

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

TINE BOROVNIK

**SODOBNI PRISTOPI K NAČRTOVANJU
PODATKOVNIH SKLADIŠČ**

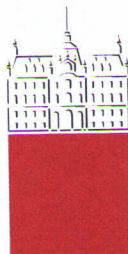
MAGISTRSKO DELO

Mentor prof. dr. Viljan Mahnič

Ljubljana, 2010

Št.: 113 -MAG-ISO/2010

Datum: 03. 09. 2010



Tine BOROVIK, univ. dipl. inž. rač. in inf.

Ljubljana

Fakulteta za računalništvo in informatiko Univerze v Ljubljani izdaja naslednjo magistrsko nalogo

Naslov naloge: **Sodobni pristopi k načrtovanju podatkovnih skladišč**

Modern approaches to data warehouse design

Tematika naloge:

Sistem poslovnega obveščanja predstavlja glavni vir informacij za različne ravni odločanja v podjetju. Podatkovno skladišče je osrednji del tega sistema. Področje podatkovnega skladiščenja je v več kot dveh desetletjih razvoja postalo zrelo. Kljub temu pa se pod vplivom razvoja informacijske tehnologije in sprememb v okolju pojavljajo novi, alternativni pristopi k načrtovanju podatkovnih modelov. Sledenje tem trendom je ena od pomembnejših nalog na področju informatike.

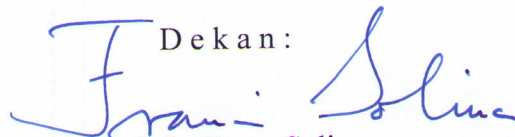
V magistrski nalogi predstavite različne tipe podatkovnih skladišč in sodobne pristope k načrtovanju njihovih podatkovnih modelov. Poleg izboljšav klasičnih pristopov predstavite tudi nove s poudarkom na prednostih, ki jih prinašajo. Med sodobnimi pristopi preučite *anchor modeling*, modeliranje v okolju DW2.0, ti. *OLAP* kocke, vključevanje struktur, namenjenih zagotavljanju kakovosti podatkov, in metode za optimizacijo podatkovnega modela. Izvedite vrednotenje pristopov in primerjavo med tistimi, ki so primerljivi. Z uporabo predstavljenih pristopov izdelajte primer modela podatkovnega skladišča za telekomunikacijsko podjetje.

Mentor:


prof. dr. Viljan Mahnič



Dekan:


prof. dr. Franc Solina

IZJAVA O AVTORSTVU

magistrskega dela

Spodaj podpisani Tine Borovnik, z vpisno številko 63970018, sem avtor magistrskega dela z naslovom »Sodobni pristopi k načrtovanju podatkovnih skladišč«.

S svojim podpisom zagotavljam, da:

- sem magistrsko delo izdelal/-a samostojno pod vodstvom mentorja prof. dr. Viljana Mahničarja,
- so elektronska oblika magistrskega dela, naslova (slov., angl.), povzetka (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko magistrskega dela
- in soglašam z javno objavo elektronske oblike magistrskega dela v zbirki »Dela FRI«.

V Ljubljani, dne 6.12.2010

Podpis avtorja: _____

Zahvala

Zahvaljujem se mentorju prof. dr. Viljanu Mahniču za vodenje in strokovno pomoč pri pripravi magistrskega dela.

Ani se zahvaljujem za podporo.

Zahvaljujem se tudi mojemu sedanjemu delodajalcu, ki mi je omogočil izobraževanja in delo na zanimivih projektih s področja podatkovnega skladiščenja.

Kazalo

Zahvala	3
Kazalo slik	9
Kazalo preglednic	9
Seznam kratic	11
Povzetek	1
Abstract.....	3
1 Uvod	5
2 Podatkovna skladišča.....	8
2.1 Osnovna opredelitev	8
2.2 Arhitektura	9
2.3 Sodobna umestitev	11
2.4 Vrste.....	13
2.4.1 Uvod	13
2.4.2 Korporativno podatkovno skladišče	13
2.4.3 Dimenzijsko podatkovno skladišče	14
2.4.4 Podatkovno skladišče kot skupek področnih podatkovnih skladišč.....	16
2.4.5 Aktivno podatkovno skladišče.....	17
2.4.6 Federativno podatkovno skladišče	18
2.4.7 Primerjava.....	19
2.5 DW 2.0.....	22
3 Dimenzijski modeli	25
3.1 Uvod.....	25
3.2 Povezava z normaliziranim modelom.....	26
3.3 Tabele dejstev	27
3.3.1 Splošno o tabelah dejstev	27
3.3.2 Transakcijske tabele dejstev	29
3.3.3 Tabele dejstev s periodičnimi posnetki stanj	29
3.3.4 Tabele dejstev z zbirnimi posnetki stanj	30
3.4 Dimenzijske tabele.....	30
3.4.1 Splošno o dimenzijskih tabelah	30
3.4.2 Skupne dimenzije	31
3.4.3 Počasi spreminjajoče se dimenzije	32
3.5 Zvezdne sheme in snežinke	33
3.6 Uporaba materializiranih pogledov.....	34
3.7 Večdimenzijske podatkovne baze.....	36
3.8 Kocke	37
3.9 Kocke in relacijski dimenzijski modeli.....	39
4 Drugi pristopi.....	42
4.1 Podatkovni modeli v okolju DW 2.0	42
4.1.1 Uvod	42
4.1.2 ERD	43
4.1.3 DIS.....	43
4.1.4 Fizični model	43
4.1.5 Izdelava.....	44
4.1.6 Primer	45
4.2 Anchor modeling	46
4.2.1 Uvod	46
4.2.2 Gradniki	47

4.2.3	Smernice pri načrtovanju	49
4.2.4	Polnjenje in zmogljivosti	49
4.2.5	Primer.....	50
4.2.6	Uporaba podatkov	51
4.2.7	Prednosti in slabosti	51
4.3	Kakovost podatkov.....	53
4.3.1	Uvod.....	53
4.3.2	Shema za spremljanje napak	53
4.3.3	Revizijska dimenzija	55
4.4	Vnaprej pripravljeni modeli	56
5	Primerjava pristopov	57
5.1	Uvod	57
5.2	Ohranjanje zgodovine.....	59
5.3	Primernost za vrste podatkovnih skladišč	59
5.4	Primernost za dele podatkovnega skladišča	60
5.5	Vzdrževanje.....	61
5.6	Uporaba	62
5.7	Sprejetost	62
5.8	Upravljanje s hierarhijami	62
6	Primer rešitve v telekomunikacijski panogi	63
6.1	Uvod	63
6.2	Informacijske potrebe.....	63
6.2.1	Uvod.....	63
6.2.2	Ključni kazalci poslovanja	64
6.4	Metodologija	65
6.4.1	Uvod.....	65
6.4.2	Relacijski modeli.....	66
6.4.3	Večdimenzijski modeli	67
6.5	Arhitektura.....	68
6.6	Celovito podatkovno skladišče.....	69
6.6.1	Uporabniki	69
6.6.2	Uporaba storitev	71
6.6.3	Glavna knjiga	73
6.6.4	Kakovost podatkov	74
6.7	Področna podatkovna skladišča.....	75
6.7.1	Uporabniki	75
6.7.2	Uporaba storitev	77
6.7.3	Glavna knjiga	78
7	Sklep.....	80
8	Priloga.....	82
8.4	Primer rešitve, stavki SQL	82
8.4.1	Tabela AT_DATUM_SKLENITVE	82
8.4.2	Tabela AT_NAZIV	82
8.4.3	Tabela AT_STEVILKA_POGODBE	82
8.4.4	Tabela AT_TELEFONSKA_STEVILKA	82
8.4.5	Tabela AT_TIP	82
8.4.6	Tabela AT_UPORABNIK_PAKET	83
8.4.7	Tabela AT_UPORABNIK_STATUS	83
8.4.8	Tabela AT_UPORABNIK_TIP	83
8.4.9	Tabela DE_GLAVNA_KNJIGA	83

8.4.10	Tabela DE_UPORABA_STORITEV.....	83
8.4.11	Tabela DIM_CAS.....	84
8.4.12	Tabela DIM_DAN.....	84
8.4.13	Tabela DIM_DRUGA_STRANKA create table DIM_DRUGA_STRANKA ..	84
8.4.14	Tabela DIM_KONTO.....	85
8.4.15	Tabela DIM_MESEC	85
8.4.16	Tabela DIM_OBDELAVA.....	85
8.4.17	Tabela DIM_OMREZJE	85
8.4.18	Tabela DIM_ORGANIZACIJSKA_ENOTA.....	85
8.4.19	Tabela DIM_POGODBA	86
8.4.20	Tabela DIM_PREVERJANJE.....	86
8.4.21	Tabela DIM_SMER.....	86
8.4.22	Tabela DIM_STORITEV	86
8.4.23	Tabela DIM_STROSKOVNI_NOSILEC	87
8.4.24	Tabela DIM_TIP_POSTAVKE.....	87
8.4.25	Tabela DIM_UPORABNIK	87
8.4.26	Tabela DE_NAPAKA	87
8.4.27	Tabela DE_NAPAKA_PODROBNOST.....	87
8.4.28	Tabela POVEZAVA_HIERARHIJA_OE.....	88
8.4.29	Tabela POVEZAVA_HIERARHIJA_KONTO	88
8.4.30	Tabela SI_POGODBA	88
8.4.31	Tabela SI_UPORABNIK	88
8.4.32	Tabela VE_UPORABNIK_POGODBA	88
8.4.33	Tabela VO_PAKET.....	89
8.4.34	Tabela VO_STATUS	89
8.4.35	Materializiran pogled MV_UPORABA_STORITEV_MESEC	89
9	Viri in literatura	90

Kazalo slik

Slika 1: Primer arhitekture podatkovnega skladišča.....	10
Slika 2: Primer umestitve podatkovnega skladišča	12
Slika 3: Korporativno podatkovno skladišče.....	14
Slika 4: Dimenzijsko podatkovno skladišče.....	15
Slika 5: Podatkovno skladišče kot skupek področnih podatkovnih skladišč.....	16
Slika 6: Aktivno podatkovno skladišče	17
Slika 7: Federativno podatkovno skladišče	18
Slika 8: Prehajanje podatkov med področji v DW 2.0	23
Slika 9: Zvezdna shema in snežinka.....	25
Slika 10: Uporaba materializiranih pogledov	35
Slika 11: Povezanost relacijskega in večdimenzijskega modela	41
Slika 12: DW 2.0, trije nivoji podatkovnega modela [22]	42
Slika 13: DW 2.0, popoln podatkovni model [22]	44
Slika 14: Primer razvoja podatkovnega modela v ogrodju DW 2.0 za področje uporabnika ..	46
Slika 15: Osnovni gradniki AM	48
Slika 16: Podatkovni model, notacija AM	50
Slika 17: Relacijski podatkovni model.....	51
Slika 18: Shema za spremljanje napak	54
Slika 19: Revizijska dimenzija v povezavi s tabelo dejstev	55
Slika 20: Arhitektura	68
Slika 21: Podatkovni model za področje uporabnikov, notacija AM	70
Slika 22: Relacijski model AM za področje naročnikov.....	71
Slika 23: Zvezdna shema za spremljanje uporabe storitev	72
Slika 24: Zvezdna shema za spremljanje podatkov v glavni knjigi	73
Slika 25: Zvezdna shema za spremljanje kakovosti podatkov	75
Slika 26: Dimenzijski tabeli za področje uporabnikov.....	76
Slika 27: Zvezdna shema za spremljanje uporabe storitev na mesečni ravni.....	78

Kazalo preglednic

Preglednica 1: Primerjava vrst podatkovnih skladišč.....	20
Preglednica 2: Počasi spreminjajoče se dimenzije, tip 2	33
Preglednica 3: Počasi spreminjajoče se dimenzije, tip 3	33
Preglednica 4: Kocke in relacijski dimenzijski modeli, prednosti	39
Preglednica 5: Primerjava pristopov.....	59
Preglednica 6: Preslikava modela AM v dimenzijske tabele	76

Seznam kratic

AM	<i>Anchor Modelling</i> Pristop k načrtovanju podatkovnih modelov
ARPU	<i>Average Revenue per User</i> Povprečni dohodek na uporabnika
CDR	<i>Call Detail Record</i> Zapis o klicu
CPM	<i>Corporate performance management</i> Upravljanje učinkovitosti poslovanja
CRM	<i>Customer relationship management</i> Upravljanje odnosov s strankami
DIS	<i>Data Item Set</i> Množica podatkovnih enot, nivo podatkovnega modela v ogrodju DW 2.0
ERD	<i>Entity Relationship Diagram</i> Entitetno relacijski model, nivo podatkovnega modela v ogrodju DW 2.0
ETL	<i>Extract, Transform, Load</i> Prenos podatkov v podatkovno skladišče
IPIS	Integriran poslovno informacijski sistem
KKP	Ključni kazalec poslovanja <i>Key Performance Indicator, KPI</i>
MDX	<i>Multidimensional Expressions</i> Poizvedovalni jezik za večdimenzijso podatkovno bazo
MMS	<i>Multimedia Messaging Service</i> Večpredstavno sporočilo
SOA	<i>Service oriented architecture</i> Storitveno usmerjena arhitektura
SMS	<i>Short Message Service</i> Kratko sporočilo
SUPB	Sistem za upravljanje s podatkovno bazo <i>Database management system, DBMS</i>

Povzetek

V magistrskem delu so predstavljena podatkovna skladišča in pristopi k njihovem načrtovanju. Na eni strani so še vedno aktualni klasični pristopi, ki se uporabljajo že skoraj dve desetletji, na drugi strani pa tisti, ki šele prihajajo, se sicer že uporabljajo, a so še vedno v fazi intenzivnega razvoja.

Predstavljena je tipična arhitektura podatkovnega skladišča in njegova umestitev v sodobno informacijsko okolje. Sledi opis različnih vrst podatkovnih skladišč, med katerimi je narejena primerjava. Predstavljeno je ogrodje za naslednjo generacijo podatkovnih skladišč, DW 2.0.

Dimenzijski modeli so danes med najbolj uporabljenimi podatkovnimi modeli. Podrobneje je opisan način takšnega modeliranja v relacijskih podatkovnih bazah. Uporaba materializiranih pogledov je predstavljena kot tehnika za izboljšanje zmogljivosti. Večdimenzionalne baze postajajo vse boljše, zato se kocke kot njihov osnovni gradnik vedno bolj uporabljajo. Poleg hitrosti so predstavljene njihove prednosti na področju napredne obravnave hierarhij.

Ogrodje DW 2.0 predvideva izgradnjo podatkovnega modela na treh nivojih, pri čemer začnemo s celovitim in manj podrobnim in zaključimo s podrobnim fizičnim modelom. *Anchor modeling* je eden od perspektivnih sodobnih pristopov k načrtovanju podatkovnega modela, ki si šele utira pot v podatkovna skladišča. Temelji na šesti normalni obliki.

Predstavljene so enostavne in pregledne strukture za zagotavljanje kakovosti podatkov, ki jih polnijo postopki *ETL*.

Posebno poglavje je namenjeno podrobni primerjavi pristopov glede na lastnosti, kot so primernost za vrste podatkovnih skladišč, zahtevnosti vzdrževanja in uporabe, sprejetost, ohranjanje zgodovine in drugih. Poglavje s primerom rešitve vsebuje praktični prikaz uporabe opisanih tehnik na primeru podatkovnega skladišča za telekomunikacijsko podjetje. Med opisanimi pristopi je, upoštevajoč njihove prednosti in slabosti ter potrebe takšnega podjetja, izbranih nekaj najprimernejših, ki so uporabljeni za izdelavo podatkovnega modela.

Ključne besede: podatkovno skladišče, podatkovni model

Abstract

In this Master's thesis data warehouses and approaches to their design are presented. On one hand, classical approaches, which have been in use for almost two decades, are still popular; on the other hand, new approaches are on the horizon, actually being used already, yet still undergoing an intensive development phase.

A typical data warehouse architecture and its placement in the modern information environment is shown, followed by a description of a variety of data warehouses and the comparison between them. The framework for the next data warehouse generation DW 2.0 is presented.

Dimensional models rank among the most widely applied data models. A detailed description of the way of such modeling in relational databases is shown. The use of materialized views is shown as a technique for performance improvement. Multidimensional databases are improving constantly, therefore the cubes as their basic component are more and more widely used. In addition to speed, their advantages in the field of advanced handling of hierarchies are presented.

DW 2.0 framework foresees the building of data model on three levels, starting with an overall less detailed model and concluding with a detailed physical model. Anchor modeling represents a prospective modern approach to the data model design which is paving its way into data warehouses. It is based on the sixth normal form.

Simple and clear structures for ensuring data quality, loaded by ETL processes are presented.

A special chapter deals with a detailed comparison of approaches regarding the properties such as suitability to the types of data warehouses, ease of maintenance and usage, acceptance, history conservation etc. Presenting a solution example, another chapter includes a practical presentation of the use of the described approaches on an example of a data warehouse for a telecommunication company. Considering their advantages and disadvantages as well as the needs of such a company, among the described the most suitable approaches have been selected to be used for the development of the data model.

Keywords: data warehouse, data model

1 Uvod

V sodobnih organizacijah se oddelki informatike soočajo z vedno bolj zahtevno nalogo oskrbovanja uporabnikov z informacijami, za kar je zadolžen sistem za poslovno obveščanje. Le-ta obsega širok nabor aplikacij in tehnologij za zbiranje, shranjevanje, analiziranje in zagotavljanje dostopa do podatkov za poslovne uporabnike, katerim se na ta način omogoči sprejemanje boljših odločitev. Pri tem ne gre le za premikanje in shranjevanje podatkov, ampak za sistematičen in strukturiran pristop z vidika celotne organizacije. Omogočiti mora dodano vrednost pri preoblikovanju podatkov v informacije in njihovo dostopnost za odločevalce na način, da jih lahko ti optimalno uporabijo. Najbolj uspešen je, kadar je sistematično povezan z dejavnostjo in poslovanjem organizacije [14].

Poslovno obveščanje omogoča hitrejši in bolj učinkovit dostop do podatkov. Uporabnikom daje odgovore na različne tipe vprašanj. Osnovni tip je povezan s tem, kaj se je zgodilo. Takšni odgovori so enostavno merljivi. Napredna uporaba da odgovore na bolj zahtevna vprašanja, kot sta zakaj se je nekaj zgodilo, in celo kaj se bo zgodilo. Taka uporaba lahko pomaga pri strateških odločitvah, kot je na primer odločitev za vstop na novo tržišče ali vpeljava novega produkta.

Razvoj poslovnega obveščanja se nadaljuje. Pojavljajo se novi trendi, kot so zajem podatkov v realnem času in spremljanje učinkovitosti poslovanja (*corporate performance management, CPM*) [17]. Tudi tehnologije se razvijajo. Tak primer so vedno boljši sistemi za upravljanje s podatkovnimi bazami, tako relacijskimi kot večdimenzijskimi.

Podatkovno skladišče je osrednji del sistema za poslovno obveščanje in kot tako podvrženo razvoju in spremembam v okolju. Gre za shrambo zgodovinskih, povezanih in konsistentnih podatkov [2]. Pojavljati so se začela kot odgovor na vedno večja pričakovanja uporabnikov. S tem imenom obstajajo že več kot dve desetletji, toliko časa pa se razvija tudi teorija za njihovo izgradnjo, pri čemer sta v ospredju Willim H. Inmon in Ralph Kimball, ki sta avtorja številnih knjig s tega področja [4], [5], [7]–[9]. Organizacija, ki je na tem področju zelo dejavna je TDWI (*The Data Warehouse Institute*).

Načrtovanje podatkovnega modela je eden večjih izzivov podatkovnega skladiščenja. Pri tem so se uveljavili določeni pristopi. Pogosto se uporablja dimenzijski model. Pristopi se razlikujejo od klasičnih relacijskih, ki jih uporabljajo transakcijski sistemi, saj je način uporabe podatkov v analitičnem okolju drugačen.

Ti pristopi se pod vplivom razvoja informacijske tehnologije in sprememb v okolju razvijajo, pojavljajo pa se tudi novi. Namen tega dela je zbrati in predstaviti pristope pri načrtovanju

podatkovnih modelov za podatkovna skladišča, s poudarkom na sodobnejših, torej tistih, ki so se začeli v pojavljati in širše uporabljati v zadnjem desetletju, ter tako zajeti trenutno stanje na tem področju.

Večdimenzijske podatkovne baze so vedno bolj aktualne. Njihova prednost so vnaprej izračunani podatki na različnih nivojih dimenzij in možnost poizvedb preko prav za analize namenjenih poizvedovalnih jezikov. Za razliko od relacijskih podatkovnih baz, ki so si med seboj podobne, je med njimi precej razlik, saj obstaja malo skupnih standardov. Z njihovim prihodom se je pojavila tudi potreba po povezavi podatkov, shranjenih v dveh različnih tipih podatkovnih baz. Hibridni pristop mora premostiti razlike med obema svetovoma, povezovanje pa lahko poteka na različne načine.

Primer alternativnega pristopa pri načrtovanju podatkovnega modela predstavlja *anchor modeling (AM)* [11]. Temelji na predpostavki, da se okolje, ki vpliva na podatkovno skladišče, neprestano spreminja. Velika sprememba izven modela se bo v modelu odrazila kot majhna sprememba. Pristop vključuje naravne lastnosti objektov, atributov in povezav, zaradi česar je model razumljiv in enostaven za uporabo. Temelji na šesti normalni obliki, zato je zelo razčlenjen in odporen na spremembe v okolju. Razčlenjenost omogoča določene prednosti v primerjavi s tradicionalnimi modeli, ima pa tudi nekatere slabosti.

William H. Inmon je leta 2008 izdal knjigo [5], v kateri predstavlja DW 2.0, arhitekturo za naslednjo generacijo podatkovnega skladiščenja. Izgradnja podatkovnega modela predstavlja jedro načrtovanja za to arhitekturo. Poteka preko treh različnih nivojev v odvisnosti od podrobnosti podatkov [22]. Vsak od teh nivojev predstavlja različen pogled na podatke.

Zagotavljanje kakovosti podatkov je pomembna naloga, predvsem pri izgradnji postopkov za prenos podatkov v podatkovno skladišče. Podatkovni model lahko omogoča spremljanje kakovosti vsakega zapisa v podatkovnem skladišču, lahko pa se zgradijo celo specializirana podatkovna skladišča, ki nam pomagajo odkrivati probleme s kakovostjo [7].

Hitrost poizvedb nad podatkovnim skladiščem je odvisna od podatkovnega modela. V relacijske podatkovne baze so vgrajeni posebni mehanizmi, ki ob izgradnji določenih struktur omogočajo, da se poizvedba izvede hitreje, kot bi se, če bi jo sistem za upravljanje s podatkovno bazo izvedel nad atomarnimi podatki.

V delu je na enem mestu zbran nabor pristopov pri načrtovanju podatkovnih modelov za podatkovna skladišča, njihovo vrednotenje in primerjava. V taki obliki lahko delo služi kot

pomoč načrtovalcem pri izbiri pristopa ob načrtovanju novega ali nadgradnji obstoječega podatkovnega skladišča.

2 Podatkovna skladišča

2.1 Osnovna opredelitev

Podatkovna skladišča danes predstavljajo pomemben del informacijske infrastrukture. Nastala so kot odziv na povečane potrebe po informacijah znotraj organizacij. Njihov glavni namen je zagotavljanje informacij. Kot taka, dopolnjena z orodji za dostop do ustreznih informacij, zagotavljajo pomoč v postopkih odločanja.

Kljub njihovi razširjenosti obstaja kar nekaj različnih opredelitev, ki določajo, kaj predstavljajo in kakšna naj bi bila njihova arhitektura. V tem poglavju so zato predstavljene vrste podatkovnih skladišč, kot jih poznamo danes. Kljub vsem opredelitvam pa se najpogosteje navaja in uporablja izvirna, ki jo je v svoji knjigi *Building the Data Warehouse* [4] podal William H. Inmon.

Podatkovno skladišče je po tej opredelitvi:

- predmetno usmerjena (angl. *subject oriented*),
- povezana (angl. *integrated*),
- časovno spremenljiva (angl. *time-variant*),
- zbirka podatkov, ustvarjena z namenom podpore postopkom odločanja,
- ki ohranja zgodovino podatkov (angl. *nonvolatile*).

Podatki v njem morajo biti predmetno usmerjeni, kar pomeni, da so tisti, ki pripadajo določenemu predmetu ali objektu, z njim tudi povezani. Na podatke je treba gledati iz uporabnega vsebinskega in manj iz tehnološkega vidika. Podatki se skozi čas spreminjajo, a se zgodovina ohranja. Iz podatkovnega skladišča je mogoče v vsakem trenutku dobiti stanje v nekem trenutku v preteklosti. Vanj se polnijo podatki iz različnih izvornih informacijskih sistemov, ki so lahko zelo raznoliki. Zato morajo biti povezani v celoto, da lahko na njih gledamo na enoten način.

Povezovanje podatkov organizacijam zagotavlja enovit pogled na njih. Uporabniki podatkov imajo možnost, da namesto ločenih podmnožic vidijo celoto in to na način, kot da vsi podatki prihajajo iz istega izvora, kljub temu, da je osnovni izvor podatkov množica informacijskih sistemov. Ena glavnih prednosti podatkovnih skladišč je ravno ta pregled nad organizacijo. Druga prednost je ta, da predstavlja zgodovinsko shrambo podatkov, v kateri so ti shranjeni za več let v preteklost. Organizacije so zakonsko obvezane hraniti nekatere podatke, za kar jim ni potrebno vzdrževati dodatnih zbirk.

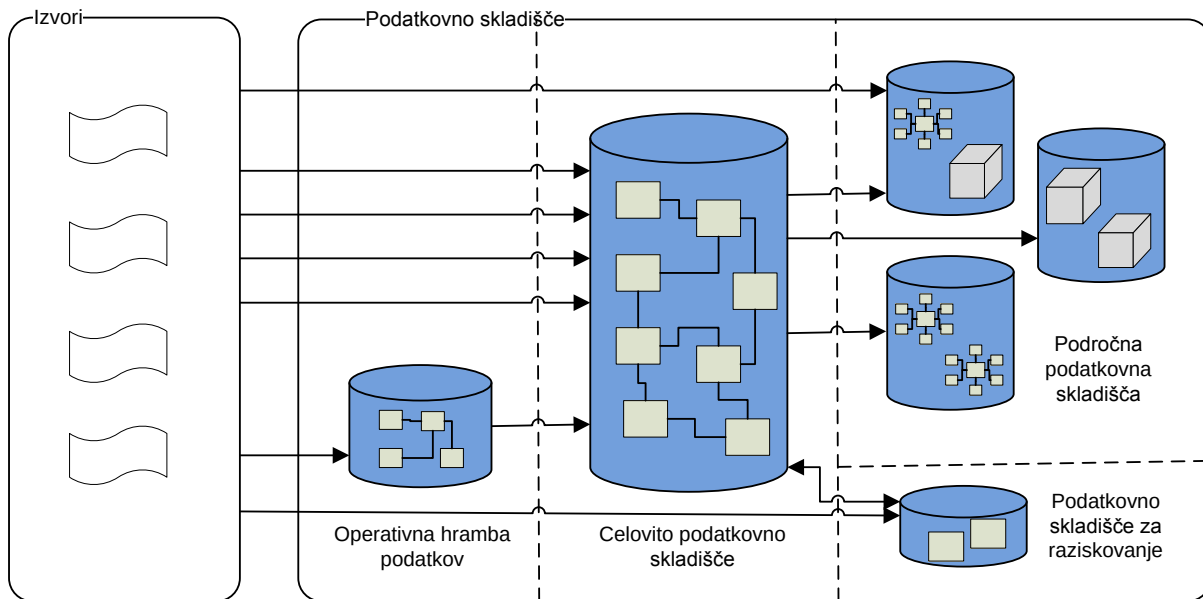
Podatki v podatkovnem skladišču naj bi bili, po mnenju glavnih teoretikov, kot sta omenjeni William H. Inmon in poleg njega Ralph Kimball, tudi na najnižjem nivoju podrobnosti. Ne vsi, saj jih je mogoče preoblikovati in združiti, morajo pa obstajati. Rezultat tega je večja prilagodljivost. Določena skupina uporabnikov jih lahko pregleduje glede na svoje potrebe, druga pa prav tako glede na svoje potrebe, ki pa niso enake kot pri prvi skupini. Tako lahko prodaja gleda na podatke na svoj način, proizvodnja pa na drugega. Osnova so v obeh primerih podatki na najnižjem nivoju podrobnosti, kar zagotavlja eno verzijo resnice. V primeru različnih pogledov se lahko ti vedno razčistijo s pomočjo teh podatkov.

Besedna zveza »podatkovno skladišče« se danes množično uporablja, vsa podatkovna skladišča pa ne sledijo petim točkam izvirne opredelitve. Predmetna usmerjenost in časovna spremenljivost sta lastna vsem, medtem ko ostale točke ne. Zgodovina se običajno ohranja, a le za določene lastnosti. Povezanost včasih ni potrebna, kjer je izvorni informacijski sistem le eden, prav tako pa obstajajo podatkovna skladišča, ki se sicer napajajo iz različnih sistemov, a so podatki v njih nepovezani. To, da s podatkovnim skladiščem ne mislimo na zbirko podatkov, je videti najbolj nenavadno. Do tega pride pri tako imenovanih federativnih podatkovnih skladiščih, kjer so določene le strukture, preko katerih se dostopa do podatkov.

Možna opredelitev podatkovnega skladišča bi tako bila: predmetno usmerjen in časovno spremenljiv vir podatkov, ustvarjen z namenom podpore postopkom odločanja. Namesto »zbirke podatkov« je uporabljena besedna zveza »vir podatkov«, saj pri federativnih podatkovnih skladiščih ne moremo govoriti o zbirkah podatkov.

2.2 Arhitektura

Tako kot obstaja več osnovnih vrst podatkovnih skladišč, tako poznamo več možnih arhitektur za njih. Na sliki 1 je prikazana najbolj tipična, kjer je podatkovno skladišče razdeljeno na več delov, od katerih lahko tudi vsak zase predstavlja podatkovno skladišče. Nenazadnje imajo nekateri od njih to tudi v imenu.



