

**UNIVERZA V LJUBLJANI**  
**EKONOMSKA FAKULTETA**

**DIPLOMSKO DELO**

**JANEZ TROŠT**



**UNIVERZA V LJUBLJANI**  
**EKONOMSKA FAKULTETA**

**DIPLOMSKO DELO**

**SAMOPOSTREŽNI SISTEM POSLOVNE INTELIGENCE KOT  
ALTERNATIVA TRADICIONALNEMU PRISTOPU**

**Ljubljana, junij 2016**

**JANEZ TROŠT**

IZJAVA

Janez Trošt izjavljam, da sem avtor tega diplomskega dela, ki sem ga napisal pod mentorstvom dr. Jurija Jakliča in dovolim objavo diplomskega dela na fakultetnih spletnih straneh.

V Ljubljani, dne \_\_\_\_\_

Podpis: \_\_\_\_\_



# Kazalo vsebine

Uvod.....	1
1 Poslovna inteligenca.....	2
1.1 Kaj je poslovna inteligenca.....	2
1.1.1 Tradicionalen sistem poslovne inteligence .....	3
1.1.2 Samopostrežna poslovna inteligenca .....	4
1.2 Uporaba poslovne inteligence .....	5
2 Podatkovno skladišče .....	8
2.1 Kaj je podatkovno skladišče .....	8
2.1.1 Operativni sistemi in podatkovna skladišča .....	8
2.2 Večdimenzijski in normaliziran pristop .....	9
2.2.1 Normaliziran pristop .....	10
2.2.2 Večdimenzijski pristop.....	10
2.2.2.1 Tabele dejstev .....	11
2.2.2.2 Dimenzijske tabele.....	13
2.2.2.3 Shema večdimenzijskega podatkovnega skladišča.....	15
2.2.3 Področno podatkovno skladišče .....	17
2.3 Postopek ETL .....	18
2.3.1 Pristopi in arhitektura .....	19
2.4 Večdimenzijske in stolpčno usmerjene baze podatkov .....	21
2.4.1 Večdimenzijska baza podatkov .....	22
2.4.2 Stolpčno usmerjeno shranjevanje.....	24
2.4.3 Sprotna analitična obdelava podatkov.....	25
3 Microsoftova orodja za podatkovno skladiščenje in poslovno inteligenco.....	26
3.1 Tradicionalna poslovna inteligenca .....	27
3.1.1 Orodja za razvoj in upravljanje sistema poslovne inteligence .....	28
3.1.2 Orodje za integracijo podatkov .....	30
3.1.3 Sistem za upravljanje baz podatkov za analize .....	30
3.1.4 Sistem za upravljanje relacijskih baz podatkov .....	33
3.1.5 Orodja za podporo poročanju .....	34
3.2 Orodje za samopostrežno poslovno inteligenco - Power BI.....	35
4 Uvedba sistema poslovne inteligence .....	36
4.1 Stanje pred uvedbo poslovne inteligence .....	36
4.2 Razvoj sistema poslovne inteligence .....	37
4.2.1 Zbiranje in analiza zahtev .....	37
4.2.2 Razvoj rešitve.....	38
4.2.2.1 Fizični načrt .....	38
4.2.2.2 Razvoj postopka ETL .....	39
4.2.2.3 Večdimenzijska baza .....	40
4.2.2.4 Poročila .....	41
4.3 Možnosti uporabe samopostrežne poslovne inteligence .....	41

4.3.1	Stroški opreme.....	42
4.3.2	Postopek ETL.....	42
4.3.3	Oblikovanje podatkovnega modela.....	43
4.3.4	Poročanje.....	45
4.3.5	Postavitev rešitve in varnost podatkov.....	45
	Sklep.....	47
	Literatura.....	48

## Kazalo slik

Slika 1	Arhitektura tradicionalnega poslovno inteligenčnega sistema.....	3
Slika 2	Uporaba poslovne inteligence na različnih nivojih organizacije.....	6
Slika 4	Podatkovno skladišče v normalizirani in večdimenzijski obliki z dodatno večdimenzijsko bazo.....	10
Slika 4	Podatkovno skladišče v dimenzijski obliki z dodatno večdimenzijsko bazo podatkov.....	10
Slika 5	Večdimenzijski model z osrednjo tabelo dejstev in štirimi dimenzijskimi tabelami... ..	11
Slika 6	Primer tabele dejstev.....	12
Slika 7	Primer dimenzijske tabele.....	13
Slika 8	Prikaz hierarhije.....	14
Slika 9	ER diagram zvezdne sheme.....	16
Slika 10	ER diagram snežinkaste sheme.....	17
Slika 11	Postopek za ekstrakcijo, preoblikovanje in nalaganje podatkov.....	18
Slika 12	ETL s preoblikovanjem podatkov v vmesnem področju.....	20
Slika 13	ETL s preoblikovanjem podatkov v pomnilniku.....	20
Slika 14	ELT, preoblikovanje podatkov v bazah podatkovnega skladišča.....	21
Slika 15	OLAP kocka s tremi dimenzijami.....	22
Slika 16	Arhitektura Microsoftovega sistema poslovne inteligence.....	28
Slika 17	Visual Studio.....	29
Slika 18	Arhitektura SQL Server Analysis Services.....	32
Slika 19	Posebna tabela dejstev nam omogoča mnogo proti mnogo relacije.....	33
Slika 20	Fizični načrt rešitve.....	39

## Kazalo tabel

Tabela 1	Koristi poslovne inteligence.....	7
Tabela 2	BUS matrika podatkovnega skladišča.....	38

## Uvod

Sodobno poslovno okolje zahteva nenehno sprejemanje odločitev na vseh ravneh podjetja od vodenja in upravljanja do operativne ravni. Za sprejemanje hitrih in pravih odločitev niso dovolj transakcijske informacije, ki jih podjetje prejema v velikih količinah od dobaviteljev in kupcev v obliki naročil, računov, plačil, reklamacij. Za ustrezno obravnavanje informacij poskrbijo sistemi poslovne inteligence (Kimball & Ross, 2013, str. 3). Uporaba teh je že dobro sprejeta, v zadnjem desetletju tudi v slovenskem prostoru. In vendar odločitve v veliko podjetjih ne temeljijo na uporabi poslovne inteligence (Imhoff & White, 2011, str. 4). Razlog za to so razmeroma visoki stroški uvedbe sistema poslovne inteligence, ki si ga večina majhnih in srednje velikih podjetij ne more privoščiti pa tudi nezmožnost dovolj hitrega uresničevanja zahtev po analizah.

Predvsem v zadnjih letih pa se pojavlja velika ponudba orodij in storitev, ki zadovoljujejo tudi takšne potrebe s postavitvijo okolja, v katerem si uporabniki informacij sami ali z minimalno podporo strokovnjakov s področja IT pripravijo zelene analize. Tak pristop je dobil ime samopostrežna poslovna inteligenca. Nova orodja narekujejo tudi poenostavljene pristope k uvajanju sistemov poslovne inteligence. Do te poenostavitve je prišlo postopoma, od začetkov pri Inmonovem pristopu, ki predvideva centralno podatkovno skladišče v normalizirani obliki z analizam namenjenimi izvedenimi področnimi podatkovnimi skladišči. Kimball to poenostavlja z uporabo samo analizam prilagojenega podatkovnega skladišča v večdimenzijski obliki. Že dolgo je podatkovnim skladiščem v relacijskih bazah dodana še posebna, analizam prilagojena baza podatkov, ki v samopostrežnih orodjih ostane edina shramba podatkov.

Namen diplomskega dela je ta razmeroma nov pristop samopostrežne poslovne inteligence primerjati s tradicionalnim pristopom. Na trgu je zaznati velik porast tako v ponudbi kot v investicijah v sisteme samopostrežne poslovne inteligence (Gartner, 2016), kar je dokaj lahka odločitev za manjše organizacije, ki sistema poslovne inteligence še nimajo. Primerjava je zato omejena na primer trgovskega podjetja, kjer smo že pred leti uvedli tradicionalni sistem poslovne inteligence s pomočjo orodja Microsoft SQL Server. Za primerjavo pa sem izbral orodje samopostrežne poslovne inteligence – Power BI.

Prvo poglavje obsega opredelitev poslovne inteligence, tako tradicionalnega kot samopostrežnega pristopa in njeno uporabnost. V drugem obravnavam glavne gradnike poslovne inteligence: podatkovno skladišče in glavne pristope pri načrtovanju tega, postopek ETL in analizam namenjene oblike shranjevanja podatkov. Obe poglavji temeljita na preučitvi strokovne literature in spletnih virov. V tretjem poglavju so predstavljena Microsoftova orodja poslovne inteligence, ki smo jih uporabili za primerjavo, to je SQL Server in Power BI, za kar sem uporabil pretežno tehnično dokumentacijo in izkušnje iz uporabe teh orodij. V zadnjem poglavju je opisana uvedba sistema poslovne inteligence od analize zahtev to same izvedbe. In končno preučitev možnosti uporabe orodja Power BI, ki je podana v opisni obliki



za vsak sestavni del sistema poslovne inteligence. Tu sem uporabil dokumentacijo in sam razvojni projekt, ki je nastal ob uvedbi sistema poslovne inteligence. Za proučitev Power BI sem uporabil tehnično dokumentacijo in praktične poizkuse uporabe orodja.

# 1 Poslovna inteligenca

## 1.1 Kaj je poslovna inteligenca

Poslovna inteligenca (angl. business intelligence, krat. BI) je razmeroma nov izraz, vendar vsekakor ne nov koncept. Je enostavno uporaba informacij, ki so že na voljo za pomoč pri boljšem in hitrejšem sprejemanju poslovnih odločitev. V strokovni literaturi najdemo kar nekaj različnih definicij:

- Poslovna inteligenca je splošen izraz, ki opisuje uporabo notranjih in zunanjih informacijskih virov organizacije z namenom sprejemanje boljših poslovnih odločitev (Kimball & Ross, 2002, str. 393).
- Poslovna inteligenca ni niti produkt niti sistem. Je arhitektura in zbirka integriranih operativnih aplikacij in aplikacij za podporo odločanju ter podatkovnih baz, ki omogočajo poslovnim uporabnikom enostaven dostop do podatkov (Moss & Atre , 2003, str. 31).
- Poslovna inteligenca predstavlja razne aktivnosti, povezane z analizami podatkov podjetja in zunanjih podatkov, za pomoč pri strateških, taktičnih in operativnih odločitvah in izvajanju potrebnih sprememb za doseg boljši rezultatov. Zajema zbiranje, analizo in razumevanje operativnih podatkov, podatkov o kupcih in dobaviteljih, finančnih podatkov, stanja na trgu in konkurence (Rainardi, 2008, str. 12).

V preteklih desetletjih se je za doseg teh ciljev uporabljalo več različno imenovanih sistemov. V začetku 80-ih let prejšnjega stoletja so bili priljubljeni direktorski sistemi (angl. executive information system, krat. EIS). EIS so v bistvu zbirali podatke iz različnih poročil v tako imenovane nadzorne plošče (angl. dashboard), ki ji je direktor lahko hitro pregledal. Pogosto je bilo to zbiranje podatkov ročno. EIS so pozneje nadomestili sistemi za podporo odločanju (angl. decision support system), ki pa so bili zelo podobni EIS (Jacobson & Misner, 2006, str. 4).

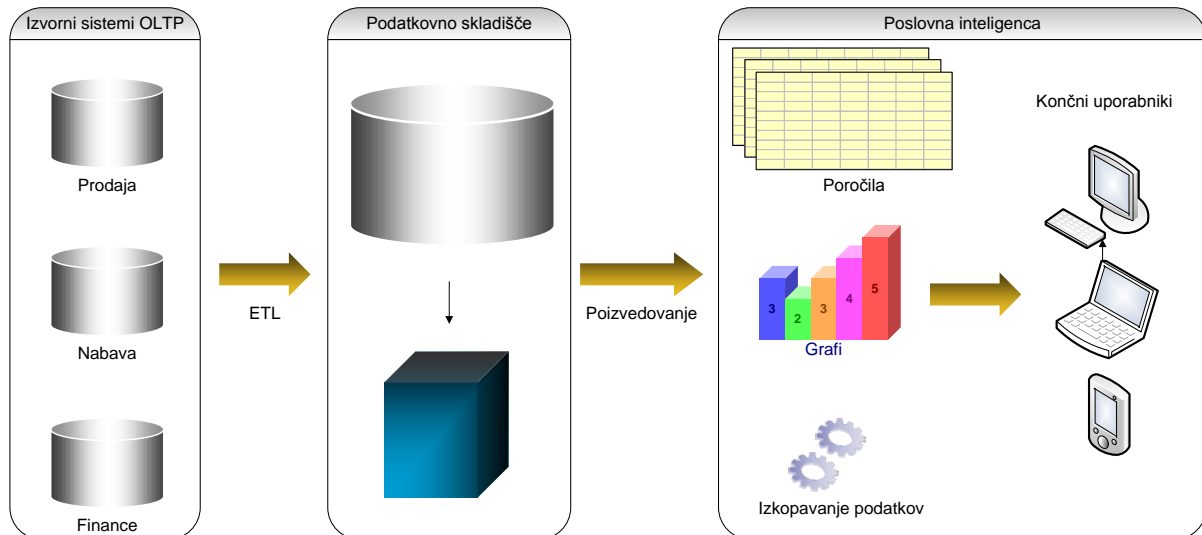
Kaj prinaša novega poslovna inteligenca? Vedno večje so potrebe po izdelavi poročil za vse nivoje organizacije in za vse odgovorne za sprejemanje odločitev. Ko je treba pripravljati poročila s hitrim odzivnim časom v velikih organizacijah, mora biti le to čim bolj avtomatizirano. Poslovno inteligenčni sistemi nudijo več fleksibilnosti in interaktivnosti ter poleg analiz preteklih podatkov nudijo tudi napovedovanje bodočih trendov.

Predvsem proizvajalci v vse bolj uporabljajo izraz poslovna inteligenca namesto izraza podatkovno skladišče. Tudi v strokovni literaturi se pri različnih avtorjih oba izraza pojmuteta

različno široko. Kimball, ki velja za enega volilnih strokovnjakov na tem področju, v svoji literaturi zelo pogosto uporablja kar kratico DW/BI sistem (Data Warehouse / Business Intelligence).

### 1.1.1 Tradicionalen sistem poslovne inteligence

*Slika 1 Arhitektura tradicionalnega poslovno inteligenčnega sistema*



Na sliki 1 so prikazani glavni sestavni deli tradicionalnega poslovno inteligenčnega sistema. Podatki organizacije so shranjeni v različnih oblikah, najbolj pogosto so to operative baze sistemov ERP pa tudi druge oblike, kot so preglednice in druge vrste datotek. Nahajajo se lahko na različnih, celo geografsko oddaljenih lokacijah.

S postopki ekstrakcije, preoblikovanja in nalaganja (angl. extract, transform, load, krat. ETL) podatke uredimo in združimo v podatkovno skladišče. Podatkovno skladišče predstavlja ena ali več relacijskih baz podatkov, ki pa jo pogosto dopolnjujejo še večdimenzijske baze podatkov, drugače poznane tudi kot OLAP kocke.

