom dizajna modela DW sustava, kao što su npr. kompleksnost implementacije, otpornost na promjene, jednostavnost korištenja i dr.Uspoređujući podatkovne modele koji se koriste u DW sustavima primjećuje se da pojedini modeli bolje od drugih ispunjavaju navedene preduvjete i karakteristike, ovisno o tome koju poziciju u DW arhitekturi zauzimaju, pa su tako neki modeli pogodniji za integraciju podataka, a neki za dostavu podataka korisnicima.

2.1.Data Vault

Data Vault model (autor Dan Linstedt, danlinstedt.com) je jedinstveno povezani skup normaliziranih tablica koje podržavaju jedno ili više poslovnih funkcionalnih područja. Orijentiran je detaljnim podacima, te omogućava povijesno praćenje promjena. To je hibridni pristup koji obuhvaća najbolje karakteristike između 3NF modela i Star schema modela.Dizajn je fleksibilan, skalabilan, dosljedan i prilagodljiv potrebama poslovanja, te je projektiran posebno za potrebe skladišta podataka.

2.1.1. Osnovne informacije

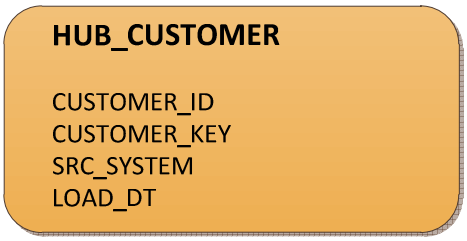
U uobičajenoj podjeli podatkovnih modela, Data Vault pripada back-end modelima (kao i ODS, Operational Data Store), što znači da nije potpuno prilagođen ad-hoc izvještajima i dostavi podataka u front-end modele, kao što je npr. Star schema.

Data Vault model fokusiran je na nekoliko aspekata skladištenja podataka. Prvo, naglašava potrebu mogućnosti praćenja izvora podataka koji su došli u data warehouse (traceability i auditability). Drugo, ne pravi razliku između ispravnih i neispravnih podataka, tj. onih koji nisu u skladu s poslovnim pravilima, time omogućavajući „jedinstvenu verziju činjenica“ umjesto „jednistvene verzije istine“. Model je također dizajniran s namjerom da bude otporan na promjene u poslovanju, kao izvoru podataka, na način da jasno razdvoji strukturne informacije (poslovni ključevi i veze među njima) od opisnih atributa. Konačno, Data Vault je zamišljen da u najvećoj mjeri omogućava paralelizam u punjenju skladišta podataka, na način da se prvo učitavaju osnovne strukturne informacije, nakon čega se paralelno učitavaju kompleksne strukturne informacije, te se na kraju paralelno učitavaju opisni atributi.

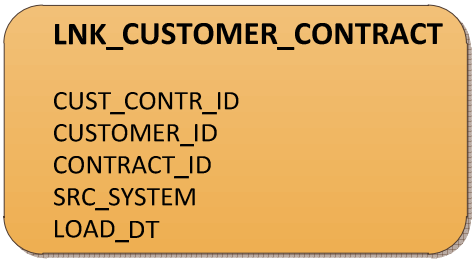
2.1.2. Elementi modela

Da bi dizajn bio jednostavan, ali elegantan, potreban je minimalni broj komponenti, konkretno Hub (os, osovina, središte, čvor), Link (veza) i Satellite (satelit).Hub entitet predstavlja primarni ključ. Link entiteti daju transakcijsku integraciju između Hubova. Satelitski entiteti daju kontekst i opis Hubovima.

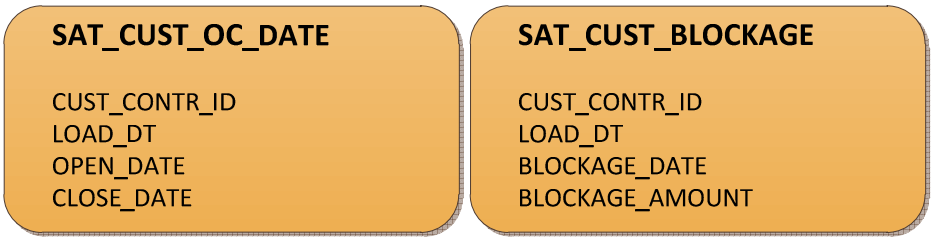
Hub se sastoji od poslovnog ključa, surogatnog ključa, datuma i vremena učitavanja podataka, te oznake izvora podataka. Surogatni ključ je uglavnom opcionalan, osim u slučajevima kad je potrebno ujediniti poslovne ključeve iz različitih poslovnih područja. U tom slučaju Hub bi imao više poslovnih ključeva koji bi bili prevedeni u jedan jedinstveni surogatni ključ.