Slika 1: Primer arhitekture podatkovnega skladišča

Za prenos iz virov in med posameznimi deli se uporabljajo postopki za pridobivanje podatkov iz virov, (ang. *Extract*), njihovo preoblikovanje in čiščenje (angl. *Transform*), ter nalaganje v ponorne strukture (angl. *Load*). Imenujemo jih postopki *ETL*. Njihova izgradnja predstavlja enega večjih izzivov podatkovnega skladiščenja. Ti postopki pogosto uporabljajo začasne strukture in metapodatke, ki pa na sliki zaradi preglednosti niso prikazani.

Izvori so lahko heterogeni. Med njimi so informacijski sistemi, podatkovne baze vseh vrst, datoteke različnih oblik, spletna mesta, servisi ter multimedijски zapisi.

Celovito podatkovno skladišče vsebuje povezane podatke z vseh področij znotraj organizacije. Ustreza vsem petim točkam osnovne opredelitve. Podatki v njem so shranjeni v relacijskih tabelah in so na najnižjem nivoju podrobnosti. Njihova količina je tipično velika.

Področno podatkovno skladišče je področje, do katerega uporabniki neposredno dostopajo. Podatki v njem so namenjeni določenim skupinam, pogosto so to oddelki organizacije. Podatkovne strukture so narejene na način, da je uporaba podatkov enostavna. Oddelek finance ima svoje področno podatkovno skladišče, prav tako ga lahko imata proizvodnja in prodaja. Polni se neposredno iz virov ali preko celovitega podatkovnega skladišča. Glede na slednjega lahko uporablja drugačne tehnologije. Količina podatkov je manjša, ker so podatki združeni in seštet na višjih nivojih podrobnosti.

Operativna hramba podatkov je podobna transakcijskim sistemom, saj se nad njenimi podatki izvajajo spremembe (angl. *update*). Je hibridno okolje v katerem se podatki iz virov preoblikujejo in prečistijo v povezano obliko. To je področje, ki lahko služi kot vir za celovito

podatkovno skladišče. Pogosto se uporablja za dostop do podatkov v realnem času, pri čemer morajo biti temu prirejeni tudi postopki *ETL*.

Podatkovno skladišče za raziskovanje predstavlja pripomoček za napredne končne uporabnike, ki bi radi odkrivali nova znanja s postopki podatkovnega rudarjenja (angl. *data-mining*). Za zapletene statistične analize se nad podatki uporabljajo hevristične metode. Oblika podatkov je temu prilagojena. Pripravijo se za potrebe posameznega projekta. Ko je leta zaključen, se vsebina izbriše. Rezultati se lahko prenesejo v celovito podatkovno skladišče.

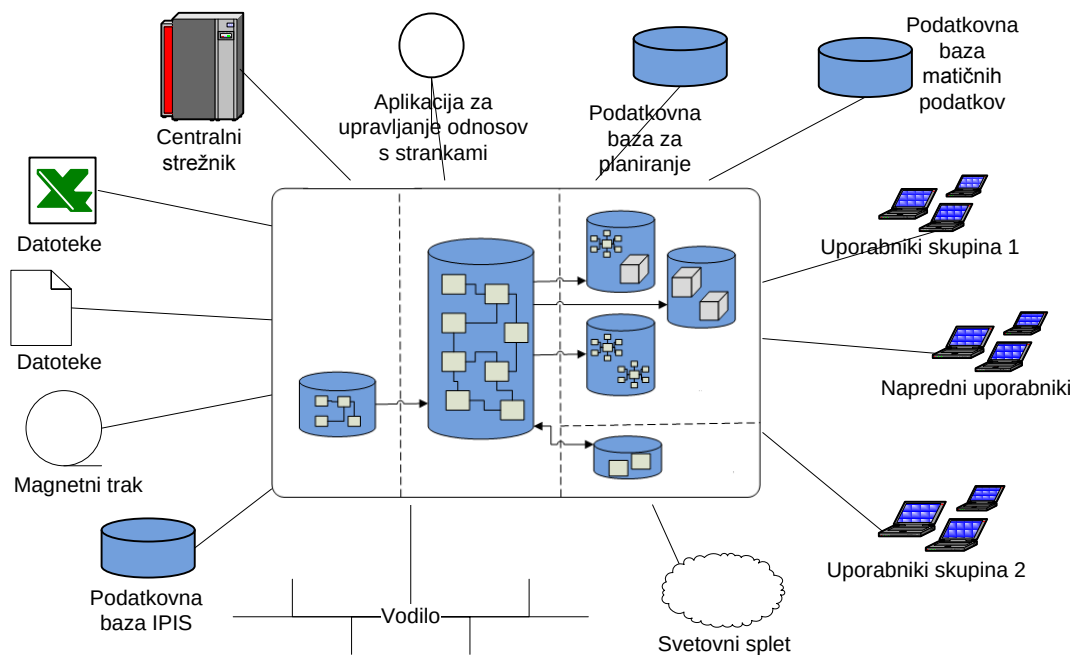
Ena od razlik med celovitim in področnim podatkovnim skladiščem je v nivoju podrobnosti podatkov. Ta nivo je v prvem primeru najnižji, v drugem pa to ni obvezno. To pomeni naslednje:

- najnižji nivo se osvežuje pogosteje,
- količina podatkov na najnižjem nivoju je večja,
- podatkovnih struktur na višjih nivojih je več in
- na višjih nivojih so podana pravila za združevanje.

Zaradi teh razlik je podatkovno skladišče pogosto zgrajeno iz več delov. En del je celovito podatkovno skladišče, potrebno za podporo odločanju pri delovanju celotne organizacije, drugi del pa je uporabniški. Ta nudi uporabnikom preprost vpogled v podatke in jim nudi možnost izvajanja učinkovitih analiz.

2.3 Sodobna umestitev

Podatkovno skladišče kot osrednji del sistema za poslovno obveščanje je podvrženo razvoju in spremembam v okolju. Kot je prikazano na sliki 2, je povezano z večino informacijskih sistemov organizacije. To pomeni, da sprememba v enem od njih posledično vpliva na spremembo v skladišču. Povezava med sistemi poteka preko različnih oblik, kot so datoteke vseh vrst, programski vmesniki, podatkovne baze, servisi, za določene podatke so to magnetni trakovi.



Slika 2: Primer umestitve podatkovnega skladišča

Povezave na sliki namenoma niso usmerjene. Podatkovno skladišče ima vpliv na okolje, ki je lahko neposreden, kadar je podatkovni tok usmerjen neposredno iz skladišča, ali posreden, kadar podatki vplivajo na odločitve uporabnikov in posredno na podatke izvornih sistemov, ki tako postanejo tudi ponorni.

Največji izvor pogosto predstavlja integriran poslovno informacijski sistem organizacije (IPIS, angl. *Enterprise resource planning*, *ERP*). V nekaterih primerih to vlogo prevzame sistem za zaračunavanje.

Starejši podatki, shranjeni na magnetnih trakovih, se prenašajo v podatkovno skladišče. Smer je obratna, kadar se izdelujejo varnostne kopije.

Aplikacija za upravljanje odnosov s strankami (angl. *customer relationship management*, *CRM*) je usmerjena k stranki, prav tako podatki v skladišču. Zato je smiselna tesna povezava med obema sistemoma. Na sliki je predstavljena povezava preko programskih vmesnikov. V praksi se za osnovni vir podatkov o uporabniku, iz katerega drugi le črpa, uporablja, odvisno od rešitve, ali eden ali drugi.

Podatkovna baza za planiranje je prav tako tesno povezana s podatki v skladišču, saj se izvaja planiranje za področja, ki imajo svoje mesto tudi v skladišču. Pri tem je zanimivo, da se danes za planiranje vedno bolj uporabljajo večdimenzijske podatkovne baze, v katerih so podatki organizirani dimenzijsko, podobno kakor v skladišču.

Podatkovna baza matičnih podatkov je v primeru na sliki narisana kot ločena hramba podatkov, lahko pa je del podatkovnega skladišča, s katerim upravlja sistem za upravljanje matičnih podatkov (angl. *master data management*, *MDM*).

V datotekah različnih oblik se nahajajo strukturirani ali nestrukturirani podatki. V zadnjih letih se krepi zavedanje o pomembnosti slednjih, saj naj bi bilo takšnih kar 80% [5]. Tehnologija, ki se uporablja danes, je prilagojena predvsem strukturiranim oblikam. Zato se za odločanje ne uporabljajo pomembne informacije, ki so shranjene v na primer zvočnih in tekstovnih zapisih, kot so pritožbe in pohvale strank.

Skladišče je lahko povezano z vodilom storitveno usmerjene arhitekture (angl. *service oriented architecture*, *SOA*). Pri tem velja upoštevati, da je *SOA* arhitektura s transakcijsko naravo [6]. Skladišče, na drugi strani, ni transakcijsko naravnano. Postopki *ETL* zato predvsem zaradi količine naenkrat prenesenih podatkov niso primerni za izvajanje v okolju *SOA*. Lahko pa podatkovno skladišče nudi informacije na zahtevo, dostopne vsem povezanim sistemom.

2.4 Vrste

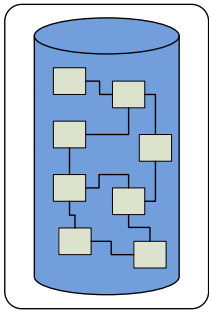
2.4.1 Uvod

V nadaljevanju so opisane osnovne vrste podatkovnih skladišč kot jih poznamo danes. Med njimi ne zadostijo vse vsem točkam osnovne opredelitve. Vzrokov za takšno heterogenost je več:

- Teorija se je razvijala neodvisno v več smereh, s strani več avtorjev. Willima H. Inmon je v osnovi zagovarjal podatkovno skladišče kot korporativno podatkovno skladišče. Vzporedno z njim je Ralph Kimball zagovarjal podatkovno skladišče kot dimenzijsko podatkovno skladišče.
- Pojavile so se nove zahteve. Tak primer je podatkovno skladiščenje v realnem času, ki zahteva drugačen model, ki mora dovoljevati spremembo podatkov.
- Ponudniki programske opreme so spoznali, da so podatkovna skladišča v vzponu in da morajo imeti na tem področju svojo rešitev. Tako so podatkovno skladišče prilagodili svojim potrebam, čeprav je bila osnovna ideja podatkovnih skladišč drugačna.

2.4.2 Korporativno podatkovno skladišče

Ta vrsta podatkovnega skladišča je izvirna in zadosti vsem točkam osnovne definicije. Poleg tega predvideva, da so podatki na najnižjem možnem nivoju podrobnosti, na kakršnem so tudi v izvornem sistemu. Utemeljitelj te vrste je William H. Inmon [5].



Slika 3: Korporativno podatkovno skladišče

Gre za osrednje skladišče, ki s podatki oskrbuje celotno organizacijo. Vsebuje eno in edino verzijo resnice. Njegov glavni cilj je zagotavljanje enotne predstavitve tako podatkov organizacije kot tudi pripadajočih pravil. Zaradi števila in različnosti sistemov ter »silosov« s podatki, ki obstajajo znotraj večjih organizacij, se mnogokatero skladišče ne uvršča v to kategorijo. Povezave med strukturami na sliki 3 kažejo na povezanost vseh podatkov in odsotnost silosov.

Sestavljeno mora biti iz več predmetno usmerjenih področij. Povezanost področij prodaje, nabave, financ, proizvodnje in drugih zagotavlja eno in edino verzijo resnice. Če bi bili podatki vsakega področja ločeni, to ne bi bilo mogoče.

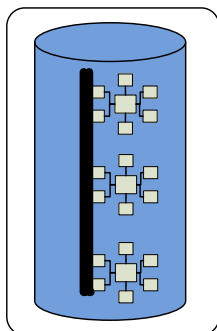
Podatkovni model mora biti normaliziran z uporabo tretje normalne oblike. Vzdrževanje je enostavno, model je prilagodljiv, na drugi strani pa manj primeren za neposredno izkoriščanje s strani uporabnikov.

Ena od značilnosti te vrste podatkovnih skladišč je razširljivost. Organizacije rastejo, z njimi raste količina njihovih podatkov. Povečuje se lahko tudi zapletenost poslovnih pravil in postopkov v organizaciji. Skladišče mora biti zasnovano tako, da je na to pripravljeno in da raste skupaj z organizacijo.

Slabost tovrstnih skladišč je v tem, da oblika podatkov ni najprimernejša za neposredno uporabo. Vzroka za to sta normaliziran podatkovni model, neprilagojen za uporabo s strani določenih skupin uporabnikov in velika količina podatkov, ki upočasni dostop do njih in naredi njihovo vzdrževanje zahtevnejše.

2.4.3 Dimenzijsko podatkovno skladišče

Ta vrsta podatkovnega skladišča zadosti vsem točkam osnovne definicije, razen v določenih primerih, ko ne ohranja zgodovine podatkov. Predvideva tudi, da so podatki na najnižjem možnem nivoju podrobnosti, na kakršnem so tudi v izvirnem sistemu. Utemeljitelj te vrste je Ralph Kimball [9].



Slika 4: Dimenzijsko podatkovno skladišče

Veliko je podobnosti s korporativnim podatkovnim skladiščem, glavna razlika pa je v podatkovnem modelu. Podatki so organizirani dimenzijsko, z določenimi tabelami v drugi in določenimi v tretji normalni obliki. Pogosto so primerni za neposredno uporabo, čeprav lahko njihova količina povzroča težave. Tabele z največjo količino podatkov so normalizirane z uporabo tretje normalne oblike, tiste z manjšo količino pa z uporabo druge. Prispevek slednjih k celotni količini podatkov je majhen, zato manjša normalizacija ni problematična.

Na sliki 4 je s tremi »zvezdami« prikazana dimenzijska organiziranost podatkov, ki pa so kljub temu povezani. Povezanost se doseže preko vodila, ki je na sliki prikazano s črno navpično črto, ki povezuje tri ločene strukture. Takšen tip skladišča torej ne predvideva »silosov« s podatki.

Dimenzijska skladišča podatke predstavljajo v obliki mer in dimenzij, ki so za uporabnike enostavno razumljive, s čimer je omogočena lažja uporaba podatkov. Vsakega od poslovnih procesov je mogoče zmodelirati na ta način. Glavna prednost je torej razumljivost podatkovnih struktur, pri čemer skladišče še vedno zagotavlja vse lastnosti osnovne opredelitve.

Poleg tega naj bi takšno skladišče omogočalo hitre rezultate [16], ker naj bi bilo zgrajeno za določene uporabniške potrebe, tipično s področja prodaje ali marketinga. To drži v primeru, kadar se uporabniške potrebe nanašajo na posamezen dogodek, ki ga podpremo z enim dimenzijskim modelom. Ta se namreč ne gradi za namene poročila, ampak za posamezen dogodek, ki odgovarja poslovnim procesom.

Slaba stran te vrste skladišč naj bi bila, kot trdijo nasprotniki [5], prirejenost modela za določeno vrsto uporabnikov. Za slednje je model optimalen, za ostale pa je slabši in s tem ne zadovolji informacijskih potreb vseh. Zagovorniki pravijo [8], da naj bi dimenzijski modeli temeljili na poslovnih procesih, kot so naročila, klici, računi, in manj na vrstah uporabnikov ali oddelkih. Različne poslovne funkcije imajo potrebo po analiziranju iste metrike v zvezi z

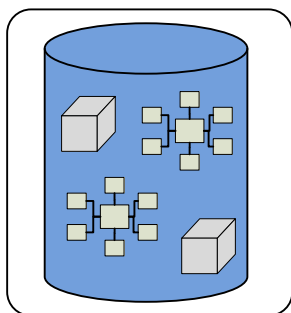
nekim poslovnim procesom. Pri izgradnji modela si je potrebno prizadevati, da je podvajanj osnovnih mer čim manj.

Druga slaba stran naj bi bila, da ta vrsta skladišč ni odporna na spremembe. Dokler zahteve ostajajo enake, je vse v redu, ko pa se tekom časa spremenijo, kar je pogosto, so potrebne velike spremembe ali pa celo zavrženje in izdelave novega dimenzijskega modela. Tudi tu zagovorniki ponujajo odgovor. Dimenzijske strukture naj bi bile prilagodljive in odporne na spremembe, saj so zrna tabel dejstev na najnižjem nivoju podrobnosti. Če to ne bi držalo, model res ne bi bil prilagodljiv.

Korporativno in dimenzijsko podatkovno skladišče imata veliko skupnih točk, največja razlika pa je v normaliziranosti podatkovnega modela. »Rivalstvo« med obema vrstama je sicer veliko, obstajajo goreči privrženci tako ene kot druge struje, kar je posledica tega, da sta zagovornika ene in druge strani glavna teoretika na področju podatkovnih skladišč.

2.4.4 Podatkovno skladišče kot skupek področnih podatkovnih skladišč

Ta vrsta podatkovnih skladišč ima podatke organizirane tako, da je njihova uporaba enostavna. Zato so pogosto organizirani dimenzijsko, zrno pa ni na najnižjem možnem nivoju, po čemer se ločijo od dimenzijskih. Povezanost in ohranjanje zgodovine nista obvezna. Na sliki 5 so prikazana med seboj neodvisna področna podatkovna skladišča, ki skupaj tvorijo podatkovno skladišče. Različni strukturi kažeta na možno uporabo ali relacijskih ali večdimenzijskih SUPB.



Slika 5: Podatkovno skladišče kot skupek področnih podatkovnih skladišč

Razširjenost te vrste pospešujejo ponudniki programske opreme, ki s takim pristopom nadgradijo svojo transakcijsko rešitev in ji dodajo analitične zmožnosti. Na ta način je rezultate mogoče doseči hitro, brez povezovanja podatkov med različnimi področji znotraj organizacije. Količina podatkov je majhna, saj so podatki že združeni in seštet na višje nivoje podrobnosti.

Na drugi strani se je potrebno pri uporabi te vrste zavedati nekaterih slabosti. Podatki posameznih področnih podatkovnih skladišč med sabo niso združljivi, kar lahko pripelje do

tega, da ni ene in edine verzije resnice. Tako lahko vsaka skupina uporabnikov na isto vprašanje poda različne odgovore. Ugotavljanje, zakaj so odgovori različni in kateri je pravi, pa je zahtevna naloga.

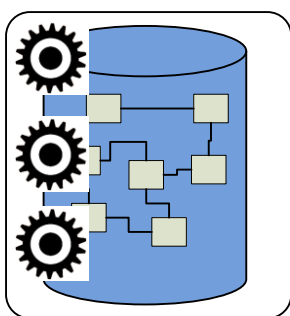
Ob pojavitvi novih zahtev je potrebno zgraditi novo področno podatkovno skladišče, kar vodi do tega, da začne njihovo število naraščati. Ob določenih spremembah jih je zato potrebno spremeniti več ali celo vsa. Več kot jih je, več dela je potrebnega in večja je verjetnost napak. Za vzdrževanja takšnega skladišča je zato potrebnega veliko časa in discipline.

Pogosto se uporabljajo večdimenzijske podatkovne baze, katere ne omogočajo ohranjanje zgodovine, ali pa je za ohranjanje potrebno uporabiti obvozne rešitve.

Takšnega skladišča ni mogoče spremeniti v celovito podatkovno skladišče, saj zrna v njegovih dimenzijskih modelih tega običajno ne omogočajo. Primerno je predvsem za manjše in srednje velike organizacije, ki ne spadajo med hitro rastoče in hitro spreminjajoče.

2.4.5 Aktivno podatkovno skladišče

Tradicionalno polnjenje podatkov v podatkovno skladišče se izvaja v rednih časovnih intervalih na paketni način, kar pomeni, da so podatki dostopni z zamikom glede na nastanek v izvornih sistemih. Določene, predvsem operativne odločitve v organizaciji, zahtevajo podatke v realnem času, ali, če že ne v realnem, pa vsaj v pravem času. To pomeni, da morajo biti podatki na voljo takrat, ko bi jih uporabniki želeli uporabiti. Aktivno podatkovno skladišče je skladišče, ki se polni tako pogosto kot je potrebno in možno, ter na ta način zadosti uporabniškim potrebam. Na sliki 6 je s kolesci prikazano neprestano spreminjanje podatkov.



Slika 6: Aktivno podatkovno skladišče

Takšno skladišče med točkami osnovne opredelitve podatkovnega skladišča pogosto ne zadosti zahtevama po ohranjanju zgodovine podatkov in povezanosti. Pogosta operacija nad takšnim skladiščem je sprememba zapisa, ki ne ohranja stare vrednosti. Zaradi kompleksnosti polnjenja so podatki posameznih področij med sabo pogosto nepovezani, saj je polnjenje enostavnejše, če ni povezovanja.

Podatkovni model je lahko dimenzijski ali ne, kar je v veliki meri odvisno od narave podatkov. Manj kompleksni podatki, kot je beleženje transakcij, omogočajo pri izbiri modela več možnosti, bolj kompleksni pa manj. Dimenzijski model je zato včasih prezahteven za določeno področje.

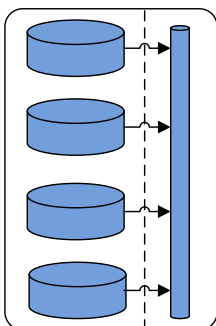
Takšno skladišče je zahtevno za vzdrževanje. Ob izvedbi napačne transakcije je potreben poseg. Takšne podatke je potrebno hitro najti in jih zbrisati ali popraviti, saj se transakcije neprestano izvajajo, ena napaka pa se lahko razmnoži, če posledic ne odpravimo dovolj hitro.

Zaradi kompleksnosti zahteva zmogljivo strojno in programsko opremo. Zmogljivosti te so izkoriščene med konicami polnjenja in izkoriščanja podatkov, preostali čas pa ne. Težava pri tem je, da sta obe konici v istem času, običajno med delovnimi urami zaposlenih. Posledica tega je visoka cena vzdrževanja.

Ta vrsta je zanimiva predvsem z vidika polnjenja, manj pa z vidika podatkovnega modeliranja. Z vidika slednjega jo lahko obravnavamo kot katero od drugih vrst skladišč, predvsem kot korporativno ali dimenzijsko podatkovno skladišče, ker ti dve vsebujeta podatke na najnižjem nivoju podrobnosti. Če so ti na višjih nivojih, je polnjenje še toliko bolj kompleksno. Podatkovno skladišče kot skupek področnih podatkovnih skladišč je zato manj primerno za polnjenje v realnem času.

2.4.6 Federativno podatkovno skladišče

Federativno podatkovno skladišče ni fizična zbirka podatkov. Gre za povezavo različnih, lahko heterogenih sistemov, tako analitičnih kot transakcijskih, med seboj na način, da so dostopni na povezan in enoten način. Povezava med podatki ni fizična, ampak logična. Desna stran slike 7 prikazuje takšno ogrodje, ki povezuje različne sisteme na levi strani. Uporabniki vidijo le enotno desno stran. Pri uporabi se jim ni potrebno zavedati arhitekture.



Slika 7: Federativno podatkovno skladišče

Glavna prednost tega pristopa je v tem, da fizično povezovanje podatkov ni potrebno. Prav povezovanje pa je ena obsežnejših nalog podatkovnega skladiščenja. Federativni pristop se zdi elegantna rešitev tega problema. Druga prednost je dostop do podatkov v realnem času.

A s tem pristopom so povezane tudi težave. Ena od njih so zmogljivosti. Podatkovna baza, ki nastopa kot izvorni sistem je lahko v trenutku dostopa obremenjena s transakcijami, ki jih nad njo izvaja transakcijski sistem. Lahko je sredi postopka izdelave varnostnih kopij ali pa je celo ugasnjena. Poleg tega je ta baza zelo obremenjena, saj vsaka poizvedba pomeni nov dostop, pa čeprav gre le za vrtanje v globino, nad enakim naborom izvornih podatkov.

Logično povezovanje zahteva kompleksne mehanizme, saj so izvori lahko heterogeni. Povezovanje mora na primer komunicirati s sistemi za upravljanje podatkovnih baz različnih proizvajalcev, ki med sabo niso kompatibilni. Tehnologija, ki jih poveže, zato ne more biti enostavna.

Ohranjanje zgodovine podatkov ni zagotovljeno. Zgodovine je le toliko, kolikor jo ohranjajo sami izvorni sistemi, ti pa pogosto niso namenjeni temu, saj bi hranjenje zgodovine omejilo njihove zmogljivosti.

Ta vrsta je primerna ob prevzemih in združitvah organizacij, ki so danes pogoste. Z organizacijami je potrebno združiti tudi njihova podatkovna skladišča in druge sisteme. Ob takšnih dogodkih so informacije hitro potrebne, prej, kot bi lahko zgradili novo fizično podatkovno skladišče. Če imajo organizacije pred tem že vsaka svoje, je federativni pristop smiseln, predvsem kratkoročno. Ta pristop je sicer smiseln vedno, kadar potrebujemo hitro in cenovno učinkovito rešitev. Čas za postavitev je med vsemi vrstami skladišč najkrajši.

2.4.7 Primerjava

Načrtovalec mora pred začetkom izgradnje podatkovnega skladišča sprejeti nekaj pomembnih odločitev. Glavna med njimi je izbira vrste, ki je lahko ena od osnovnih vrst ali kombinacija le-teh. Ta izbira je pomembna, ker je cena človeških in finančnih virov za izgradnjo običajno visoka. V primeru napačne odločitve je potrebno delo kasneje opraviti še enkrat, kar zahteva porabo dodatnih virov. Slaba izraba slednjih pa vodi v visoke stroške, katerim bi se organizacije rade izognile. Preglednica 1 ima v stolpcih vrste skladišč, v vrsticah pa lastnosti, na osnovi katerih se mora načrtovalec odločiti.

Vrsta podatkovnega skladišča / Lastnost	Korporativno	Dimenzijsko	Skupek področnih	Aktivno	Federativno
Predmetna usmerjenost	Da	Da	Da	Da	Da
Povezanost	Da	Da	Ni nujno	Ni nujno	Ni nujno
Časovna spremenljivost	Da	Da	Da	Da	Da
Ohranjanje zgodovine	Da	Da	Ni nujno (pogojeno tudi s tehnologijo)	Običajno ne (zaradi enostavnosti postopkov polnjenja)	Običajno ne (odvisno od izvornih sistemov)
Najnižji nivo podrobnosti podatkov	Da	Da	Običajno ne	Običajno da	Običajno da
Normaliziran model	Da	Ne	Običajno ne	Ni nujno	Običajno ne
Dimenzijski model	Ne	Da	Običajno da	Ni nujno	Običajno da
Zahtevnost izgradnje (5: najbolj zahtevna, 1: najmanj)	3	4	2	5	1
Zahtevnost vzdrževanja (5: najbolj zahtevna, 1: najmanj)	1	2	3	5	4
Količina podatkov (5: največ, 1: najmanj)	4	5	2	3	1
Fizična zbirka podatkov	Da	Da	Da	Da	Ne
Podatki v realnem času	Ni nujno	Ni nujno	Običajno ne	Da	Da

Preglednica 1: Primerjava vrst podatkovnih skladišč

Če določena lastnost vedno velja za posamezno vrsto, potem je označena z da. Če ne velja nikoli, je označena z ne. Nekatere lastnosti se lahko pri posamezni vrsti pojavljajo ali pa tudi ne. Te so ocenjene glede na verjetnost pojavljanja. Lestvica je naslednja: da – običajno da – ni nujno – običajno ne – ne. Zahtevnost izgradnje, vzdrževanja in količina podatkov sta ocenjeni od 1 do 5, pri čemer so vse vrednosti porazdeljene med vse vrste, ki so na ta način razvrščene

glede na lastnost. Podana je splošna ocena, vrstni red se lahko v določenih posebnih primerih tudi razlikuje.

Z zahtevnostjo izgradnje in vzdrževanja je mišljen celoten postopek, vključno z izgradnjo postopkov *ETL* in zagotavljanjem kakovosti podatkov. Večja zahtevnost pomeni daljši čas izgradnje, pri vzdrževanju pa večjo porabo časa. Aktivno skladišče je najbolj zahtevno z vidika obeh lastnosti, zato pa omogoča dostop do podatkov v realnem času. Federativno je najlažje za izgradnjo in najtežje za vzdrževanje. Dimenzijsko skladišče je glede na obe lastnosti bolj zahtevno od korporativnega, a ta razlika ni velika. Je posledica večjih razhajanj med modelom izvornega sistema in modela v skladišču pri dimenzijskih skladiščih. Pri primerjavi ni upoštevano dejstvo, da je verjetnost, da bo potrebno narediti dodatna področna podatkovna skladišča, večja pri korporativnem skladišču, kot pri dimenzijskem. Zahtevnosti izgradnje in vzdrževanja sta neposredno povezani s kadri in finančnimi viri, ki jih organizacija namenja za podatkovno skladiščenje.

Količina podatkov je pri korporativnem in dimenzijskem skladišču, ki zagotavljata enako količino informacij, podobna. Večjo oceno ima dimenzijsko skladišče, ker ima denormaliziran model, kar pa nima bistvenega vpliva, saj je delež podatkov v dimenzijah majhen. Aktivno skladišče običajno ne vsebuje zgodovine, zato je količina podatkov manjša od prej omenjenih. Skladišče kot skupek področnih pa ima podatke običajno že združene na višjih nivojih, tako da je količina zato manjša. Federativno samo po sebi sploh ni fizična zbirka podatkov, zato ima oceno 1.

Podatkovna skladišča so se od svojih začetkov zelo razvila in danes, odvisno od potreb, uporabnikom zagotavljajo povezane, zgodovinske in pravočasne informacije. Vsaka od osnovnih vrst ima svoje prednosti in slabosti, ki se jih je potrebno pri načrtovanju zavedati in jih upoštevati.

Nekaterim slabostim se je mogoče izogniti s skladiščem kot kombinacijo osnovnih vrst. V praksi je takšnih primerov največ. V poglavju arhitektura je opisan eden od njih. Za celovito podatkovno skladišče se tako lahko uporabi korporativna ali dimenzijska vrsta, kadar je potrebno, pa se zaradi prevelike količine ali razdrobljenosti podatkov izdelajo posamezna področna podatkovna skladišča, ki se lahko napajajo iz celovitega skladišča ali neposredno iz izvorov. Operativna hramba podatkov je lahko narejena kot aktivno skladišče, ki s tem nudi uporabnikom dostop do podatkov, ki jih potrebujejo v realnem času. Zavedati se je potrebno, da se ta potreba ne pojavlja povsod. Nekatera poročila se izdelujejo na mesečnem ali celo letnem nivoju in prenos podatkov v realnem času za njih ni smiseln. Federativna vrsta pa se

na primer začasno uporabi ob priključitvi novega izvirnega sistema, ko uporabniki ne morejo čakati na podatke tako dolgo, kot traja postopek običajne vključitve v skladišče.