Podatkovno skladišče nam služi kot vir podatkov za različne obdelave. To so večinoma predpripravljene poizvedbe in poročila ter modeli za izkopavanje podatkov. Za prikaz in upravljanje s temi poročili služi razna programska oprema, kot so razni pregledovalniki poročil in grafov ter orodja za brskanje po OLAP kockah in modelih za izkopavanje podatkov. Navaden uporabnik navadno nima neposrednega dostopa do podatkovnega skladišča, ampak mora uporabljati vnaprej pripravljena poročila, ki pa vseeno nudijo visoko stopnjo interaktivnosti. Še posebej to velja za OLAP. Dostop do podatkovnega skladišča je dovoljen le informatikom in naprednim uporabnikom za pripravo ad-hoc analiz.

### 1.1.2 Samopostrežna poslovna inteligenca

Samopostrežna poslovna inteligenca je pristop, ki končnim uporabnikom omogoča primerljive možnosti, kot jih ponuja tradicionalni pristop, vendar večjo samostojnost in neodvisnost od IT strokovnjakov. To omogočajo namenska orodja in storitve, ki v zadnjih letih zelo napredujejo in se osredotočajo na štiri glavne cilje: omogočiti lažji dostop do virov podatkov, lažjo analiza podatkov, hitrejšo postavitev rešitve in enostavne in prilagodljive končne uporabniške vmesnike (Imhoff & White , 2011, str. 5). So vse v enem: podpirajo postopke ETL, so shramba podatkov in orodje za izdelavo in objavo poročil. Ker so namenjena poslovnim uporabnikom, jih odlikuje intuitiven uporabniški vmesnik in popolna interaktivnost v vseh korakih. Na primer: hiter ogled izvornih podatkov in podatkov po preoblikovanju ter hiter prikaz izdelanih poročil.

Samopostrežna poslovna inteligenca je odgovor na slabosti tradicionalnega pristopa: razvoj in spremembe so dokaj počasne, za razvoj pa so potrebni strokovnjaki s področja IT. Prav nenehno spreminjanje poslovnih zahtev in nezmožnost IT oddelkov, da jih uresničijo, sta glavna razloga za vpeljavo samopostrežne poslovne inteligenca (Imhoff & White , 2011, str. 10). Podobno kot pri razvoju programske opreme je v trendu hiter, agilni razvoj rešitev, ki ga lahko opravijo tudi poslovni uporabniki. To pa je zasuk od nekaterih vrednot tradicionalnega pristopa, kot je centraliziran razvoj rešitve in zbiranje podatkov na enem mestu, kar pomeni, da podatkovnega skladišča, kot ga določa tradicionalni pristop, ni več. Seveda stvari niso črno bele, potrebe po uporabi poslovne inteligenca so lahko od enostavne do zelo zahtevne. V skladu se tem se odločamo za uporabo ustreznega pristopa.

Glavne cilje samopostrežne poslovne inteligenca lahko strnemo v (Imhoff & White , 2011, str. 7):

- Omogočanje lažje uporabe in izboljševanje rezultatov sistema poslovne inteligenca. To je doseženo z objavo rezultatov preko uporabniških vmesnikov, ki so enostavni za uporabo, omogočajo skupinsko delo, napredne ponazoritve in iskanje informacij.
- Narediti orodja enostavnejša za uporabo. Poleg lažje končne uporabe morajo biti orodja lažja za uporabo tudi pri razvoju rešitve, torej pri pridobivanju in preoblikovanju podatkov, pripravi izračunov, kazalcev in poročil.
- Omogočanje hitrejših postavitev in upravljanja rešitev. To je omogočeno predvsem s ponudbo storitev ob oblaku, ki nudijo tako strojne kot aplikacijske zmogljivosti za postavitev sistema poslovne inteligenca.
- Omogočanje enostavnega dostopa do izvorov podatkov. Ključno pri samopostrežnem pristopu je, da omogočimo dostop do potrebnih podatkov tudi brez vnaprejšnjega shranjevanja v podatkovno skladišče ali pa s kombinacijo obojega.

Te lastnosti samopostrežne poslovne inteligenca omogočajo hitrejši razvoj postavitev rešitve ter razbremenitev IT oddelkov. Nižje cene orodij in poslovni modeli najema opreme pa nižje začetne ali celo končne stroške, če omogočijo celovito rešitev sistema poslovne inteligenca.

Prinaša pa tudi nevarnosti, ki jih tradicionalni pristop dobro rešuje. Te so posledica hitrega razvoja, samega pristopa in orodij, ki se uporabljajo pri samopostrežni poslovni inteligenci. Še posebej se pokažejo v večjih organizacijah.

- Potencialno slabša kakovost informacij. To je lahko posledica slabih izvornih podatkov ali slabše sledljivosti do izvora podatkov.
- Podvojenost podatkov in poročil, ki je predvsem posledica enostavnega objavljanja, deljenja z drugimi uporabniki.
- Manjša varnost in nadzor nad podatki, ki je posledica prej omenjenega lahkega objavljanja podatkov.
- Zelo velike količine in zapletene strukture podatkov so težko obvladljive izključno s samopostrežnimi orodji.

## **1.2 Uporaba poslovne inteligence**

Predvsem večje organizacije, ki se srečujejo z veliko količino podatkov, lahko z uporabo poslovne inteligence veliko pridobijo. Takih organizacij pa je vse več, saj se je količina podatkov, ki so na voljo organizaciji v zadnjih desetletjih zelo povečala. Večja količina podatkov sama po sebi ne pomeni prednosti, ker je iz njih zelo težko izluščiti uporabne informacije, s katerimi si pomagamo pri sprejemanju odločitev. Ta problem, torej razliko med naraščajočo količino podatkov in relativno malo uporabnimi informacijami imenujemo informacijska vrzel. Uporaba poslovne inteligence učinkovito zmanjšuje vrzel in tako pomaga pri vse večjem številu odločitev, ki so posledica trenda povečane hitrosti poslovanja.

Problem, povezan z veliko količino podatkov, je tudi degradacija hitrosti delovanja informacijskega sistema. Večina sistemov ERP ponuja tudi poročanje, ki se pogosto vrši neposredno nad operativnimi bazami podatkov. Pri velikih količinah podatkov takšno poročanje preveč ovira operativno delo v organizaciji. Vpeljava DW/BI sistema razbremeni operativni sistem in omogoči nemoteno in hitro delo, tako uporabnikom operativnega sistema, kot uporabnikom poročil.

Drugo veliko področje, ki ga naslavlja uporaba poslovne inteligence, predvsem tradicionalni pristop, je razdrobljenost podatkov. Podatki, potrebni za analize, se navadno nahajajo v različnih virih. V praksi so to baze različnih sistemov ERP, na primer ločeni sistemi za finance, nabavo in prodajo. Problem še stopnjujejo stari sistemi ERP, ki niso več v uporabi, vendar hranijo zgodovinske podatke. Veliko podatkov je tudi v manj strukturiranih oblikah, kot so preglednice in tekstovne datoteke. Vse to pomeni velike neuskklajenosti podatkov v različnih virih. Ko dodamo še geografsko razdrobljenost podatkov, ki nastopa v organizacijah z mnogo poslovnimi enotami, dobimo stanje s problemi, kot so:

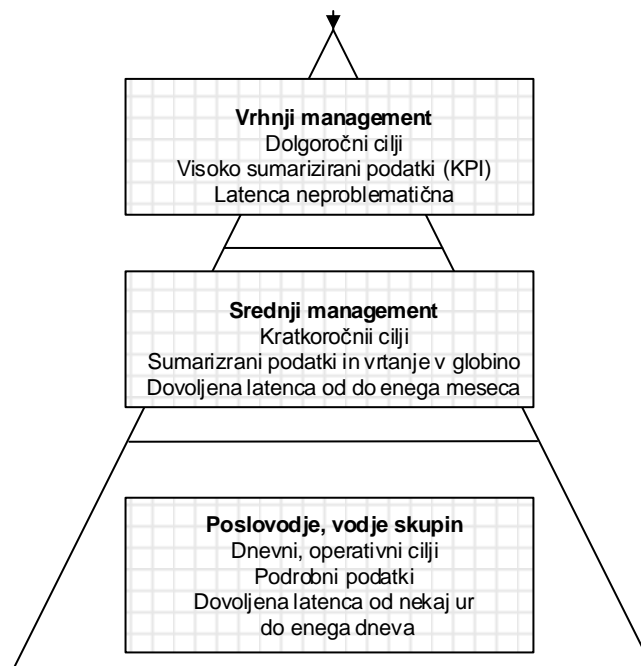
- Težavna integracija podatkov.
- Zahtevna, dolgotrajna priprava poročil, ki je za povprečnega uporabnika skorajda neizvedljiva.
- Velika možnost napake pri izdelavi poročil, kar vodi v nekonsistentne analize.

Uporaba poslovne inteligence nam v tej točki ponuja popolnoma avtomatiziran proces integracije, ki zagotavlja visoko kakovost podatkov, torej točnost informacij, usklajenost, hitrost dostopa in ažurnost. Odločitven postopek je posledično mnogo boljši in krajši.

Poročevalski kaos je problem, povezan z razdrobljenostjo podatkov. Poročila seveda najlaže nastanejo tam, kjer je izvor podatkov in to privede do neorganiziranega poročanja v organizaciji z množico poročil iz različnih virov. Da uporabniki pridejo do želenih informacij, se v poročila združujejo podatki iz različnih virov, vendar so to navadno le parcialne rešitve. Posledica je težaven dostop do poročil in netočnost, nekonsistentnost podatkov. Zelo pomembna stvar, ki jo ponujajo DW/BI sistemi, so orodja za centralno upravljanje in distribucijo poročil, kar prinese enotno infrastrukturo poročanja. Tak poročevalni sistem je dobra nadgradnja integracije podatkov. Zagotavlja udoben in hiter dostop do poročil na različne načine (spletni brskalnik, mobilne naprave, elektronska pošta).

Poslovna inteligenca je uporabna na vseh nivojih odločanje v organizaciji, vendar so za posamezne nivoje potrebni različni tipi informacij (Larson, 2009, str. 18). Takšne prilagoditve pa je v sistemih poslovne inteligence mogoče razmeroma enostavno implementirati.

*Slika 2 Uporaba poslovne inteligence na različnih nivojih organizacije*



*Vir: Prirejeno po (Larson, 2009, str. 18)*

- Vrhni management organizacije mora dobiti širok pogled na situacijo, ker je zadolžen za sprejemanje dolgoročnih odločitev. Poslovna inteligenca na tem nivoju mora biti prilagojena tem potrebam. Podatki morajo biti visoko sumarizirani. Pogosto sploh niso predstavljeni s števkami, ampak z različnimi indikatorji, ki povedo, ali je vrednost znotraj sprejemljivega območja, ali malo zaostaja oz. je popolnoma nesprejemljiva. Tako

predstavljeni podatki so imenovani ključni kazalniki poslovanja (angl. key performance indicators, krat. KPI). Navadno so predstavljeni z različnimi grafičnimi ikonami, kot so števcji, semaforji ali enostavnimi grafi. KPI tako uporabnikom zelo hitro pokažejo stanje bistvenih indikatorjev zdravja organizacije. Ker uporabniki na tem nivoju sprejemajo dolgoročne rešitve, jih zanima gibanje vrednosti v daljšem časovnem obdobju in jim tekoči podatki niso tako pomembni. Zato je za takšna poročila sprejemljiva tudi višja latenca. Latenca v poslovni inteligenci pomeni čas, ki preteče od nastanka podatka v poslovni transakciji in nalaganjem tega podatka v poslovno inteligenčni sistem.

- Srednji management postavlja kratkoročne cilje in plane za področja, za katera je odgovoren. Za svoje delo še vedno potrebujejo sumarizirane podatke, vendar je pogosto za te sumarne podatke treba preveriti podrobnosti, zato mora poslovna inteligenca na tem nivoju omogočati interaktivno vrtanje po podatkih v globino. Uporabniki poslovne inteligence na tem nivoju potrebuje bolj sveže podatke in zato je dovoljena latenca pogosto največ en dan.
- Na najnižjem nivoju so ljudje, ki skrbijo za dnevne posle. Odločati se morajo o razporeditvah virov in postavljanju ciljev za naslednji teden ali dan. Poslovna inteligenca na tem nivoju mora omogočati pregled sumariziranih podatkov, vendar tudi vrtanje v globino do najbolj podrobnih podatkov. Dovoljena latenca na tem nivoju je kvečjemu en dan navadno pa še bistveno manj.

Uporaba poslovne inteligence prinaša številne koristi. Najti je moč veliko poročil o koristih, ki so posamezne organizacije uresničile z vpeljavo poslovne inteligence. Tudi raziskava potrjuje (Thompson, 2004, str. 1), da je 19 % organizacij preseglo cilje po vpeljavi poslovne inteligence, še 60 % organizacij pa je vsaj približno doseglo zastavljene cilje. Raziskava tudi navaja, v kolikšni meri so organizacije dosegle posamezne koristi:

*Tabela 1 Koristi poslovne inteligence*

Korist	% organizacij, ki so dosegle korist
Hitrejše in natančnejše poročanje	81
Izboljšano sprejemanje odločitev	78
Izboljšana podpora strankam	56
Povečani prihodki	49
Prihranki pri neinformatiki	50
Prihranki pri informatiki	40

*Vir: (Thompson, 2004, str. 1)*

Zaradi široke uporabnosti poslovna inteligenca danes ni več razkošje, ki si ga privoščijo le največje organizacije. Postala je sestavni del informacijskega sistema in je v določenih panogah nepogrešljiva.

## 2 Podatkovno skladišče

### 2.1 Kaj je podatkovno skladišče

Podatkovno skladišče je definirano kot sistem, ki zbira in združuje podatke periodično iz izvornih sistemov v večdimenzijsko ali normalizirano podatkovno shrambo. Navadno hrani podatke dolga leta in se uporablja za poslovno inteligenco in ostale analitične aktivnosti. Podatki se navadno osvežujejo v paketih in ne ob posameznih spremembah v izvornih sistemih (Rainardi, 2008, str. 1). To je širša definicija podatkovnega skladišča, ki vključuje tudi poslovno inteligenco. Proizvajalci programske opreme in tudi nekatera strokovna literatura pa definirajo podatkovno skladišče precej ožje in s poudarkom na hranjenju podatkov, torej kot shrambo elektronskih podatkov organizacije, katere namen je olajšanje izdelave poročil in analiz. Kot že omenjeno, pa Kimball v svoji literaturi uporabi kar kratico DW/BI sistem in tako vključi oba pojma.

Tudi Inmon, ki velja za utemeljitelja podatkovnega skladiščenja, podobno definira podatkovno skladišče kot »predmetno usmerjeno, integrirano, nespremenljivo in časovno variantno zbirko podatkov za podporo odločanju« (Harinath, Zare, Meenakshisundaram, Carroll, & Guang-Yeu Lee, 2009, str. 4). Kaj pomenijo ti izrazi v kontekstu podatkovnega skladišča?

- **Predmetno usmerjena zbirka podatkov:** Operativne baze podatkov so organizirane okoli poslovnih procesov in velikokrat posamezni predmeti, kot so na primer stranke, izdelki med bazami podatkov posameznih procesov niso konsistentni. Nasprotno so podatkovna skladišča organizirana okoli predmetov.
- **Integrirana:** Operativne baze podatkov so pogosto nekonsistentne v veliko podrobnostih glede zapisa podatkov. V podatkovnem skladišču morajo biti podatki glede formatov in drugih vidikov konsistentni.
- **Nespremenljiva:** Ko so podatki enkrat naloženi v podatkovno skladišče, jih ne spreminjamo več. Seveda pa se podatkovno skladišče periodično dopolnjuje z novimi podatki.
- **Časovno variantna:** S tem je mišljeno, da podatkovno skladišče hrani tako zgodovinske kot tudi tekoče podatke. Nasprotno operativni sistemi navadno hranijo podatke le v okviru poslovnega leta.