Link predstavlja relaciju između dva ili više poslovnih entiteta, odnosno njihovih poslovnih ključeva, npr. kupca i računa. U skladu s tim, Link se sastoji od dva ili više surogatnih ključeva Hubova koje povezuje, oznake izvora podataka, datuma i vremena učitavanja podataka, te surogatnog ključa Linka.



Satelit predstavlja kontekst Huba ili Linka, odnosno atribute koji opisuju poslovni ključ ili vezu između ključeva. Kako su opisni atributi podložni promjenama, satelit mora biti u mogućnosti pratiti povijest promjena podataka. Satelit se sastoji od ključa Huba ili Linka i datuma i vremena učitavanja, koji zajedno čine primarni ključ Satelita. Opcionalni elementi Satelita su surogatni ključ i oznaka izvora podataka. Osim navedenih elemenata, Satelit sadrži i jedan ili više opisnih atributa, te najviše podsjeća na Kimballovu dimenziju tipa 2. Broj satelita vezanih na Hub ili Link nije ograničen, već njihov broj ovisi o kriteriju podjele atributa, koji može biti međusobna povezanost atributa, frekvencija promjene, frekvencija dohvata podataka ili neki drugi kriterij.



Sateliti se u model mogu dodavati po potrebi, bilo zbog proširenja opsega modela, bilo zbog promjene u poslovnom sustavu, bilo zbog nekog drugog razloga.

2.1.3. Pravila modeliranja

Data Vault model mora slijediti određena pravila da bi se iskoristile prednosti modela:

- Ključevi Huba ne mogu semigrirati u druge Hubove (ne postojichild/parent Hub)
- Hubovi moraju biti spojeni preko Linkova
- Više od dvaHuba može biti spojeno preko Linka
- Linkovi mogu biti povezani s drugim Linkovima
- Link mora imati najmanje dva povezana Huba
- Surogatni ključevi mogu se koristiti za Hubovei Linkove.
- Surogatni ključevine mogu se koristiti za Satelite
- Hub ključevi uvijek migriraju prema van
- Poslovni i surogatni ključevi Huba se nikada ne mijenjaju
- Sateliti mogu biti povezani s Hubom ili Linkom
- Sateliti moraju imati datum i vrijeme unosa ili ključ na samostalnu tablicu koja sadrži datum i vrijeme unosa
- Stand-alone tablice, kao što su kalendari, vrijeme, šifarnici i opisne tablice se mogu koristiti

- Linkovi mogu imati surogatni ključ
- Ako Hub ima dva ili više satelita, može se dodati point-in-time tablica
- Sateliti se uvijek pune deltama podataka, dupli redovi ne bi se trebalipojavljivati
- Podaci se razdvajaju po Satelitima na temelju: 1) vrstepodataka 2) frekvencije promjene

2.2. Anchor model

Anchor modeliranje (autor Lars Ronnback, www.anchormodelling.com) je open source tehnika modeliranja podataka razrađena na pretpostavci da se okruženjedata warehouse sustava konstantno mijenja. Međutim, velike promjeneizvan modela će rezultirati malim promjenama unutar modela. Model se temelji na šestoj normalnoj formi, što rezultira implementacijom pomoću velikog broja komponenti, čime se izbjegavajumnogi nedostaci povezani s tradicionalnim modelima baza podataka.

2.2.1. Osnovne informacije

Zahvaljujući svojoj modularnoj prirodiAnchor model podržava jasno odvajanje interesnih područja i pojednostavljuje definiranje opsega projekta. Na ovaj način dizajn modela skladišta podataka može početi s malim prototipovima, te se kasnije razviti u korporativno skladište podataka bez potrebe za reinženjeringom dosad napravljenog modela.

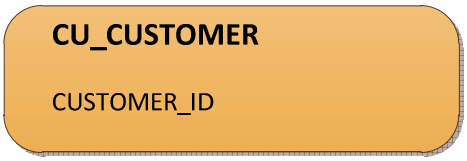
Iako su korijeni Anchor modela proizašli iz zahtjeva koje su postavila skladišta podataka, ovo je generički pristup modeliranju koji je također pogodan i za druge vrste sustava. Svaka promjena modela se provodi kao neovisnoi nedestruktivno proširenje postojećeg modela. Kao rezultat ovog načina modeliranja, sve trenutne aplikacije ostat će nepromijenjene, odnosno sve verzije aplikacija koje se naslanjaju na modelmogu se koristiti u isto vrijeme kada se provode proširenja modela. Ukratko, svaka prethodna verzija modela još uvijek postoji kao podskup unutar cjelokupnog Anchor modela.