2.5 DW 2.0

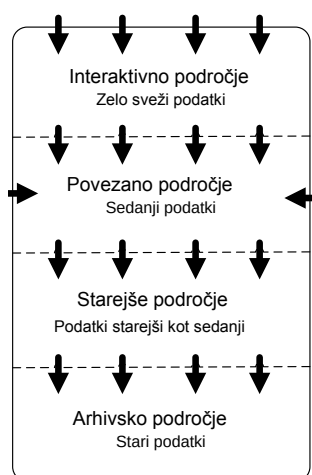
Willima H. Inmon je prvi uporabil besedno zvezo podatkovno skladišče, leta 2008 pa je izdal knjigo [5], v kateri predstavlja DW 2.0, ogrodje za naslednjo generacijo podatkovnih skladišč. Z njo je želel razrešiti zmedo zaradi različnih pogledov na to področje, ko se je začelo s podatkovnimi skladišči imenovati tudi to, kar prvotno ni bilo predvideno – zato je tudi prišlo do različnih vrst. Gre torej za enega od naslednjih korakov na tem področju. Za DW 2.0 je za razliko od podatkovnih skladišč pridobil ustrezne pravice, tako da je opredelitev le ena in so različne razlage nemogoče.

DW 2.0 ima veliko značilnosti, ki jih prva generacija skladišč ni imela. Poleg povezanih transakcijskih podatkov vsebuje še [21]:

- urejene nestrukturirane podatke v različnih oblikah,
- povezane metapodatke, tako poslovne kot tehnične,
- podatke v realnem času, ki se lahko spreminjajo,
- matične podatke organizacije (angl. *reference master data*) in
- skupne združene podatke (angl. *profile data*).

Ob tem DW 2.0 vsebuje še zaporedne podatke v časovnih razponih (angl. *continuous time span data*).

Pomembno je, v katerem življenjskem obdobju se nahaja določen podatek. Podatek najprej nastane, zatem se uporabi in nato zavrže. Ko vstopi v sistem je svež in nov, nato se začne starati. Najprej preide v srednje življenjsko obdobje, nato v starejše, nazadnje pa se arhivira. Tehnologije in principi, ki to starost upoštevajo, so tesno povezani v ogrodje DW 2.0. Podatki pri tem prehajajo skozi različna področja, kot je prikazano na sliki 8.



Slika 8: Prehajanje podatkov med področji v DW 2.0

Področja so štiri. Prvo je interaktivno, drugo povezano, tretje starejše (orig. angl. *near line*) in četrto arhivsko. Podatki vstopijo v sistem v interaktivno področje v obliki aplikacij. Po pravilu se v aplikacijah izvajajo transakcije. Te zahtevajo hiter odzivni čas in visoko razpoložljivost. Podatki v njih med sabo niso tesno povezani. Pri prehodu v povezano področje se povežejo. Lahko vstopijo tudi neposredno v povezano področje, brez prehoda skozi interaktivno področje. Tisti v povezanem področju pripadajo različnim tipom. Podatki iz aplikacij, ki se prenesejo skozi interaktivno področje se povežejo in preoblikujejo v predmetne skupine. Ostali tipi so še tekstovni podatki, podatki v časovnih razponih, skupni združeni podatki, ter poslovni in tehnični metapodatki.

Iste oblike podatkov se nahajajo tudi v starejšem in arhivskem področju.

Interaktivno področje vsebuje zelo sveže podatke, stare do enega meseca, v povezanem področju so stari od dneva do treh let, v starejšem od šestih mesecev do desetih let in v arhivskem od petih let pa naprej do neskončnosti.

Lokacija podatka je v največji meri odvisna od verjetnosti njegove uporabe in potrebne hitrosti dostopa. V interaktivnem področju je ta verjetnost velika, dostop pa mora biti zelo hiter. V povezanem področju je verjetnost srednja, dostop pa razumno hiter. Za starejše področje velja, da je verjetnost dostopa manjša, zato tudi dostop ni najhitrejši. Do podatkov v arhivskem področju pa se dostopa zelo redko, zato je dostop lahko počasen. Pravzaprav je običajno, da je v arhivskem področju verjetnost dostopa blizu ničle. Organizacije hranijo te podatke zaradi zakonskih predpisov in do njih dostopajo le v izrednih razmerah ali pa za vsak slučaj, če bi jih kdaj potrebovali zaradi nekega posebnega dogodka. Čeprav je verjetnost dostopa blizu ničle, je podatke brez večjega napora še vedno mogoče obnoviti (angl. *restore*).

Količina podatkov v interaktivnem področju je najmanjša. Po področjih nato narašča. Starejši kot so podatki, večja je. S tega vidika so razlike med področji velike, zato je različna tudi tehnologija. Ne obstaja »ena za vse«.

V splošnem podatki prehajajo med področji nespremenjeni in v paketih. Prva izjema je prenos med interaktivnim in povezanim področjem, ko se podatki povežejo. Druga izjema je prenos v arhivsko okolje. Pri tem se lahko preoblikujejo zaradi prenosa podatkov v strukture in tehnologijo, ki bo podprta še čez nekaj desetletij.

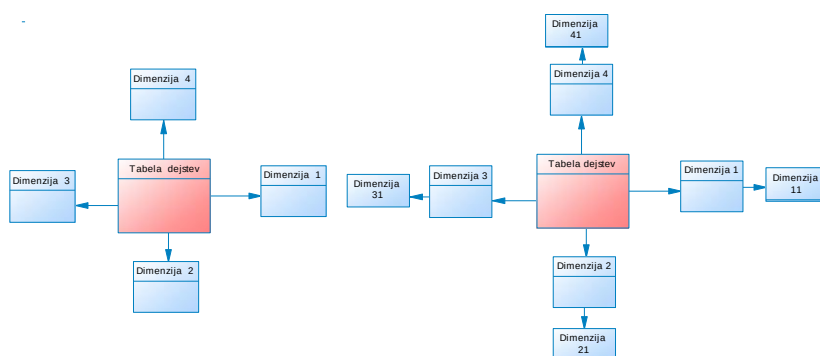
3 Dimenzijski modeli

3.1 Uvod

Dimenzijski modeli so široko sprejeta tehnika pri načrtovanju podatkovnih modelov podatkovnih skladišč. Njihova glavna prednost je razumljivost. Uporabnikom, običajno neveščim dela s podatkovnimi strukturami, omogočajo lažje razumevanje, posebna namenska programska oprema pa omogoča njihovo neposredno uporabo. S tem, ko je model enostaven, uporabnikom razumljiv, hkrati pa omogoča hitro izvedbo poizvedb, je omogočena učinkovita raba podatkov v analitične namene. S takšnimi modeli znajo upravljati sistemi za upravljanje relacijskih in sistemi za upravljanje večdimenzijskih podatkovnih baz. Slednji so posebej namenjeni takšnim modelom.

Dimenzijski modeli so sestavljeni iz mer in njihovega konteksta. Mere nastanejo v poslovnem procesu organizacije in se zajamejo s pomočjo transakcijskih izvornih sistemov. Običajno so to številčne vrednosti, ki jim rečemo tudi dejstva. Ta nastanejo v določenem kontekstu, ki velja v trenutku, ko se dejstvo zabeleži. Kontekst lahko razdelimo v neodvisne skupine, ki jim rečemo tudi dimenzije. Te odgovarjajo na vprašanja kot so kdo, kaj, kdaj, kje, zakaj in kako.

Vsakega od poslovnih procesov organizacije je mogoče predstaviti z dimenzijskim modelom, ki je v relacijskem okolju sestavljen iz tabele dejstev s številčnimi merami, katera je povezana z dimenzijskimi tabelami z večinoma tekstovno vsebino. Zaradi značilnega izgleda takšnega entitetno relacijskega modela se imenuje tudi zvezdna shema. V primeru, da so dimenzijske tabele naprej povezane z drugimi dimenzijskimi tabelami, govorimo o snežinkah. Na sliki 9 je na levi strani prikazana zvezdna shema, na desni pa snežinka. Dimenzijski modeli v večdimenzijskih podatkovnih bazah se imenujejo kocke.



Slika 9: Zvezdna shema in snežinka

Dimenzijski model je uporabnikom bolj razumljiv kot običajen normaliziran izvorni sistem, ker so podatki združeni v razumljive skupine ali dimenzije, ki so uporabnikom znane. Z

njihovo pomočjo lahko upravljajo s podatki modela, pri čemer lahko uporabijo le tiste dimenzije, ki so za določeno poročilo smiselne, ostalih pa ne. Vsaka dimenzija je enakovredna vstopna točka v model. Takšna enakovrednost omogoča številne spremembe v poizvedbah, kakršne se izvajajo na primer med spontanim domenskim poizvedovanjem v specifične namene (angl. *ad hoc*). Model mora biti enostaven, a ne bolj, kot je to potrebno, saj mora še vedno predstavljati poslovanje podjetja, tudi če je to zapleteno.

3.2 Povezava z normaliziranim modelom

Normaliziran podatkovni model je entitetno relacijski model v tretji normalni obliki, ki pa za dimenzijski model ni značilna. S takšnim modeliranjem se izognemo redundancam v podatkih. Podatki so razdeljeni in normalizirani v zapise posameznih tabel. Takšna normalizacija je zelo koristna v transakcijskih okoljih, saj omogoča enostavne in hitre transakcije.

Normalizirani modeli zajamejo vse povezave med podatkovnimi elementi, število entitetnih tipov je tipično veliko, rezultat tega pa je zapletena mreža povezav. Normaliziran model večjega sistema v organizaciji je lahko sestavljen iz na stotine entitetnih tipov, vsak pa se pri prenosu v fizično obliko preslika v tabelo.

Za predstavitev tako normaliziranega kot relacijskega dimenzijskega modela se največkrat uporabljajo entitetno relacijski diagrami, saj sta oba sestavljena iz povezanih relacijskih tabel. Glavna razlika med obema je v stopnji normalizacije.

Normaliziran model je zelo primeren za transakcijske sisteme. V celovitem podatkovnem skladišču sta mogoča oba, normaliziran in dimenzijski pristop. Oba sta mogoča tudi v področnih podatkovnih skladiščih, a se pri slednjih normaliziran pristop uporablja redko.

Pretvorba iz normaliziranega v dimenzijski model je mogoča. Najprej je potrebno v normaliziranem modelu odkriti tabele, ki vsebujejo številčne mere in jih je mogoče pretvoriti v tabele dejstev. Le-te so normalizirane tudi v dimenzijskem modelu, saj je pripadajoči kontekst ločen v dimenzijah. Naslednji korak je denormalizacija ostalih tabel in njihova pretvorba v dimenzijske tabele, s ključi na najnižjem nivoju podrobnosti, ki so povezani neposredno s tabelo dejstev. Dimenzijske tabele so običajno v drugi normalni obliki.

V članku »Iz korporativnih modelov do dimenzijskih modelov: metodologija za načrtovanje podatkovnih skladišč in področnih podatkovnih skladišč« [10] je opisana metoda za izgradnjo skladišča na osnovi podatkovnega modela na nivoju organizacije, katere del je tudi pretvorba v dimenzijski model. Koraki te metode so:

1. Izdelava podatkovnega modela na nivoju organizacije (v primeru ko ta še ne obstaja). Gre za model na najvišjem nivoju abstrakcije.
2. Izdelava modela za celovito podatkovno skladišče. Ta je podmnožica modela na nivoju organizacije, s podatki, ki so pomembni z vidika podpore postopkom odločanja. Priporoča se postopni razvoj. Začne se z najpomembnejšimi področji in nato nadaljuje. Eden od možnih pristopov k tej točki je opisan v članku »Creating The Data Warehouse Data Model From The Corporate Data Model« [20].
3. Razvrščanje entitetnih tipov v celovitem skladišču v tri razrede: transakcijske, sestavne (angl. *component*) in razvrstitvene (angl. *classification*).
4. Določanje hierarhij v podatkovnem modelu.
5. Načrt modela za področna podatkovna skladišča. Transakcijski entitetni tipi se preslikajo v tabele dejstev, sestavni in razvrstitveni pa v dimenzijske tabele. Sestavne tvorijo najnižji nivo podrobnosti v dimenzijskih tabelah, razvrstitvene pa višje nivoje.

Uporaba te metodologije ima kar nekaj prednosti. Celovito skladišče in področna skladišča tako odražajo osnovne povezave med podatki. Postopki polnjenja so enostavnejši, saj ponorni model temelji na izvornem. Postopek izgradnje skladišča sicer ni popolnoma avtomatičen, zagotavlja pa smernice za načrtovalce, ki ga poenostavijo. Še vedno je potrebna temeljita analiza za določitev in razvrstitev entitetnih tipov, ko pa je to opravljeno, je razvoj dimenzijskega modela relativno enostaven. Uporaba osnovnega normaliziranega modela za izgradnjo dimenzijskega modela zagotavlja močno osnovo in tako omogoča, da se izognemo mnogim nevarnostim in težavam, na katere bi naleteli, če bi začeli model izdelovati od začetka. Takšna in podobne metodologije se zaradi navedenih prednosti veliko uporabljajo.

3.3 Tabele dejstev

3.3.1 Splošno o tabelah dejstev

Tabele dejstev hranijo dejstva, ki nastanejo ob posameznih dogodkih, ali pa so plod poslovnih aktivnosti organizacije. Dejstvo je lahko katerakoli mera, pomembna za organizacijo. Pojavi se ob izdaja računa, zabeležki pritožbe ali ob telefonskem klicu. Pri slednjem se pojavijo na primer čas trajanja, cena in popust na ceno, zabeležijo pa se za vsak klic. Tipično so to številčne vrednosti, ki lahko zavzamejo katerokoli vrednost znotraj nekega intervala.

Najbolj uporabna dejstva so tista, ki so poleg tega, da so številčna, tudi aditivna. To je pomembno, ker orodja za poslovno obveščanje redko obdelujejo le en zapis naenkrat. Običajno delajo nad množico zapisov, seštevanje dejstev v teh zapisih pa je najbolj naravna operacija. Večina številčnih mer je zato aditivnih, ne pa vsa. Številčne mere v času, kot so stanja na dan, so le deloma aditivne, saj jih ni smiselno seštevati skozi čas. Nekatere številčne mere pa sploh niso aditivne, kar pomeni da jih ne moremo seštevati glede na nobeno

dimenzijo. Takšna je na primer temperatura. Večina tekstovnega konteksta pripada diskretnemu seznamu domenskih vrednosti, zato njegovo mesto ni v tabelah dejstev v obliki tekstovnih dejstev, ampak v dimenzijskih tabelah.

Vsako številčno dejstvo ni primerno za tabelo dejstev. Takšen primer je delovna površina objekta, ki je bolj kot dejstvo statična lastnost objekta in spada v dimenzijo objektov. Pri določanju, kam spada številčna mera, si lahko pomagamo z naslednjim pravilom. Mera spada v dimenzijo, kadar se uporablja kot omejitev (filter), kot začetni stolpec (angl. *column header*) ali začetna vrstica (angl. *row header*) v poročilih, ali pa ima diskretno zalogo vrednosti.

Tabele dejstev lahko vsebujejo veliko količino podatkov z milijoni ali milijardami vrstic. Zato morajo biti načrtovane tako, da je mogoče iz njih pridobiti podatke na učinkovit način. Redundantni podatki tako za njih niso primerni. Če je mogoče, potem se za organizacijo za iste mere izdela le ena tabela dejstev, tudi če je uporaba od oddelka do oddelka drugačna.

Pomembno je, da se tabele dejstev ne gradijo primarno na osnovi pogostih poročil in poslovnih povpraševanj, ampak glede na dogodke, ki odgovarjajo poslovnim procesom. S tem omogočimo, da je podatkovno skladišče razširljivo in uporabno za različne namene, ter za različne oddelke in skupine uporabnikov.

Ključ v tabeli dejstev je sestavljen iz ključev posameznih dimenzijskih tabel. Vsak tuji ključ v tabeli dejstev se mora ujemati s primarnim ključem v pripadajoči dimenzijski tabeli. Polje s tujim ključem naj torej ne bi bilo prazno. Težava nastane, ko v dimenzijski tabeli ne obstaja zapis, ki bi odgovarjal vrednosti v tabeli dejstev. To preprečimo s posebnim zapisom, na katerega se lahko takšna vrednost poveže, z opisom na primer »neznan predmet« ali podobno.

Zrno tabele dejstev predstavlja poslovno opredelitev dogodka, katerega rezultat je natanko en zapis v tabeli dejstev. Zrno posledično določa primarni ključ, če le-ta obstaja, poslovno pa ga je mogoče opredeliti z vsemi dimenzijami, povezanimi s tabelo dejstev. Zrno je enako za vse zapise v tabeli.

Ralph Kimball razvršča tabele dejstev v tri vrste [7]–[9]. Ne glede na dejavnost in pripadajoče poslovne procese so dejstva, ki se beležijo, treh vrst: transakcije, periodični posnetki stanj in zbirni posnetki stanj. Na tej osnovi se delijo tudi tabele dejstev. Transakcije in posnetki stanj predstavljajo bistvo podatkovnega skladišča. Prve nam nudijo popoln vpogled v dogajanje, drugi pa omogočajo enostavno spremljanje. Pogosto rabimo oboje, saj ne obstaja preprost način za kombiniranje teh dveh kontrastnih pogledov.

3.3.2 Transakcijske tabele dejstev

Transakcijske tabele dejstev so najbolj pogoste. Zrno je določeno s transakcijo, kar pomeni da vsak zapis predstavlja natanko eno transakcijo. Mera je veljavna v natančno določenem trenutku.

Transakcijski podatki na najnižjem nivoju podrobnosti so običajno povezani z mnogo dimenzijami. Ko se zgodi dogodek, se hkrati s transakcijo zabeleži tudi njen kontekst. Zapisi se vnesejo v tabelo le, če se je dogodek zgodil. Ko se zapis zabeleži, se največkrat ne popravlja več.

Takšne tabele za uporabnika predstavljajo močno orodje za pridobivanje informacij, saj beležijo najbolj elementarne aktivnosti poslovanja. Težava lahko nastane zaradi količine podatkov, kar pa je mogoče rešiti s posnetki stanj, ki omogočajo lažji pregled na dogajanjem.

Enostavnost polnjenja takšnih podatkov je zagotovljena s tem, da se zapisi, ko se enkrat pojavijo v tabeli, ne spreminjajo več. Zato je takšna vrsta primerna med drugim tudi za aktivno podatkovno skladišče. Tipična je sicer za celovito podatkovno skladišče dimenzijske vrste. V področnih podatkovnih skladiščih se zaradi najnižjega nivoja podrobnosti podatkov pojavlja manj, je pa z določenimi prijemi, kot je seštevanje podatkov na višje nivoje, primerna tudi za to vrsto skladišč.

3.3.3 Tabele dejstev s periodičnimi posnetki stanj

Tabele dejstev s periodičnimi posnetki stanj so glede na pogostost pojavljanja na drugem mestu. Kot pove že ime tabel, zapis v njih predstavlja posnetek stanja v določenem trenutku. Stanje se zajema periodično v vnaprej določenih intervalih. Interval je lahko poljuben, običajno pa je dnevni, nekaj dnevni, tedenski ali mesečni. Vsi posnetki zajemajo podatke na enakem nivoju podrobnosti.

Nekateri poslovni procesi so že naravno predstavljeni kot periodični posnetki. Tak primer je stanje na bančnem računu, ki se računa dnevno ali mesečno. Tudi najbolj pogosta finančna in računovodska poročila so te vrste.

Te tabele dejstev dopolnjujejo transakcijske, a jih ne nadomeščajo. Podatke na najnižjem nivoju podrobnosti iz slednjih so še vedno potrebni za odgovore na najbolj precizna povpraševanja. Tako kot za transakcijske tabele velja tudi za te, da se podatki, ko so enkrat v tabeli, ne spreminjajo več.

Takšne tabele so zelo primerne za področna podatkovna skladišča, primerne pa so tudi za celovita podatkovna skladišča, še posebej, kadar so dejstva že na izvoru periodičnega tipa.

3.3.4 Tabele dejstev z zbirnimi posnetki stanj

Za razliko od periodičnih posnetkov stanj predstavljajo zbirni posnetki stanj dogajanje v postopkih skozi nedoločeno časovno obdobje, ki ima svoj začetek in konec. Primer je zadeva na sodišču, ki jo sodišče prejme, nato se čez nekaj časa skliče obravnava in na koncu se zadeva zaključi. Zadevo predstavlja en zapis, ki se sproti popravlja, tako da vedno odraža zadnje stanje.

Tabele dejstev z zbirnimi posnetki stanj imajo glede na drugi dve vrsti nekaj svojstvenih značilnosti. Običajno imajo več polj, nad katerimi so tuji ključi s povezavo na časovno dimenzijo. Na ta način so datumsko določeni mejniki v življenju postopka. Večina teh polj nima vrednosti, ko se zapis prvič pojavi v tabeli. Določijo se kasneje.

Poleg tipičnih zneskov in aditivnih števil takšne tabele vsebujejo še dve vrsti dejstev. Prvo takšno je indikator, ki nosi informacijo o tem ali je bil mejnik že dosežen ali ne in ima zalogo vrednosti ena in nič, pri čemer ena pomeni, da je bil mejnik dosežen. Omogoča nam, da enostavno ugotovimo, koliko zapisov je že prišlo do določenega mejnika, na primer ali je zadeva na sodišču že imela obravnavo ali ne. Druga takšna vrsta dejstev so tista, ki predstavljajo trajanje med dvema mejnikoma, na primer med prejemom na sodišče in obravnavo.

Glavna značilnost teh tabel dejstev je spreminjanje zapisov. Skoraj vedno jih dopolnjuje ena ali več transakcijskih tabel. Zbirni postopek predstavlja ključne transakcijske aktivnosti skupaj zbrane v enem zapisu. To zagotavlja hitrost poizvedb in posledično veliko analitičnih zmožnosti, a ne more nadomestiti transakcijskih tabel, saj se nekatere podrobnosti neizogibno izgubijo.

Zaradi možnosti spreminjanja zapisov so te tabele dejstev primerne za operativno hrambo podatkov, zaradi preprostega poizvedovanja pa za področna podatkovna skladišča. Samostojne so manj primerne za celovito podatkovno skladišče, saj se določeni podatki izgubijo, če pa jih dopolnjujejo druge tabele, so primerne tudi za to vrsto.

3.4 Dimenzijske tabele

3.4.1 Splošno o dimenzijskih tabelah

Za razliko od tabel dejstev, ki so sestavljene predvsem iz ključev in številčnih mer, je v dimenzijskih tabelah mnogo opisnih polj, ki se uporabljajo predvsem za omejitve in

označevanje stolpcev in vrstic na poročilih. Uporabnost podatkovnega skladišča je na mnogo načinov povezana s kakovostjo podatkov v teh poljih. Ti opisi so ali tekstovni, ali pa številčni z diskretno zalogo vrednosti. Koristna lastnost dimenzijskih modelov je, da se lahko vsa polja povsem enakovredno uporabijo za omejitve in označevanja. Polja v dimenzijskih tabelah naj bi bila [9]:

- obširna (oznake, sestavljene iz celih besed),
- opisna,
- popolna (brez manjkajočih vrednosti),
- z diskretno zalogo vrednosti (le ena vrednost na zapis) in
- kakovostna (brez zatipkanih napak, neveljavnih vrednosti, zastarelih vrednosti, ali še tako majhnih razlik med vrednostmi, ki naj bi bile enake).

Dimenzijske tabele so lahko sestavljene iz več deset polj, a to ni pravilo. V določenih poljih se lahko nahajajo tudi kode in okrajšave, tem pa je priporočljivo dodati pripadajoče opisno polje, tako da imajo uporabniki možnost dela neposredno s takšno vrednostjo, brez da bi se ta prilagajala v orodju za poročanje. Določene operative kode nosijo v sebi nek dodatni pomen, dodatno kodo, kot je lahko na primer iz prvih dveh znakov kode zaposlenega razvidno, v katerem oddelku je zaposlen. Takšne dodatne kode je smiselno zabeležiti v ločenem polju, da jih lahko uporabimo na enak način, kot vse druge vrednosti v dimenziji.

V dimenzijskih tabelah se pojavljajo tudi hierarhične povezave. Na ta način so v eni tabeli shranjeni odvisni podatki, med katerimi velja odvisnost ena proti mnogo (angl. *one-to-many*). To je naravna posledica denormalizacije. Če produkti spadajo v skupino produktov in ta v kategorijo, lahko to predstavimo v eni tabeli produktov, ki poleg polj, ki pripadajo produktu, vsebuje še tista, ki pripadajo skupini produktov in kategoriji. Podatki so torej redundantni, a zato razumljivi, poizvedbe nad njimi pa delujejo hitreje. Ni pa to edini način, o čemer je več napisano v podpoglavju »Zvezdne sheme in snežinke«.

3.4.2 Skupne dimenzije

Skupne dimenzije so ključnega pomena za dobro organizirano dimenzijsko podatkovno skladišče. Skupne so z vidika celotnega podatkovnega skladišča, saj si jih deli množica tabel dejstev in različni poslovni procesi. Na nek način tvorijo skupno vodilo.

Dve tabeli dejstev se tako povezujeta z isto dimenzijsko tabelo, lahko pa se povezujeta z dvema različnima, med katerima je ena podmnožica druge. Tak primer nastopi, kadar je zrno v eni tabeli dejstev glede na isto dimenzijo na drugačem nivoju podrobnosti, kot v drugi tabeli dejstev. V tem primeru je smiselno, da imajo polja enaka imena in da vsebina v eni tabeli natančno odgovarja vsebini v drugi.

Skupne dimenzije prinašajo naslednje prednosti [9]:

- Doslednost. Zagotavljajo, da je omejevanje nad tabelami dejstev dosledno, rezultati povpraševanj pa so ustrezno označeni.
- Povezanost. Omogočajo, da celotno podatkovno skladišče deluje kot povezana celota. Možno je vrtanje med različnimi tabelami dejstev, ki predstavljajo različne procese.
- Krajši čas izvedbe. Za uvedbo je sicer potreben določen čas, a se vložek kasneje povrne. Vsi nadaljnji projekti lahko uporabijo že obstoječo infrastrukturo, kar pomeni, da ni potrebno na novo odkrivati že obstoječih dimenzij.

3.4.3 Počasi spreminjajoče se dimenzije

Dimenzijske tabele naj bi imele dokaj statično vsebino, kar pa pogosto ne drži. Drži pa, da tudi kadar se podatki v njih spreminjajo, se spreminjajo počasneje kot tisti v tabelah dejstev. Dimenzijski model mora omogočati sledenje in beleženje sprememb, kadar za to obstaja potreba. Obstaja več tipov za upravljanje počasi spreminjajočih se dimenzij v dimenzijskih tabelah. Osnovni (izvirni) tipi so trije [9].

- Tip 1: prepis vrednosti. Kadar se vrednost spremeni, se naredi prepis vrednosti v odgovarjajočem polju. Polje vedno vsebuje zadnje stanje, zgodovinski podatki pa so pri tem izgubljeni. Ta tehnika je nadvse primerna za popravke kot posledice napak, koristna pa je tudi, ko stara vrednost nima nobene poslovne vrednosti. Tip 1 torej ne ohranja vrednosti, ki je bila veljavna, ko je bil nek zapis v tabeli dejstev ustvarjen. Prednost takega pristopa je v enostavnosti polnjenja, težava pa lahko nastane, kadar so bili podatki že pred spremembo združeni v neko drugo strukturo glede na staro vrednost. Ti združeni podatki potem ne odgovarjajo trenutnemu stanju osnovnih podatkov. V tem primeru bi bilo sinhronost podatkov potrebno zagotoviti s ponovnim združevanjem.
- Tip 2: dodajanje novega zapisa. Ohranja se celotna zgodovina. Ob spremembi vrednosti se vnese nov zapis z novo vrednostjo. Tako obstaja za vsako spremembo vrednosti nov zapis, ki pa ima enak naravni ključ kot starejši zapisi. Vsi zapisi so poleg tega opremljeni še s podatkom o začetku in koncu veljavnosti. Tip 2 se uporablja, kadar je potrebno spremembe beležiti in za vsak trenutek v času vedeti, kdaj je bila katera vrednost veljavna. Vsak zapis predstavlja verzijo, ki je bila veljavna v določenem obdobju. Preglednica 2 na primeru treh zapisov prikazuje spreminjanje kontaktne številke, ki je veljavna od trenutka, ki ga določa polje »velja od«, pa do trenutka, ki ga določa polje »velja do«.

Ključ zapisa	Naravni ključ (številka računa)	Kontaktne tel. številka	Velja od	Velja do
3	05100-	01 1234567	1.1.2000	8.6.2007

	1234567890			
14	05100- 1234567890	040 123456	9.6.2007	1.10.2010
39	05100- 1234567890	030 123456	2.10.2010	odprto

Preglednica 2: Počasi spreminjajoče se dimenzije, tip 2

Slaba stran tega tipa je v kompleksnosti polnjenja in v kompleksnosti morebitnih popravkov.

- Tip 3: novo polje v dimenzijski tabeli. Ta tip se uporablja, kadar bi še vedno radi omogočili poizvedovanje o tem, kaj bi bilo, če se sprememba ne bi zgodila. Primer je spreminjanje prodajnih okrožij. Uporabnike zanima, kakšni so rezultati glede na spremenjene okoliščine, kakšni bi bili rezultati, če bi bile okoliščine spremenjene že prej in kakšni še bodo rezultati glede na obe možnosti. Sami lahko preklaplajo med obema možnostma. Ob spremembi se stara vrednost prepíše v polje s staro vrednostjo, nova vrednost pa se zapiše v polje z novo vrednostjo namesto stare. Pri tej tehniki ni potrebno dodajanje novih zapisov, omogoča pa le spremljanje omejene količine sprememb. Poleg dodatnega polja s staro vrednostjo se lahko doda še polje z datumom spremembe vrednosti.