#### 2.1.1 Operativni sistemi in podatkovna skladišča

Podatki organizacije so lahko shranjeni na dva načina. Vedno so shranjeni v operativnih sistemih, lahko pa tudi v podatkovnih skladiščih. Operativni sistemi so namenjeni sprotni obdelavi transakcij. Imenujemo jih tudi operativne baze podatkov ali sistemi za sprotno obdelavo transakcij (ang. online transaction processing systems; krat. OLTP). Podatke, ki so rezultat operativnih sistemov, imenujemo operativni podatki. Razlikujejo se glede na dejavnost organizacije, tipično pa so to računovodski podatki, podatki o prodaji, podatki o nabavi in podobno.

Operativni sistemi so optimizirani za hitro beleženje poslovnih dogodkov in ohranjanje celovitosti podatkov. To je doseženo z normalizacijo podatkov in uporabo entitetno relacijskega modela. Operativne baze podatkov so navadno polno normalizirane, torej izpolnjujejo večino od petih Coddovih pravil glede normalizacije podatkov. Posledično so podatki shranjeni v stotinah med seboj povezanih tabel. Za hranjenje se uporabljajo relacijski sistemi za upravljanje baz podatkov, ki nudijo veliko hitrost pri zapisovanju in popravljanju. Da se ta hitrost čez čas tudi ohrani, se stari podatki navadno periodično brišejo. Zaradi teh lastnosti se operativni sistemi slabše odrežejo pri poročanju in analizah (Rainardi, 2008, str. 10). Podatke je treba v poizvedbah združiti iz velikega števila tabel, kar je precej potratno in zato počasno. Veliko število tabel pomeni tudi preveliko zahtevnost za poslovnega uporabnika (Kimball & Ross, 2002, str. 11). V praksi se sicer navadno naredi kompromis in se zagotovi tudi hitrejše branje podatkov, na primer z uporabo indeksov. Seveda gre to na račun počasnejšega zapisovanja.

Nasprotno so podatkovna skladišča optimizirana za branje podatkov. To je navadno doseženo z denormalizacijo, kot to narekuje večdimenzijski model podatkovnega skladišča. Za še večjo hitrost so podatki dodatno shranjeni še v sešteti obliki, to so tako imenovani agregati. Navadno to dosežemo z uporabo večdimenzijskih baz podatkov. Podatkovno skladišče hrani tudi zgodovinske podatke še dolgo potem, ko so bili izbrisani iz operativnih sistemov (Rainardi, 2008, str. 11). Zaradi postopka nalaganja podatkov v podatkovno skladišče, ki se dogaja periodično, sveži podatki uporabnikom niso na voljo takoj. Zelo običajno je dnevno osveževanje podatkovnega skladišča, lahko pa so intervali zelo kratki. Podatkovno skladišče se glede na zahteve in možnosti dopolnjuje z novimi tabelami.

## **2.2 Večdimenzijski in normaliziran pristop**

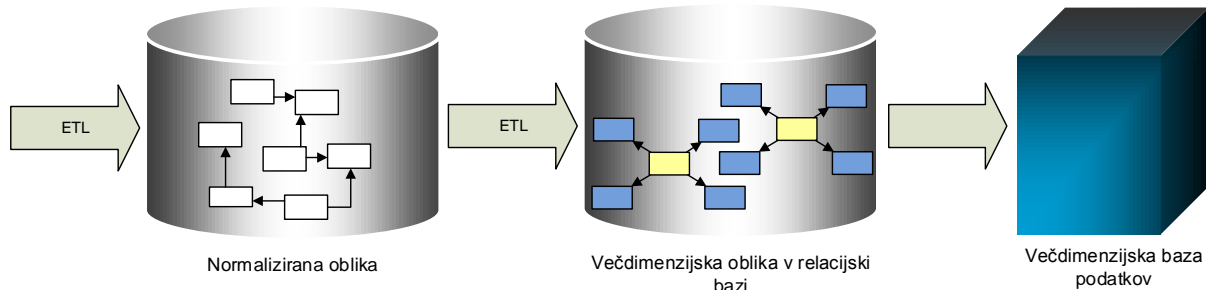
Obstajata dva pristopa k načinu oblikovanja podatkov v podatkovnem skladišču – večdimenzijski in normaliziran, ki sta povezana s Kimballovim in Inmonovim pristopom. Nasprotovanje med Kimballovo in Inmonovo filozofijo je znano in ga v grobem lahko povzamemo:

- **Kimball** zagovarja, da morajo biti podatki v podatkovnem skladišču obvezno v večdimenzijski obliki, organizirani v področna podatkovna skladišča, ki vsa skupaj tvorijo enotno podatkovno skladišče z uporabo konformnih dimenzij (Kimball & Ross, 2002, str. 12). Bistvo je, da večdimenzijska oblika omogoča boljšo in enostavnejšo analizo, kar priznava tudi Inmon.
- **Inmon** pa zagovarja enotno podatkovno skladišče v normaliziran obliki. Zaradi lažje analize pa se potem iz normalizirane oblike dodatno ustvarijo večdimenzijska področna podatkovna skladišča, ki pa nimajo nujno konformnih dimenzij (Inmon, 2005, str. 376).

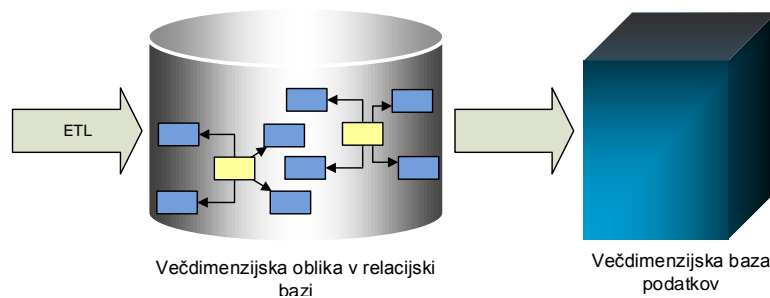


Vsak pristop ima svoje prednosti, velika večina sodobnih DW/BI sistemom pa hrani podatke v večdimenzijski obliki, lahko pa še dodatno v normalizirani. Za shranjevanje podatkov v normalizirani obliki vedno uporabimo relacijsko bazo, pri večdimenzijski obliki pa lahko poleg relacijske baze uporabimo tudi analizam prilagojeno večdimenzijsko bazo podatkov.

*Slika 3 Podatkovno skladišče v normalizirani in večdimenzijski obliki z dodatno večdimenzijsko bazo*



*Slika 4 Podatkovno skladišče v dimenzijski obliki z dodatno večdimenzijsko bazo podatkov*



### 2.2.1 Normaliziran pristop

Normalizirano podatkovno skladišče predstavlja ena ali več relacijskih baz podatkov, v katerih so podatki dokaj visoko normalizirani – navadno vsaj v tretji normalni formi in zato je podvojevanje podatkov majhno. Tabele so organizirane na področja, ki odražajo glavne skupine podatkov (na primer kupci, prodaja, finance). Glavne prednosti so enostavnejše in hitrejšo dodajanje podatkov in visoka stopnja prilagodljivosti tako glede dodajanje podatkov kot priprave poročil in analiz (Inmon, 2005, str. 359). Prav normalizirana oblika podatkov pa prinaša tudi slabosti. Veliko število tabel pomeni preveliko zahtevnost za poslovne uporabnike, podatke je treba združevati v pomenljivo obliko, kar negativno vpliva na hitrost delovanja.

### 2.2.2 Večdimenzijski pristop

V strokovni literaturi prevladuje širok konsenz, da večdimenzijski pristop ponuja boljše strukturo za predstavitev informacij uporabnikom. Z večdimenzijskim modelom na najboljši način dosežemo osnovne cilje podatkovnega skladišča (Mundy, Thornthwaite, & Kimball, 2006, str. 57):

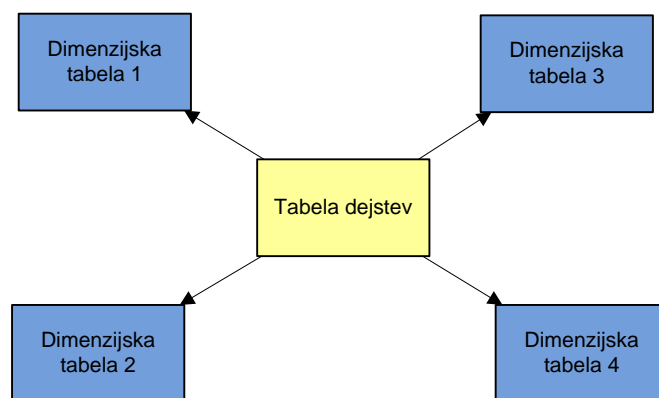
- Predstaviti zahtevane informacije uporabnikom na kar se da enostaven način.

- Ponuditi zahtevane informacije čim bolj hitro.
- Zagotoviti, da informacije iz podatkovnega skladišča kar se da točno sledijo izvornim poslovnim procesom.

Večdimenzijski model je razmeroma enostavnejši in lažje razumljiv uporabniku v primerjavi z normaliziranim modelom. Seveda tudi »kar se da enostaven« velikokrat ne pomeni, da je res enostaven. Tudi večdimenzijski model mora odražati poslovne procese, ki pa so v velikih organizacijah precej zapleteni. S prevelikim poenostavljanjem model izgubi podrobnosti, ki so nujne za razumevanje poslovnih procesov.

Tudi pri večdimenzijskem pristopu lahko podatke shranjujemo v relacijsko bazo podatkov. Vendar dosežemo večjo hitrost pri poizvedbah z denormalizacijo, ki jo uporabimo pri izdelavi dimenzij. Z denormalizacijo že pri pripravi podatkovnega skladišča združujemo podatke več tabel. Poizvedbe proti tako pripravljenih tabelah so hitrejšje kot proti polno normaliziranemu modelu.

*Slika 5 Večdimenzijski model z osrednjo tabelo dejstev in štirimi dimenzijskimi tabelami*



Večdimenzijski model sestavlja osrednja tabela dejstev (ali več tabel), ki hranijo transakcijske podatke in pripadajoče tabele dimenzij, v katerih so podatki, ki opisujejo dejstva. Tabele dejstev so popolnoma normalizirane, ker morajo hraniti veliko količino podatkov. Navadno predstavljajo več kot 95 odstotkov velikosti podatkovnega skladišča. Dimenzijske tabele pa so navadno denormalizirane, saj navadno ne vsebujejo veliko vrstic, vendar pa veliko stolpcev.

### 2.2.2.1 Tabele dejstev

V večdimenzijskem podatkovnem skladišču je tabela, ki hrani podrobne podatke o merah ali dejstvih, poimenovana tabela dejstev (angl. fact table). Zapis v tabeli pomeni mero, navadno številski podatek, ki pomeni velikostni razred nekega poslovnega dogodka. To je lahko na primer število prodanih izdelkov, znesek prodaje, količina zaloge in podobno. Te številke

imenujemo dejstva ali mere. Dejstvo se zabeleži skupaj s pripadajočimi podatki o dimenzijah, na primer vrednost prodaje skupaj s številko artikla, datum prodaje, številko kupca.

Slika 6 Primer tabele dejstev

FactProdaja	
FK1	ArtikelID
FK2	DatumRacunaID
FK5	PoslovalnicaID
FK6	StrankaID
FK4	KnjigaID
FK7	OperaterID
FK3	DokumentID
	ProdanaKolicina
	NetoProdanaVrednost
	ProdanaVrednostZDDV
	BrutoProdVrednostzDDV
	NabavnaVrednost

} Tuji ključi

} Dejstva

Tabelo dejstev poleg podatkov o dejstvih sestavljajo še tuji ključi, ki povezujejo dejstvo s točno določenimi primarnimi ključi v dimenzijskih tabelah in tako ohranjajo referenčno integriteto podatkov. Primarni ključ tabele je sestavljen iz tujih ključev, to je tako imenovani sestavljeni ključ.

Večina dejstev je številskih in zelo pomembna lastnost je aditivnost, kar pomeni, da jih je možno seštevati po dimenzijah. Večina dejstev je aditivnih kot na primer prodana količina ali vrednost prodaje v evrih. Aditivnost je pomembna, ker pri poizvedbah le redko iščemo samo eno vrstico iz tabele dejstev. Tipično gre za tisoče, lahko tudi milijone vrstic hkrati, ki se seštevajo (Mundy, Thornthwaite, & Kimball, 2006, str. 58). Enostavno poročilo prodaje po mesecih za eno leto res vsebuje samo dvanajst vrstic, vendar so te seštete iz tisočev vrstic iz tabele dejstev. Nekatera dejstva so pol aditivna, kar pomeni, da jih ne moremo seštevati po vseh dimenzijah. Največ je takšnih, ki jih ne moremo seštevati po časovni dimenziji kot na primer stanja zalog na dan. Obstajajo pa tudi dejstva, ki sploh niso aditivna, na primer cena izdelka ali devizni tečaj.

Pomembna je tudi stopnja podrobnosti, ki ji jih vsebuje tabela dejstev. To imenujemo granulacija. Vsi zapisi v tabelo dejstev morajo imeti enako granulacijo, to pomeni, da noben zapis ne sme biti seštet po kateri od dimenzij, ki jo vsebuje tabela dejstev (Mundy, Thornthwaite, & Kimball, 2006, str. 59). Tipičen primer bi bil, da bi hoteli v isto tabelo shraniti dnevne podatke o prodaji in letne plane prodaje. Zelo priporočljivo je tudi, da ohranimo čim višjo stopnjo podrobnosti, kolikor le omogočajo izvorni podatki. Tako zgrajene tabele dejstev ponujajo največjo prilagodljivost, podatke namreč lahko vedno seštejemo na višji nivo, obratno pa je nemogoče. Tako se zavarujemo proti morebitnim prihodnjim zahtevam za analizo podatkov po kakšni novi dimenziji.

### 2.2.2.2 Dimenzijske tabele

Dimenzije so osnova večdimenzijskega modela. Opisujejo objekte, ki nastopajo v poslovnih procesih, kot so izdelki, kupci, zaposleni. Z dejstvi zabeležimo poslovne dogodke, z dimenzijami pa te dogodke dodatno opišemo. Pri dogodku prodaje tako povemo, kdaj se je zgodil, komu smo prodali, katere izdelke smo prodali in podobno. Da dosežemo čim boljši opis poslovnega dogodka, dimenziji priprnemo čim več atributov, ki opisujejo zapis v dimenzijski tabeli. Povsem običajno je, da imajo dimenzije tudi po sto atributov.