Kao i Data Vault, i Anchor model pripada back-end modelima. Osim ove očite sličnosti, postoje i mnoge druge, prvenstveno temeljene na sličnim konceptima strukturalnih podataka.

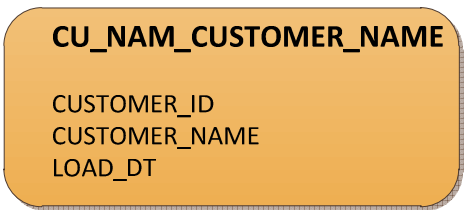
2.2.2. Elementi modela

Anchor model ima četiri osnovnakoncepta modeliranja: sidra (Anchor), atribute (Attribute), veze (Tie) i čvorove (Knot). Svaki od ovih logičkih koncepata se transformira u jednu tablicu fizičkog modela. Zbog ovakvog pristupa implementaciji dolazi do eksplozije broja tablica, pa se za tradicionalnu dimenzijsku tablicu s 10 atributa može očekivati najmanje isto toliko samostalnih tablica. Da bi se olakšalo snalaženje među velikim brojem tablica, metodologija Anchor modela predviđa i posebna pravila imenovanja tablica, ovisno o njihovoj namjeni.

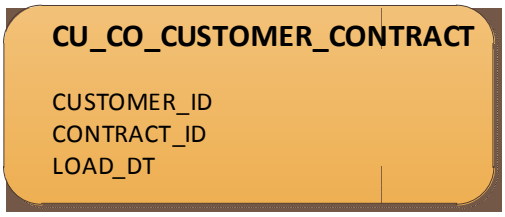
Sidro se koristi za modeliranjeentiteta i događaja, a fizička tablica sastoji se samo od surogatnog ključa. Za razliku od Data Vault modela, poslovni ključ se modelira kao atribut Sidra.Naziv Sidra počinje dvoslovnom skraćenicom praćenom opisom Sidra.



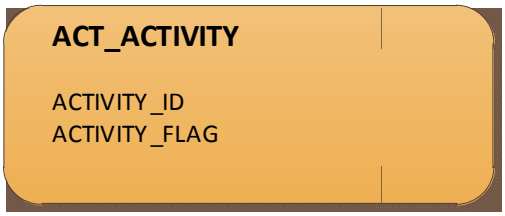
Atributi se koriste za modeliranje svojstava Sidra, pa fizička tablica osim ključa Sidra sadrži i opis svojstva, te datum i vrijeme unosa, ukoliko je potrebno čuvati povijest promjene Atributa. Naziv Atributa sastoji se od dvoslovne skraćenice naslijeđene od Sidra, troslovne skraćenice atributa, opisa naslijeđenog od Sidra, te opisa Atributa.



Veza služi za modeliranje odnosa između dva ili više Sidra. Fizička implementacija sastoji se od dva ili više ključeva Sidra i opcionalnog datuma i vremena unosa.Naziv Veze sastoji se od dvoslovnih skraćenica naslijeđenih od Sidara, te opisa naslijeđenih od Sidara.



Čvorovi se koriste za modeliranje zajedničkih osobina, tipova ili stanja veze, te šifarnika. Sadrže ključ i opis zajedničke osobine. Naziv Čvora sastoji se od troslovne skraćenice i opisa Čvora.