Ključ zapisa	Naravni ključ (številka računa)	Nova vrednost (kontaktna tel. številka)	Stara vrednost (kontaktna tel. številka)	Datum spremembe
3	05100- 1234567890	030 123456	040 123456	1.10.2010

Preglednica 3: Počasi spreminjajoče se dimenzije, tip 3

Preglednica 3 prikazuje spremembe s tipom 3 na enakem primeru kot za predhodno predstavljeni tip 2. Beleži se le ena sprememba, zato se informacija o kontaktni številki do 9.6.2007, prikazana na prejšnji preglednici, izgubi.

3.5 Zvezdne sheme in snežinke

Zvezdne sheme in snežinke so si med seboj podobne. V obeh primerih govorimo o dimenzijskem modelu. Količina informacij je enaka, ne glede na to, ali se odločimo za eno ali drugo vrsto. Snežinko lahko opredelimo kot zvezdno shemo, ki jo preoblikujemo v tretjo normalno obliko in na ta način dobimo večnivojsko strukturo z več dimenzijskimi tabelami in na videz večjo kompleksnostjo (slika 9).

Njena prednost so jasno izražene strukture in povezave med njimi, ki nadomestijo nepovezane zbirke polj. Nekoč je veljalo, da so snežinke nezaželene [8], [6], ker povečajo kompleksnost, poslabšajo hitrost poizvedb in le neznatno zmanjšajo porabo prostora. Danes se takšno mišljenje spreminja, saj empirični testi dokazujejo, da so zmogljivosti odvisne od sistemov za upravljanje podatkovnih baz in orodij za poročanje. Nekateri dosegajo boljše zmogljivosti z zvezdnimi shemami, drugi s snežinkami [13].

Hierarhije so v snežinkah jasno razvidne, medtem ko se jih mora v zvezdnih shemah zavedati uporabnik, ali pa jih je potrebno določiti v orodju za poročanje. Zato je včasih koristno nad zvezdnimi shemami narediti poglede s pomočjo katerih lahko uporabniki vidijo strukturo [11].

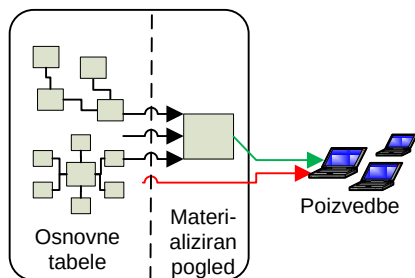
Pri odločanju za eno ali za drugo je potrebno sprejeti odločitev glede na več dejavnikov. Kadar imamo dve tabeli dejstev, ki uporabljata isto dimenzijo, a na drugačem nivoju podrobnosti, je smiselno uporabiti snežinko, torej za vsak nivo podrobnosti svojo dimenzijsko tabelo. Kadar je del hierarhije skupen več dimenzijam, kot je na primer država skupna tako dimenziji kupcev kot dobaviteljev, je prav tako smiselno uporabiti snežinko in s tem za državo uporabiti svojo tabelo. Naslednji primer v prid snežinkam je visoko razmerje med kardinalnostjo primarne dimenzijske tabele in sekundarne. Če obstaja veliko produktov in zelo malo vrst produktov, lahko z denormalizacijo pridobimo na prostoru. Zvezdne sheme na drugi strani pa prinašajo predvsem razumljivost in, odvisno od tehnologije, včasih tudi zmogljivosti.

3.6 Uporaba materializiranih pogledov

V tem podpoglavju je predstavljena uporaba materializiranih pogledov za namene izboljšave hitrosti poizvedovanja nad relacijskimi dimenzijskimi modeli. V relacijskih sistemih za upravljanje s podatkovno bazo predstavlja pogled navidezno tabelo z rezultatom neke poizvedbe. Kadar se nad pogledom izvede poizvedba, se podatki prenesejo iz pripadajočih tabel. Pri materializiranem pogledu je pristop drugačen. Rezultat poizvedbe se shrani v stvarno tabelo, ki se občasno osveži iz osnovnih tabel. To omogoča bolj učinkovit dostop, pri čemer se je potrebno zavedati, da so podatki lahko zastareli. Čas izvedbe za pogoste poizvedbe se na ta način zelo zmanjša. Materializiran pogled je torej predstavljen kot tabela, kar pomeni, da lahko dodajamo indekse in tako še dodatno pridobimo na hitrosti.

Ena glavnih prednosti materializiranih pogledov je, da ni potrebe, da bi se jih pisci poizvedb sploh zavedali. Poizvedba se napiše neposredno nad osnovnimi tabelami, sistem za upravljanje podatkovne baze pa, če obstaja v materializiranem pogledu zahtevana množica podatkov, naredi poizvedbo kar nad njim. Slika 10 prikazuje takšno uporabo. Rdeča puščica

prikazuje, nad katerimi tabelami je napisana poizvedba, zelena puščica prikazuje, da se podatki dejansko zajamejo iz materializiranega pogleda, črne puščice pa predstavljajo občasno osveževanje.



Slika 10: Uporaba materializiranih pogledov

Materializirane poglede uporablja vedno več podatkovnih baz. Prva na tem področju je bila podatkovna baza Oracle [25]. V IBMovi bazi DB2 [19] se imenujejo tabele materializiranih poizvedb (angl. *materialized query tables*).

V dilemi o uporabi zvezdnih shem ali in snežink materializirani pogledi odprejo dodatno možnost. Izdelava se snežinka, če pa njena izdelava povzroči slabše zmogljivosti, se dimenzijske tabele, ki predstavljajo hierarhijo ene dimenzije, združijo v materializiran pogled. Na ta način lahko uporabimo strukturo snežinke, ki ohranja dobre lastnosti zvezdne sheme. Ta materializiran pogled je lahko uporabnikom dostopen kot denormalizirana dimenzija, ali pa jim je neviden in služi le kot pomoč sistemu za upravljanje podatkovnih baz za hitrejšo izvedbo poizvedb.

Glavni namen materializiranih pogledov pa je optimizacija poizvedb nad večjo količino podatkov, kakršna se običajno nahaja v tabelah dejstev. Pri tem se moramo odločiti, katere poglede bomo materializirali. Poskušati moramo čim bolj zmanjšati ceno in zagotoviti skladnost z sistemskimi in uporabniškimi omejitvami. Pri zmanjševanju cene moramo biti pozorni na dva vidika.

- Ocena obremenitve. Mnogo tehnik za izbor pogledov temelji na njihovi pričakovani obremenitvi [3], kjer se za izračun cene uporabi utežene seštevke cen različnih poizvedb. Uteži so odvisne od pogostosti poizvedb in njihove pomembnosti za uporabnike. Nekatere tehnike za izračun cen posameznih poizvedb so preproste in predvidevajo, da cena poizvedbe odgovarja velikosti pogleda, nad katerim se izvajajo, nekatere pa predvidevajo še druge dejavnike, kot je na primer število prebranih podatkovnih blokov.
- Cena vzdrževanja. Materializirani pogledi morajo odgovarjati stanju v osnovnih tabelah, zato se morajo redno osveževati. Cena vzdrževanja je cena tega osveževanja. Njen izračun je zaradi mnogih možnih tehničnih rešitev težaven. Najbolj enostaven približek je izvedba

poizvedbe pogleda nad osnovnimi tabelami, a v primeru inkrementalnega osveževanja pogledov ni primeren.

Sistemske omejitve so dveh vrst.

- Prostor na disku. Na disku ni neomejene količine prostora, zato je potrebna ocena velikosti posameznega pogleda. Materializiran pogled ni edina rešitev za izboljšanje zmogljivosti, zato je včasih smiselno razmisliti o uvedbi indeksov, ki pa prav tako zahtevajo svoj prostor. Materializiran pogled tipično zasede manj prostora kot osnovne tabele, je pa ob tem potrebno upoštevati, da je nad nekim naborom podatkov, na primer nad eno zvezdno shemo, narejenih več takšnih pogledov.
- Čas osveževanja. Osveževanje materializiranih pogledov pomeni veliko obremenitev za sistem. Ob tem je potrebno upoštevati, da to ni edina obremenitev, saj so tu še varnostno kopiranje (angl. *backup*), obnovitev (angl. *recovery*), postopki polnjenja in drugi, tako da je čas za osveževanje omejen. Število materializiranih pogledov je torej omejeno s časom, ki je na voljo za njihovo osveževanje. To osveževanje je lahko vključeno v postopke polnjenja.

Uporabniške omejitve so povezane z uporabniškimi zahtevami.

- Čas za izvedbo poizvedb. Uporabniki lahko zahtevajo, da se njihove poizvedbe izvršijo v določenem času. Ta čas se lahko od ene do druge razlikuje. Na osnovi tega časa je mogoče sklepati o pomembnosti poizvedbe.
- »Svežina« podatkov. Uporabniki lahko zahtevajo, da se podatki v materializiranih pogledih osvežujejo v določenem intervalu. S tem zagotovijo, da podatki do katerih dostopajo ne odražajo starega stanja.

Ti cilji so v navzkrižju drug z drugim. Najboljša rešitev je zato uravnotežen kompromis. Če so omejitve prehude, potem se lahko zgodi, da materializirani pogledi niso primerna rešitev.

Uporaba kock je ena od nadgradenj uporabe materializiranih pogledov. Kocke imajo podatke izračunane na več nivojih podrobnosti, a njihovo osveževanje traja dlje. Kot tehnika za izboljšavo hitrosti poizvedovanja nad relacijskimi tabelami niso tako razširjene, ni pa še jasno, ali bodo kdaj tako uveljavljene, kot so materializirani pogledi danes.

3.7 Večdimenzijske podatkovne baze

Večdimenzijski sistemi za upravljanje s podatkovno bazo (v angl. tudi *OLAP engines*) so namenski sistemi za podporo izvajanju analiz. Poizvedbe so hitre, saj so podatki vnaprej združeni na več nivojih podrobnosti. Napisane so v analitično usmerjenem jeziku, kar omogoča napredno rabo podatkov. Primer takega jezika je MDX [23] (angl. *multidimensional*

expressions). Osnovna struktura v takšnih podatkovnih bazah je kocka, sestavljena iz dimenzij in dejstev.

Večdimenzijski SUPB so se v zadnjem desetletju zelo razvili. Pri združevanju podatkov se nekateri omejujejo le na podmnožice, s čimer se izognejo preveliki količini podatkov, ki jih je omejevala v prejšnjih verzijah. Tudi zato lahko sedaj obvladujejo več dimenzij, ki so lahko tudi večje, z milijoni članov.

Napredek se pozna tudi v obvladovanju dimenzij. Relacijske dimenzije se lahko sedaj neposredno preslikajo v večdimenzijske in pri tem ohranijo vsa polja oz. attribute. V nekaterih starejših verzijah je lahko imel en član dimenzije poleg ključa le en atribut, zaradi česar je bilo potrebno ločevati dimenzije. Na primer stranko z demografskimi podatki, lokacijo in tipom je bilo potrebno razdeliti v toliko dimenzij, kolikor je bilo atributov.

Večdimenzijske podatkovne baze so pogosto eden ključnih delov podatkovnega skladišča in niso več le izboljšava relacijskih baz, saj obvladujejo združevanje podatkov in zagotavljajo izboljšane analitične zmožnosti. Na drugi strani pa to omogočajo tudi nekatere relacijske baze same. Če poleg njih to omogoča še orodje za poročanje, potem večdimenzijske baze ne rabimo.

Vrednotenje takšnih baz mora poteka vzporedno z vrednotenjem orodja za poslovno obveščanje, s pomočjo katerega se do takšne baze dostopa. Nekateri ponudniki trdijo, da lahko z njihovim orodjem dostopamo tako do relacijskih kot do večdimenzijskih baz, kar je res, a je ta dostop in združevanje takšnih podatkov še precej okorno. Najboljša orodja za poizvedovanje nad večdimenzijskimi podatkovnimi bazami so še vedno tista, ki so bila narejena posebej za ta namen. Z vključevanjem večdimenzijske podatkovne baze v arhitekturo skladišča si lahko omejimo izbiro orodja za poslovno obveščanje.

Večina ponudnikov SUPB ponuja večdimenzijske zmožnosti, a nekateri temu posvečajo več pozornosti kot drugi. Tehnologije se med seboj še vedno precej razlikujejo in le malo je standardov, ki jih med sabo povezujejo.

3.8 Kocke

Načrtovanje kock je v veliki meri odvisno od izbrane tehnologije. Te sicer imajo nekatere skupne značilnosti, a je trg na področju večdimenzijskih SUPB še razmeroma nezrel, tako da je med orodji še vedno precej razlik kar se tiče zmožnosti, zmogljivosti in skalabilnosti.

Kocke so strukturirane strogo dimenzijsko. Mere in dimenzije so analogne tistim v relacijskih dimenzijskih modelih. Razlika pri dimenzijah je v določenosti hierarhij. Le-te so pomembne tudi v relacijskih modelih, saj omogočajo vrtanje v globino in nazaj, a so v primeru zvezdnih shem lahko tudi nepravilne, pri snežinkah in v večdimenzijskem okolju pa to ni mogoče, saj so določene z omejitvami.

Pomembna odločitev pri načrtovanju je odločitev o nivoju podrobnosti, ki bo določala tudi velikost kocke. Odvisna je od skalabilnosti, ki jo omogoča izbrana tehnologija. Skalabilnost je pogosto odvisna tudi od verzije orodja, saj se ta orodja trenutno zelo izboljšujejo in z vsako verzijo omogočajo več.

Ko je kocka na voljo in uporabniki nad njo izvajajo poizvedbe, se skoraj obvezno pojavi tudi potreba po dostopu do podrobnih podatkov. En način za rešitev tega je izgradnja velike kocke na najnižjem nivoju podrobnosti. Ni pa to edini način. Z vidika uporabnikov je lahko kocka videti večja kot je v resnici, če v orodju za poslovno obveščanje omogočimo vrtanje do podatkov iz katerih se kocka polni. Običajno so to relacijski podatki, ki vsebujejo več informacij ali podrobnosti, a je dostop do njih počasnejši, zato je nad tiste najbolj uporabljane smiselno postaviti indekse.

Naravno polnjenje dimenzij poteka preko dimenzijskih tabel, ni pa to nujno. Pri tem je mogoče v te tabele dodati določene meta podatke, ki dodatno razlagajo vlogo in namen dimenzije.

Nad dimenzijami je pri nekaterih orodjih mogoče določiti več hierarhij. Tak primer je časovna dimenzija, ki vključuje tako koledarsko kot fiskalno hierarhijo. Ali pa organizacija planskih postavk, ki se pogosto menja z leti.

Neuravnoteženost hierarhij je v večdimenzijskem okolju nekaj normalnega, medtem ko se je potrebno v takih primerih v relacijskem okolju posluževati trikov, kot je umetno uravnoteževanje. Tudi orodja za poslovno obveščanje, ki delujejo nad kockami, člane dimenzij obravnavajo enakovredno, ne glede na to, na katerem nivoju v hierarhiji se nahajajo, medtem ko je v relacijskem okolju ta nivo pomemben.

Kocke omogočajo uporabo izračunanih mer ali dejstev. Večdimenzijske baze so zasnovane z namenom čim boljše uporabe dimenzijskega modela, kar vključuje tudi izračunane mere, ki se pojavljajo na poročilih. Tipičen primer je donosnost, ki se jo da izračunati na osnovi ostalih mer. Izračun v kocki je boljši kot izračun na poročilu, saj se s tem izognemo nevarnosti, da bi

dve poročili uporabljali dve različni opredelitvi, poleg tega pa je izračun narejen le enkrat in to v posebej za to predvidenem okolju.

Ena od prednosti uporabe kock, spet le pri nekaterih orodjih, je v združevanju podatkov po hierarhiji navzgor, kjer seštevanje ni več edina operacija, ampak so možne poljubne operacije, določene s pravili za združevanje (angl. *aggregation rules*). Ta veljajo za določene mere glede na posamezne dimenzije. Vrednost nadrejenega člana v hierarhiji je običajno res seštevek vrednosti podrejenih članov, ni pa to pravilo. Tako je na primer dobiček, kot nadrejeni član, enak znesku prihodkov, od katerih se odšteje znesek odhodkov, in še to le v primeru, kadar so prihodki večji od odhodkov. Stanje na dan tudi ni aditivno, tako da je na letnem nivoju potrebno vzeti zadnjo vrednost, možno pravilo za združevanje pa je tudi povprečje.

Slabost kock je daljši čas polnjenja, saj vključuje tudi različne izračune ter združevanje na višje nivoje. Druga lastnost je nezmožnost upravljanja s počasi spreminjajočimi se dimenzijami. Vedno je na voljo le zadnje stanje, za ohranjanje zgodovine pa je potrebno uporabiti katerega od trikov, kot je na primer obravnavo spremenjenega člana dimenzije na način, kot da je to povsem nov član. Nekateri ponudniki programske opreme sicer že napovedujejo možnost obravnave spreminjajočih se dimenzij na način tipa 2, kot se uporablja v relacijskem okolju.

3.9 Kocke in relacijski dimenzijski modeli

Vsaka od tehnik iz naslova tega poglavja ima svoje prednosti in slabosti. Pri odločanju za eno ali drugo si lahko pomagamo s preglednico 4.

Kocka	Relacijski dimenzijski model
	Zrelost relacijske tehnologije
Hitrost proizvodovanja	
	Mnogo standardov
Uporaba alternativnih hierarhij	
	Upravljanje z veliko količino podatkov
Uporaba pravil za združevanje	
	Mnogo orodij za poslovno obveščanje
Uporaba neuravnoteženih hierarhij	
	Zgodovina
	Primernost za transakcije

Preglednica 4: Kocke in relacijski dimenzijski modeli, prednosti

Relacijske podatkovne baze so danes nepogrešljive. Uporabljajo se tako za transakcijske kot za analitične sisteme in se v svojem bistvu le malo spreminjajo. Na drugi strani so večdimenzijske baze manj zrele, saj se še vedno močno razvijajo in nadgrajujejo. Zrelost tudi

pomeni, da obstaja več strokovnjakov, ki takšno tehnologijo poznajo, kar pomeni manjše tveganje in nenazadnje vodi v manjše stroške.

Glavna prednost kock je hitrost poizvedovanja, saj so podatki na več nivojih izračunani vnaprej. Zvezdne sheme in snežinke je sicer mogoče optimizirati z uporabo indeksov, materializiranih pogledov in particij, a s tem optimiziramo le določen nabor poizvedb, ne pa vseh.

Vodilne relacijske podatkovne baze zagotavljajo podporo za mnogo standardov, kar pomeni, da so si navzven podobne. Posledica tega je, da je mogoče en SUPB zamenjati z drugim, kar pa v realnosti ni tako enostavno. Le redki so primeri, ko organizacija zamenja ponudnika, saj vseeno obstaja precej posebnosti, poleg tega pa to zahteva tudi drugačna znanja skrbnikov teh sistemov. Zamenjava večdimenzijske podatkovne baze pa danes pomeni, da je potrebno pripadajočo infrastrukturo postaviti na novo, vključno z logiko in podatkovnim modelom.

Uporaba alternativnih hierarhij je sicer mogoča tudi v relacijskih bazah, a je pri tem potrebno uporabiti trike in sprejemati določene kompromise, največkrat v škodo preglednosti in hitrosti poizvedovanja. Znotraj večdimenzijske baze, ki to omogoča, je to nekaj naravnega.

Relacijske baze znajo upravljati z ogromno količino podatkov, sposobnosti večdimenzijskih baz pa se tudi zelo izboljšujejo. Vseeno velja trditev, da za zelo velike količine podatkov slednje niso primerne. Odgovor na vprašanje, kje je ta meja, je odvisen od orodja. Nekateri ponudniki trdijo, da lahko njihova orodja obvladujejo terabajte podatkov.

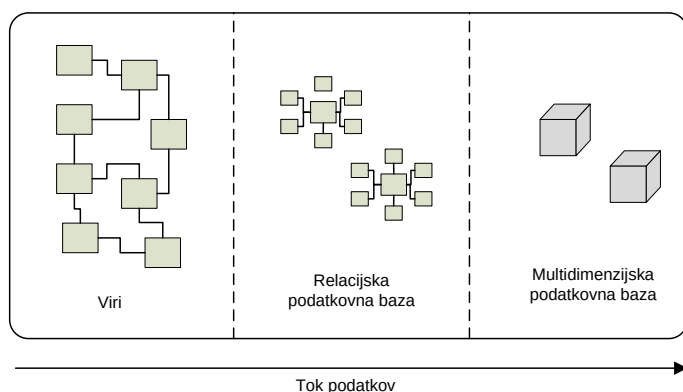
Pravila za združevanje so močno orodje kock, ki mu relacijski dimenzijski modeli niso kos. V slednjih se je potrebno za iste zmožnosti posluževati trikov, bodisi z dodatnimi meta podatki, bodisi z določanjem pravil za združevanje v orodju za poslovno obveščanje, ali celo na poročilih. Enako velja za neuravnotežene hierarhije.

Na trgu je veliko orodij za poslovno obveščanje. Mnogo od njih ponuja možnost dela s kockami in možnost dela s kockami in relacijskim modelom naenkrat, a je ta možnost še vedno v začetni fazi, saj je možno le delo s kockami na način, kot da bi bile relacijske. Velja, da so najboljša orodja za poizvedovanje nad večdimenzijskimi podatkovnimi bazami še vedno tista, ki so bila narejena posebej za ta namen. Orodja, ki delajo nad relacijskimi bazami se lahko brez težav priklopijo na vse večje relacijske SUPB. Na drugi strani so tista, ki delajo nad kockami, pogosto omejena le na en tip večdimenzijskega SUPB.

Ohranjanje zgodovine sprememb v dimenzijah je v relacijskem okolju enostavno, v večdimenzijskem pa na tem področju ni veliko možnosti. Orodja ponujajo le spreminjaje tipa 1.

Kocke so manj primerne za transakcije, saj podatki v njih pogosto niso na najnižjem nivoju podrobnosti, poleg tega pa bi morale vsako transakcijo združiti še na višje nivoje podrobnosti, kar zahteva svoj čas. Zaradi tega niso najbolj primerne za operativno hrambo podatkov ali celo za transakcijske sisteme, čeprav jih najdemo tudi v njih. Tak primer je sistem za planiranje, ki dela s hierarhičnimi podatki, kjer je dimenzijski model nekaj naravnega. So pa kocke zelo primerne za področna podatkovna skladišča, kjer se tudi v praksi najpogosteje uporabljajo.

Relacijski dimenzijski model in kocke so v podatkovnem skladišču običajno povezani. Slika 11 prikazuje tak primer.



Slika 11: Povezanost relacijskega in večdimenzijskega modela

Podatki se iz virov prenesejo v dimenzijski model, dimenzijske tabele pa se uporabijo za izdelavo dimenzij v kockah, kar se zgodi ob določenih periodah. Dejstva se v kocko redno prenašajo iz tabel dejstev. Dimenzijske tabele so lahko opremljene z dodatnimi meta podatki, ki natančneje opredeljujejo obnašanje dimenzij. Tak primer je dodatno polje s pravilom za združevanje, ki za samo dimenzijsko tabelo ne pomeni nič, ob prenosu v kocko pa določa način združevanja. Prenosi dejstev se tipično izvajajo pogostejše kot prenosi dimenzij, katerih naloga je pravzaprav osveževanje strukture kocke. Prenosi dejstev se lahko izvajajo inkrementalno, lahko pa vedno v celotnem obsegu.

4 Drugi pristopi

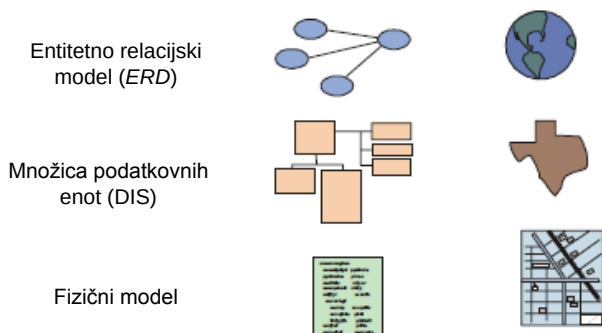
4.1 Podatkovni modeli v okolju DW 2.0

4.1.1 Uvod

Ogrodje DW 2.0 je široko. Predstavlja vse podatke v organizaciji in širše. Poleg tega predstavlja tako starejše kot današnje podatke. V primerih velikih, mednarodnih organizacij je obseg lahko ogromen. Kljub temu DW 2.0 predvideva najnižji možni nivo podrobnosti. Iz takega seznama zahtev sledi, da načrtovanje in izgradnja takšnega sistema nista preprosta. Modeliranje zahteva pregled nad celotno sliko, medtem ko se osredotočamo na podrobnosti.

Podatkovni model, ki mu lahko rečemo tudi podatkovni model skladišča, se uporablja za povezano, starejše in arhivsko področje. Interaktivno področje je posebno, saj se ob izdelavi uporabljajo tudi procesni modeli, ki jih sestavljajo strukturni diagrami, diagrami podatkovnih tokov in podobni, ki se uporabljajo v transakcijskih sistemih.

Podatkovni model, ki predstavlja jedro načrtovanja za spodnja tri področja v ogrodju DW 2.0 ima tri različne nivoje [22]. Prikazani so na sliki 12.



Slika 12: DW 2.0, trije nivoji podatkovnega modela [22]

Najvišji nivo predstavlja entitetno relacijski model (angl. *entity relationship diagram*, *ERD*), množica podatkovnih enot (angl. *data item set*, *DIS*) predstavlja srednji nivo, na najnižjem nivoju pa je fizični model. Pravzaprav so vsi lahko predstavljeni z entitetno relacijskim modelom, zato bo v izogib zmedi za sklicovanje na najvišji nivo odslej uporabljena kratica *ERD*. Vsak od treh nivojev je pomemben del podatkovnega modela. Vsak je tudi edinstven, a povezan z ostalima. Predstavlja svojstven pogled na podatke. Znotraj organizacije obstaja samo en *ERD*, *DIS*ov in fizičnih modelov pa je več. Na osnovi *ERD* se izdelajo *DIS*i, na osnovi slednjih pa fizični modeli.

4.1.2 ERD

Slika 12 kaže, da nivo modeliranja *ERD* odraža najvišjo stopnjo abstrakcije. Če povlečemo vzporednice, je analogen pogledu na zemeljsko oblo. Kot tak *ERD* predstavlja vse podatke, tako kot globus prikazuje vse kontinente in večino držav. Predstavlja grobe povezave ene skupine podatkov z drugo skupino podatkov, tako kot so kontinenti postavljeni eden zraven drugega. Bistvo *ERD* je popolnost modela in visoka stopnja abstrakcije.

Sestavljen je iz dveh glavnih gradnikov. Prvi so entitetni tipi organizacije ali njena glavna področja, drugi pa povezave med njimi. Entitetni tip je tu mišljen na najvišji stopnji abstrakcije. Z njim lahko predstavimo na primer stranko, ki lahko vključuje tako fizične kot pravne osebe in tako pretekle kot sedanje stranke. Na najvišjem nivoju so posledično tudi povezave, ki so lahko opremljene s krajšim opisom. Vedno povezujejo natanko dve entiteti. Med tremi ali več niso mogoče. Poseben tip povezave je rekurzivna povezava, ki povezuje dve entiteti istega tipa.

Zaradi visoke stopnje abstrakcije je *ERD* obvladljivo majhen, ne glede na velikost organizacije. Kot tak mora biti zaključen, preden nadaljujemo z modeliranjem ostalih nivojev podatkovnega modela.

4.1.3 DIS

DIS je ekvivalenten zemljevidu velikega ozemlja, kot je na primer Evropa. Popolnost ni njegova lastnost. Ob pogledu na zemljevid Evrope ne pričakujemo, da nam bo pomagal pri iskanju mesta na Filipinih. Je pa tak zemljevid mnogo bolj podroben kot globus. Najti je mogoče mesta kot so Gorica, Celovec, Kranj in podobna, ki jih na globusu ni. *DIS* je torej osredotočen na podmnožico *ERD* in vsebuje bolj podrobne podatke.

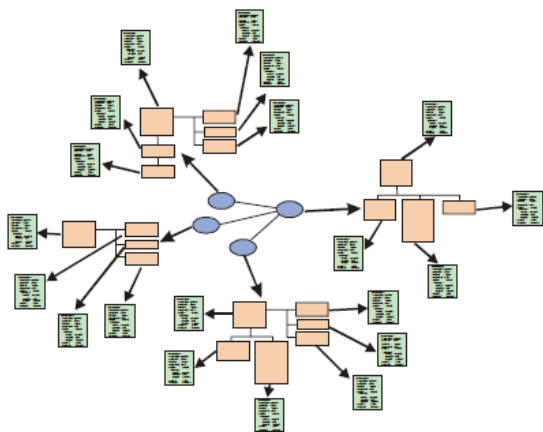
Predstavlja nivo, na katerem se določijo posamezna polja in ključi, skupaj z njihovimi medsebojnimi povezavami. Vsak *DIS* odgovarja natanko enemu entitetnemu tipu v *ERD*, s tem da je bolj podroben.

4.1.4 Fizični model

Najnižji nivo podrobnosti podatkov je predviden v fizičnem modelu. Ta je ekvivalent zemljevidu mesta, kakršno je Gorica. Na njem je zajeto majhno področje, a je zato mogoče na njem najti odgovore na vprašanja, kot kje se nahaja univerza in kakšna imena imajo ulice. Ta model je med vsemi tremi najmanj popoln, a najbolj podroben.

Fizični model odgovarja enemu delu v *DIS*.

Celoten podatkovni model s fizičnim modelom za vse možne entitetne tipe nikakor ni predpogoj za začetek izgradnje okolja DW 2.0. Če bi veljal ta predpogoj, potem nobeno podatkovno skladišče nikoli ne bi bilo zgrajeno. Način gradnje je inkrementalen. Okolje v katerega je skladišče umeščeno je namreč preveč dinamično, da bi bilo mogoče vse narediti naenkrat, brez da bi v okolju medtem prišlo do sprememb. Najprej se zgradi en del skladišča, nato naslednji in tako naprej. Za izgradnjo popolnega skladišča in modela je potrebnih mnogo iteracij. Sestavljanje poteka po koščkih skozi daljše časovno obdobje.



Slika 13: DW 2.0, popoln podatkovni model [22]

Slika 13 prikazuje popoln podatkovni model. V njegovi sredini je en in edini *ERD* s štirimi entitetnimi tipi, iz katerih so narejeni *DISi*. Posamezni deli teh se nato pretvorijo v fizične podatkovne modele, na osnovi katerih se zgradi sistem DW 2.0.