Slika 7 Primer dimenzijske tabele

DimArtikli	
PK	<u>ArtikelID</u>
	NazivArtikla KratekNazivArtikla Barva Teza Velikost Cena Klasifikacija Gruca Proizvajalec Dobavitelj Opis

Dimenzijska tabela mora imeti en unikatni atribut, ki je primarni ključ tabele in jo povezuje s tabelo dejstev. Ostali atributi se lahko ponavljajo in to lahko izkoristimo za grupiranje zapisov v dimenzijski tabeli po ponavljajočih se atributih. Ni pa grupiranje vedno smiselno, čeprav se atributi ponavljajo. Artikel lahko enostavno grupiramo po atributih Barva, Klasifikacija, Velikost. Po atributu Cena je že težje - vsaj neposredno, lahko pa ustvarimo atribut Cenovni razred. Po atributu Opis pa je grupiranje nesmiselno, ker se verjetno ta razlikuje od artikla do artikla (Jacobson & Misner, 2006, str. 12)

Nekatere attribute, ki jih je mogoče grupirati, lahko združimo v naravne hierarhije. Tipičen primer so kategorije izdelkov. Izdelek spada natančno v eno podkategorijo, določena podkategorija pa v eno kategorijo. V poročilu lahko prikažemo kategorije izdelkov, uporabnik pa potem vrta do podkategorije in končno do izdelka.

*Slika 8 Prikaz hierarhije*



Pri naravnih hierarhijah člani nižjega nivoja določajo člana višjega nivoja. Obstajajo pa tudi hierarhije, ki nimajo te lastnosti. Izdelke lahko grupiramo najprej po Barvi nato pa še po Velikosti. Barva in Velikost prav tako tvorita hierarhijo, čeprav Velikost ne določa Barve. Tu gre samo za dejstvo, da lahko neka Velikost izdelka nastopa v več Barvah.

Za implementacijo naravnih hierarhij obstajata dve osnovni fizični strukturi podatkov:

- Rekurzivno strukturo uporabimo za hierarhije, ki nimajo vnaprej znane in stalne globine (Mundy, Thornthwaite, & Kimball, 2006, str. 79). Rekurzivni način je edini primeren za pravilen prikaz takšne hierarhije. S takšnimi hierarhijami se poslovnem okolju velikokrat srečamo, na primer: organizacijska struktura zaposlenih, postavke v poslovnih izkazih, sestavnice izdelkov.
- Nerekurzivne strukture so precej enostavnejše, uporabiti pa jih je mogoče le v primeru znane globine. Primer je zgoraj omenjena hierarhija kategorij in podkategorij izdelkov.

Hierarhije igrajo pomembno vlogo pri navigaciji po podatkih in pri načrtovanju agregacij. Hierarhije uporabniki najprej omogočijo pregled seštetih podatkov na visokem nivoju, na primer letna prodaja. S pomočjo hierarhije pa lahko uporabnik preide na podrobnejšo analizo, na primer prodaje v izbranem mesecu ali celo dnevu. Takemu prehodu iz višjega nivoja hierarhije v nižji nivo pravimo vrtanje v globino.

Izvorni podatki za dimenzije povečini nastopajo v normalizirani obliki torej v več tabelah. S postopkom denormalizacije lahko združimo vse attribute dimenzije, vključujoč hierarhije v eno dimenzijsko tabelo. Tako poenostavimo model z uporabniškega vidika, hkrati pa tudi izboljšamo hitrost poizvedb v primerjavi s polno normaliziranimi tabelami. Denormalizirana dimenzija še vedno vsebuje vse podatke in relacije kot normaliziran model, tako da z analitskega vidika ni izgubljeno nič. Stranki učinek je podvojevanje podatkov tabeli, kar pa z vidika velikosti ni problematično, saj dimenzijske tabele praviloma ne vsebujejo veliko zapisov. Poveča pa se seveda zapletenost priprave dimenzijske tabele zaradi postopka denormalizacije.

Ena bistvenih lastnosti dobro načrtovanega poslovno-inteligenčnega sistema je uporaba enakih dimenzij za vse poslovne procese v organizaciji. Na primer dimenzijo Artikli uporabimo tako

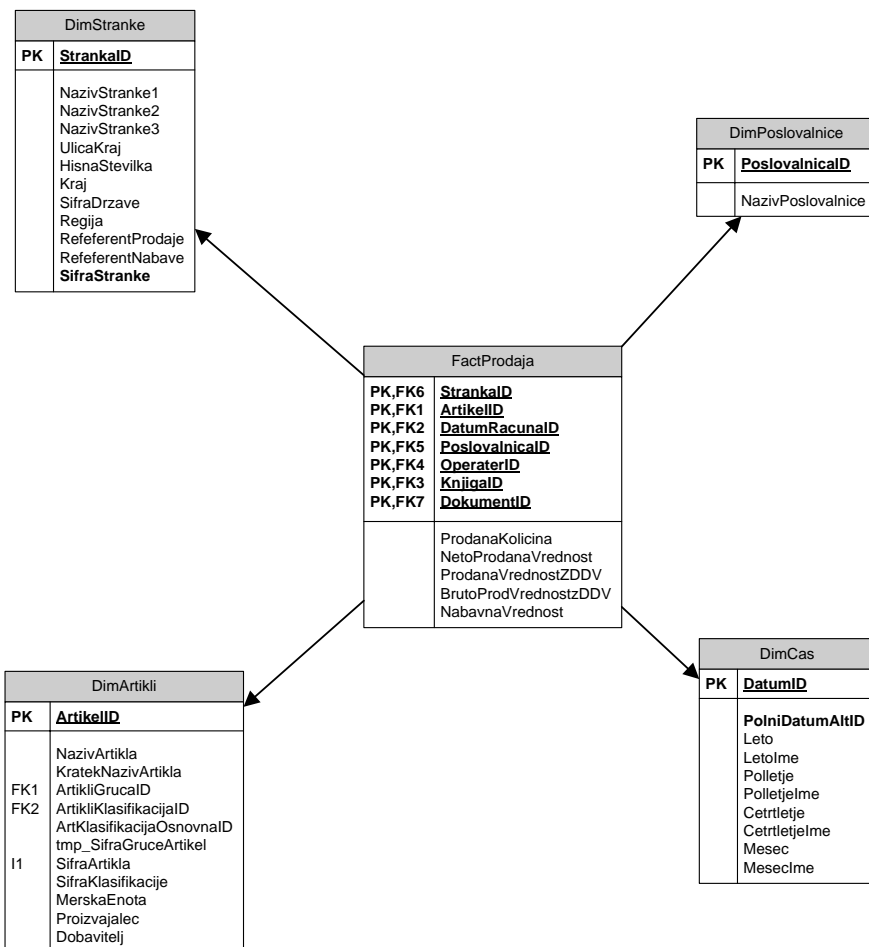
dimenzijskem modelu prodaje kot v večdimenzijskem modelu skladišča. Ker v obeh procesih nastopajo popolnoma enaki artikli, morata oba dimenzijska modela vsebovati isto dimenzijsko tabelo z enakimi ključi. V tem primeru govorimo o **konformnih dimenzijah**. Omogočajo nam analizo podatkov iz več poslovnih procesov. Tako lahko na primer iz večdimenzijskega modela prodaje dobim prodajno vrednost po artiklih, iz skladišča pa nabavno vrednost in iz teh dveh podatkov izračunamo razliko v ceni po artiklih. Tako analizo podatkov čez več poslovnih procesov imenujemo vrtnje počez. Natančna definicija konformnih dimenzij je, da sta dve dimenziji konformni, če vsebujeta eno ali več polj z enakimi imeni in vsebino. Ta polja so potem osnova za operacije vrtnja počez (Mundy, Thornthwaite, & Kimball, 2006, str. 64).

### 2.2.2.3 Shema večdimenzijskega podatkovnega skladišča

Pri oblikovanju večdimenzijskega podatkovnega skladišča lahko uporabimo dve različni shemi, zvezdno shemo in shemo snežinke, ki se razlikujeta predvsem stopnji denormalizacije dimenzijskih tabel:

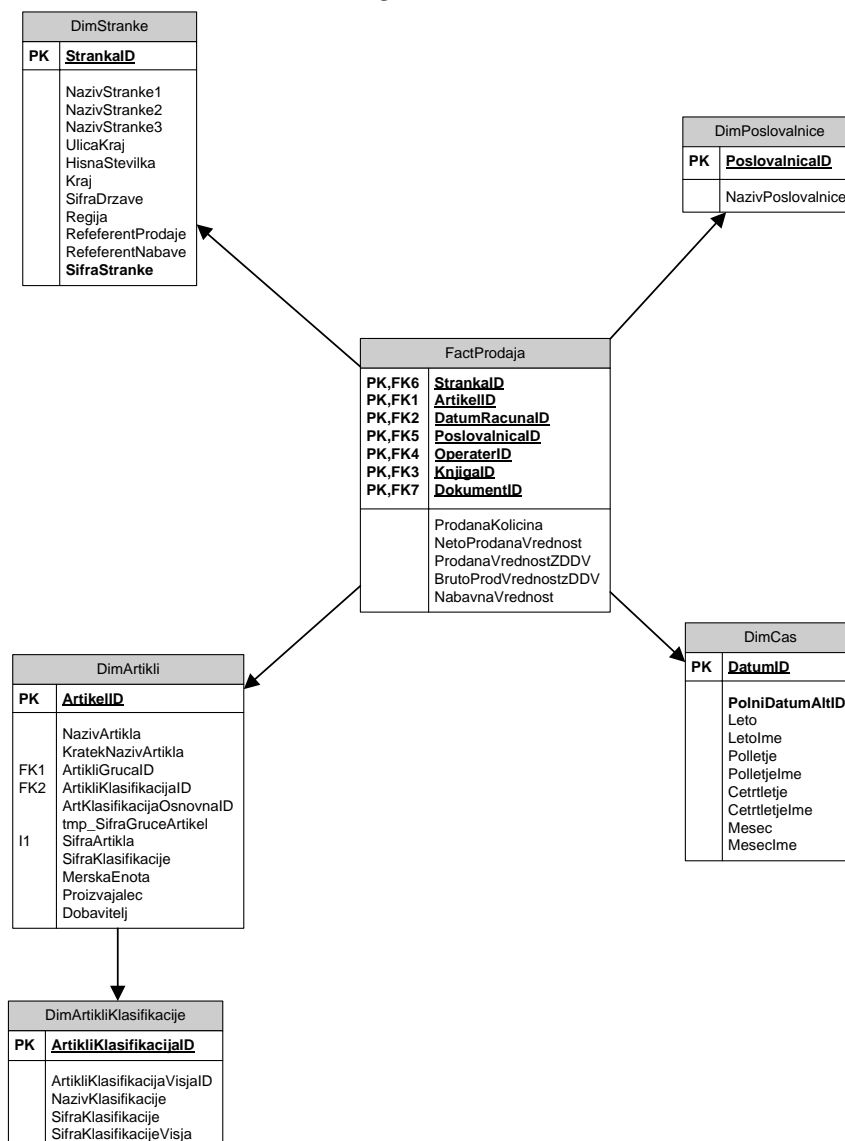
- **Zvezdna shema** je sestavljena iz osrednje tabele dejstev in več dimenzijskih tabel. Na sliki spodaj vidimo, da ta predstavlja poslovne entitete in povezave med njimi na zelo jasen in lahko razumljiv način. Dimenzijske tabele so popolnoma denormalizirane in tako vsebujejo vse nivoje dimenzije v eni tabeli. Tipičen primer na sliki spodaj je tabel DimCas.

Slika 9 ER diagram zvezdne sheme



- **Shema snežinke** je razširitev zvezdne sheme, v kateri vsaj nekatere dimenzijske tabele niso popolnoma denormalizirane, ampak ima vsak nivo dimenzije svojo tabelo. Na sliki spodaj to velja za dimenzijo Artikli, ki je sestavljen iz dveh tabel. DimArtikliKlasifikacije predstavlja višji nivo dimenzije, DimArtikli pa nižjega. Pri poizvedbah je tak model malo zapletenejši, ni pa treba izvesti denormalizacije ob prenosu podatkov iz operativne baze.

Slika 10 ER diagram snežinkaste sheme



### 2.2.3 Področno podatkovno skladišče

Različni oddelki znotraj organizacije imajo lahko povsem različne potrebe po analizah podatkov. Oddelek nabave za svoje delo potrebuje povsem drugačne podatke kot oddelek prodaje in spet drugačne kot oddelek financ. Potrebam posameznih oddelkov lahko zadostimo z izgradnjo manjših verzij podatkovnega skladišča, ki so prilagojena zahtevam posameznih oddelkov.

Področno podatkovno skladišče je tako definirano kot namensko, predmetno orientirano odlagališče podatkov, ki je namenjeno točno določeni skupini uporabnikov. Organizacija ima tako več področnih podatkovnih skladišč, ki služijo potrebam finančnega, prodajnega in drugih oddelkov. Področno podatkovno skladišče je navadno organizirano v obliki zvezdne sheme, ki jo tvori osrednja tabela dejstev in več dimenzijskih tabel. V nasprotju je enotno



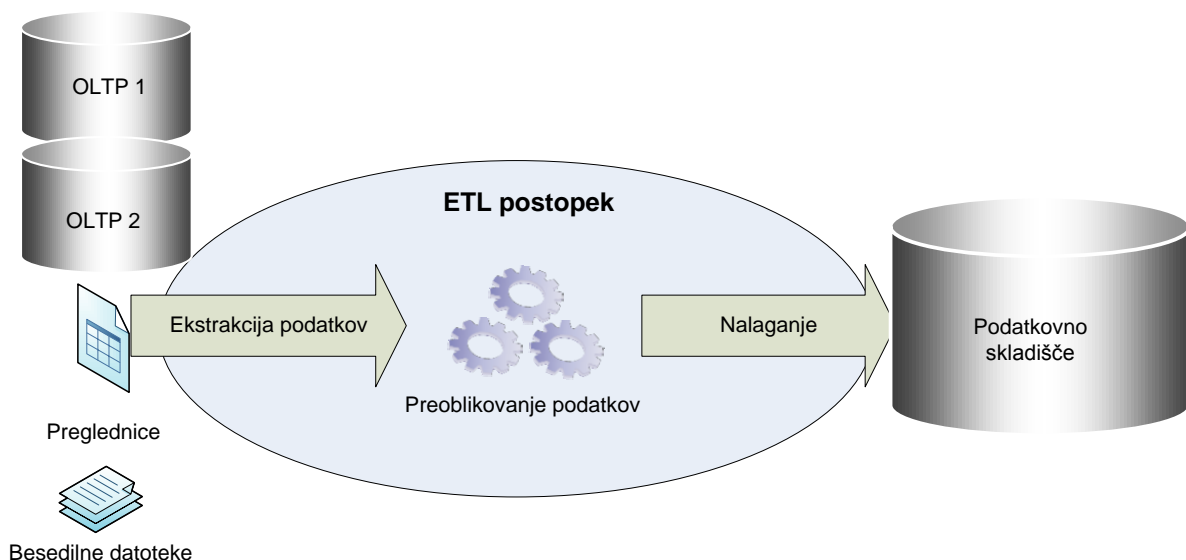
podatkovno skladišče eno in edino odlagališče podatkov organizacije, ki zajema vsa področja poslovanja in vse oddelke organizacije.

Ralph Kimball začne z izgradnjo področnega podatkovnega skladišča kot večdimenzijskega modela za podatke posameznega oddelka organizacije. Na podatkovno skladišče pa gleda kot zbirko vseh področnih. To je tako imenovani pristop od spodaj navzgor. Pri takem pristopu je treba uporabiti konformne dimenzije v posameznih področnih podatkovnih skladiščih. Z uporabo konformnih dimenzij vzpostavimo relacije med področnimi podatkovnimi skladišči in omogočimo tako imenovane analize vrtnja počez. Takih potreb je v praksi kar veliko. Na primer nekoga iz prodajnega oddelka zanima, kakšne so bile vrednosti prodaje po posameznih kupcih, zanima ga pa tudi ali so ti kupci redno plačevali. Prvo informacijo lahko dobi v področnem podatkovnem skladišču prodaje, drugo pa na enostaven način iz financ, ker obe podatkovni skladišči uporabljata skupno dimenzijo Kupci. Tak pristop je tudi lažje implementirati, saj imamo opravka z manjšimi področji. Projekt lahko začnemo z implementacijo za oddelek prodaje, in postopoma nadaljujemo z drugimi oddelki organizacije.