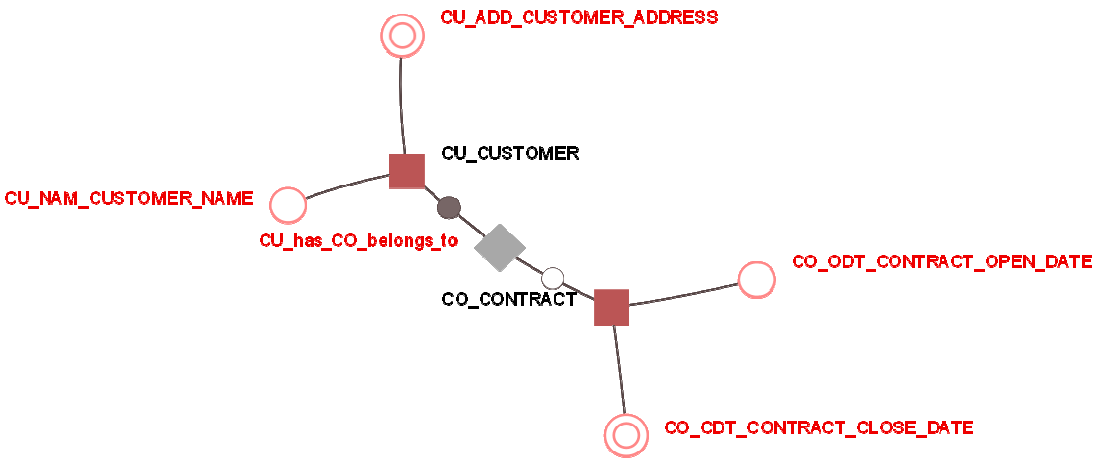


Osim pravila imenovanja tablica, metodologija modeliranja predviđa i posebne view-ove za olakšano praktično korištenje modela:

- Posljednji podatak
- Određena točka u vremenu
- Interval
- Prirodni ključ

Kao i tablice, i viewovi i kolone moraju slijediti pravila imenovanja.

Vrlo poželjna osobina ovako detaljnog i normaliziranog modela je i mogućnost automatskog generiranja objekata u bazi podataka, od tablica i viewova, do ETL procedura. Lako je zamisliti definiciju modela u Excel tablici ili XMLu koja se dinamički pretvara u skladište podataka, u potpunosti bez potrebe za ručnim kodiranjem.



2.2.3. Pravila modeliranja

Kao i kod Data Vaulta, postoje pravila kojih se potrebno pridržavati da bi se iskoristile sve prednosti modela:

- Sidra se koriste za modeliranje osnovnih entiteta i transakcija
- Ako je potrebno čuvati povijest promjena, koristi se povijesni Atribut, a inače statični
- Ako vrijednosti atributa imaju mali broj fiksnih vrijednosti, koristi se Atribut s Čvorom, a inače statični Atribut
- Ako vrijednosti atributa imaju mali broj fiksnih vrijednosti i potrebno je čuvati povijest promjena, koristi se povijesni Atribut s Čvorom
- Ako se veze između Sidara mogu mijenjati kroz vrijeme, koristi se povijesna Veza, a inače statična Veza
- Ako tipovi veza imaju mali broj fiksnih vrijednosti, koristi se Veza s Čvorom, a inače statična Veza
- Ako tipovi veza imaju mali broj fiksnih vrijednosti i potrebno je čuvati povijest promjena, koristi se povijesna Veza s Čvorom

2.2.4. Performansni aspekti

Kao posljedica velikog broja fizičkih tablica koje rezultiraju iz Anchor modela, prvo pitanje koje se nameće su performanse spajanja (join) velikog broja tablica. Odgovor na pitanje je da se, uz specifične karakteristike samog modela, Anchor model čvrsto oslanja na određene preduvjete i funkcionalnosti baze podataka.

U prvom redu tu su primarni ključevi i referencijalni integritet. Apsolutno je neophodno definirati primarne ključeve nad Sidrima i Čvorovima, te vanjske ključeve nad Atributima i Vezama. Indeksi moraju biti implementirani kao indeksno organizirane tablice (Index-Organized tables), tako da ne zauzimaju dodatni prostor na diskovima. Dodatni indeksi u većini slučajeva nisu potrebni.

Pošto se podaci u Anchor modelu ne dupliciraju, veličina tablica će biti puno manja u usporedbi s denormaliziranim modelima.

Uobičajeni upiti nad tablicama skladišta podataka u većini slučajeva dohvaćaju samo manji podskup svih atributa u tablici, što znači da se nepotrebno dohvaćaju podaci iz cijelog reda, samo da bi se na kraju odbacili. Ovakvom normalizacijom postiže se efekt vertikalnog particioniranja tablica karakterističnog za column-based baze podataka. Međutim, za dohvat samo potrebnih podataka iz predefiniranih viewova potrebno je da baza podataka podržava mogućnost eliminacije tablica/joina iz upita, što bi značilo da baza jednostavno izbacuje iz selecta kolone, odnosno u ovom slučaju tablice, koje se ne pojavljuju u krajnjem rezultatu. Oracle RDBMS od verzije 11 potpuno podržava ovu funkcionalnost.

Uzimajući u obzir navedeno, može se zaključiti da je Anchor model u suštini manje I/O intenzivan od denormaliziranih modela, a I/O je često ključni problem u data warehouse sustavima.