4.1.5 Izdelava

Prvi korak pri izdelavi podatkovnega modela je prepoznavanje glavnih področij ali entitetnih tipov na najvišjem nivoju abstrakcije. Ko so določeni, jih je potrebno razdeliti na tiste z osnovnimi in tiste z izpeljanimi podatki. Osnovni podatki so tisti, ki jih ne moremo razdeliti na bolj podrobne, izpeljani pa tisti, ki jih lahko izračunamo iz bolj podrobnih. Primer osnovnih je ime stranke, primer izpeljanih pa število strank.

Na osnovi osnovnih podatkov se zgradi podatkovni model, na osnovi izpeljanih pa ne. Razlog je v tem, da se izpeljani podatki spreminjajo hitreje kot jih lahko modeliramo. Njihove opredelitve, izračuni in uporaba niso konstantni. Poleg tega bi napihnil model, ga naredili nepreglednega in neprilagodljivega. Še več, to bi povzročilo, da model nikoli ne bi bil popoln, saj se ti podatki nenehno spreminjajo.

Naslednji korak je izbira področja, ki ga bomo razvili. V nekaterih primerih je področje preveliko, zato je potrebno izbrati njegovo podmnožico. Pri izbiri velja upoštevati, da

izberemo področje, ki še ni razvito in je čim bliže jedru dejavnosti organizacije. Na tej osnovi se izdelava *DIS*.

Med izdelavo na osnovi poslovnih zahtev za področje izdelamo tabele, polja in ključe. Slednji so na tem nivoju le identifikatorji in lahko določajo enoličnost, lahko pa je ne. Omenjeni elementi so razdeljeni na tiste za:

- temeljne podatke (angl. *root data*),
- tipske podatke (angl. »type of« *data*) in
- ponavljajoče se podatke (angl. *multiply occurring data*).

Temeljni podatki so tisti, ki opisujejo osnovne enkratne značilnosti področja. Če je področje stranka, je takšna značilnost njen naziv. Tipski podatki opisujejo različne tipe podatkov znotraj področja. Tip stranke je tako lahko fizična ali pravna oseba. Ponavljajoči podatki opisujejo večkratne značilnosti področja. Stranka se tako lahko ukvarja z več dejavnostmi.

Polja, z operativnimi podatki, ki niso primerni za analitične namene, se zbrisejo. Tak primer je lahko telefonska številka, a ne vedno. To se naredi po tem, ko je *DIS* že zgrajen. Pri pregledu se polja brez analitične vrednosti odstranijo. Na drugi strani se, če še ni prisotna, doda časovna komponenta.

Izvorni podatki za DW 2.0 lahko imajo vse potrebne vrednosti, lahko pa ne. Kjer jih na izvoru ni, moramo zagotoviti privzete vrednosti.

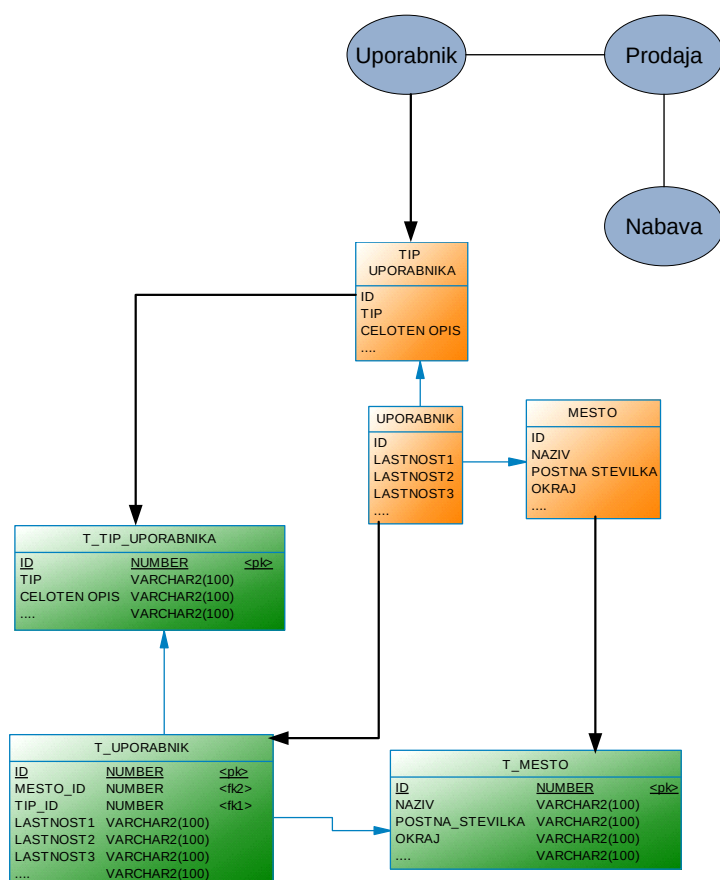
Obstaja tudi možnost združevanja skupin podatkov. Tako združevanje ima smisel, kadar se podatki vedno ali skoraj vedno uporabljajo skupaj, v paketu, pa čeprav se logično razlikujejo. Torej, če se ob običajni uporabi dve skupini podatkov uporabljata skupaj, se lahko ti skupini združita v eno fizično skupino. Tak primer je združevanje zneskov po mesecih v en zapis, kljub temu, da na izvoru obstaja za vsak mesec svoj zapis.

Fizični model izdelamo, ko je *DIS* zaključen. Vsebuje vse značilnosti, ki so potrebne za fizično izgradnjo. Tako so opredeljena imena tabel, indeksi, particije, materializirani pogledi in podobno.

4.1.6 Primer

Na sliki 14 je prikazan poenostavljen primer razvoja podatkovnega modela v ogrodju DW 2.0. Najprej je potrebno izdelati *ERD*, model na najvišji stopnji abstrakcije, ki pokriva vsa področja znotraj organizacije. Prikazan je z modro barvo. V primeru na sliki se nato odločimo za razvoj najbolj pomembnega področja, področja uporabnikov, za katerega se naredi *DIS*, v

katerem so že določena posamezna polja in ključi. Področje *DIS* je na sliki prikazano z oranžno barvo. Na njegovi osnovi se nato izdela fizični model v katerem so določene tudi fizične lastnosti tabel. Fizični model je prikazan z zeleno barvo.



Slika 14: Primer razvoja podatkovnega modela v ogrodju DW 2.0 za področje uporabnika

4.2 Anchor modeling

4.2.1 Uvod

Vzdrževanje in razvijanje podatkovnih skladišč je zahtevna, obsežna in časovno zahtevna aktivnost. Glavni razlog za to je nenehno spreminjajoče se okolje, medtem ko mora samo skladišče predstavljati stabilen in dosleden vmesnik do informacij, ki so razporejene skozi daljše časovno obdobje. Izvori, ki zagotavljajo podatke, se skozi čas nenehno spreminjajo, občasno tudi v večjem obsegu. Analitične in poročevalske potrebe po informacijah prav tako niso konstantne. Zaradi teh izzivov morajo biti podatkovni modeli skladišč modularni in prilagodljivi, da se lahko prilagajajo spremembam. Če model temu ne ustreza, lahko to predstavlja večjo težavo. Tretjina izdelanih skladišč v določenem trenutku, običajno v prvih štirih letih, spremeni svojo arhitekturo in manj kot tretjina teh navaja uvedbo skladišča kot uspeh [15].

Anchor modeling (AM) je tehnika za načrtovanje podatkovnih modelov, ki uporablja razširljive, nedestruktivne mehanizme, s katerimi zagotavlja robustno in prilagodljivo vpeljavo sprememb. Kadar v okolju pride do večjih sprememb, kot je menjava izvirnega sistema, je v takšnem modelu potrebna le manjša sprememba. Manjša potreba po spremembah modela podaljša življenjski čas skladišča, zmanjša čas za implementacijo in poenostavi vzdrževanje.

Podatkovno skladišče zgrajeno na osnovi AM zahteva le razširitve, ne pa sprememb modela, kar je ena ključnih prednosti. Orodja za poslovno obveščanje in druge aplikacije, ki uporabljajo skladišče, tako niso obremenjeni s spremembami modela. Poleg tega razvoj skozi razširitve in ne spremembe povzroči modularnost, kar omogoča razdelitev modela na manjše, stabilne sestavne dele, s katerimi je mogoče upravljati na enostaven način. Takšna modularnost je lahko še posebej koristna ob agilnem razvoju programske opreme, kjer so zaželeni kratki cikli. Začetni model je mogoče sestaviti na osnovi majhnega števila poslovnih terminov, ki ga lahko kasneje razširimo do končnega modela [12].

Model AM je relacijski model z visoko stopnjo normalizacije in z zmožnostjo shranjevanja zgodovinskih podatkov. Uporablja majhno število konstruktov, ki jih dopolnjujejo smernice za njihovo sestavljanje. Velika razčlenjenost je posledica tega, da se vsak atribut preslika v svojo tabelo. Gre torej za diametralno nasproten pristop v primerjavi z dimenzijskimi modeli.

Temelji na šesti normalni obliki [27], kar pomeni, da je vsako dejstvo ali lastnost predstavljeno s svojo tabelo s ključem in vrednostjo ter neobvezno zgodovinsko informacijo. V tretji normalni obliki lahko ena tabela vsebuje več polj z lastnostmi, kar pa v šesti normalni obliki ni dovoljeno. Tabela je v šesti normalni obliki, kadar je ni mogoče dodatno razčleniti na tabele z manj atributi [12].

4.2.2 Gradniki

Osnovni gradniki podatkovnega modela so sidra (angl. *anchor*), vozli (angl. *knots*), atributi in vezi (angl. *ties*). Metapodatki, kot so izvor, čas nastanka, čas spremembe, številka obdelave in podobni, so sicer tudi del modela AM, a je njihova uporaba povsem enaka, kot pri vseh drugih tehnikah načrtovanja. Slika 15 prikazuje grafično predstavitev osnovnih gradnikov AM [12]. Predstavitev atributov in vozlov lahko grafično še dodatno razširimo z dodatnim okvirjem, s katerim določimo, da gre za zgodovinski gradnik, kot je opisan v nadaljevanju.



Slika 15: Osnovni gradniki AM

Sidro

Sidro v podatkovnem skladišču predstavlja množico dogodkov ali predmetov, kot sta na primer stranka in nakup. Vedno je tehnične narave in predstavljeno z umetnim in ne naravnim ključem. Zaloga vrednosti so zato običajno naravna števila. Edino obvezno polje je umetni ključ, dopolnjujejo ga le meta podatki.

Vozel

Vozel se uporablja za predstavitev nespremenljive množice lastnosti, ki se skozi čas ne spreminja. Sidra predstavljajo poljubne dogodke ali predmete, vozli pa se uporabljajo za lastnosti, ki so skupne več instancam istega sidra. Tipični primer vozla je spol, ki vsebuje le dve vrednosti - moški in ženska, vsako sidro človek pa pripada enemu od njih. Uporaba vozla tako zmanjša redundanco. Enkratni zapis je dovolj, ponavljanje ni potrebno. Vozel ima obvezno vsaj dve polji. To sta umetni ključ in vrednost.

Atribut

Atributi predstavljajo lastnosti sider. Delijo se na štiri vrste:

- statični,
- zgodovinski,
- statični z vozli in
- zgodovinski z vozli.

Statični se uporabljajo za predstavitev poljubnih lastnosti sider, kadar ni potrebe po beleženju zgodovine sprememb vrednosti atributa. Za to beleženje se uporabljajo zgodovinski atributi, katerih vrednost je veljavna, dokler je ne nadomesti naslednja. Obdobje veljavnosti ima točno določen čas začetka. Čas zaključka veljavnosti pa je enak času začetka veljavnosti naslednje vrednosti. Statičen atribut z vozli se uporablja za predstavitev povezav med sidri in vozli, torej za povezovanje sider s statičnimi lastnostmi. Zgodovinski atribut z vozli se uporablja, kadar povezave z vozli niso stalne, torej se lahko skozi čas spreminjajo. Atribut vedno

vsebuje vsaj tuji ključ, ki se navezuje na pripadajoči umetni ključ v sidru in natanko eno vrednost. Če gre za katerega od zgodovinskih vrst, vsebuje tudi zgodovinsko informacijo, to je datum začetka veljavnosti. Kadar gre za vrsto z vozli, vsebuje tudi tuji ključ, ki se navezuje na pripadajoči umetni ključ v vozlu. Primer atributa je barva (na primer avtomobila).

Vez

Za povezave med gradniki se uporabljajo vezi, ki med sabo povezujejo dve, lahko pa tudi več gradnikov. Tako kot atributi se delijo na enake štiri vrste, ki imamo tudi enako vlogo:

- statične,
- zgodovinske,
- statične z vozli in
- zgodovinske z vozli.

Vez vsebuje vsaj tuja ključa, ki se navezujeta na pripadajoča umetna ključa v sidru. Zgodovinski vrsti vsebujeta še zgodovinsko informacijo, vrsti z vozli pa tuji ključ, ki se navezujeta na pripadajoči umetni ključ v vozlu, na enak način kot pri atributu. Vez lahko povezuje na primer eno stranko z drugo.

4.2.3 Smernice pri načrtovanju

Ključni entitetni tipi področja morajo biti v modelu *AM* predstavljeni s sidri. Pogosto vprašanje modeliranja je, ali naj transakcijo predstavimo kot sidro ali kot vez. Dobra praksa pravi: za modeliranje ključnih entitetnih tipov in transakcij uporabimo sidra. Transakcijo lahko modeliramo kot vez, a le v primeru, ko nima nobenih lastnosti.

Zgodovinske attribute uporabimo, kadar je njihovo verzioniranje smotno. Eden od namenov podatkovnega skladišča je tudi ohranjanje zgodovine sprememb, ki so se zgodile skozi čas. Z zgodovinskimi atributi dosežemo prav to. Kadar instanca atributa predstavlja kategorijo ali kadar ima njena zalog vrednosti majhen nabor vrednosti, potem se uporabi atribut z vozli. Enako velja za vezi.

4.2.4 Polnjenje in zmogljivosti

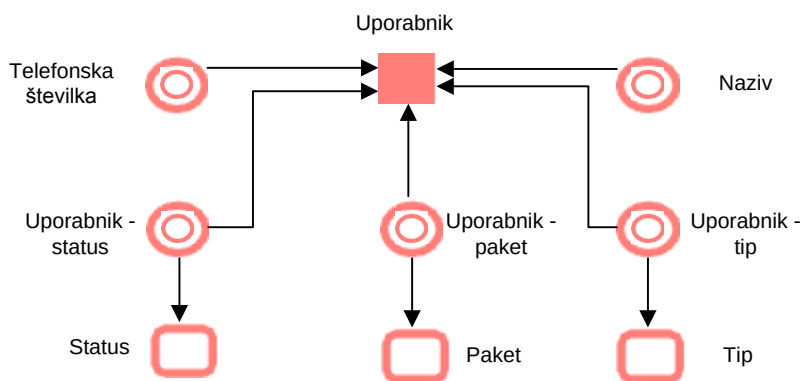
Model *AM* omogoča polnjenje brez sprememb že obstoječih zapisov. To pomeni, da se dogajajo le vnosi. Podatki se le dodajajo, nikoli spreminjajo. Brisanje podatkov ni dovoljeno, možno je le v primeru odprave napak. Polnjenje brez sprememb omogoča, da se postopki polnjenja izvedejo hitreje, saj je vnos v primerjavi s spremembo cenejša operacija.

Na drugi strani moramo zagotoviti hitre odzivne čase za poizvedovanje. Eden od načinov je uporaba grupnih (angl. *clustered*) indeksov nad primarnimi ključi, ki ne porabljajo dodatnega prostora. Organizirani morajo biti nad umetnimi ključi in zgodovinskimi polji v padajočem

redu. Na ta način dosežemo, da so podatki na nosilcu fizično organizirani tako, da je zadnja verzija za katerikoli ključ dostopna prva.

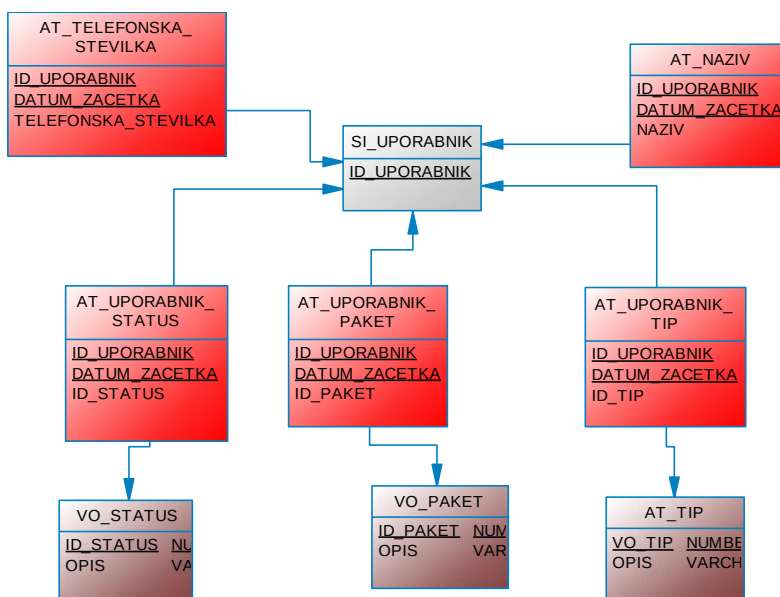
4.2.5 Primer

Na sliki 16 je primer modela *AM* za področje uporabnikov. Informacije, ki jih vsebuje so naziv uporabnika, njegova telefonska številka, tip, status in paket, ki ga uporablja. Jedro modela je sidro, s katerim je predstavljen uporabnik. Status, paket in tip so predstavljeni z vozli, saj gre za nespremenljive zaloge vrednosti. Tako ima tip na primer le dve možni vrednosti, ki sta »Poslovni uporabnik« in »Zasebni uporabnik«. Telefonska številka in naziv sta na drugi strani predstavljeni neposredno z atributi, saj nimata nespremenljive zaloge vrednosti. Uporabnika in lastnosti predstavljene z vozli povezujejo atributi. Vsi atributi so obkroženi, kar pomeni, da se lahko s časom spreminjajo, torej vsebujejo tudi zgodovinsko komponento.



Slika 16: Podatkovni model, notacija AM

Tak model se neposredno preslika v entitetno relacijski model prikazan na sliki 17. Vsak od gradnikov je v njem predstavljen z enim entitetnim tipom, ki se nato preslika v eno fizično tabelo. To velja tudi za attribute in vezi. Tabele so razporejene enako kot v notaciji AM.



Slika 17: Relacijski podatkovni model

Vse tabele vsebujejo umetni primarni ključ, katerega ime se začne s predpono ID_. Tiste, ki se preslikajo iz vozlov in atributov brez vozlov imajo navedeno lastnost. To so opisi, naziv in telefonska številka. Vsi atributi so zgodovinski, zato imajo še datum začetka. Vse tabele lahko vsebujejo še morebitne metapodatke, ki pa na sliki niso prikazani.

4.2.6 Uporaba podatkov

Zaradi velikega števila tabel in pripadajoče zapletenost modela je smiselno nad njim uvesti dodatni sloj, ki poenostavi poizvedovanje, ga naredi razumljivega uporabnikom in prijaznega orodjem za poslovno obveščanje. Ta sloj pretvori model v tretjo normalno obliko ali v dimenzijsko obliko. Lahko obstaja v obliki pogleda, materializiranega pogleda, ali pa celo v obliki novih tabel. V zadnjem primeru je potrebno pripraviti tudi postopke polnjenja podatkov iz modela AM vanj. Obstajajo trije različni tipi tega sloja:

- **Celoten pogled** na sidro je denormalizacija sidrne tabele in pripadajočih tabel z atributi. Narejen je z levim zunanjim stikom (angl. *left outer join*) sidra z atributi.
- **Zadnji pogled** na sidro je denormalizacija sidrne tabele in pripadajočih tabel z atributi, pri čemer se upoštevajo le zadnje vrednosti atributov.
- **Trenutni pogled** na sidro je denormalizacija sidrne tabele in pripadajočih tabel z atributi, pri čemer se upoštevajo le atributi, ki so bili veljavni v določenem trenutku. Datum, ki določa ta trenutek, je osnova za ta pogled.

4.2.7 Prednosti in slabosti

Prednosti AMja, povzetih po [12] in [26] je veliko.

- Prilagodljivost. Velika sprememba v okolju ima kot posledico le majhno spremembo modela.

- Neodvisnost. Model je neodvisen od poslovne logike. Pravila so opisna in ne fizična, kar zagotavlja daljšo življenjsko dobo skladišča.
- Določanje obsega. Modularna zasnova omogoča ločevanje in olajša določanje obsega projekta.
- Modularnost. Določeni moduli se lahko razvijejo neodvisno od drugih. Lahko začnemo z osnovnimi in nato dodajamo ostale.
- Razširljivost. Vsaka sprememba je implementirana kot neodvisna nadgradnja. Na obstoječe aplikacije zato nima vpliva.
- Enostavnost vzdrževanja. Dosledna uporaba *AM* pomeni enostavno vzdrževanje. Popravki napak na enem atributu ne vplivajo na drugega.
- Enostavni koncepti in notacija. Modeli *AM* so zgrajeni z uporabo malega števila enostavnih konceptov.
- Ohranjanje zgodovine. Ohranjanje različnih verzij je enostavno, saj za to obstajajo temu prilagojeni elementi.
- Primernost za iterativni in inkrementalni razvoj. *AM* omogoča iterativni in agilni pristop k razvoju, saj dovoljuje neodvisno delo na majhnih podmnožicah modela.
- Ponovna uporaba. Na voljo je majhno število gradnikov, kar zagotavlja čvrsto strukturo, ki se lahko ponovno uporabi.
- Primernost za polnjenje v realnem času. Podatki se nikoli ne spreminjajo, kar olajša polnjenje.
- Učinkovito shranjevanje. *AM* zahteva manj prostora v podatkovni bazi. Vzroki so visoka stopnja normalizacije, odsotnost praznih vrednosti, in dejstvo, da ohranjanje zgodovine ne zahteva podvajanja podatkov. Manj normalizirani modeli zahtevajo več prostora.

Na drugi strani je tudi nekaj slabosti:

- Zahtevno poizvedovanje. Za poenostavitev poizvedovanj je potrebno nad modelom izdelati dodatni sloj.
- Orodja za poslovno obveščanje niso združljiva z *AM*. Narejena so za delo z normaliziranim modelom ali modelom v tretji normalni obliki. Tudi zato je potreben dodatni sloj.
- Majhna razširjenost. Takšnih modelov je v praksi malo, zato je tudi izkušenj in praks manj.
- Nepoznavanje med načrtovalci. Ta točka je povezana s prejšnjo. Če se bo število implementacij *AM* povečalo, se bo to spremenilo. Okrepilo se bo tudi zaupanje v takšen način dela.

Tak model je najbolj primeren za celovito podatkovno skladišče. Odgovarja vsem petim točkam osnovne opredelitve. Zaradi svojih lastnosti je primeren tudi za polnjenje v realnem času, tako da se lahko uporabi za aktivno podatkovno skladišče. Pri tem je zanimivo, da takšno aktivno skladišče pri tem še vedno zadosti zahtevi po ohranjanju zgodovine, kar sicer ni njegova tipična lastnost.

Med vsemi načini modeliranja predvideva *AM* največjo stopnjo normalizacije. Po tem kriteriju si najbolj uporabljeni načini od največje stopnje proti najmanjši sledijo tako:

- 1) *AM*,
- 2) tretja normalna oblika in
- 3) dimenzijski modeli.

Razlike v količini podatkov sicer niso velike, saj največjo količino podatkov predstavljajo dejstva, za katera pa nobeden od treh načinov ne predvideva denormalizacije.

4.3 Kakovost podatkov

4.3.1 Uvod

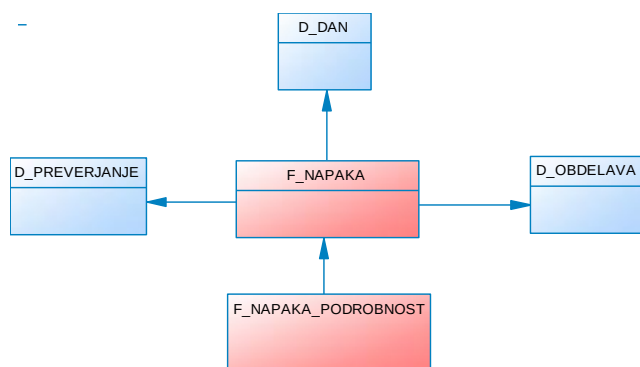
Zagotavljanje kakovosti je v prvi vrsti naloga podatkovnih izvorov in postopkov *ETL*. Na prve pogosto nimamo vpliva, v drugih pa je to ena ključnih nalog. Koraki postopkov, ki zagotavljajo kakovost podatkov, so tisti, ki prinašajo dodano vrednost. Ostale aktivnosti postopkov so seveda obvezne, a podatke le prenašajo in nalagajo [9].

Z zagotavljanjem kakovosti želimo omogočiti, da bo podatkovno skladišče vir kakovostnih informacij. Te so opredeljene kot pravi podatki, s pravo mero popolnosti, v pravem kontekstu, s pravo točnostjo, v pravem formatu, ob pravem času, na pravem mestu in s pravim namenom [1].

Z dobrim podatkovnim modelom lahko veliko pripomoremo h kakovosti, saj model med drugim uveljavlja razne omejitve preko normalizacije, ključev in povezav med podatki. Vključimo lahko tudi strukture, namenjene izključno zagotavljanju kakovosti. Dva takšna načina [9] sta opisana v nadaljevanju, uporabljena pa sta tudi v primeru rešitve v telekomunikacijski panogi.

4.3.2 Shema za spremljanje napak

Shema za spremljanje napak predstavlja centraliziran dimenzijski model, katerega namen je beleženje napak, ki jih vračajo preverjanja vgrajena v postopke *ETL*. Ta pristop lahko poleg podatkovnih skladišč uporabimo za vsako premikanje podatkov, na primer pri prenosu iz starega sistema v novega. Model je prikazan na sliki 18.



Slika 18: Shema za spremljanje napak

Glavna tabela je tabela dejstev z napakami *F_NAPAKA*. Njeno zrno je napaka, ki jo vrne preverjanje, ki se lahko nahaja kjerkoli v postopku *ETL*. To pomeni da vsaka napaka, kot posledica preverjanja, prispeva natančno en zapis v tabelo, en zapis v tabeli pa pripada natanko eni napaki.

Dimenzije te tabele so časovna dimenzija *D_DAN*, ki nosi informacijo o koledarskem datumu pojavitve napake, dimenzija obdelav *ETL* *D_OBDELAVA*, ki pove znotraj katere obdelave se je napaka pojavila in dimenzija preverjanj *D_PREVERJANJE*, ki pove katero preverjanje je povzročilo napako. Natančnost časovne dimenzije je en dan. Na ta način zagotavlja način za omejevanje in seštevanje napak po običajnih poljih koledarja, kot so delovni dan ali zadnji dan v nekem obdobju. Podrobnejši trenutek v dnevu je zapisan neposredno v tabeli dejstev v svojem polju kot dejstvo. Z njegovo pomočjo lahko na primer določimo točen čas med dvema napakama.

Dimenzijo obdelav *ETL* lahko posplošimo v dimenzijo korakov prenosa, kadar se podatki ne prenašajo v skladišče preko obdelav ampak preko drugačnega toka podatkov, kot na primer v primeru prenosa v realnem času. Dimenzija preverjanj natančno določa preverjanja in kje se nahaja koda za njih. Prav tako določa kaj se zgodi v primeru ko preverjanje najde napako. Postopek se lahko zaustavi, zapiše poročilo o napaki, označi podatke ali le pošlje kratko sporočilo odgovornim osebam.

Tabela dejstev ima tudi polje s primarnim ključem. Ta umetni ključ je, tako kot umetni ključi v dimenzijskih tabelah, predstavljen z naravnim številom, ki ga proizvede sekvenca. Tak ključ je obvezen zaradi situacij, ko se v tabelo naenkrat prenese velika množica napak.

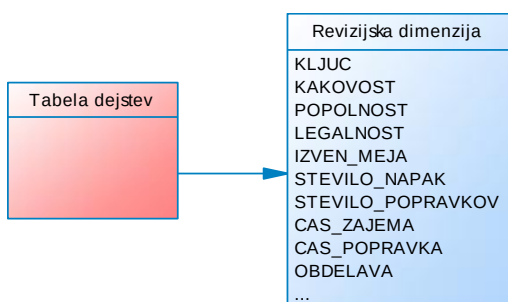
Shema vključuje tudi drugo bolj podrobno tabelo dejstev *F_NAPAKA_PODROBNOST* z zrnem na nižjem nivoju. Vsak zapis v tej tabeli predstavlja posamezno polje v določenem zapisu, ki je povzročil napako. Tako lahko preverjanje, ki povzroči le en zapis v glavni tabeli

napak, v tej tabeli povzroči več zapisov. Ti dve tabeli sta povezani preko primarnega ključa tabele napak, na katerega se povezuje polje v podrobnejši tabeli. Ta vsebuje tabelo, zapis, in polje z napako, poleg tega lahko podeduje tudi dimenzije nadrejene tabele.

S takšnim modelom je mogoče enostavno beležiti napake, mogoče je enostavno dostopati do posameznih napak, mogoče pa je tudi analizirati napake v daljšem obdobju. Tako lahko na enostaven način ugotavljamo, kateri izvor povzroča največ napak, katere vrste napak so se dogajale, kaj je za napake pomenila vpeljava novega modula in podobno. Ker je model dimenzijski, ga orodja za poročanje znajo izkoriščati, kar pomeni, da lahko nad njim enostavno izdelamo poročila.

4.3.3 Revizijska dimenzija

Revizijska dimenzija je posebna oblika dimenzijske tabele, ki služi kot pomoč za zagotavljanje kakovosti. Vsaka pripada eni tabeli dejstev, polni pa se preko postopkov ETL. Dimenzija na sliki 19 vsebuje kontekst meta podatkov v trenutku, ko nastane določen zapis v tabeli dejstev. Meta podatke smo na ta način pretvorili v prave podatke, saj postanejo del dimenzijskega modela.