### 2.3 Postopek ETL

Postopek, s katerim pridobimo podatke iz izvornih sistemov in jih ustrezno preoblikujemo, je poznan s kratico ETL (angl. Extract, Transform, Load) ali ekstrakcija, preoblikovanje in nalaganje podatkov. Pri tradicionalnih pristopih je v tej fazi že določena struktura podatkovnega skladišča. V skladu s to preoblikujemo podatke in jih naložimo v podatkovno skladišče. Pri orodjih samopostrežne poslovne inteligence je pristop malo drugačen: podatke preoblikujemo, dokler ne dobimo zelene strukture. Zapisa v podatkovno skladišče, kot ga določa tradicionalni pristop, ni, podatki se nahajajo v delovnem pomnilniku. Cilj pa je v obeh primerih enak: iz izvornih sistemov prenesti podatke v obliko primerno za analize.

Slika 11 Postopek za ekstrakcijo, preoblikovanje in nalaganje podatkov



Na sliki zgoraj vidimo, da se postopek začne z ekstrakcijo podatkov iz različnih virov, pridobljeni podatki se ustrezno preoblikujejo in na koncu zapišejo v podatkovno skladišče. Postopek se izvaja periodično, glede na zahteve po svežih podatkih. Najbolj običajno je izvajanje postopka vsako noč, ko sistemi OLTP niso v uporabi. Dnevno ali tedensko izvajanje ETL je v praksi najbolj pogosto, vendar so zahteve lahko tudi višje in načeloma lahko ETL vršimo tudi ob vsaki spremembi v sistemu OLTP. Zelo pomembno je, da sistema OLTP ne oviramo preveč. Postopek ETL izvedemo lahko na dva načina:

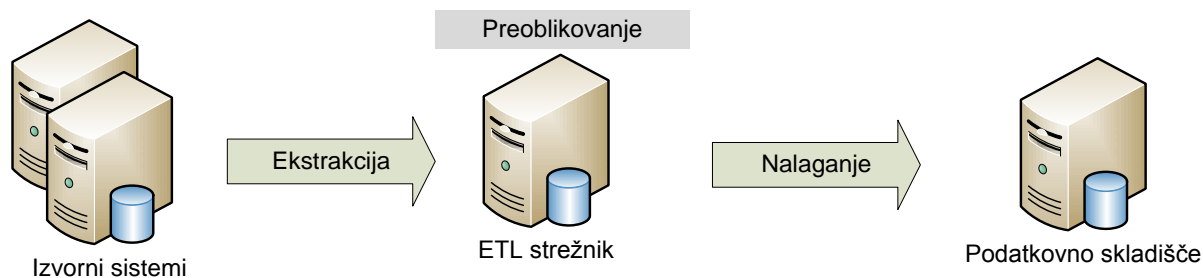
- **Vsakokratni prenos vseh podatkov** je primeren, kadar imamo opraviti z relativno majhnimi količinami podatkov in je časovno okno za postopek ETL dovolj veliko. Prednost takšnega načina je v enostavnosti, zato ga je vredno upoštevati, če zahteve in razmere to dopuščajo.
- **Prenos samo novih podatkov** je običajen pri velikih količinah podatkov in pogostejšem osveževanju podatkov v podatkovnem skladišču, kar navadno pomeni tudi krajša časovna okna za postopek ETL. Tak postopek je precej bolj zapleten in je sestavljen iz:
  - **Nalaganja začetnega stanja:** Ob uvedbi DW/BI sistema je potrebno najprej naložiti obstoječe podatke. Ti se nahajajo v operativnih bazah, lahko pa tudi kot zgodovinski podatki v raznih arhivih. Postopek se izvede samo enkrat.
  - **Inkrementalno nalaganje:** Ob vsakem zagonu postopka ETL v podatkovno skladišče prenašamo samo nove in spremenjene podatke. Ugotavljanje, kateri podatki so novi, je dokaj preprosto, kadar je vir relacijska baza podatkov, veliko težje pa, kadar so vir podatkov razne datoteke. Najtežje pa je obravnavati izbrisane podatke. Navadno je treba brisanja v bazi OLTP sprti beležiti v dnevnik, sicer se sled za izbrisanimi zapisi izgubi. Zaradi zapletenosti je postopek dovzeten za napake, zato je pri takem načinu obvezno potrebo vršiti kontrole pravilnosti podatkov (Moss & Atre , 2003, str. 218).

### 2.3.1 Pristopi in arhitektura

ETL postopek lahko naredimo na več načinov (Rainardi, 2008, str. 175):

**Tradicionalni pristop** se začne z ekstrakcijo podatkov v vmesno področje (angl. staging area), kjer se podatki preoblikujejo. Od tu se na koncu naložijo v podatkovno skladišče. Vmesno področje predstavljajo navadno besedilne datoteke ali pa baza podatkov. To pomeni, da je treba iz izvornih sistemov podatke najprej zapisati na disk, jih spet prebrati in šele na to preoblikovati. Branje in pisanje iz diska je v primerjavi z branjem iz pomnilnika veliko počasneje, vendar je tak pristop nujen, če imamo opraviti z velikimi količinami podatkov. V takem primeru se navadno uporabi namenski strežnik za ETL.

Slika 12 ETL s preoblikovanjem podatkov v vmesnem področju



**Preoblikovanje v pomnilniku:** Alternativa je, da izvornih podatkov ne zapisujemo v vmesno področje, ampak se preoblikovanje zgodi v pomnilniku. Od tu se preoblikovani podatki zapišejo v podatkovno skladišče. Ker smo vmesni korak zapisovanja in branja iz diska izpustili, je postopek veliko hitrejši, saj je delo s pomnilnikov neprimerno hitrejše od dela z diskom. Tak postopek se uporablja lahko na namenskem strežniku za ETL ali pa se ETL odvija na strežniku, kjer je podatkovno skladišče.

Slika 13 ETL s preoblikovanjem podatkov v pomnilniku



**ELT:** Kot že kratica namiguje, pri takem postopku podatke najprej prepišemo v podatkovno skladišče in jih tam preoblikujemo s pomočjo posodabljanja (angl. update) v bazah podatkovnega skladišča. Tudi tu se izognemo vmesnemu področju, vendar tudi popravljanje podatkov v bazi intenzivno bere in piše na disk in je zato počasnejše od preoblikovanj v pomnilniku. Tak pristop bi uporabili, če bi podatkovno skladišče uporabljalo zelo zmogljivo strojno opremo, ker se nanj prenese skoraj celotno breme postopka ETL.

Slika 14 ELT, preoblikovanje podatkov v bazah podatkovnega skladišča



Glede na zgoraj opisane pristope moramo izbrati še primerno orodje za izdelavo postopka ETL:

- **Namenska orodja ETL**, kot so Microsoft SQL Server Integration Services, Oracle Data Integator, so v večini primerov najboljša izbira. Imajo namreč že pripravljena ogrodja, kjer je poskrbljeno za običajne težave procesa ETL, kot je povezovanje na razne baze podatkov, razne formate datotek in druge vire. Ponujajo orodja za tipična opravila, kot so razvrščanje, združevanje, preračunavanje. Poleg tega pa dopuščajo izdelavo skript za preoblikovanje podatkov po meri. Omogočajo nam izdelavo postopka ETL po kateremkoli pristopu, opisanem zgoraj. Tudi ETL v samopostrežnih orodjih omogoča primerljive možnosti, sam postopek pa se izvaja v celoti v delovnem pomnilniku.
- **Poljuben programski jezik** omogoča zelo prilagodljive rešitve, vendar je treba narediti postopek ETL popolnoma od začetka, kar je zelo zahtevno.

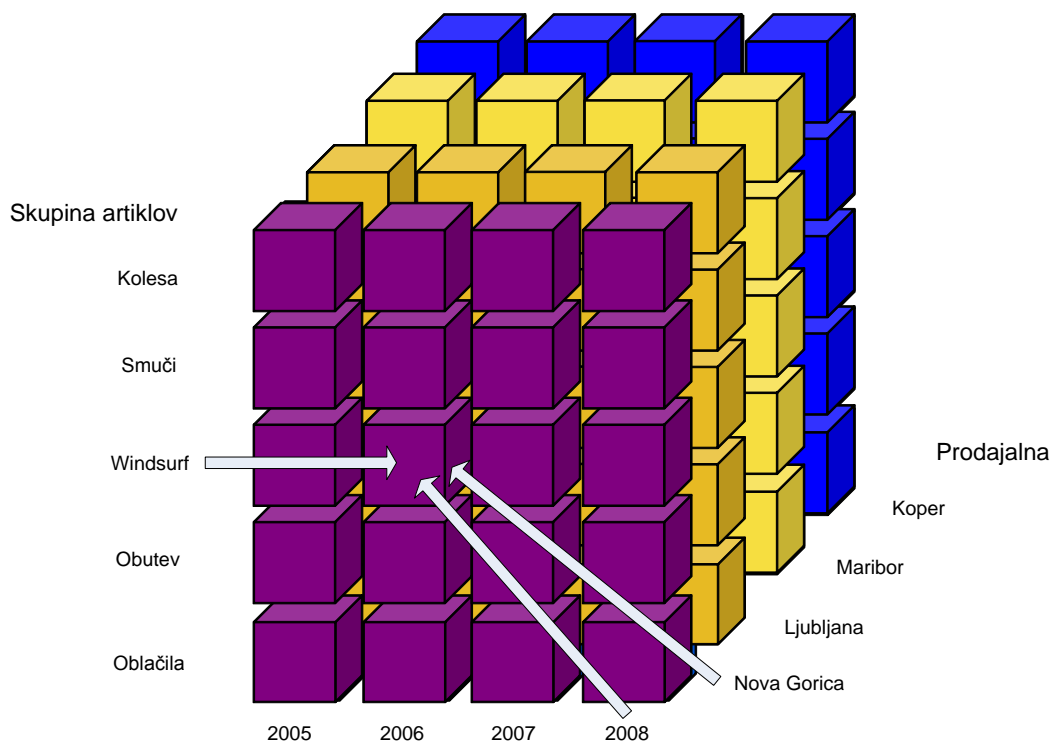
## 2.4 Večdimenzijske in stolpčno usmerjene baze podatkov

Večdimenzijske in v zadnjih letih stolpčno usmerjene baze podatkov najdemo praktično v vseh sistemih poslovne inteligence. Pri tradicionalnih pristopih kot nadgradnjo podatkovnega skladišča v relacijski bazi, lahko pa tudi kot edino shrambo podatkov, predvsem pri samopostrežni poslovni inteligenci. Shramba podatkov v takšni obliki omogoča hitrejše poizvedbe. Pomeni tudi dodaten logični sloj nad podatkovnim skladiščem v relacijski bazi, ki omogoča razširitve večdimenzionalnega modela. Omogočeno je prirejanje dodatnih metapodatkov, ki izboljšajo navigacijo po podatkih, kot je določanje hierarhij v dimenzijah. Prinašata tudi analizam prilagojene poizvedovalne jezike, ki so v pomoč tudi pri izračunih novih mer, ki jih dodamo v podatkovni model. Prednost se pokaže že pri agregaciji podatkov, poenostavljeno je na primer obravnavanje pol aditivnih mer in zahtevnejših izračunov, kot so tekoče vsote, primerjave s preteklimi obdobji. V primeru uporabe samo podatkovnega skladišča v relacijski bazi zgornjih funkcionalnosti ne moremo vključiti v sam podatkovni model in jih moramo obravnavati na strani odjemalca.

## 2.4.1 Večdimenzijska baza podatkov

Večdimenzijska baza podatkov (krat. VBP) je posebna oblika baze podatkov. V relacijski bazi podatkov so podatki shranjeni v tabelah. Nahajališče posameznega podatka je določeno z imenom tabele, znotraj tabele pa z imenom stolpcev in ključem vrstice. V VBP so podatki v celicah večdimenzionalnega polja (angl. multidimensional array), pozicija vsake celice pa je določena z več indeksi, ki jih predstavljajo dimenzije (Rainardi, 2008, str. 378). Vsaka celica predstavlja poslovni dogodek, ki je številsko ovrednoten in ga imenujemo mera (angl. measure). VBP je poznana tudi kot OLAP kocka ali hiperkocka (angl. hypercube), ali kar kocka, kar je takoj jasno, če pogledamo spodnjo sliko OLAP kocke s tremi dimenzijami. Izraz hiperkocka natančno pomeni kocko s štiri ali več dimenzijami, in v praksi ima tipična OLAP kocka res veliko več kot tri dimenzije. Ker pa bi to težko grafično predstavili, je OLAP kocka s tremi dimenzijami idealen primer.

Slika 15 OLAP kocka s tremi dimenzijami



Na zgornji sliki je predstavljena OLAP kocka s tremi dimenzijami Skupina artiklov, Leto in Prodajalna in merama Prodana količina in Prodana vrednost. Vsaka celica, na sliki predstavljena kot majhna kocka, nosi torej količino in vrednost prodaje.

VBP navadno črpa podatke iz relacijskega podatkovnega skladišča v večdimenzijski obliki. Tabele dejstev so vir podatkov za mere, dimenzijske tabele pa za dimenzije kocke. In če že imamo podatkovno skladišče, kakšen je smisel vpeljave VBP? VBP prinaša veliko prednosti (Mundy, Thornthwaite, & Kimball, 2006, str. 124):

- **Izboljšano navigacijo** po podatkih. Če želimo prikazati skupno prodajo windsurf opreme, prodane leta 2008 v prodajalni Nova Gorica, je to zelo enostavno. Iz kocke samo izberemo določeno mero in člane dimenzije. Že v večdimenzijskem modelu v relacijski bazi bi morali te podatke združiti vsaj iz štirih tabel (tabele dejstev in treh dimenzijskih tabel), podatke ustrezno filtrirati in sešteti. V normaliziranem modelu pa bi bilo tabel še več. Enako enostavno v kocki pridemo seštevkom. Če bi želeli vsoto vse prodaje v Novi Gorici (na sliki označeno z vijoličasto barvo) enostavno izberemo celotni dimenziji Čas in Skupina artiklov ter člana Nova Gorica dimenzije Prodajalna. Omogočajo pa še veliko drugih operacij, kar si bomo ogledali v o poglavju OLAP.
- **Hitrejša poizvedbe:** OLAP kocke so veliko hitrejša predvsem pri vračanju visoko sumarnih podatkov. Najpomembnejši dejavnik je agregacija podatkov. Drugi razlog je manjša velikost na disku v primerjavi z relacijskimi bazami podatkov, saj podatke hrani v posebni stisnjeni obliki. Manjša velikost pomeni manj branja iz diska, za kar pa vemo, da je relativno počasno.
- **Agregacija podatkov:** To so seštevki več celic, ki se izračunavajo in shranijo med pripravo ali tako imenovanim procesiranjem kocke. Agregati se pripravijo iz tabele dejstev s spreminjanjem granularnosti posamezne dimenzije. Na primer za dimenzijo Čas, ki ima nivoje Leto, Mesec, Dan, se bodo pripravili agregati po dnevih, mesecih, letih in za vsa leta skupaj. Sistemi za upravljanje VBP že sami poskrbijo za agregate, lahko popolnoma samodejno z vgrajenimi algoritmi, lahko jih ročno vsilimo, lahko pa se agregati izračunavajo tudi na podlagi najpogostejših poizvedb. Čeprav so agregati zelo koristni, se ne shranjujejo vsi, saj je kombinacij vseh dimenzij preprosto preveč in bi zavzeli preveč prostora na disku, kar bi spet negativno vplivalo na hitrost. To pa seveda ne pomeni, da nekaterih seštevkom s poizvedbo ne moremo dobiti. Agregati, ki niso shranjeni, se namreč izračunajo sproti. (Veerman, Lachev, Sarka, & Loria, 2006, str. 222).
- **Lažje vzdrževanje** v primerjavi z relacijsko bazo podatkov. VBP potrebujejo veliko manj vzdrževanja kot relacijske in z naraščanjem količine podatkov ali intenzivnosti uporabe še vedno zagotavljajo veliko hitrost.