Dodatne uobičajene opcije za poboljšanje performansi su i particioniranje i kompresiranje tablica, a imajući na umu da je ovo ipak back-end model, uvijek je moguće konkretizirati viewove u obliku tablica ili materijaliziranih viewova.

3. USPOREDBA DATA MODELA

Kad se govori o stupnju normaliziranosti data modela, vidljivo je da su stariji modeli više denormalizirani od modernih. Ovaj trend se može objasniti i time što su noviji modeli dizajnirani s namjenom da olakšaju back-end procese integracije podataka, dok su stariji modeli naglašavali jednostavnost i front-end procese, bliže poslovnim korisnicima. Dok je Star schema primarno front-end model, Inmonov CIF u obliku historiziranog 3NF modela jednako je koristan (i nekoristan) i kao front-end i kao back-end model, a Data Vault i Anchor su primarno back-end modeli. U skladu s tim, pristup dizajnu i izgradnji skladišta podataka kod Kimballovog modela je top-down, a fokus je na poslovnim korisnicima kao motivu za izgradnju skladišta.

Moderni modeli su također agilniji od tradicionalnih po pitanju otpornosti na promjene, koje često imaju za posljedicu velike zahvate nad modelom i podacima kod tradicionalnih modela, dok su promjene kod modernih modela uglavnom nedestruktivne prirode. Tako je i vrijeme razvoja i održavanje znatno duže i kompleksnije kod tradicionalnih modela. Kompleksnost ETLa pada što je model normaliziraniji, pa je za Anchor model ETL moguće čak i potpuno automatizirati.

Normalizirani modeli imaju i prednost manje potrošnje diskovnog prostora jer po svojoj definiciji uglavnom ili u potpunosti ne dupliciraju podatke.

Auditabilnost, odnosno sposobnost praćenja porijekla podataka, u tradicionalnim modelima nije eksplicitno definirana, što ne znači da ne postoji, dok je u Data Vaultu i Anchoru već ugrađena kao jedna od važnih značajki tih modela.

	Star schema	CIF	Data Vault	Anchor
Normaliziranost	Niska	Srednja	Visoka	Ekstremna
Dohvat podataka	Odličan	Vrlo dobar	Dobar	Ovisi
Namjena	Front-end	Oboje	Back-end	Back-end
Pristup razvoju	Top-down	Bottom-up	Bottom-up	Bottom-up
Fokus	Korisnici	Podaci	Podaci	Domena
Otpornost na promjene	Niska	Niska	Visoka	Visoka
Vrijeme razvoja	Dugo	Dugo	Kratko	Kratko
Održavanje	Kompleksno	Kompleksno	Jednostavno	Jednostavno
ETL	Vrlo kompleksan	Kompleksan	Jednostavan	Automatiziran
Auditabilnost	Nije definirana	Nije definirana	Ugrađena u model	Ugrađena u model
Veličina	Velika	Srednja	Mala	Ekstremno mala

4. ZAKLJUČAK

Nije sporno da su današnji poslovni sustavi sve dinamičniji i podložniji promjenama koje se događaju pod utjecajem i unutrašnjih i vanjskih čimbenika. U takvoj okolini organizacije imaju i veću potrebu za bržim i jednostavnijim uvidom u značajne poslovne informacije iz svojih data warehouse sustava, što zahtijeva i specifična rješenja tih sustava.

Upravo ovi ekstremno normalizirani data warehouse modeli jedno su od rješenja navedenih problema. U kombinaciji s agilnim projektnim metodologijama omogućavaju faznu implementaciju, nerijetko u malim, ali brzim koracima. Posljedica takve implementacije je vrlo kratko vrijeme dostave korisnog sustava, odnosno dijela sustava, te brži povrat uloženih sredstava.

Ne postoji najbolji model podataka za data warehouse. Svi postojeći modeli imaju svoje prednosti i mane, a izbor modela bi trebao ovisiti primarno o potrebi poslovanja, te uzimati u obzir stanje podataka u izvornim sustavima, funkcionalna područja, predviđene troškove analize, implementacije i održavanja data warehouse sustava i druge aspekte.

Međutim, u današnjem dinamičnom poslovnom okruženju, prepoznaje se rastuća potreba za prilagođenijim data warehouse modelima kao što su Data Vault i Anchor model, unatoč nekim njihovim nedostacima, pa je moguće da su,u kombinaciji s modernim projektnim metodologijama, ovakvi modeli budućnost data warehousinga.