Slika 19: Revizijska dimenzija v povezavi s tabelo dejstev

Uporabo je najlažje opisati s primerom. Recimo da je tabela dejstev na sliki tabela zadev na sodišču, ki se polni vsak dan. Na dan, ko postopek *ETL* ne zabeleži nobenih napak, se v revizijsko dimenzijo vnese natanko en zapis, ki pomeni, da je z zapisom vse v redu, nanj pa so povezani vsi zapisi iz tabele dejstev, ki so nastali na ta dan. Torej imajo vsi enake vrednosti za vsa polja. Če pa postopki zabeležijo napako, se v dimenzijo za vsako vnese dodatni zapis. Zapisi v tabeli dejstev, nad katerimi se je zgodila napaka, se povežejo nanj. Tak zapis v dimenziji nosi podatke o napaki.

Poleg ključa so lahko v revizijski dimenziji še različna polja, ki so za pripadajočo tabelo dejstev smiselna. Splošna so naštetna na sliki, njihov pomen pa določijo postopki polnjenja. Izven meja je klasična kontrola, s katero na primer ulovimo negativne ali prevelike zneske na računih. Dodamo lahko še poljubna polja, vezana na konkretno tabelo dejstev. Za tabelo zadev na sodišču bi bila smiselna polja, ki bi nosila informacijo o tem, ali so postopki *ETL*

uspeli določiti sodnika, ali je zadeva že sodni zaostanek, ali obstaja številka zadeve in podobno.

Kontrole iz sheme za spremljanje napak lahko uporabimo tudi za polnjenje revizijske dimenzije. Na ta način lahko oba opisana načina učinkovito združimo.

Ta dimenzija ne spreminja zrna tabele dejstev. Uporabniške poizvedbe je običajno ne vključujejo, medtem ko je za skrbnika skladišča zelo uporabna za odkrivanje napak in za določanje stopnje kakovosti v celotnem skladišču. Cilj je seveda imeti čim več zapisov brez napak.

4.4 Vnaprej pripravljeni modeli

Podatkovni modeli se lahko zgradijo posebej za konkretno podatkovno skladišče, torej so »pisani na kožo«. Slabost takšnega pristopa je draga izgradnja, saj lahko vzame veliko časa. Ni pa to edini pristop. Če izhajamo iz tega, da sta podatkovna modela za dve organizaciji znotraj iste dejavnosti podobna, lahko podatkovni model zgradimo na osnovi nekega drugega, že obstoječega, kar je tudi eden od sodobnih pristopov k izgradnji.

Kot telekomunikacijsko podjetje lahko uporabimo model, ki ga je zgradilo drugo telekomunikacijsko podjetje, namesto da na novo odkrivamo težave. Njihov model lahko bodisi kupimo ali najamemo in ga preuredimo tako, da bo ustrezal našim potrebam. Tak splošen model lahko uporabimo hitreje, poleg tega pa še prihranimo v primerjavi s pristopom, ko začnemo od začetka.

Vnaprej pripravljen model lahko najdemo na spletu, ali pa ga kupimo od enega od ponudnikov, kakršnih je danes kar nekaj. Uporabne brezplačne modele najdemo na spletni strani *Corporate Information Factory* [28]. Nekatera od podjetij, ki ponujajo plačljive modele, so oracle, IBM in sybase. Nekatera podjetja ponujajo tudi rešitve v kombinaciji s postopki za polnjenje in celo z že izdelanimi vnaprej pripravljenimi poročili. Takšne rešitve v ponudbi običajno dopolnjujejo posamezen transakcijski sistem in so navezane izključno nanj. Zato so manj primerne za splošno uporabo.

Ko se odločamo za vnaprej pripravljen model, se moramo zavedati, da ne bo zadostil vsem našim potrebam, pa naj bo še tako popoln in preizkušen. Do določene mere ga bomo morali spremeniti in prilagoditi. Obstajajo tako modeli za transakcijske modele in tisti za podatkovna skladišča. Če gradimo skladišče moramo seveda paziti, da uporabiti slednjega, prvi pa lahko služi le kot pomoč, a moramo v tem primeru model za podatkovno skladišče zgraditi sami, saj med njima vedno obstajajo razlike.

5 Primerjava pristopov

5.1 Uvod

Vsaka vrsta modelov je primerna za določene zahteve. Obstaja precej lastnosti, na osnovi katerih se mora načrtovalec odločiti za enega izmed njih ali za njihovo kombinacijo. V preglednici 5 je zbranih šest vrst in njihove lastnosti. Prvi štiri vrste so dimenzijske, razdeljene glede na vrsto podatkovne baze in znotraj relacijske glede na predstavljene vrste tabel dejstev. Kocke zahtevajo večdimenzijske podatkovne baze, ostale vrste relacijske. Normaliziran model predstavlja model Inmonovega korporativnega podatkovnega skladišča. Vse vrste so primerne za podatkovna skladišča, podatki v njih so predmetno usmerjeni, povezani in časovno spremenljivi. V primerjavah med relacijskimi modeli je vedno predviden model na najnižjem nivoju podrobnosti, razen v primerih, ko gre za lastnost področnih podatkovnih skladišč, kjer so podatki običajno že združeni na višje nivoje. Relacijski dimenzijski modeli so vsaj v drugi normalni obliki, v normaliziranem modelu v tretji in v modelu AM v šesti. Primernost vrst modelov za določene lastnosti je opisana v nadaljevanju.

Vrsta podatkovnega modela / Lastnost	Dimenzijski model, transakcijske tabele dejstev	Dimenzijski model, tabele dejstev s periodičnimi posnetki stanj	Dimenzijski model, tabele dejstev z zbirnimi posnetki stanj	Kocke	Normaliziran model	Anchor modeling
Vrsta podatkovne baze	Relacijska	Relacijska	Relacijska	Večdimenzijska	Relacijska	Relacijska
Stopnja normalizacije	Druga normalna oblika	Druga normalna oblika	Druga normalna oblika	Ni relacijski model	Tretja normalna oblika	Šesta normalna oblika
Ohranjanje zgodovine	Da (odvisno tudi od tipa dimenzij)	Da (odvisno tudi od tipa dimenzij)	Ne	Ne	Da	Da
Primernost za korporativno podatkovno skladišče	Ne	Ne	Ne	Ne	Da	Ne
Primernost za dimenzijsko podatkovno skladišče	Da	Da	Da	Ne	Ne	Ne

Vrsta podatkovnega modela / Lastnost	Dimenzijski model, transakcijske tabele dejstev	Dimenzijski model, tabele dejstev s periodičnimi posnetki stanj	Dimenzijski model, tabele dejstev z zbirnimi posnetki stanj	Kocke	Normaliziran model	Anchor modeling
Primernost za podatkovno skladišče kot skupek področnih podatkovnih skladišč	Ne	Da	Da	Da	Ne	Ne
Primernost za aktivno podatkovno skladišče	Da	Ne	Da	Ne	Da	Da
Primernost za federativno podatkovno skladišče	Da	Da	Da	Da	Da	Ne
Primernost za celovito podatkovno skladišče	Da	Da	Da (kot dopolnitev transakcijskih tabel)	Ne	Da	Da
Primernost za področno podatkovno skladišče	Ne	Da	Da	Da	Ne	Ne
Primernost za operativno hrambo podatkov	Da	Da	Da	Da	Da	Da
Primernost za polnjenje v realnem času	Da	Ne	Da	Ne	Da	Da
Zahtevnost polnjenja	Nezahtevno (odvisno tudi od tipa dimenzij)	Srednje (odvisno tudi od tipa dimenzij)	Srednje (odvisno tudi od tipa dimenzij)	Zahtevno	Nezahtevno	Nezahtevno
Zahtevnost odprave napak	Nezahtevno	Zahtevno	Srednje	Zahtevno	Srednje	Nezahtevno
Primernost za orodja za poslovno obveščanje	Primerno	Primerno	Primerno	Primerno	Srednje	Neprimerno

Vrsta podatkovnega modela / Lastnost	Dimenzijski model, transakcijske tabele dejstev	Dimenzijski model, tabele dejstev s periodičnimi posnetki stanj	Dimenzijski model, tabele dejstev z zbirnimi posnetki stanj	Kocke	Normaliziran model	Anchor modeling
Preglednost za uporabnike	Pregledno	Pregledno	Pregledno	Pregledno	Srednje	Manj pregledno
Hitrost izvajanja poizvedb	Srednja	Srednja	Srednja	Velika	Srednja	Majhna
Sprejetost	Velika	Velika	Velika	Srednja	Velika	Majhna
Uporaba alternativnih hierarhij	Težka	Težka	Težka	Enostavna	Srednja	Enostavna
Uporaba neuravnoteženih hierarhij	Težka	Težka	Težka	Enostavna	Srednja	Enostavna
Uporaba pravil za združevanje	Težka	Težka	Težka	Enostavna	Težka	Težka

Preglednica 5: Primerjava pristopov

5.2 Ohranjanje zgodovine

Normaliziran model in model AM vedno zagotavljata ohranjanje zgodovine, pa naj gre za transakcijske ali katerekoli druge podatke. Med relacijskimi dimenzijskimi modeli, prvi dve vrsti zagotavljata zgodovino v tabelah dejstev, medtem ko je narava podatkov v modelu s zbirnimi posnetki stanj taka, da se lahko tudi spreminjajo, saj je sprememba zapisa običajna operacija. Zgodovina v spreminjajočih se dimenzijskih tabelah pa se ohranja le v primeru, ko je tip spreminjanja drugi, ali, v primeru le ene spremembe, tretji. V kockah se že zabeležena dejstva lahko spreminjajo ali ne, dimenzije pa ne ohranjajo zgodovine. Tip spreminjanja 2 ni podprt – način uporabljen v relacijskih bazah v kockah ni mogoč. Tudi kocke se glede na ohranjanje zgodovine lahko razdelijo v enake tri skupine kot tabele dejstev.

5.3 Primernost za vrste podatkovnih skladišč

Korporativno skladišče ima že po definiciji model v tretji normalni obliki, ki ohranja vso zgodovino, tako da je zanj primeren le tako normaliziran model. Podobno je dimenzijsko skladišče sestavljeno iz vseh treh relacijskih dimenzijskih modelov. Kocke niso primerne ne za enega, ne za drugega, saj so podatki v njih običajno že združeni, kar pa ni pravilo. Mogoče bi bilo celo zgraditi celotno dimenzijsko skladišče iz kock, a to ni običajno.

Področna podatkovna skladišča vsebujejo podatke, ki so tipično združeni na višji nivo podrobnosti, zato so za njih primerne zadnje tri vrste dimenzijskih modelov, pri čemer so kocke primerne prav za to vrsto skladišča. Transakcijske tabele dejstev so na najnižjem nivoju podrobnosti, zato niso primerne za njih. Postanejo pa primerne s prijemi, kot so uporaba materializiranih pogledov za hitrejše poizvedovanje, a tudi takrat bolj sodijo v celovito dimenzijsko skladišče.

Za aktivno podatkovno skladišče so primerni modeli, ki predvidevajo najnižji nivo podrobnosti podatkov in zajemajo ali transakcije ali stanje v določenem trenutku. Kocke niso primerne zaradi nivoja podatkov, modeli s periodičnimi posnetki stanj pa zaradi periodične narave polnjenja. Transakcijski modeli ne predvidevajo sprememb zapisov, podatki v njih se le nalagajo, kar omogoča enostavnejše transakcije. Zbirni posnetki stanj predvidevajo spremembe zapisov, kar na eni strani zaplete njihovo vzdrževanje, saj je treba zapis pred spremembo najti, a na drugi strani poenostavi strukturo podatkov, saj je zapis o enem objektu vedno en sam. Normaliziran model je na videz zaradi za eno stopnjo večje normalne oblike primernejši za to vrsto skladišča, v resnici pa je od vrste podatkov v njem odvisno kako primeren je, saj lahko nosi tako transakcijske podatke, kot posnetke stanj, tako da je primer podoben kot v primeru relacijskih dimenzijskih modelov. Modeli *AM* so tudi primerni, saj nikoli, ne glede na vrsto podatkov, ne predvidevajo sprememb zapisov, kar omogoča dovolj hitro polnjenje, pri tem pa ohranjajo zgodovino podatkov. Ohranjanje zgodovine sicer ni obvezna lastnost aktivnih skladišč, v primeru transakcijskih podatkov pa jo je enostavno doseči.

Model federativnega podatkovnega skladišča se ne preslika v hrambo fizičnih podatkov, ampak služi le kot vmesnik za dostop do pod njim ležečih podatkovnih baz. Smiselno je ta model načrtovati na način, da bodo uporabniki preko orodij enostavno dostopali do njega. Zaradi tega ni smiselno uporabiti le modela *AM*, ostali so primerni.

5.4 Primernost za dele podatkovnega skladišča

Pod deli podatkovnega skladišča so opisani deli tipične arhitekture podatkovnega skladišča. To so celovito podatkovno skladišče, področno podatkovno skladišče in operativna hramba podatkov. Za celovito podatkovno skladišče so primerni vsi naštetih modeli z izjemo kock, pa tudi te bi bilo mogoče uporabiti v primeru celovitega dimenzijskega podatkovnega skladišča.

Področna podatkovna skladišča so optimizirana za poizvedovanje, zato so kocke za njih zelo primerne, normaliziran model in še posebej model *AM* pa bistveno manj. Transakcijske dimenzijske tabele so sicer organizirane za poizvedovanje, a so na najnižjem nivoju podrobnosti, zato tudi niso primerne. Postanejo pa primerne s prijemi, kot so uporaba

materializiranih pogledov za hitrejše poizvedovanje, a tudi takrat bolj sodijo v celovito skladišče.

Za operativno hrambo podatkov so primerni vsi modeli, saj pravila ne zahtevajo ohranjanja zgodovine podatkov in povezanosti, ter ne določajo nivoja podrobnosti podatkov.

5.5 Vzdrževanje

Lastnost primernost za polnjenje v realnem času je podobna lastnosti primernost za aktivno podatkovno skladišče. Tudi primernost vrst modelov je zato enaka.

Zahtevnost polnjenja transakcijskih tabel dejstev je najmanjša, saj se transakcije le dodajajo. Takšne tabele so običajno velike, a pri polnjenju to ne predstavlja težave, saj ni sprememb zapisov, kar pomeni, da zapisa pred vnosom ni treba poiskati. Poleg tega je glede na izvor potrebno le manjše preoblikovanje transakcije, ki običajno vključuje dodajanje izračunanih mer in povezave na umetne ključe dimenzijskih tabel. Tabele dejstev s periodičnimi posnetki stanj vsebujejo podatke, ki zasedejo malo prostora, poleg tega pa se njihovo polnjenje izvaja ob redkejših časovnih intervalih. Zbirni posnetki stanj so z vidika polnjenja zahtevni, ker je dovoljena sprememba zapisa, tega pa je treba pred spremembo tudi poiskati. Je pa količina podatkov majhna v primerjavi s transakcijskimi tabelami. Polnjenje vseh treh relacijskih dimenzijskih modelov je odvisno tudi od tipov spreminjajočih se dimenzij. Tip 1 je za implementacijo najlažji, sledita mu tip 3 in kot zadnji od treh tip 2. Polnjenje večjih dimenzij slednjega tipa je lahko časovno bolj zahtevno od polnjenja pripadajoče tabele dejstev. Zahtevnost polnjenja kock je podobno polnjenju tabele dejstev s periodičnimi posnetki stanj. Nekoliko večja je zato, ker je potrebno izračunati več različnih nivojev podrobnosti podatkov, poleg tega pa se za polnjenje pogosto uporabljajo manj zrele tehnologije, kakršne so v večdimenzijskih SUPB. Polnjenje normaliziranih modelov je običajno nezahtevno, saj je tak model najbolj podoben tistemu v izvornih sistemih. V modelih AM so podatki mnogo bolj razčlenjeni od izvornih podatkov v tretji normalni obliki, a polnjenje vseeno ni zahtevno, saj se podatki vedno le dodajajo, poleg tega pa se vedno doda le atribut, ki se je tudi spremenil, brez vpliva na ostale.

Zaradi te lastnosti modelov AM je njihovo vzdrževanje enostavno. Prav tako je enostavno vzdrževanje transakcijskih tabel dejstev. Vzdrževanje kock in tabel dejstev s periodičnimi posnetki stanj pa je bolj zahtevno, saj je včasih težko vrniti podatke v stanje, kot je bilo v trenutku zajema.

5.6 Uporaba

Orodja za poslovno obveščanje so najbolj uporabna v kombinaciji z dimenzijskimi modeli. Normalizirani modeli so prav tako primerni, a zaradi svoje strukture manj. Modeli *AM* so za takšna orodja neprimerni, uporabni postanejo šele po preoblikovanju podatkov s pomočjo pogleda ali prepisa podatkov v drugačne strukture.

Tudi z vidika preglednosti za uporabnike so dimenzijski modeli najbolj uporabni, saj sledijo načinu razmišljanja uporabnika o določenem področju. Normalizirani modeli so s tega vidika manj primerni, razen v primerih, ko je uporabnik vaje dela s takšnim modelom. Modeli *AM* so z vidika preglednosti najslabši. Težko so berljivi tudi za tehnično bolje podkovane uporabnike.

Hitrost poizvedovanja je pri kockah najboljša, saj so vsi podatki izračunani vnaprej. Ostale vrste modelov so si glede na to lastnost enakovredne, negativno odstopa le *AM*, kjer je pogosto potrebno ob poizvedovanju povezati veliko tabel.

5.7 Sprejetost

Relacijski dimenzijski in normalizirani modeli so široko sprejeti, saj se uporabljajo že od začetkov podatkovnega skladiščenja. Uporaba kock je novejša, zato je tudi njihova sprejetost manjša in zato je manjše tudi število strokovnjakov za tak način modeliranja. Modeli *AM* so najbolj redki, saj jih uporablja le majhno število skladišč.

5.8 Upravljanje s hierarhijami

Napredna uporaba hierarhij je lastnost kock. Te v tem pogledu prekašajo vse ostale načine modeliranja. Sicer je mogoče v vseh načinih omogočiti uporabo tako alternativnih in neuravnoteženih hierarhij, kot uporabo pravil za združevanje, a se je treba v tem primeru poslužiti nekaterih trikov, kot sta umetno uravnoteževanje dimenzij in uporaba pogledov. V modelih *AM* in tudi v normaliziranih modelih je uporabo alternativnih in neuravnoteženih hierarhij sicer mogoče relativno enostavno podpreti, a je uporaba takšnih podatkov preko orodij za poslovno obveščanje zato težja.

6 Primer rešitve v telekomunikacijski panogi

6.1 Uvod

V tem poglavju je prikazana praktična uporaba opisanih pristopov na primeru splošnega podatkovnega skladišča za organizacijo, ki se ukvarja s telekomunikacijami. Za posamezna področja so izbrani pristopi, ki so za njih najbolj primerni.

Podatkovni model je narejen za naslednja področja:

- uporaba storitev,
- uporabniki ter
- glavna knjiga.

Določene strukture so namenjene še zagotavljanju kakovosti podatkov, ki pokriva vsa omenjena področja.

Pri izdelavi modela so upoštevane smernice, ki veljajo za panogo. Pri tem se je potrebno zavedati, da ima vsaka organizacija še svoje specifične potrebe, zato tega modela, kot vsakega splošnega, ni mogoče neposredno uporabiti, ampak so za uporabo potrebne določene razširitve in spremembe, lahko pa služi kot osnova.

6.2 Informacijske potrebe

6.2.1 Uvod

Telekomunikacijska podjetja po svetu se soočajo s številnimi izzivi, ki jih pred njih postavlja konkurenca, tehnološke novosti in vedno večje zahteve strank. Uspešno obvladovanje teh izzivov pripomore h konkurenčnosti, zadržanju ali povečanju baze strank in s tem k zelenim rezultatom.

Telekomunikacijska panoga, za razliko od preteklosti, ni več v toliki meri tehnološko usmerjena. Njena osnovna naloga je postalo upravljanje odnosov s strankami (CRM). Zadovoljstvo strank, pospešeno trženje in napredna tehnologija predstavljajo osnovo poslovanja. Brezobzirna konkurenca in velikanske naložbe odžirajo dobiček, zaradi česar ima sprejemanje pravih odločitev kritično vlogo. Odločitve se sprejemajo na osnovi informacij, ki jih nudi sistem za poslovno obveščanje preko poročil, analiz in ključnih kazalcev poslovanja.

Telekomunikacijska podjetja morajo zadovoljiti potrebe strank – uporabnikov njihovih storitev, saj v nasprotnem primeru tvegajo, da jih bodo izgubili (angl. *churn*). To tveganje je še posebej veliko pri ponudnikih mobilne telefonije, saj njihovi uporabniki prehajajo od

ponudnika k ponudniku že ob manjših spremembah cen ali drugih stimulacijah. Strošek pridobitve novega uporabnika je do trikrat višji od stroška zadržanja obstoječega, zato se ponudniki zavedajo, da je potrebno spodbujati zvestobo.

Istočasno nima smisla zadrževati uporabnikov, ki ne prinašajo dobička. Za prepoznavanje takšnih uporabnikov morajo odločevalci imeti na voljo ustrezne informacije, na osnovi katerih lahko sklepajo o njihovi donosnosti. Poznavanje uporabnikov in njihovih zahtev je zato ena večjih prednosti ponudnika telekomunikacijskih storitev.

Takšno poznavanje vključuje odgovore na osnovna vprašanja, kot so:

- Kdo so najbolj donosni uporabniki?
- Katere produkte uporabljajo ti uporabniki in v kakšnih kombinacijah?
- Kateri produkt je najbolj donosen in zakaj?
- Ali so cene in storitve privlačne za najbolj donosne uporabnike?

Podatkovno skladišče nudi odgovore na takšna in druga vprašanja. Odločevalci lahko z njegovo pomočjo analizirajo uporabnike in jih razdelijo v skupine glede na njihovo uporabo storitev ali glede na demografske značilnosti. Na ta način imajo ponudniki dostop do znanja s katerim lahko bolj natančno oblikujejo svojo ponudbo in cene, tako da te zadostijo zahtevam določene skupine uporabnikov.

Poslovno obveščanje je učinkovito, če omogoča primerjavo med ciljno in dejansko skrbjo za uporabnike. Ponudniki lahko tako nudijo proaktivno podporo, kar je bolje kot odzivanje na težave, ko so se te že zgodile v večji meri ali pa je določena priložnost že mimo. Tako lahko ponudnik meri odstotek uspešno opravljenih klicev med vsemi klici in na ta način določi težavo pri zagotavljanju določene storitve, ki bi lahko kot posledico imela nezadovoljno in posledično lahko celo izgubljeno stranko. Poslovno obveščanje lahko opozori na težave, ki zahtevajo takojšen odziv.

6.2.2 Ključni kazalci poslovanja

Ključni kazalci poslovanja (KKP) so mere učinkovitosti, s katerimi organizacije vrednotijo svojo uspešnost pri doseganju dolgoročnih ciljev. Določijo se z odgovori na vprašanje, kaj je zares pomembno za posamezne interesne skupine v zvezi z organizacijo.

V članku na spletni strani [29] je objavljenih sto KKPjev. Od teh so tu, razdeljeni po področjih, naštet tisti, ki jih je mogoče dobiti iz modela, predstavljenega v nadaljevanju.

Uporabniki:

- Število uporabnikov
- Členjenje uporabnikov (predplačniki, naročniki, poslovni uporabniki, zasebni uporabniki)
- Izguba uporabnikov na mesec
- Število uporabnikov na zaposlenega

Uporaba storitev:

- Čas uporabe storitev (trajanje klicev)
 - Členjenje glede na tip uporabnika
 - Členjenje glede na dohodne in izhodne klice
- Število izhodnih kratkih sporočil (SMS) stranke na mesec
- Število izhodnih multimedijskih sporočil (MMS) stranke na mesec
- Čas gostovanja uporabnikov
- Število klicev
- Število klicev na uporabnika
- Povprečna dolžina klica
- Čas gostovanja uporabnika v drugi državi
 - Členjenje glede na dohodne in izhodne klice

Prihodki:

- Povprečni prihodek na uporabnika (angl. *ARPU*, *Average Revenue per User*)
 - Členjenje glede na storitev (glasovni klici, podatkovne storitve)
 - Členjenje glede na tip uporabnika
- Povprečni prihodek na klic
- Delitev prihodkov po storitvah

Rast:

- Število novih uporabnikov
- Delež novih uporabnikov
- Rast uporabe storitev

6.4 Metodologija**6.4.1 Uvod**

Prikazan je fizični model, na osnovi katerega je možno neposredno izdelati podatkovne strukture. V relacijskih modelih so opredeljene tabele, pogledi, materializiran pogled in polja.

Določeni so tako primarni kot tuji ključi. V prilogi so stavki *SQL* za izdelavo tabel. V večdimenzijskih modelih so kocke določene z dejstvi in opisom dimenzij, ki jih sestavljajo.

Model ne vsebuje struktur, ki so namenjene izključno lažjemu polnjenju, kot so prevajalne tabele, razni šifranti in tabele začasnega področja. Prav tako ne vsebuje tabel z metapodatki.

6.4.2 Relacijski modeli

Referenčna integriteta

Tuji ključi vsiljujejo referenčno integriteto in na ta način zagotavljajo upoštevanje določenih poslovnih pravil in s tem prispevajo h kakovosti podatkov. V dimenzijskih podatkovnih modelih se pojavljajo za opredelitev povezave med tabelami dejstev in pripadajočimi dimenzijami. Zaradi količine podatkov jih včasih izključimo, s čimer pridobimo na hitrosti polnjenja, a moramo pri tem v postopkih polnjenja sami zagotavljati povezavo.

V primeru so tuji ključi vedno omogočeni. Če se pojavijo težave z zmogljivostjo polnjenja, jih je mogoče enostavno izključiti.

Umetni ključi

Vsi primarni ključi dimenzijskih tabel so narejeni nad enim samim poljem numeričnega tipa, ki ga polni sekvenca. Vrednosti v teh poljih uporabniku ne nudijo nobene informacije, služijo le za povezavo med tabelami. Vrednosti so obvezne, enako velja za polja, ki se na njih navezujejo preko tujih ključev. Takšni umetni ključi so boljši od naravnih (tistih iz izvornih sistemov), zaradi [9]:

- Zmogljivosti. Umetni ključ nad enim poljem omogoča učinkovitejše stike kot naravni ključ, še posebej če je slednji narejen nad več polji.
- Povezovanje različnih virov. V različnih izvornih sistemih lahko ima ista entiteta več različnih naravnih ključev. Umetni ključ deluje povezovalno saj se dodeli entiteti ne glede na vir. Pri polnjenju se v ta namen uporabljajo prevajalne tabele.
- Upravljanje z neznanimi vrednostmi. Polja, nad katerimi so narejeni tuji ključi, naj nikoli ne bi bila prazna. Kadar vrednosti ne znamo določiti, se lahko vedno povežemo na nek vnaprej določen zapis z naprej določeno umetno vrednostjo, kot je na primer 0 ali -1.
- Spremljanje zgodovine. V dimenzijskih tabelah tipa 2 obstaja več zapisov, ki predstavljajo enega izvornega skozi čas. Zato naravni ključ ne more opravljati vloge primarnega ključa, ampak je potreben umetni ključ.

Prazne vrednosti

Večina polj ne dopušča praznih vrednosti, saj lahko te zmedejo uporabnike. Še posebej je to nevarno, kadar se nad takšnimi polji izvajajo omejitve, saj primerjava vrednosti s prazno

vrednostjo v določenih SUPB ne deluje tako, kot bi uporabniki pričakovali. Zato je namesto praznih vrednosti bolj smiselno uporabljati privzete vrednosti, kot na primer »Ni opredeljeno«, »Nedoločeno«, »Ni na voljo« ali podobno.

Informacija o polnjenju

Vsaka tabela vsebuje polje z informacijo o obdelavi, ki je vnesla ali dodala zapis. Kadar se tabela vzdržuje ročno, vsebuje polje z informacijo o času spremembe in uporabniku, ki je vnesel ali zadnji popravil zapis.

Poimenovanje

Zaradi preglednosti se za poimenovanje struktur uporabljajo standardi. Predpone tabel in drugih objektov so naslednje:

- Tabele dejstev: DE_
- Dimenzijske tabele: DIM_
- Sidro: SI_
- Vozel: VO_
- Atribut: AT_
- Vez: VE_
- Polje nad katerim je narejen primarni ključ: ID_
- Polje ki predstavlja naravni ključ: IDN_
- Pogledi P_
- Materializirani pogledi MP_

6.4.3 Večdimenzijski modeli

Predstavitev kocke

Kocke so predstavljene opisno, z navedbo dejstev in dimenzij, ki jih sestavljajo. Kjer je potrebno napredno obvladovanje hierarhij, je to posebej poudarjeno.

Predpostavke

Lastnosti kock so v različnih večdimenzijskih SUPB lahko zelo različne. Model predpostavlja, da je omogočeno napredno obvladovanje hierarhij, pri čemer je mogoča uporaba neuravnoteženih hierarhij, uporaba pravil za združevanje in uporaba alternativnih hierarhij.

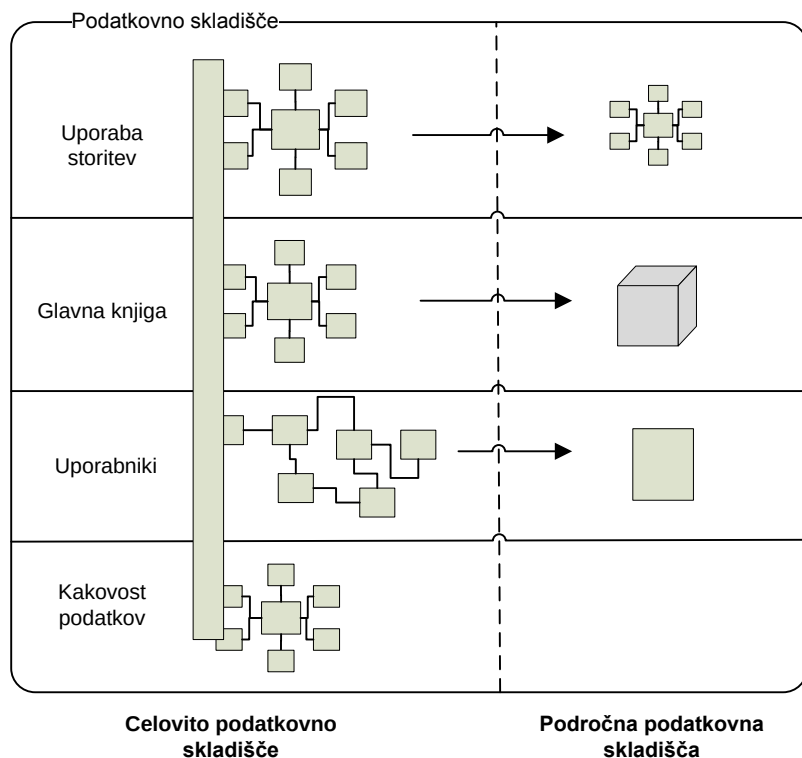
Poimenovanje

Kocka je poimenovana s predpono K_, dimenzije so poimenovane brez posebne predpone na način, kot jih poznajo uporabniki.