Šibka točka VBP pa je čas, potreben za procesiranje kock. Kadarkoli je izvorna baza podatkov ažurirana, je potrebno ponovno procesiranje VBP, torej agregati se morajo ponovno izračunati. Vendar tudi to ni tako velika slabost, saj procesiranje izvedemo kot zadnji korak postopka ETL, potreben pa je seveda dodaten čas. To je lahko večji problem le pri zelo pogostih osveževanjih podatkov DW/BI sistema. Kljub temu pa VBP prinaša toliko prednosti, da danes skoraj ni več DW/BI projekta, ki ne bi implementiral tudi VBP.

Prav tako kot sistemi za upravljanje relacijskih baz skrbijo za upravljanje z relacijskimi bazami podatkov, podobno obstajajo sistemi za upravljanje VBP, poznani tudi kot strežniki OLAP. Na trgu obstaja kar velika ponudba: Microsoft SQL Analysis Services, Hyperion Essbase, Cognos Powercube, Pentaho Mondrian in drugi. In tudi podobno kot za relacijske baze obstaja standard SQL, za delo z VBP obstaja industrijski standard imenovan XML for

Analysis (krat. XMLA). Z XMLA upravljamo objekte v VBP, vključuje pa tudi poseben poizvedovalni jezik imenovan Multidimensional Expressions, bolj poznan pod kratico MDX. MDX je poizvedovalni jezik prirejen za delo z VBP in je precej podoben SQL, nekateri elementi pa spominjajo na izraze za izračune v preglednicah (kot je na primer Excel).

## 2.4.2 Stolpčno usmerjeno shranjevanje

Bistvo stolpčno usmerjenega shranjevanja (angl. column oriented storage) je v drugačni organizaciji zapisa (Ali, 2012). Tradicionalni zapis v sistemih za upravljanje baz podatkov je vrstično usmerjen. Razliko lahko ponazorimo s poenostavljenim primerom spodnje tabele.

Vrstica	Šifra	Izdelek	Cena
1	51	Kolo	500
2	60	Žoga	40
3	67	Lopar	40

Vrstično usmerjen zapis na disk:

1: 51, Kolo, 500;

2: 60, Žoga, 40;

3: 67, Lopar, 40;

Tak sistem zapisa je učinkovit pri vračanju in spreminjanju celotnih podatkov ene vrstice, kar je pogosto v operativnih sistemih. Ni pa učinkovit za vračanje večjih nizov podatkov in samo določenih stolpcev, kar je običajno pri poizvedovanju in analizah.

Stolpčno usmerjen zapis:

51:1, 60:2, 67:3;

Kolo:1, Žoga:2, Lopar:3

**500:1, 40:2, 3;**

Tak zapis je učinkovit pri vračanju velikih nizov podatkov za samo določene stolpce. Enake vrednosti v stolpcu so zapisane samo enkrat, navedene so samo reference do različnih vrstic, kjer se vrednost nahaja, kar pomeni manjšo porabo prostora in hitrost pri iskanju vrednosti. Poleg tega se uporabljajo še standardni algoritmi za stiskanje podatkov. Zaradi navadno večje uniformnosti podatkov v stolpčno usmerjenem zapisu dosežejo večjo učinkovitost kot pri vrstično orientiranemu (Ali, 2012). Običajne stopnje stiskanje podatkov iz relacijskih baz so okoli 1:10, so pa zelo odvisne od stopnje ponavljanja enakih vrednosti v stolpcih (Russo, 2013). V podatkovnih skladiščih je predvsem v tabelah dejstev, ki običajno predstavljajo večino podatkov, tako ponavljanje vrednosti veliko.

Zaradi velikega stiskanja podatkov in večje cenovne dostopnosti glavnega računalniškega pomnilnika se velikokrat uporablja skupaj s tehnologijo pomnilniških baz podatkov (angl. in-

memory database). To je sistem za upravljanje baze podatkov, ki uporablja glavni računalniški pomnilnik za shranjevanje podatkov, kar je mnogo hitreje od uporabe diska.

Zaradi svojih lastnosti sta tehnologiji stolpčno usmerjenega shranjevanja in pomnilniških baz zelo primerni za DW/BI sisteme in predstavljata skoraj revolucijo na tem področju:

- Kot alternativa večdimenzijski bazi podatkov, ki se v tradicionalnih DW/BI sistemih uporablja poleg podatkovnega skladišča v relacijskem SUBP. Pri samopostrežnih sistemih pa kot edina shramba podatkov.
- Tudi novejši relacijski SUBP so začeli vključevati to tehnologijo v obliki stolpčno usmerjenih indeksov (angl. columnstore index). Ti so sicer namenjeni operativnim sistemom. Ker se ti indeksi samodejno ažurirajo, so zelo hitri pri poizvedbah in imajo majhen vpliv na transakcijski sistem, so lahko nadomestilo za DW/BI sistem, če je razlog njegove vpljave razbremenitev transakcijskega sistema.

V primerjavi z večdimenzijskimi bazami so stolpčno usmerjene pomnilniške baze hitrejšje v večini primerov (Ingham, 2012):

- Pri osveževanju podatkov ni potrebno procesiranje – izračunavanje agregatov in zapisovanje teh na disk. Podatki se naložijo neposredno v pomnilnik, kjer se agregati sproti izračunavajo, kar poteka hitreje. Prednost sprotnega izračuna je tudi možnost izbire agregacijske funkcije neposredno ob analizi. V večini primerov je to seštevanje, omogočeno pa je lahko tudi štetje, izračun povprečja, najmanjša ali največje vrednosti in drugo.
- Poizvedbe so primerljivo hitre, hitrejšje pri vračanju podrobnejših podatkov, vendar nekoliko počasnejše pri visoko agregiranih podatkih in pri podatkih, kjer v okviru stolpcev ponavljanja vrednosti ni.

Produkti, zasnovani na tej tehnologiji, kot je Microsoft Power BI, so še dokaj novi in ne podpirajo dobro nekaterih funkcionalnosti povezanih z modeliranjem podatkov (na primer več relacij med tabelami, rekurzivne hierarhije) in implementiranjem poslovne logike, na primer funkcije za kompleksne izračune in navigacijo po hierarhijah in podrobno omejevanje dostopa do podatkov (Russo, 2014). Vendar zaradi intenzivnega razvoja na tem področju lahko pričakujemo, da bodo v prihodnosti te pomanjkljivosti odpravljene. Slabost je tudi odsotnost standardnega poizvedovalnega jezika, kot je MDX za večdimenzijske baze.

### **2.4.3 Sprotna analitična obdelava podatkov**

Podatke, ki smo jih pripravili za analize v kateri izmed oblik, opisanih v prehodnih poglavjih, je treba dostaviti končnemu uporabniku. To lahko storimo s klasičnimi vnaprej določenimi poročili, vendar predvsem večdimenzijske in stolpčno orientirane baze omogočajo veliko več. Zaradi njihove hitre odzivnosti, pregledne navigacije po podatkih in vnaprej določenimi izračuni lahko uporabnik poročila interaktivno sproti oblikuje in spreminja. To imenujemo sprotna analitična obdelava podatkov (angl. online analytical processing, krat. OLAP), ki je



definirana kot dejavnost interaktivne analize podatkov shranjenih v večdimenzijskem podatkovnem skladišču z namenom sprejemanja taktičnih in strateških odločitev (Rainardi, 2008, str. 380). Pri analizi si pomagamo z orodji OLAP, imenovanimi tudi OLAP odjemalci, ki se povezujejo na podatkovno skladišče. OLAP na pri odločitvah pomaga z naslednjimi operacijami (OLAP Council, 1995):

- Rezanje (angl. slice and dice): Celotno OLAP kocko razrežemo na manjše po izbranih dimenzijah.
- Vrtanje v globino (angl. drill down): Pomeni navigacijo po nivojih določene dimenzije. Na primer znesek celotne prodaje lahko razbijemo po letih, prodajo izbranega leta na mesece, mesec pa naprej na dneve.
- Vrtanje skozi (angl. drill trough): Kadar z zgornjo operacijo pridemo do najnižjega nivoja, si lahko ogledamo podrobne podatke, ki tvorijo vsoto v celici.
- Vrtanje počez (angl. drill across): na podlagi skupne dimenzije prikažemo podatke iz več table dejstev.
- Zvijanje (angl. roll-up): Pomeni seštevanje podatkov po nivojih hierarhije. Namesto seštevanja je lahko določena tudi drugačna operacija.
- Vrtenje (angl. pivot): Pomeni vrtenje kocke, tako da zamenjujemo dimenzije, prikazane na določeni osi.

Glede na to ali OLAP uporablja relacijsko ali multidimenzionalno bazo podatkov poznamo:

- **ROLAP** (angl. relational OLAP): Kot pove ime, uporablja za delovanje relacijsko bazo podatkov, v večini primerov mora biti večdimenzionalna. Agregati se navadno izračunavajo sproti ob poizvedbi, nekateri sistemi pa ustvarijo nove tabele s sumarnimi podatki. ROLAP je počasnejši ob poizvedbah, vendar so podatki takoj pripravljene za uporabo. Tak način bi izbrali, če so potrebna pogosta osveževanja podatkovnega skladišča.
- **MOLAP** (angl. multidimensional OLAP) za svoje delovanje rabi VBP. Kot smo že spoznali, so tu agregati predhodno shranjeni, zato je MOLAP hitrejši pri poizvedbah, vendar je potrebno procesiranje VBP.
- **HOLAP** (angl. hybrid OLAP) uporablja tako relacijsko kot VBP. V takem primeru so agregati, shranjeni v VBP, podrobni podatki pa v relacijski bazi podatkov, kar nam nudi kompromis med obema načinoma.

### 3 Microsoftova orodja za podatkovno skladiščenje in poslovno inteligenco

Na trgu je veliko ponudnikov sistemov poslovne inteligence, za predstavitev Microsoftovih orodij sem se odločil, ker sem jih uporabil v več implementacijah, en primer bo opisan v nadaljevanju. Se pa Microsoft redno uvršča med vodilne na trgu, kar potrjujejo periodične Gartnerjeve raziskave Magic Quadrant for Business Intelligence and Analytics Platforms. Tako je tudi po zadnji raziskavi, kjer je zaznan velik premik trga k investicijam v

samopostrežne sisteme in smo priča izpadu nekaterih velikih ponudnikov kot je Oracle (Gartner, 2016).

Microsoftova ponudba na tem področju je zelo raznolika, od brezplačnih in zato omejenih orodij za najmanjše organizacije do podpore največjim. Podprte so lokalne postavitve in postavitve v oblaku, ki poleg programske nudijo tudi najem strojne opreme in so zaradi nizkih vstopnih stroškov primerne za manjše organizacije in organizacije, ki prvič uvajajo sistem. poslovne inteligence.

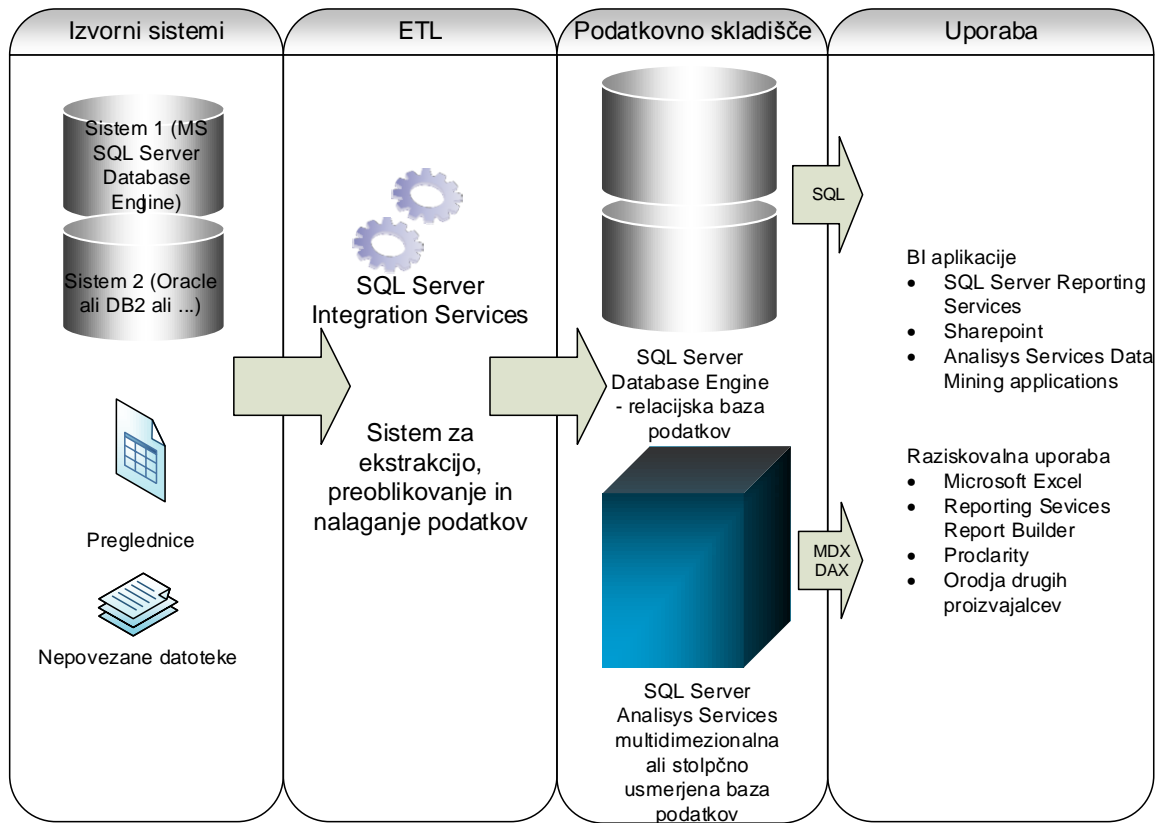
### **3.1 Tradicionalna poslovna inteligenca**

Microsoftova orodja tradicionalne poslovne inteligence so v glavnem združena v paketu Microsoftov SQL Server. Paket sestavljajo naslednje komponente

- **Sistem za upravljanje podatkovnih baz:** V paketu SQL Server je ta komponenta imenovana Database Engine. S pomočjo te komponente shranjujemo in upravljamo podatke podatkovnega skladišča. Poleg tega vsebuje še veliko dodatnih orodij, ki nam pomagajo pri podatkovnem skladiščenju: SQL Agent nam omogoča samodejno zaganjanje opravil, na primer postopka ETL.
- **SQL Server Integration Services** (krat. SSIS), s katerim izgradimo sistem za ekstrakcijo, preoblikovanje in nalaganje podatkov.
- **SQL Server Analysis Services** (krat. SSAS), ki omogoča shranjevanje v večdimenzijske in stolpčno usmerjene baze podatkov.
- **SQL Server Reporting Services** (krat. SSRS) omogočajo izdelavo klasičnih vnaprej določenih poročil. Ponuja pa tudi podporo ad-hoc poizvedbam s pomočjo orodja Report Builder.
- **SQL Server Management Studio** in **Visual Studio** pa sta orodji za razvoj in upravljanje BI sistema.