6.5 Arhitektura

Podatkovno skladišče je razdeljeno na dva dela. Jedro predstavlja celovito podatkovno skladišče prikazano na levi strani slike 20. Podatki v njem so predmetno usmerjeni, povezani, zgodovinski, nespremenljivi in na najnižjem nivoju podrobnosti. Zadnja omenjena lastnost omogoča enostavno razširljivost in nezahtevno vzdrževanje, pri tem so podatki predstavljeni na enak način ne glede na njihovo izvorno lokacijo.

Področna podatkovna skladišča na desni strani slike so namenjena enostavnejši uporabi podatkov. Ti so v obliki, ki omogoča hiter dostop na pregleden način. V ta namen so združeni na višji nivo podrobnosti, preoblikovani v obliko, s kakršno delajo orodja za poslovno obveščanje ali preoblikovani v napredno obliko, podprto z drugim tipom SUPB. Za področje kakovosti podatkov ne obstaja področno skladišče, saj model že sam po sebi omogoča enostavno poročanje. Tudi količina podatkov na področju kakovosti ni prevelika, podatki pa so tako ali tako že organizirani dimenzijsko in na ta način primerni za uporabo.



Slika 20: Arhitektura

Celoten model sestavljajo področja uporabnikov, uporabe storitev, glavne knjige in kakovosti podatkov. V celovitem skladišču so vsa področja razen uporabnikov podprta z dimenzijskim modelom, področje uporabnikov, ki je najbolj kompleksno, pa z modelom AM. Podatki v vsakem področju so z drugimi področji povezani preko vodila, ki ga sestavljajo dimenzijske tabele in sidro.

6.6 Celovito podatkovno skladišče

6.6.1 Uporabniki

Področje uporabnikov je narejeno z modelom *AM*. Sidri na sliki 21 sta uporabljeni za predstavitev samih uporabnikov in pogodb. Pogodba ima dva atributa:

- datum pogodbe in
- številko pogodbe.

Oba atributa se na pogodbi ne spreminjata. Zaloga vrednosti obeh ni vnaprej določena, zato nista povezana z vozlom.

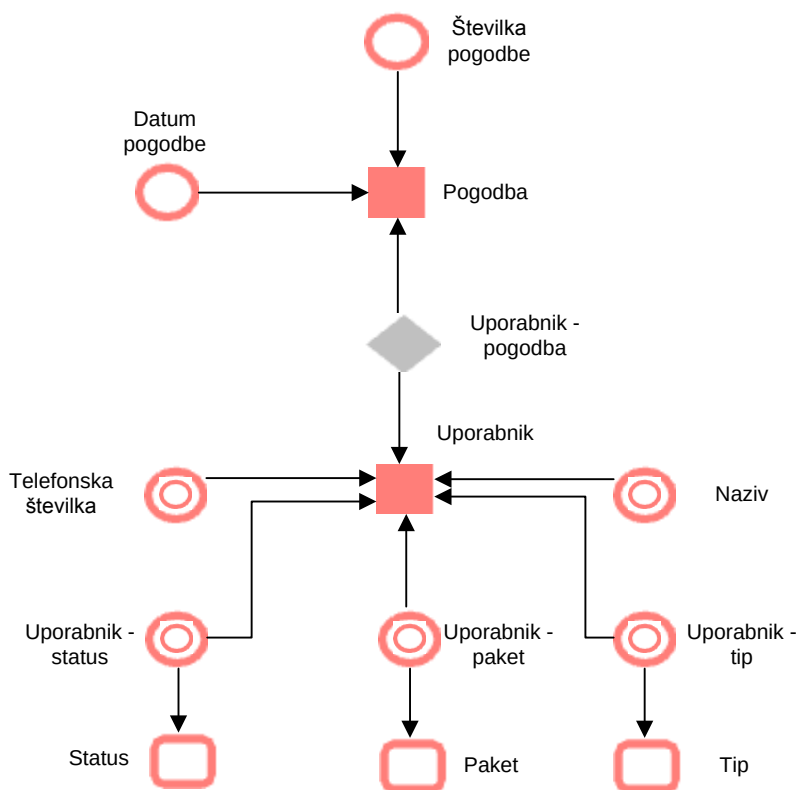
Uporabnik ima več atributov:

- telefonsko številko,
- status,
- paket,
- tip in
- naziv.

Status nosi informacijo o tem ali je uporabnik aktiven, preklican, zamrznjen in podobno. Paket predstavlja uporabnikove pogoje za klice in njihovo obračunavanje, tip pa pove ali gre za poslovnega ali zasebnega uporabnika.

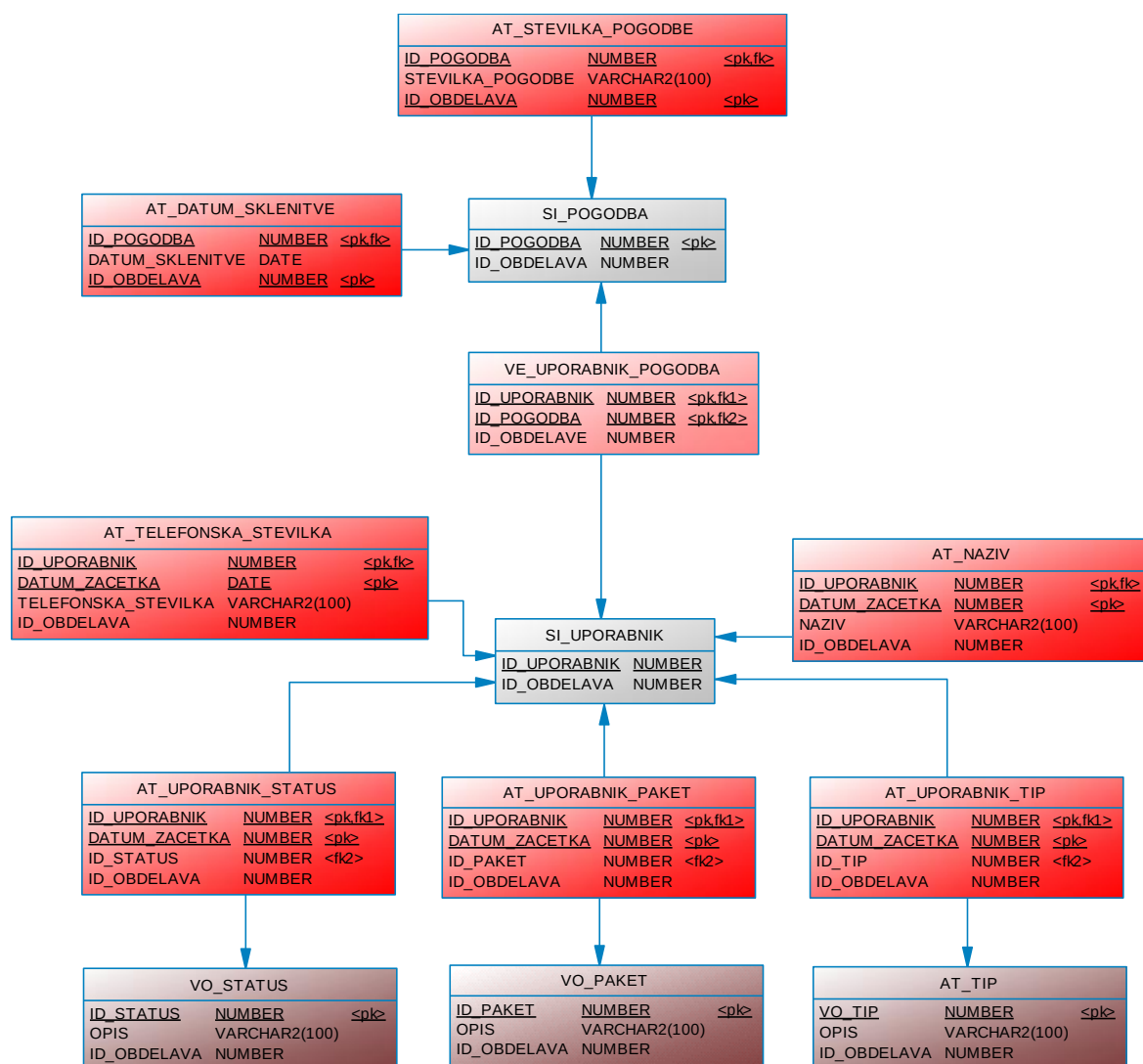
Uporabnik lahko spremeni kateregakoli od naštetih atributov, zato vsi vsebujejo še zgodovinsko komponento. Status, paket in tip so atributi z vozli, saj imajo nespremenljivo ali zelo počasi spremenljivo zalogo vrednosti. Za enkratni zapis teh vrednosti so uporabljeni vozli.

Tak model hrani nespremenljive podatke o pogodbi in podatke o uporabnikih, pri katerih se vse spremembe ohranjajo. Stanje v vsakem trenutku v preteklosti je vedno na voljo.



Slika 21: Podatkovni model za področje uporabnikov, notacija AM

Vsak gradnik na sliki 21 se preslika v eno tabelo na sliki 22. Lokacija glede na ostale gradnike je na obeh slikah enaka. Vse tabele vsebujejo informacijo o obdelavi, ki je vnesla ali zadnja spremenila določen zapis. Ta informacija je v polju z imenom ID_OBDELAVA. Sidi poleg tega vsebujeta le še polje z umetnim ključem. Tabele s statičnimi atributi imajo poleg umetnega ključa še polje z vrednostjo. Tabele z atributi z vozli imajo poleg povezave na uporabnika še povezavo na pripadajoče tabele z vozli, v katerih se nahajajo vrednosti. Zgodovinski atributi vsebujejo še datum začetka veljavnosti, ki je del ključa. Vrednost atributa je veljavna vse dokler se ne pojavi nova vrednost z novim začetkom veljavnosti.



Slika 22: Relacijski model AM za področje naročnikov

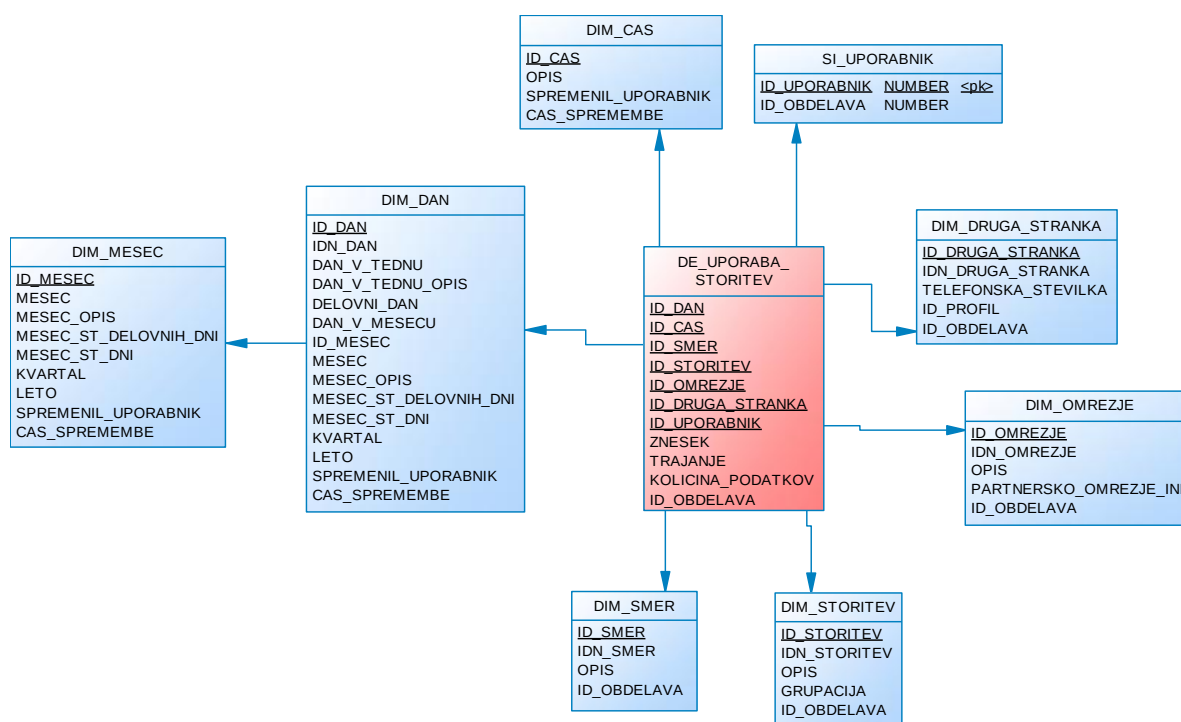
Tabela SI_UPORABNIK je v primeru uporabljena tudi na področju uporabe storitev, lahko pa bi se uporabila tudi na kateremkoli drugem z uporabnikom povezanim področju. Na ta način ima podobno vlogo kot skupne dimenzijske tabele. To ji omogoča tudi umetni primarni ključ, ki ima enako vlogo kot umetni primarni ključi dimenzij.

6.6.2 Uporaba storitev

Za spremljanje uporabe storitev je predvidena uporaba dimenzijskega modela s transakcijsko tabelo dejstev, v kateri je s svojim zapisom predstavljena vsaka transakcija. Predvidena je zvezdna shema na sliki 23, na kateri je prikazana tudi povezava z dimenzijsko tabelo DIM_MESEC, ki pa je uporabniki pri poizvedovanju nad tem dimenzijskim modelom ne uporabljajo. Uporabljajo jo v pripadajočem področnem podatkovnem skladišču, kjer so podatki združeni na višji nivo podrobnosti.

Tabeli dejstev pripadajo naslednje dimenzijske tabele:

- DIM_DAN predstavlja časovno dimenzijo na dnevnem nivoju.
- DIM_MESEC predstavlja časovno dimenzijo na mesečnem nivoju. Vsa njena polja obstajajo tudi v tabeli DIM_DAN.
- DIM_CAS je tabela, v kateri so deli dneva na nivoju minut. Vsebuje torej $24 \cdot 60 = 1440$ zapisov.
- DIM_SMER določa smer transakcije. Gre bodisi za vhodno ali izhodno transakcijo. Določa ali je uporabnik klical ali bil poklican. Pri podatkovnih storitvah gre vedno za izhodno transakcijo.
- DIM_STORITEV nosi podatke o storitvah. Primeri storitev so: glasovni klic, podatkovna storitev, sporočilo SMS, sporočilo MMS...
- DIM_OMREZJE vsebuje omrežja, v katerih se uporabnik lahko nahaja v času klica.
- DIM_DRUGA_STRANKA predstavlja drugo stranko v transakciji, na primer prejemnika sporočila SMS.
- SI_UPORABNIK ni prava dimenzijska tabela. Je sidro iz področja uporabnikov.



Slika 23: Zvezdna shema za spremljanje uporabe storitev

Dejstva so naslednja:

- znesek transakcije,
- trajanje transakcije in
- količina prenesenih podatkov znotraj transakcije.

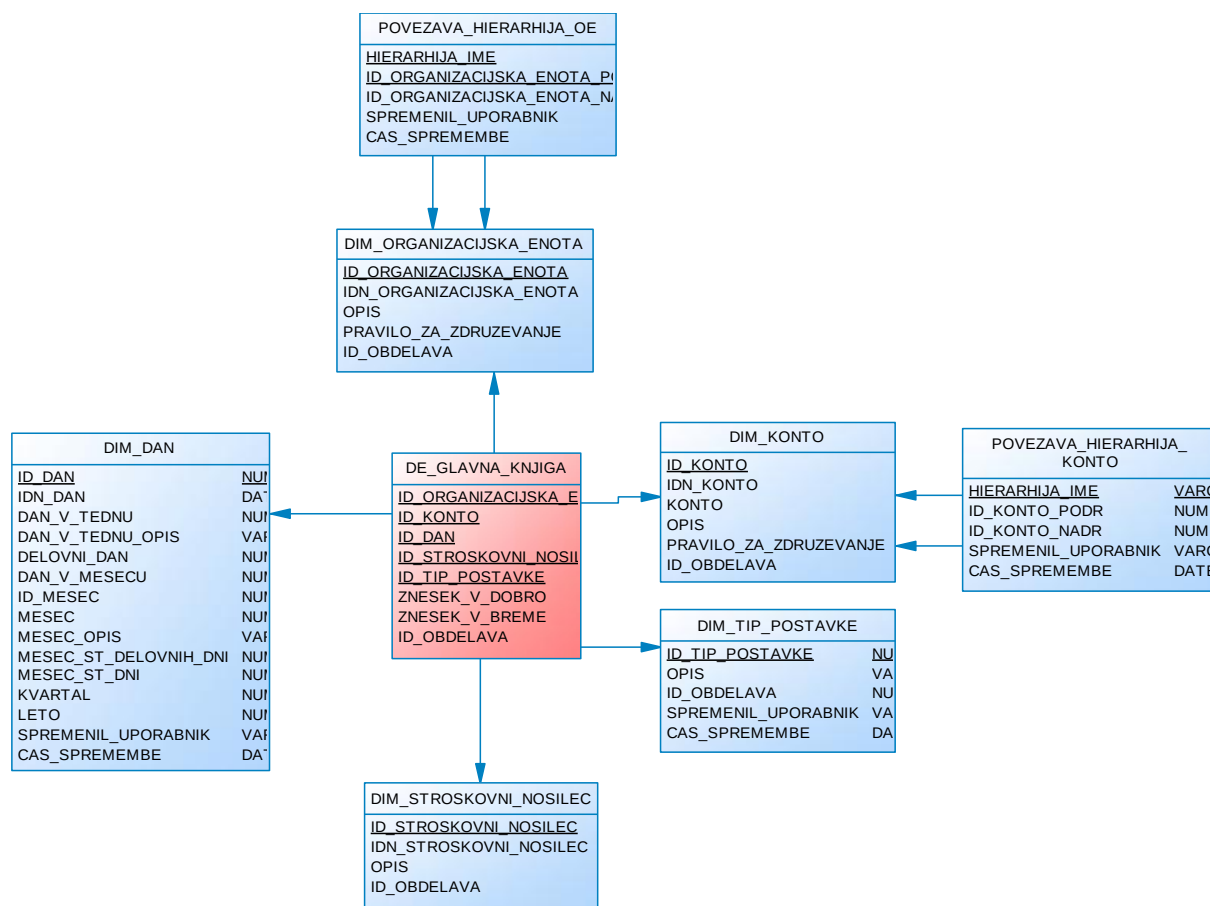
Količina podatkov v takšni tabeli dejstev je že v primeru večjega slovenskega telekomunikacijskega podjetja izredno velika, saj se na dan v to tabelo prenese nekaj milijonov zapisov. Pri polnjenju to dejstvo ne predstavlja težave, saj se podatki le dodajajo, pri uporabi pa lahko to pomeni počasno izvajanje poizvedb.

6.6.3 Glavna knjiga

Za spremljanje podatkov v glavni knjigi je predvidena uporaba dimenzijskega modela s transakcijsko tabelo dejstev, v kateri je s svojim zapisom predstavljena vsaka transakcija. Predviden je podatkovni model na sliki 24.

Dimenzijske tabele so naslednje:

- DIM_DAN predstavlja časovno dimenzijo na dnevnem nivoju.
- DIM_ORGANIZACIJSKA_ENOTA vsebuje organizacijske enote znotraj organizacije.
- DIM_KONTO vsebuje računovodske konte.
- DIM_STROSKOVNI_NOSILEC vsebuje stroškovne nosilce.



Slika 24: Zvezdna shema za spremljanje podatkov v glavni knjigi

Dejstvi sta dve:

- znesek v dobro in
- znesek v breme.

Poleg dimenzijskih tabel in tabele dejstev model sestavljata še dve povezovalni tabeli. To sta POVEZAVA_HIERARHIJA_OE in POVEZAVA_HIERARHIJA_KONTO. Njuna vsebina določa različne hierarhije nad organizacijskimi enotami in konti. Primer dveh različnih hierarhij nad konti sta hierarhiji kontnega plana in planskih postavk. Hierarhije nad organizacijskimi enotami se dodajajo ob spremembah v organizaciji. Ohranjanje starih hierarhij uporabnikom omogoča poizvedovanje na star in na nov način in jim na ta način omogoča odgovor na vprašanje, kaj bi bilo, če bi bila hierarhija organizacijskih enot še takšna, kot je bila v preteklem letu. Vsak zapis iz teh dveh tabel povezuje dva zapisa iz iste dimenzijske tabele- podrejenega in njegovega nadrejenega.

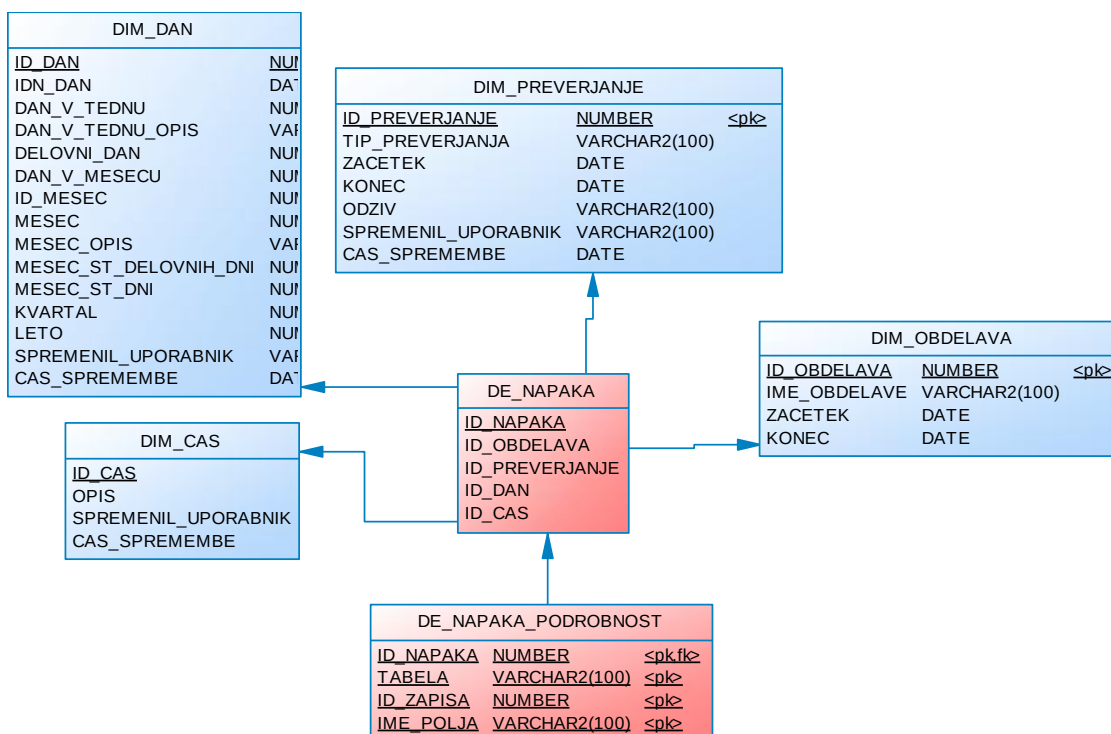
Ti povezovalni tabeli omogočata tudi obstoj neuravnoteženih hierarhij, saj je število nivojev v tako določenih hierarhijah poljubno.

Pri vrtanju navzgor glede na dimenzijo kontov ali planskih postavk se pri združevanju dejstva iz tabele dejstev ne le seštevajo, kot je običajno, ampak so mogoče tudi druge operacije. Pravilo za združevanje je v pripadajočih tabelah podano v polju PRAVILO_ZA_ZDRUZEVANJE. Tako je lahko na primer poslovni rezultat enak znesku prihodkov od katerega se odšteje znesek odhodkov, a le, kadar je to pozitivna številka, sicer se pomnoži z minus ena.

Opisana napredna obravnava hierarhij je pri uporabi podatkov lahko problematična, saj orodja za poslovno obveščanje nad relacijskim podatkovnim modelom za takšne operacije niso najbolj primerna.

6.6.4 Kakovost podatkov

Področje kakovosti podatkov sicer ni poslovno področje znotraj organizacije, se pa dotika vseh poslovnih področij. Predvidena je uporaba dimenzijskega modela s transakcijskima tabelama dejstev, kot je prikazano na sliki 25. Poenostavljen podatkovni model je že predstavljen med drugimi pristopi v podpoglavju »Kakovost podatkov«.



Slika 25: Zvezdna shema za spremljanje kakovosti podatkov

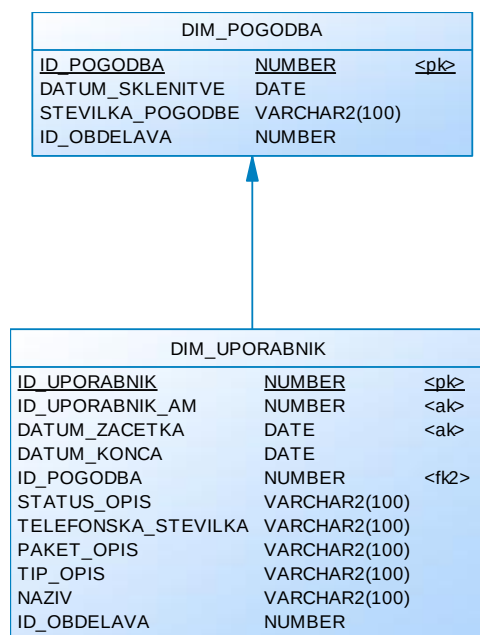
Količina podatkov v takšnem modelu naj bi bila relativno majhna. V tabeli DE_NAPAKA_PODROBNOST, ki vsebuje največ zapisov, naj jih ne bi bilo več kot milijon. Tudi povezava med dvema tabelama dejstev zaradi razmeroma majhne količine podatkov ne predstavlja težav.

Če je število zapisov vseeno preveliko, je mogoče tiste zastarele brez poslovne škode tudi izbrisati. Podatki so v obliki, ki je enostavna za uporabo, zato posebno področno podatkovno skladišče ni predvideno.

6.7 Področna podatkovna skladišča

6.7.1 Uporabniki

Področje uporabnikov je v celovitem podatkovnem skladišču narejeno z uporabo modela AM, katerega slabost je neprimernost za neposredno uporabo. Zato se model pretvori v dve dimenzijski tabeli na sliki 26, ki lahko služita kot skupni dimenziji za vsa z uporabnikom povezana področja. Pogodba ni spreminjajoča se dimenzija, za uporabnika pa je uporabljen tip spreminjanja 2. Veljavnost je pri tem določena z datumom začetka in datumom zaključka.



Slika 26: Dimenzijski tabeli za področje uporabnikov

Preslikava iz modela *AM* na sliki 22 v dimenzijsko tabelo DIM_UPORABNIK je prikazana v preglednici 6.

Polje v tabeli DIM_UPORABNIK	Izvorno polje iz modela AM
ID_UPORABNIK	Umetni ključ uporabnika
ID_UPORABNIK_AM	SI_UPORABNIK.ID_UPORABNIK
DATUM_ZACETKA	DATUM_ZACETKA iz ene od naslednjih tabel: - VO_STATUS - AT_TELEFONSKA_STEVILKA - VO_PAKET - AT_TIP - AT_NAZIV
DATUM_KONCA	DATUM_ZACETKA naslednje vrednosti iz ene od naslednjih tabel: - VO_STATUS - AT_TELEFONSKA_STEVILKA - VO_PAKET - AT_TIP - AT_NAZIV
ID_POGODBA	SI_POGODBA.ID_POGODBA
STATUS_OPIS	VO_STATUS.OPIS
TELEFONSKA_STEVILKA	AT_TELEFONSKA_STEVILKA.TELEFONSKA_STEVILKA
PAKET_OPIS	VO_PAKET.OPIS
TIP_OPIS	AT_TIP.OPIS
NAZIV	AT_NAZIV.NAZIV
ID_OBDELAVA	Umetna številka obdelave

Preglednica 6: Preslikava modela *AM* v dimenzijske tabele

V tabeli se beleži vsa zgodovina atributov uporabnika. Ob spremembi kateregakoli atributa v izvornem modelu se le-ta ob prenosu odrazi tudi v dimenzijski tabeli DIM_UPORABNIK kot nov zapis. Za razliko od modela AM se v tej tabeli ob spremembi le enega atributa vsi atributi prenesejo v nov zapis.

Izvorno polje ID_UPORABNIK iz tabele SI_UPORABNIK se preslika v polje SI_UPORABNIK_AM, ki skupaj z datumom začetka veljavnosti tvori enolični ključ. Primarni ključ je umeten.

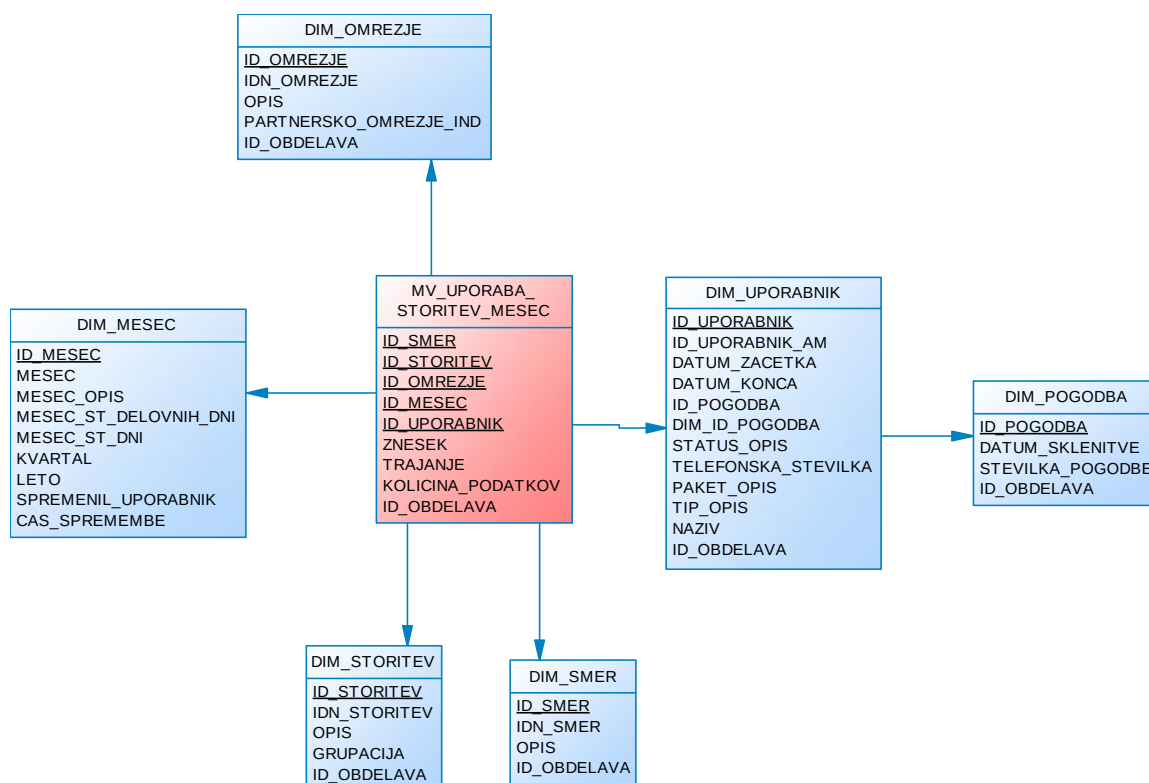
Uporaba obeh dimenzijskih tabel v dimenzijskem modelu je prikazana v naslednjem podpoglavju.

6.7.2 Uporaba storitev

Zaradi količine podatkov in posledično počasnega izvajanja poizvedb je potrebno pohitriti izvajanje z uvedbo področnega podatkovnega skladišča. Slika 27 prikazuje takšno skladišče. Vlogo tabele dejstev ima materializirani pogled MV_UPORABA_STORITEV_MESEC, v katerem so podatki združeni na višji nivo podrobnosti. Gre za periodični posnetek stanja.

Osnovna časovna enota je en mesec. V celovitem podatkovnem skladišču je en dan. Dimenzijski tabeli DIM_CAS in DIM_DRUGA_STRANKA sta odstranjeni, saj predvidevamo, da se pri poročanju uporabljata redkeje. Najbolj se uporabljajo dimenzije smer, storitev, omrežje in uporabnik.

Prednost uporabe materializiranega pogleda je v tem, da ni potrebe, da bi se jih pisci poizvedb sploh zavedali. Poizvedba se lahko napiše neposredno nad osnovnimi tabelami iz celovitega podatkovnega skladišča, sistem za upravljanje podatkovne baze pa, če obstaja v materializiranem pogledu zahtevana množica podatkov, naredi poizvedbo kar nad njim. Tako so lahko vse poizvedbe napisane nad tabelo DE_UPORABA_STORITEV. Če bodo želeni podatki obstajali v materializiranem pogledu MV_UPORABA_STORITEV_MESEC se bo poizvedba izvedla nad njim. Vsebuje malo podatkov, zato bo rezultat hitro na voljo. Če bo poizvedba zahtevala podrobnejše podatke, ki jih v materializiranem pogledu ni, se bo izvedla nad osnovno tabelo dejstev, za kar bo potrebno več časa.



Slika 27: Zvezdna shema za spremljanje uporabe storitev na mesečni ravni

Količina podatkov v materializiranem pogledu je nekaj desetkrat manjša kot v tabeli dejstev. To je posledica tega, da zrno tabele ni več posamezna transakcija, temveč je določeno s smerjo, storitvijo, omrežjem, uporabnikom in mesecem.

6.7.3 Glavna knjiga

Za predstavitev podatkov v glavni knjigi je predvideno področno podatkovno skladišče z uporabo kocke. Glavna knjiga iz relacijskega modela celovitega podatkovnega skladišča se preslika v kocko K_GLAVNA_KNJIGA. Podatki so združeni na mesečni nivo, ostale dimenzije ostanejo na istem nivoju.

Dejstvi:

- Znesek v dobro
- Znesek v breme

Dimenzije:

- Obdobje
- Mesec
- Konto
- Stroškovni nosilec
- Organizacijska enota

- Tip postavke

Dimenzije mesec, organizacijska enota in konto so hierarhične. Nad člani na najnižjem nivoju so zgrajene različne hierarhije. Nad dimenzijo obdobje, ki je na mesečnem nivoju sta potrebni vsaj dve hierarhiji:

- Mesec -> Kvartal -> Polletje -> Leto
- Mesec -> Fiskalni kvartal -> Fiskalno polletje -> Fiskalno leto

Hierarhije nad dimenzijama kontov in organizacijskih enot so poleg tega lahko tudi neuravnotežene, kar pomeni, da nimajo vedno enakega število nivojev. Organizacijska enota ima tako lahko večinoma šest nivojev, kadrovska služba pa je na primer razčlenjena le do drugega nivoja. V celovitem podatkovnem skladišču sta predvideni povezovalni tabeli v katerih se določijo hierarhije, ki jih je lahko nad eno dimenzijo tudi več, poleg tega pa so lahko neuravnotežene.

Ob vrtanju navzgor seštevaje ni edina operacija. Možne so tudi druge. Zato je v celovitem podatkovnem skladišču predvideno polje s pravilom za združevanje, ki se nahaja v obeh dimenzijskih tabelah, kjer je to smiselno.

Predvidena je uporaba večdimenzijskega SUPB, ki omogoča zahtevano napredno obravnavo dimenzij, kot je uporaba alternativnih hierarhij, neuravnoteženih hierarhij in pravil za združevanje. Predvidena je tudi raba orodja za poslovno obveščanje, s katerim je poizvedovanje nad takšno kocko enostavno.

Uporaba kocke je za to področno podatkovno skladišče v prvi vrsti smiselna predvsem zaradi zmožnosti obvladovanja hierarhij. Ob tem je poizvedovanje nad njo tudi zelo hitro, kar jo naredi še koristnejšo.

7 Sklep

V delu so predstavljene različne vrste podatkovnih skladišč in tehnike za načrtovanje pripadajočih podatkovnih modelov. Poleg klasičnih skladišč, katerih predstavnik sta dimenzijska in korporativna vrsta, so danes aktualni še aktivno in federativno skladišče ter skladišče kot skupek posameznih področnih podatkovnih skladišč. Zadnje tri vrste so se pojavile kot posledica novih zahtev, med katerimi so dostop do podatkov v realnem času, združevanje skladišč in drugih sistemov brez izdelave nove fizične zbirke podatkov in hitra izgradnja. Vsaka vrsta ima svoje prednosti in slabosti, na osnovi katerih se je glede na potrebe smiselno odločiti za eno od njih, ali, kot je najbolj pogosto, za njihovo kombinacijo.

Vsaka zahteva tudi svojo tehniko podatkovnega modeliranja. Še vedno sta najbolj aktualni dve. To sta uporaba dimenzijskega modela in normaliziranega modela v tretji normalni obliki. Za izboljšanje zmogljivosti se lahko v relacijskih SUPB poleg indeksov in particij uporabljajo materializirani pogledi, ki omogočajo večjo hitrost poizvedb, brez da bi se jih uporabniki sploh zavedali. Novejša tehnika je *anchor modeling*, ki temelji na šesti normalni obliki. Prinaša številne prednosti, kot so prilagodljivost, modularnost, razširljivost in enostavnost vzdrževanja. Na drugi strani takšen model ni najbolj primeren za neposredno uporabo s strani orodij za poslovno obveščanje, ampak je v ta namen nad njim potrebno zagotoviti dodaten sloj. Pri izbiri tehnike modeliranja si morajo načrtovalci pomagati s številnimi lastnostmi posameznih tehnik.

Večdimenzijski SUPB postajajo vedno boljši, prav tako orodja za uporabo podatkov v kockah. Prednost kock ni le v hitrejšem dostopu do podatkov, ampak je pomembna tudi napredna obravnava dimenzij s pravili za združevanje na višje nivoje ter možnost alternativnih in neuravnoteženih hierarhij.

Ogrodje DW 2.0, katerega avtor predstavlja kot naslednjo generacijo podatkovnega skladiščenja, predvideva štiri področja v katerih se nahajajo podatki glede na njihovo starost. Poleg tega med drugim vključuje še nestrukturirane podatke in podatke v realnem času. Izgradnja modela v tem ogrodju poteka preko treh nivojev, odvisno od nivoja abstrakcije. Začne se z modelom na najvišjem nivoju, nivoju celotne organizacije, ki mora zajemati vsa poslovna področja, končni model pa je fizični model, katerega praviloma izdelamo postopoma, eno poslovno področje za drugim.

V praktičnem delu je prikazan model podatkovnega skladišča za organizacijo v telekomunikacijski panogi. Podprta so področja uporabnikov, uporabe storitev, glavne knjige in kakovosti podatkov. Skladišče je razdeljeno na celovito in na področna. V celovitem skladišču so podprta vsa področja, v njem so povezani podatki na najnižjem možnem nivoju

podrobnosti, kar omogoča enostavno razširljivost in daljši rok uporabe skladišča. Uporabljeni so dimenzijski modeli in model *anchor modeling*. Slednji se uporablja zaradi narave podatkov, kateri se pogosto spreminjajo, poleg tega pa obstaja zapletena mreža povezav med njimi. Področna podatkovna skladišča so narejena z namenom optimizacije uporabniških poizvedb. Poleg relacijskih SUPB je predvidena tudi uporaba večdimenzijskih.

Anchor modeling se je v praktičnem delu izkazal kot enostavna tehnika za podatkovno modeliranje. Za transakcijske podatke zaradi razčlenjenosti struktur sicer ni najbolj primeren, je pa zelo primeren za bolj kompleksne podatke. Le-te je kasneje mogoče enostavno pretvoriti v uporabnikom bolj prijazno obliko, kakršna je na primer dimenzijska. Zahtevnost polnjenja v tako razčlenjene strukture bi bilo zanimivo preveriti v nadaljnjih raziskavah.

V bodoče bo zanimivo spremljati tudi, koliko organizacij bo prevzelo ogrodje DW 2.0 in kako se bo le-to razvijalo ter kateri konstrukti se bodo uporabljali več in kateri manj. Še posebej zanimivo bo spremljati, kako se bodo v bodoče uporabljali nestrukturirani podatki, katerih je veliko, njihova uporaba v analitične namene pa je še v povojih.

Večdimenzijske SUPB se še razvijajo. V prihodnosti lahko pričakujemo izboljšave v smeri še večje hitrosti, tako polnjenja kot poizvedb, ter v smeri vpeljave standardov, ki bi zmanjšali razlike med produkti različnih ponudnikov.

Izbira vrste podatkovnega skladišča in tehnike podatkovnega modeliranja sta pomembni aktivnosti pred začetkom izgradnje skladišča. Pri tem je potrebno upoštevati veliko lastnosti, saj lahko s tem bistveno podaljšamo čas uporabe skladišča, povečamo zadovoljstvo uporabnikov ter zmanjšamo stroške izgradnje in vzdrževanja.

8 Priloga

8.4 Primer rešitve, stavki SQL

8.4.1 Tabela AT_DATUM_SKLENITVE

```
create table AT_DATUM_SKLENITVE
(
  ID_POGODBA      NUMBER      not null,
  DATUM_SKLENITVE  DATE        not null,
  ID_OBDELAVA      NUMBER      not null
);
```

8.4.2 Tabela AT_NAZIV

```
create table AT_NAZIV
(
  ID_UPORABNIK      NUMBER      not null,
  DATUM_ZACETKA     NUMBER      not null,
  NAZIV              VARCHAR2(100) not null,
  ID_OBDELAVA       NUMBER      not null
);
```

8.4.3 Tabela AT_STEVILKA_POGODBE

```
create table AT_STEVILKA_POGODBE
(
  ID_POGODBA      NUMBER      not null,
  STEVILKA_POGODBE VARCHAR2(100) not null,
  ID_OBDELAVA      NUMBER      not null
);
```

8.4.4 Tabela AT_TELEFONSKA_STEVILKA

```
create table AT_TELEFONSKA_STEVILKA
(
  ID_UPORABNIK      NUMBER      not null,
  DATUM_ZACETKA     DATE        not null,
  TELEFONSKA_STEVILKA VARCHAR2(100) not null,
  ID_OBDELAVA       NUMBER      not null
);
```

8.4.5 Tabela AT_TIP

```
create table AT_TIP
(
  VO_TIP          NUMBER      not null,
  OPIS             VARCHAR2(100) not null,
  ID_OBDELAVA      NUMBER      not null
);
```

8.4.6 Tabela AT_UPORABNIK_PAKET

```
create table AT_UPORABNIK_PAKET
(
  ID_UPORABNIK      NUMBER          not null,
  DATUM_ZACETKA     NUMBER          not null,
  ID_PAKET          NUMBER          not null,
  ID_OBDELAVA       NUMBER          not null
);
```

8.4.7 Tabela AT_UPORABNIK_STATUS

```
create table AT_UPORABNIK_STATUS
(
  ID_UPORABNIK      NUMBER          not null,
  DATUM_ZACETKA     NUMBER          not null,
  ID_STATUS         NUMBER,
  ID_OBDELAVA       NUMBER
);
```

8.4.8 Tabela AT_UPORABNIK_TIP

```
create table AT_UPORABNIK_TIP
(
  ID_UPORABNIK      NUMBER          not null,
  DATUM_ZACETKA     NUMBER          not null,
  ID_TIP            NUMBER          not null,
  ID_OBDELAVA       NUMBER          not null
);
```

8.4.9 Tabela DE_GLAVNA_KNJIGA

```
create table DE_GLAVNA_KNJIGA
(
  ID_ORGANIZACIJSKA_ENOTA NUMBER          not null,
  ID_KONTO           NUMBER          not null,
  ID_DAN             NUMBER          not null,
  ID_STROSKOVNI_NOSILEC NUMBER          not null,
  ID_TIP_POSTAVKE    NUMBER          not null,
  ZNESEK_V_DOBRO    NUMBER          not null,
  ZNESEK_V_BREME    NUMBER          not null,
  ID_OBDELAVA       NUMBER          not null
);
```

8.4.10 Tabela DE_UPORABA_STORITEV

```
create table DE_UPORABA_STORITEV
(
  ID_DAN            NUMBER          not null,
  ID_CAS            NUMBER          not null,
  ID_SMER           NUMBER          not null,
  ID_STORITEV       NUMBER          not null,
);
```

```

ID_OMREZJE      NUMBER      not null,
ID_DRUGA_STRANKA  NUMBER      not null,
ID_UPORABNIK     NUMBER,
ZNESEK          NUMBER      not null,
TRAJANJE        NUMBER      not null,
KOLICINA_PODATKOV  NUMBER      not null,
ID_OBDELAVA      NUMBER      not null
);

```

8.4.11 Tabela DIM_CAS

```

create table DIM_CAS
(
  ID_CAS          NUMBER      not null,
  OPIS            VARCHAR2(100) not null,
  SPREMENIL_UPORABNIK VARCHAR2(100) not null,
  CAS_SPREMEMBE   DATE
);

```

8.4.12 Tabela DIM_DAN

```

create table DIM_DAN
(
  ID_DAN          NUMBER      not null,
  IDN_DAN         DATE        not null,
  DAN_V_TEDNU     NUMBER      not null,
  DAN_V_TEDNU_OPIS VARCHAR2(100) not null,
  DELOVNI_DAN     NUMBER      not null,
  DAN_V_MESECU    NUMBER      not null,
  ID_MESEC        NUMBER      not null,
  MESEC           NUMBER      not null,
  MESEC_OPIS      VARCHAR2(100) not null,
  MESEC_ST_DELOVNIH_DNI NUMBER not null,
  MESEC_ST_DNI    NUMBER      not null,
  KVARTAL         NUMBER      not null,
  LETO            NUMBER      not null,
  SPREMENIL_UPORABNIK VARCHAR2(100) not null,
  CAS_SPREMEMBE   DATE        not null
);

```

8.4.13 Tabela DIM_DRUGA_STRANKA create table DIM_DRUGA_STRANKA

```

(
  ID_DRUGA_STRANKA  NUMBER      not null,
  IDN_DRUGA_STRANKA VARCHAR2(100) not null,
  TELEFONSKA_STEVILKA VARCHAR2(100) not null,
  ID_PROFIL         NUMBER      not null,
  ID_OBDELAVA       NUMBER      not null
);

```

8.4.14 Tabela DIM_KONTO

create table DIM_KONTO

```
(
  ID_KONTO          NUMBER          not null,
  IDN_KONTO         VARCHAR2(100)   not null,
  KONTO             VARCHAR2(100)   not null,
  OPIS              VARCHAR2(100)   not null,
  PRAVILO_ZA_ZDRUZEVANJE VARCHAR2(100),
  ID_OBDELAVA       NUMBER          not null
);
```

8.4.15 Tabela DIM_MESEC

create table DIM_MESEC

```
(
  ID_MESEC          NUMBER          not null,
  MESEC            NUMBER          not null,
  MESEC_OPIS        VARCHAR2(100)   not null,
  MESEC_ST_DELOVNIH_DNI NUMBER          not null,
  MESEC_ST_DNI      NUMBER          not null,
  KVARTAL           NUMBER          not null,
  LETO              NUMBER          not null,
  SPREMENIL_UPORABNIK VARCHAR2(100) not null,
  CAS_SPREMEMBE     DATE            not null
);
```

8.4.16 Tabela DIM_OBDELAVA

create table DIM_OBDELAVA

```
(
  ID_OBDELAVA       NUMBER          not null,
  IME_OBDELAVE      VARCHAR2(100)   not null,
  ZACETEK           DATE            not null,
  KONEC             DATE            not null
);
```

8.4.17 Tabela DIM_OMREZJE

create table DIM_OMREZJE

```
(
  ID_OMREZJE        NUMBER          not null,
  IDN_OMREZJE       VARCHAR2(100)   not null,
  OPIS              VARCHAR2(100)   not null,
  PARTNERSKO_OMREZJE_IND NUMBER          not null,
  ID_OBDELAVA       NUMBER          not null
);
```

8.4.18 Tabela DIM_ORGANIZACIJSKA_ENOTA

create table DIM_ORGANIZACIJSKA_ENOTA

```
(
```

```

ID_ORGANIZACIJSKA_ENOTA NUMBER          not null,
IDN_ORGANIZACIJSKA_ENOTA VARCHAR2(100)    not null,
OPIS          VARCHAR2(100)    not null,
PRAVILO_ZA_ZDRUZEVANJE VARCHAR2(100),
ID_OBDELAVA   NUMBER          not null
);

```

8.4.19 Tabela DIM_POGODBA

```
create table DIM_POGODBA
```

```

(
  ID_POGODBA      NUMBER          not null,
  DATUM_SKLENITVE DATE          not null,
  STEVILKA_POGODBE VARCHAR2(100) not null,
  ID_OBDELAVA     NUMBER          not null
);

```

8.4.20 Tabela DIM_PREVERJANJE

```
create table DIM_PREVERJANJE
```

```

(
  ID_PREVERJANJE  NUMBER          not null,
  TIP_PREVERJANJA VARCHAR2(100)    not null,
  ZACETEK         DATE          not null,
  KONEC           DATE,
  ODZIV           VARCHAR2(100)    not null,
  SPREMENIL_UPORABNIK VARCHAR2(100) not null,
  CAS_SPREMEMBE   DATE          not null
);

```

8.4.21 Tabela DIM_SMER

```
create table DIM_SMER
```

```

(
  ID_SMER      NUMBER          not null,
  IDN_SMER     VARCHAR2(100)    not null,
  OPIS         VARCHAR2(100)    not null,
  ID_OBDELAVA  NUMBER          not null
);

```

8.4.22 Tabela DIM_STORITEV

```
create table DIM_STORITEV
```

```

(
  ID_STORITEV  NUMBER          not null,
  IDN_STORITEV VARCHAR2(100)    not null,
  OPIS         VARCHAR2(100)    not null,
  GRUPACIJA   VARCHAR2(100)    not null,
  ID_OBDELAVA  NUMBER          not null
);

```

8.4.23 Tabela DIM_STROSKOVNI_NOSILEC

```
create table DIM_STROSKOVNI_NOSILEC
(
  ID_STROSKOVNI_NOSILEC NUMBER          not null,
  IDN_STROSKOVNI_NOSILEC VARCHAR2(100)  not null,
  OPIS          VARCHAR2(100)  not null,
  ID_OBDELAVA   NUMBER          not null
);
```

8.4.24 Tabela DIM_TIP_POSTAVKE

```
create table DIM_TIP_POSTAVKE
(
  ID_TIP_POSTAVKE   NUMBER          not null,
  OPIS          VARCHAR2(100)  not null,
  ID_OBDELAVA   NUMBER          not null,
  SPREMENIL_UPORABNIK VARCHAR2(100)  not null,
  CAS_SPREMEMBE   DATE          not null
);
```

8.4.25 Tabela DIM_UPORABNIK

```
create table DIM_UPORABNIK
(
  ID_UPORABNIK      NUMBER          not null,
  ID_UPORABNIK_AM   NUMBER          not null,
  DATUM_ZACETKA    DATE          not null,
  DATUM_KONCA      DATE          not null,
  ID_POGODBA       NUMBER          not null,
  STATUS_OPIS      VARCHAR2(100)  not null,
  TELEFONSKA_STEVILKA VARCHAR2(100)  not null,
  PAKET_OPIS       VARCHAR2(100)  not null,
  TIP_OPIS         VARCHAR2(100)  not null,
  NAZIV            VARCHAR2(100)  not null,
  ID_OBDELAVA      NUMBER
);
```

8.4.26 Tabela DE_NAPAKA

```
create table DE_NAPAKA
(
  ID_NAPAKA      NUMBER          not null,
  ID_OBDELAVA    NUMBER          not null,
  ID_PREVERJANJE NUMBER,
  ID_DAN         NUMBER,
  ID_CAS         NUMBER
);
```

8.4.27 Tabela DE_NAPAKA_PODROBNOST

```
create table DE_NAPAKA_PODROBNOST
```

```
(
  ID_NAPAKA      NUMBER      not null,
  TABELA        VARCHAR2(100) not null,
  ID_ZAPISA      NUMBER      not null,
  IME_POLJA      VARCHAR2(100) not null
);
```

8.4.28 Tabela POVEZAVA_HIERARHIJA_OE

create table POVEZAVA_HIERARHIJA_OE

```
(
  HIERARHIJA_IME  VARCHAR2(100) not null,
  ID_ORGANIZACIJSKA_ENOTA_PODR NUMBER not null,
  ID_ORGANIZACIJSKA_ENOTA_NADR NUMBER not null,
  SPREMENIL_UPORABNIK VARCHAR2(100) not null,
  CAS_SPREMEMBE   DATE          not null
);
```

8.4.29 Tabela POVEZAVA_HIERARHIJA_KONTO

create table POVEZAVA_HIERARHIJA_KONTO

```
(
  HIERARHIJA_IME  VARCHAR2(100) not null,
  ID_KONTO_PODR   NUMBER      not null,
  ID_KONTO_NADR   NUMBER      not null,
  SPREMENIL_UPORABNIK VARCHAR2(100) not null,
  CAS_SPREMEMBE   DATE          not null
);
```

8.4.30 Tabela SI_POGODBA

create table SI_POGODBA

```
(
  ID_POGODBA      NUMBER      not null,
  ID_OBDELAVA     NUMBER      not null
);
```

8.4.31 Tabela SI_UPORABNIK

create table SI_UPORABNIK

```
(
  ID_UPORABNIK     NUMBER      not null,
  ID_OBDELAVA      NUMBER      not null
);
```

8.4.32 Tabela VE_UPORABNIK_POGODBA

create table VE_UPORABNIK_POGODBA

```
(
  ID_UPORABNIK     NUMBER      not null,
  DATUM_ZACETKA    DATE        not null,
  ID_POGODBA       NUMBER      not null,
);
```

```

    ID_OBDELAVE      NUMBER          not null
);

```

8.4.33 Tabela VO_PAKET

```

create table VO_PAKET
(
    ID_PAKET          NUMBER          not null,
    OPIS              VARCHAR2(100)   not null,
    ID_OBDELAVA       NUMBER          not null
);

```

8.4.34 Tabela VO_STATUS

```

create table VO_STATUS
(
    ID_STATUS         NUMBER          not null,
    OPIS              VARCHAR2(100)   not null,
    ID_OBDELAVA       NUMBER          not null
);

```

8.4.35 Materializiran pogled MV_UPORABA_STORITEV_MESEC

```

create table MV_UPORABA_STORITEV_MESEC
(
    ID_SMER           NUMBER          not null,
    ID_STORITEV       NUMBER          not null,
    ID_OMREZJE        NUMBER          not null,
    ID_MESEC          NUMBER          not null,
    ID_UPORABNIK      NUMBER          not null,
    ZNESEK            NUMBER          not null,
    TRAJANJE          NUMBER          not null,
    KOLICINA_PODATKOV NUMBER          not null,
    ID_OBDELAVA       NUMBER          not null
);

```


9 Viri in literatura

- [1] L.P. English, Improving Data Warehouse and Business Information Quality, Methods for Reducing Costs and Increasing Profits, Wiley Publishing, Inc., 1999.
- [2] M. Golfarelli, S. Rizzi, »Data Warehouse Design, Modern Principles and Methodologies«, McGraw-Hill 2009.
- [3] H. Gupta, I.S. Mumick, »Selection of Views to Materialize in a Data Warehouse«, IEEE Computer, Januar 2005, str. 24-43.
- [4] W.H. Inmon, »Building the Data Warehouse«, New York, John Wiley & Sons, 1992.
- [5] W.H. Inmon, D. Strauss, G. Neushloss »DW 2.0, The Architecture for the Next Generation of Data Warehousing«, Morgan Kaufman 2008.
- [6] R. Kimball, »The Data Warehouse Toolkit, Practical Techniques for Building Dimensional Data Warehouses«, Wiley Inc. 1996.
- [7] R. Kimball, J. Caserta, »The Data Warehouse ETL Toolkit«, Wiley Inc. 2004.
- [8] R. Kimball, M. Ross, »The Data Warehouse Toolkit, The complete guide to dimensional modeling«, Wiley Inc. 2002.
- [9] R. Kimball, M. Ross, W. Thornthwaite, J. Mundy, B. Becker, »The Data Warehouse Lifecycle Toolkit«, Wiley Inc. 2008.
- [10] D. Moody, M.A.R. Kortink, "From Enterprise Models to Dimensional Models: A Methodology for Data Warehouse and Data Mart Design", Članek z dogodka DMDW 2000, Stockholm, Sweden, June 5-6, objavljen na CiteSeerX, 2000.
- [11] D. Moody, M.A.R. Kortink, »From ER Models to Dimensional Models: Bridging the Gap between OLTP and OLAP Design, Part I«, Business Intelligence Journal, 2003, str. 7-24.
- [12] O. Regardt, L. Rönnbäck, M. Bergholtz, P. Johansson, P. Wohed, »Anchor Modeling – An Agile Modeling Technique Using the Sixth Normal Form for Structurally and Temporally Evolving Data«, Lecture Notes In Computer Science, Vol. 5829, November 2009, pp. 234-250.
- [13] T. Spencer, T. Loukas, "From Star to Snowflake to ERD," Enterprise Systems Journal, 1999.
- [14] S. Viaene, »Linkin Business Intelligence into Your Business«, IT Pro, November/December 2008, str. 28 – 34.
- [15] H.J. Watson, T. Ariyachandra, »Selection Decision and the Success of the Architectures«, Tehnično poročilo, College of Business, University of Georgia, Athens, GA, July 2005.
- [16] H.J. Watson, »Recent Developments in Data Warehousing«, Communications of the Association for Information Systems, Volume 8, 2001.

- [17] H.J. Watson, B.H. Wixom, »The Current State of Business Intelligence«, IEEE Computer, September 2007, str. 96-99.

Ostali viri:

- [18] Anchor Modeler, orodje za razvoj anchor modela, <http://www.anchor modeling.com/modeler/>.
- [19] IBM, »DB2 Basics, an Introduction to Materialized Query Tables«, <http://www.ibm.com/developerworks/data/library/techarticle/dm-0509melnyk/>, September, 2005.
- [20] W. H. Inmon, »Creating The Data Warehouse Data Model From The Corporate Data Model«, članek objavljen na spletni strani, <http://www.inmoncif.com/registration/whitepapers/ttdwdmod-1.pdf>.
- [21] W. H. Inmon, »DW 2.0«, članek objavljen na spletni strani, <http://inmoncif.com/news/pdf/dw20FN.pdf>.
- [22] W. H. Inmon, »Modeling in the DW 2.0 Environment«, članek objavljen na spletni strani, <http://inmoncif.com/news/pdf/modelFN.pdf>.
- [23] Microsoft, »Multidimensional Expressions (MDX) Reference«, <http://msdn.microsoft.com/en-us/library/ms145506.aspx>.
- [24] O'Brien, John, »Emerging Technologies Shaping the Future of Data Warehouses and Business Intelligence«, TDWI, 2009.
- [25] Oracle, »Oracle9i Data Warehousing Guide«, http://download.oracle.com/docs/cd/B10501_01/server.920/a96520.pdf, 2002.
- [26] O. Regardt, L. Ronnback, »Anchor Modeling, Building a Future Proof Data Warehouse«, članek objavljen na spletni strani, http://www.anchor modeling.com/wp-content/uploads/2010/08/AM_Cheat_Sheet.pdf.
- [27] O. Regardt, L. Rönnbäck, M. Bergholtz, P. Johannesson, P. Wohed, »Analysis of normal forms for anchor-tables«, članek objavljen na spletni strani www.anchor modeling.com, April, 2009.
- [28] <http://www.inmoncif.com>, »Corporate Information Factory«, spletna stran.
- [29] »100 KPI's for Mobile Telecom Operators«, članek objavljen na spletni strani, <http://consultantvalueadded.com/2010/04/14/100-kpi's-for-mobile-telecom-operators/>, April, 2010.