Paket SQL Server vsebuje programsko opremo potrebno za razvoj, namestitve, nalaganje in upravljanje BI sistema. Druga večja skupina Microsoftovih orodij za BI je namenjena poslovnemu uporabniku. Pomemben je predvsem Microsoft Office Excel, kot OLAP odjemalec za SSAS.

Slika 16 Arhitektura Microsoftovega sistema poslovne inteligence

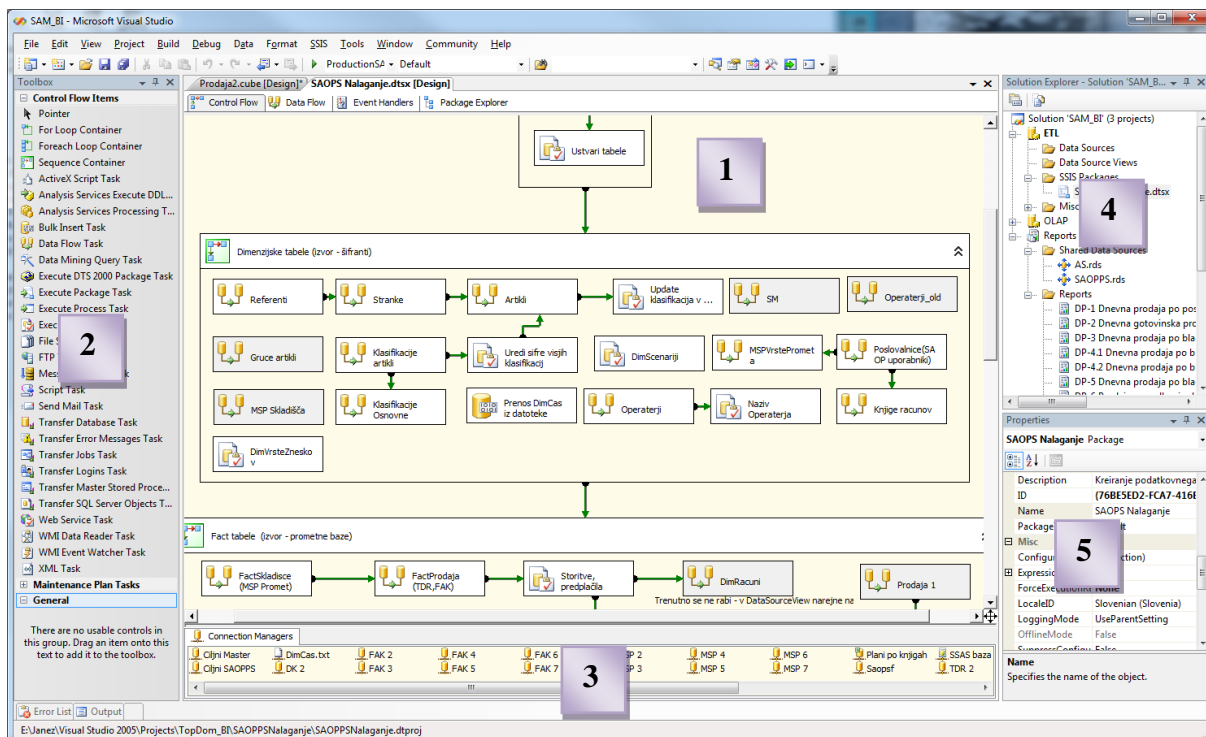


Vir: (Mundy, Thornthwaite, & Kimball, 2006, str. 123)

### 3.1.1 Orodja za razvoj in upravljanje sistema poslovne inteligence

Kot del paketa SQL Server ponuja dve orodji za delo z BI sistemi, Visual Studio in SQL Server Management Studio. Visual Studio je splošno razvojno orodje, ki je tudi namenjeno načrtovalcem in razvijalcem BI sistema. Ponuja uporabniški vmesnik za načrtovanje, razvoj in razhroščevanje baz v SSAS in modelov za rudarjenje podatkov, postopka ETL in poročil.

Slika 17 Visual Studio



Uporabniški vmesnik na grobo razdelimo na:

1. Menijsko in orodne vrstice, v kateri so meniji oziroma ikona lahko standarden ali pa se nanašajo na objekt, s katerim delamo, na primer poročilo, OLAP kocka, ETL.
2. Orodna vrstica, ki vsebuje orodja za ETL, kot so povezave na vire podatkov in transformacije in gradnike poročil kot to tabele in grafi.
3. Delovna površina, kjer gradimo in urejamo ETL pakete, OLAP kocke, poročila.
4. Raziskovalec: V DW/BI sistemu nastopa namreč veliko objektov, kot so OLAP kocke, dimenzije, poročila, viri podatkov, ETL paketi in drugo. V Visual Studiu so ti objekti organizirani v programsko rešitev (angl. solution), ki jo sestavlja več projektov. Tipično bodo nastopali vsaj trije projekti: postopek ETL, baza SSAS skupaj z morebitnimi modeli za rudarjenje podatkov in projekt, ki vsebuje poročila. Vsak projekt pa vsebuje posamezne objekte.
5. Okno, ki prikazuje lastnosti izbranega objekta. Omogočana na pregled in urejanje lastnosti, ki so odvisne od izbranega objekta.

SQL Server Management Studio (krat. SSMS) je primarno orodje za administratorje podatkovnih baz. Omogoča nam upravljanje relacijskih in baz SSAS, paketov ETL in poročil. V SSMS opravimo naloge, kot sta vzdrževanje podatkovnih baz (varnostne kopije, obnavljanje indeksov), dodeljevanje dostopa do objektov podatkovnih baz in poročil. SSMS pa uporabljamo tudi kot razvojno orodje, saj vsebuje tudi urejevalnik z razhroščevalnikom, ki ga uporabimo za pisanje poizvedb proti relacijskim in bazam SSAS. Te poizvedbe potem uporabimo za poročila ali pa v postopku ETL. SSMS nam omogoča tudi načrtovanje in

izdelavo objektov relacijske baze, kot so tabele, indeksi, primarni ključ. Te objekte SSMS lahko pretvori v skripte, ki jih lahko uporabimo v postopku ETL za ustvarjanje objektov v podatkovnem skladišču.

### 3.1.2 Orodje za integracijo podatkov

SQL Server Integration Services (krat. SSIS) je orodje, ki ima široko uporabo na področju migracij podatkov in avtomatizaciji vzdrževanja podatkovnih baz. V podatkovnem skladiščenu SSIS uporabljamo za izvedbo postopka ETL za nalaganje podatkovnega skladišča in tudi za pripravo podatkov za rudarjenje ter avtomatizacijo procesiranja večdimenzijskih baz podatkov.

Razvoj za SSIS poteka v Visual Studiu, rezultat pa so tako imenovani paketi SSIS. Za izvajanje paketov pa potrebujemo SSIS, ki teče kot storitev (angl. service) v operacijskem sistemu. Paket sestavljajo:

- Povezave na izvore podatkov in cilj, ki ga navadno predstavljajo baze podatkovnega skladišča.
- Kontrola toka (angl. control flow), ki določa zaporedje izvajanje opravil v paketu in pod kašnim pogojem se bo izvedlo naslednje opravilo.
- Opravila: SSIS ponuja vrsto opravil, ki jih potrebujemo pri izdelavi postopka ETL, kot so delo z datotekami, FTP, izvajanje skript .NET, izvajanje poizvedb SQL. Najpomembnejše opravilo pa je pretok podatkov.
- Pretok podatkov je osrednje opravilo v postopku ETL, s pomočjo katerega izvedemo skoraj celotno ekstrakcijo podatkov, celotno transformacijo in nalaganje. Paket lahko vsebuje poljubno število pretokov podatkov. Opravilo pretoka podatkov lahko razdelimo na:
  - Povezave na izvore podatkov: SSIS nam omogoča povezave na vse baze podatkov, ki podpirajo povezljivost po protokolih OLEDB ali ODBC, razne datoteke (Excel, XML, tekstovne datoteke).
  - Transformacije: Podprte so praktično vse tipične transformacije, ki potrebujemo pri postopku ETL.
  - Zapis v podatkovno skladišče.

### 3.1.3 Sistem za upravljanje baz podatkov za analize

SQL Server Analysis Services (krat. SSAS) je sistem za upravljanje večdimenzijskih in stolpčno usmerjenih baz podatkov, ki ga odlikujejo zanesljivost, nadgradljivost (angl. scalability) in varnost. Poleg vloge OLAP strežnika pa nam omogoča tudi rudarjenje podatkov z ducatom algoritmov, ki so namenjeni temu.

SSAS je v primarni vlogi OLAP strežnik in zagotavlja uporabniško orientiran sloj nad podatkovnim skladiščem in opravlja štiri osnovne funkcije (Kimball, 2005, str. 124):

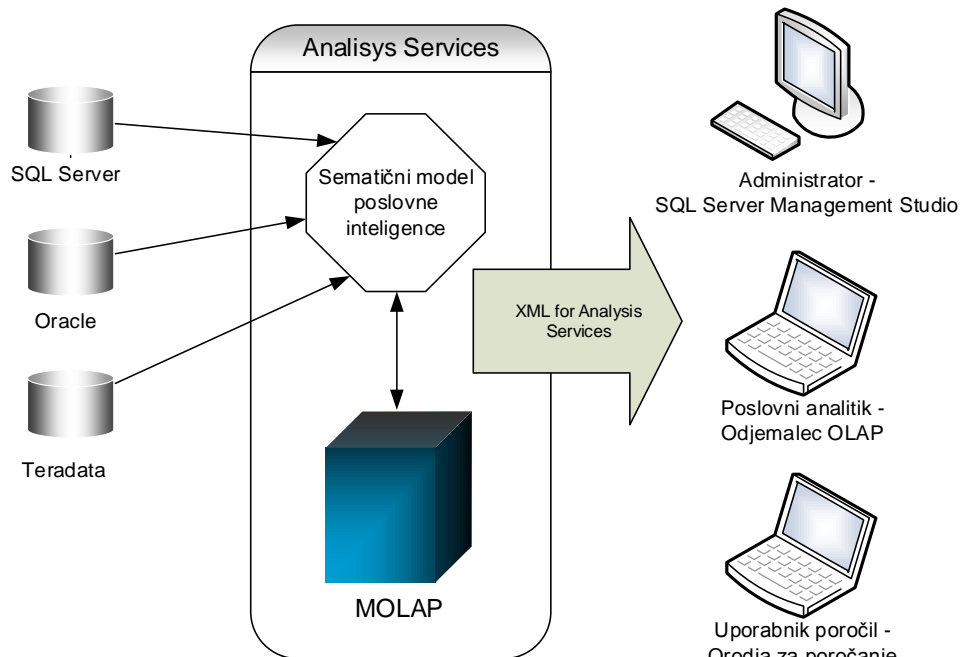
- Zagotavlja enostavno navigacijo po podatkih z uporabo uporabniku prijaznih imen za objekte v bazi podatkov in z jasnimi in dokaj enostavnimi povezavami med dimenzijskimi tabelami in tabelami dejstev.
- Omogoča centralno hranjenje izračunov nad osnovnimi podatki in njihovo enotno poimenovanje. To je mnogo bolj pregledno, kot če so izračuni raztreseni v klasičnih poročilih.
- Velika hitrost poizvedb, ki je v največji meri zagotovljena z agregacijami.
- Varnost podatkov, ki jo omogoča centralni varnostni sistem, s katerim je mogoče zelo podrobno določiti, do katerih podatkov lahko dostopa uporabnik ali skupina uporabnikov.

Osrednji koncept SSAS je tako imenovan BISM (angl. business intelligence semantic model). Nudi vmesni logični sloj nad virom podatkov, navadno podatkovnim skladiščem v večdimenzijski obliki. Možna sta dva načina delovanja: večdimenzijska ali stolpčno usmerjena pomnilniška baza podatkov imenovana tabelarni model (angl. tabular model).

Ključni elementi semantičnega modela so (Harinath, Zare, Meenakshisundaram, Carroll, & Guang-Yeu Lee, 2009, str. 20):

- **Podpora različnim virom podatkov:** pomaga združiti podatke iz različnih virov (na primer iz različnih področnih podatkovnih skladišč). Omogoča nam združevanje različnih shem v enoten logičen model, kar daje uporabnikom možnost, da poizvedujejo po enem samem podatkovnem modelu.
- **Možnost dostopa do podatkov v realnem času in visoka zmogljivost:** zagotavlja hiter dostop do podatkov skupaj z zmogljivostmi, ki jih prinaša MOLAP. Kadarkoli pride do sprememb podatkov v izvornih relacijskih bazah, se ažurira tudi baza MOLAP. V času ažuriranja MOLAP se podatki za tekoče poizvedbe pridobivajo iz relacijskih baz.
- **Enostavna navigacija in raziskovanje podatkov:** ponuja enoten pogled na različne vire podatkov. To pa je še nadgrajeno z bogatimi metapodatki, ki jih ponuja OLAP, kot so določanje hierarhij v dimenzijah. K enostavnosti pripomore tudi možnost ustvarjanja pogledov na določene dele OLAP kocke. To precej olajša delo uporabniku, ki ga zanimajo samo določeni podatki iz kompleksne OLAP kocke.
- **Bogata podpora analizam:** zagotavlja podporo kompleksnim izračunom iz izvornih podatkov. Za definicijo izračunov uporablja MDX, ki je standardni jezik za poizvedovanje po večdimenzijskih bazah. Za tabelarni model je na voljo namenski jezik DAX, ki je po funkcionalnostih primerljiv z MDX.

Slika 18 Arhitektura SQL Server Analysis Services



Vir: (Harinath, Zare, Meenakshisundaram, Carroll, & Guang-Yeu Lee, 2009, str. 19)

Na sliki zgoraj vidimo, da lahko v SSAS združimo podatke iz različnih relacijskih baz, vendar bo v veliki večini primerov izvor le baza podatkovnega skladišča. Združevanje različnih virov je lahko zelo uporabno, kadar moramo hitro izdelati prototip rešitve in preskočimo ETL. V načinu MOLAP ali HOLAP se bodo podatki dodatno shranjevali še v večdimenzijsko bazo podatkov. Vsi odjemalci pa komunicirajo po že omenjenem standardu XMLA (krat. XML for analysis), kar odpira vrata uporabi ne-Microsoftovim orodjem.

Kljub možnostim semantični model, da združuje različne podatkovne vire pa bo v veliki večini primerov edini podatkovni vir relacijska baza v večdimenzijski obliki. Semantični model ima namreč tudi svoje omejitve (Larson, 2009, str. 112):

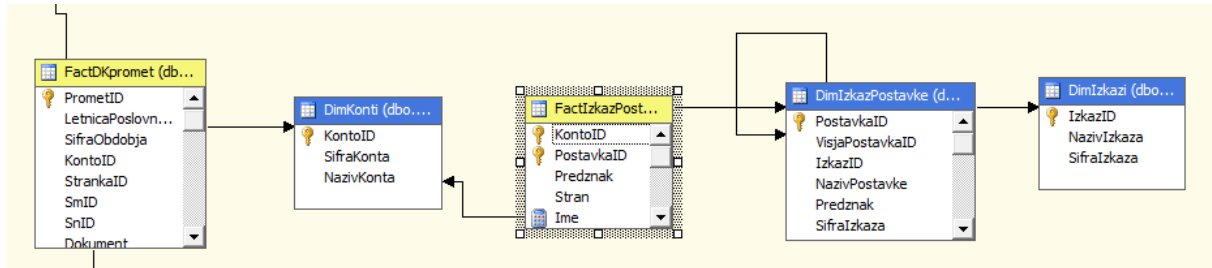
- Kot smo že omenili, so viri podatkov lahko tudi v raznih datotekah. Tudi teh ni mogoče neposredno uporabiti v SSAS.
- Zahteva tudi stalno fizično povezavo na podatkovni vir. To v praksi velikokrat ni mogoče, saj so lahko podatki na geografsko različnih lokacijah.

Vsi zgoraj naštetih razlogi zahtevajo uporabo postopka ETL in podatkovnega skladišča, ki je potem vir za SSAS.

SSAS podpira tudi zahtevnejše podatkovne modele. Zvezdna shema in shema snežinke sta najbolj pogosti obliki, vendar nekaterih poslovnih problemov tako ne moremo pravilno predstaviti. Primer je lahko analiza poslovnih izkazov. Vsak poslovni izkaz sestavlja več postavk izkaza, te postavke pa so sestavljene iz kontov. Konti pa so povezani s prometnimi podatki. Z običajnim večdimenzijskim modelom si ne moremo pomagati, saj različne izkaze in postavke izkaza sestavlja več kontov, konti pa nastopajo v več postavkah izkaza. Rešitev, ki jo ponuja SSAS, je mnogo proti mnogo relacija med dimenzijo in tabelo dejstev. (Russo,

The many-to-many revolution, 2011). V ta namen vpeljemo posebno tabelo dejstev, ki definira mnogo proti mnogo relacijo. V našem primeru med dimenzijo Konti in dimenzijo Postavke izkaza.

Slika 19 Posebna tabela dejstev nam omogoča mnogo proti mnogo relacije



Zaradi naštetih lastnosti je uporaba SSAS v podatkovnem skladiščanju zelo pogosta. Vseeno pa obstaja nekaj argumentov proti uporabi SSAS (Mundy, Thornthwaite, & Kimball, 2006, str. 126):

- **Podvojevanje podatkov:** Veliko uporabnikom ni všeč dejstvo, da se podatki iz relacijskega podatkovnega skladišča podvojijo še v večdimenzijski bazi. Vendar pa se podvojevanju podatkov lahko izognemo z uporabo ROLAP – a. Pa tudi če se odločimo še za večdimenzijsko bazo, je izgradnje le – te popolnoma samodejna in nanjo lahko gledamo kot na indekse pri relacijskih bazah.
- **Sprememba uporabniških aplikacij za poročanje:** Organizacije, ki že uporabljajo DW/BI sisteme, osnovane na relacijskih bazah, morajo ob uvedbi SSAS zamenjati uporabniške aplikacije za poročanje. Nakup novih orodij in izobraževanje uporabnikov lahko predstavlja velik strošek.

### 3.1.4 Sistem za upravljanje relacijskih baz podatkov

SQL Server Database Engine, najpogosteje imenovan kar SQL Server je sistem za upravljanje relacijskih baz podatkov in je osrednja komponenta paketa Microsoft SQL Server. Database Engine uporabljamo za upravljanje baz za namene OLTP ali OLAP. Tu vključuje izdelavo tabel za hranjenje podatkov in ostalih objektov, kot so indeksi, shranjene procedure in ostalo. V DW/BI sistemu Database Engine uporabimo za upravljanje relacijskih baz podatkovnega skladišča, pa naj bodo v normalizirani ali večdimenzijski obliki.

Kot je že bilo omenjeno, se za veliko večino DW/BI sistemov priporoča uporabo tako relacijske, kot večdimenzijske baze podatkov. Argumenti za uporabo večdimenzijske baze so pojasnjeni v prejšnjem poglavju. Zakaj pa uporabiti še relacijsko bazo? Načeloma SSAS omogoča polnjenje OLAP kock iz heterogenih baz podatkov, tudi če te niso v večdimenzijski obliki. V praksi pa se pokaže, da je tako način izvedljiv le v izjemnih primerih. Za večino DW/BI sistemov se priporoča uporaba relacijske baze zaradi naslednjih razlogov (Mundy, Thornthwaite, & Kimball, 2006, str. 127):



- **Uporabo narekuje postopek ETL:** V poglavju o postopku ETL so bile naštetе vse njegove prednosti, kot so čiščenje in urejanje podatkov. Rezultate postopka ETL pa je treba shraniti v relacijsko bazo.
- **Večdimenzijski model:** Kljub temu da SSAS podpira heterogene podatkovne vire, iz katerih bi bilo mogoče s samimi poizvedbami izvleči podatke v večdimenzijsko obliko, je to praksi neizvedljivo. Da ustvarimo dober večdimenzijski model, je potreben dokaj zapleten postopek ETL, katerega rezultate je treba shraniti v relacijsko bazo.
- **Hitrost poizvedb:** Poizvedbe, ki vračajo veliko vrstic, so hitrejše v relacijski bazi. Večdimenzijska baza se odlično obnaša pri vračanju visoko summariziranih podatkov in vračanju relativno majhnega števila vrstic, kar je sicer najpogostejša uporaba pri analizah podatkov. Kadar pa mora poizvedba vrniti več deset tisoč vrstic, je relacijska baza mnogo hitrejša.
- **Večja prilagodljivost:** Sistemi za upravljanje relacijskih baz so med seboj relativno dobro združljivo. Če bi se odločili za drugega ponudnika, bi bilo podatke dokaj lahko migrirati. V primeru, da uporabljamo samo večdimenzijsko bazo, je to težje.

### 3.1.5 Orodja za podporo poročanju

Sodobni DW/BI sistemi ponujajo veliko možnosti za analizo podatkov, kot je OLAP in rudarjenje podatkov. Vendar večina uporabnikov še vedno dostopa do podatkov skozi klasična poročila, zato je sistem za ustvarjanje in dostavo poročil bistven del DW/BI sistema. Za takšna poročila uporabimo SQL Server Reporting Services (krat. SSRS), ki je strežniško osnovan sistem za ustvarjanje, upravljanje in dostavo poročil. Še posebej je uporaben v organizacijah z velikim številom uporabnikov poslovne inteligence, saj poskrbi za enotno infrastrukturo poročanja, hkrati pa ponuja dostop do poročil širokemu krogu uporabnikov, saj ni omejitve na določeno platformo. Za osnovno uporabo popolnoma zadostuje vsaka naprava, ki lahko prikazuje standardne spletne strani oziroma prejema elektronsko pošto. Kljub temu pa omogoča več kot le statična poročila. Interaktivnost dosežemo z uporabo različnih gradnikov poročil, ki omogočajo vrtanje v globino in s parametri poročil.

SSRS izpolnjuje vse bistvene zahteve glede poročanja:

- **Ustvarjanje in popraviljanje poročil:** Na voljo sta dve orodji. Visual Studio je namenjen razvijalcem in ponuja veliko možnosti za izdelavo poročil. Naprednim poslovnim uporabnikom, ki hočejo ustvarjati svoja ad hoc poročila, je namenjeno orodje Report Builder. Ta ponuja poenostavljeni podatkovni model, tako da avtorju poročila ni treba podrobno poznati baze podatkov in pisanja poizvedb. Tudi sam uporabniški vmesnik je zelo podoben tistemu in Microsoft Excela, ki ga veliko poslovnih uporabnikov dobro pozna. Zaradi enostavnosti uporabe seveda ne podpira vseh možnosti, ki jih ponuja SSRS. Tako ustvarjena poročila damo uporabnikom na voljo z namestitvijo na strežnik.
- **Iskanje in pregledovanje poročil:** Pregledovanje poročil je možno na več načinov.

- Najpogosteje se do poročil dostopa preko spletnega portala, kjer so poročila organizirana v mape in jih je mogoče tudi iskati po imenu. Varnostni sistem poskrbi za dodeljevanje dostopa do poročil ali map s poročili posameznim uporabnikom. Poročila so vedno prikazana v HTML obliki, vendar jih je mogoče izvoziti tudi v druge oblike.
- Potiskanje poročil k uporabnikom je možno s tako imenovanimi naročninami. Z naročnino na poročilo določimo, da se poročilo samodejno pošilja po elektronski pošti ali shranjuje na datotečni sistem. Pošiljanje poročila lahko sprožimo po urniku ali pa, ko so na voljo zahtevani podatki.
- SSRS je implementiran kot spletna storitev (angl. web service), zato lahko poročila uporabimo tudi v drugih aplikacijah. Lahko so to spletne aplikacije, kot so različni portali ali pa običajne namizne aplikacije. To so lahko standardne aplikacije, kot je Microsoftov spletni portal SharePoint ali pa lastne aplikacije.
- **Upravljanje poročil:** Za naloge, kot je nameščanje novih poročil na strežnik, organiziranje poročil v mape, dodeljevanje dostopa in naročnin, lahko uporabimo spletni portal SSRS ali pa orodje SQL Server Management Studio. Omogoča centralno upravljanje in nadzor nad celotno infrastrukturo poročanja v organizaciji.

### **3.2 Orodje za samopostrežno poslovno inteligenco - Power BI**

Power BI je aktualna Microsoftova ponudba v samopostrežni poslovni inteligenci. Je kombinacija orodij in storitev (krat. SaaS, angl. software as a service), ki pokrivajo večino funkcionalnosti tradicionalnih sistemov poslovne inteligence. Hkrati pa so orodja mnogo preprostejša za uporabo in zato primerna tudi za poslovne uporabnike. Začetki segajo v dodatek za Microsoft Office Excel imenovan Power Pivot.

Power BI Desktop, je predvsem razvojno orodje, vendar ga lahko uporabimo tudi za same analize v enouporabniškem okolju. Vsebuje:

- Orodja za ETL: Omogoča pridobivanje in združevanje podatkov iz različnih virov, lokalnih in v oblaku. Nabor je velik in se redno dopolnjuje. Še posebej so zanimivi viri podatkov iz storitev, kot so ERP, CRM in podobni poslovni sistemi (na primer Salesforce, Dynamics CRM). Lahko so že v obliki primerni za analize in ne potrebujejo dodatnih transformacij. Podprte so tudi najbolj pogoste transformacije podatkov, vendar manj kot v SSIS.
- Vse delo poteka preko grafičnega vmesnika in je popolnoma interaktivno. Ob izbiri podatkovnega vira lahko takoj pregledujemo podatke, prav tako po vsaki transformaciji.
- Vključeno je orodje za modeliranje podatkov, s katerim upravljamo relacije med tabelami, dodajamo novo izračunane stolpce ali mere v tabele.

- Shramba podatkov je stolpčno usmerjena pomnilniška baza. Vključuje tudi vse metapodatke o virih, transformacijah, podatkovnem modelu in definicij poročil. Omogočeno je tudi neposredno poizvedovanje po nekaterih podatkovnih virih.
- Izdelava poročil je močnejša točka Power BI: Na voljo je več deset grafičnih ponazoritev podatkov, število pa se redno povečuje. Možen je tudi uvoz ponazoritev po meri, obstaja celo zbirka ponazoritev, ki jo ustvarja skupnost uporabnikov. Izdelava poročil je enostavna, podobna delu z OLAP orodji: izbira dejstev, dimenzij, filtriranje in razvrščanje podatkov in izbira zelene ponazoritve. Med posameznimi poročili lahko ustvarimo interakcije in jih vizualno združimo v celoto – nadzorno ploščo.
- Objava celotne rešitve, to je podatkov in metapodatkov o podatkovnih virih, transformacijah, podatkovnem modelu, definicij poročil, je enostavna s storitvijo Power BI.

Power BI Service je storitev narejena na osnovi Azure, Microsoftove infrastrukture in platforme za računalništvo v oblaku. Ponuja:

- Računalniške zmogljivosti za v obliki diskovnega prostora, pomnilnika, procesorskih zmogljivosti in pretoka podatkov.
- Dostop preko GUI, ki je skoraj enak Desktop različici. Omogoča nam pregled nad nadzornimi ploščami, posameznimi poročili in podatkovnimi viri. Omogoča tudi povezavo na nove podatkovne vire in izdelavo poročil. Bistvena je možnost delitve podatkov, poročil in nadzornih plošč, navadno znotraj organizacije. Nabore podatkov, poročila in nadzorne plošče lahko združimo v tako imenovane pakete vsebine, in jih kot celoto ponudimo drugim uporabnikom. Ta možnost je še posebej zanimiva za ponudnike aplikacij, ki lahko v taki obliki ponudijo analize svojih podatkov s storitvijo Power BI.
- Omogočeno je samodejno periodično osveževanje iz podatkovnih virov.
- Na voljo je tudi programski vmesnik, s katerimi lahko ustvarimo nove nabore podatkov ali v obstoječe nabore dodajamo nove podatke iz svoje aplikacije. Omogoča pa tudi integracijo poročil v lastno aplikacijo.

## 4 Uvedba sistema poslovne inteligence

### 4.1 Stanje pred uvedbo poslovne inteligence

Podjetje SAM d.o.o. se ukvarja z veleprodajo in maloprodajo izdelkov za gradnjo in dom. Ponudba obsega področje kritin, gradbenega materiala, fasadnih sistemov in izolacijskega materiala. S svojo ponudbo zadovoljuje potrebe graditeljev, obrtnikov in investitorjev objektov ter individualnih kupcev. Za to skrbi mreža 6 prodajnih mest (SAM d.o.o.).

Pred uvedbo sistema poslovne inteligence je bilo stanje ne SAM naslednje:

- Sistem ERP je razdeljen na več modulov: knjigovodstvo in finance, maloprodaja, veleprodaja, materialno in skladiščno poslovanje ter naročanje dobaviteljem. Vsak

modul uporablja za shranjevanje svojo bazo podatkov. ERP nima dovolj uporabnih poročil, ki bi združevala podatke iz več baz.

- Podatki se nahajajo na šestih poslovnih enotah, razen baze knjigovodstva, ki je samo na upravi. Poročanje o stanju na poslovnih enotah povzroča težave, saj so potrebne dolgotrajne ročne priprave poročil. Rešitev te težave je prišla na vrsto najprej.
- Količina podatkov predvsem iz maloprodaje (POS blagajne) in materialnega poslovanja hitro narašča. Ob analizi stanja velikost vseh OLTP baz skupaj že preseže 15 GB (v začetku 2010 pa 50 GB). Izdelava poročil iz ERP sistema postane počasna in začne resno ovirati operativno delo, predvsem prodajo na POS blagajnah. Pojavi se nezadovoljstvo s hitrostjo delovanja ERP sistema.
- Vodstvo podjetja in poslovnih enot ni zadovoljno z obstoječimi poročili iz ERP sistema. Poročila ne vsebujejo vseh potrebnih informacij, primanjkuje jim predvsem interaktivnosti. Zanimajo jih na primer različni pogledi na podatke prodaje: prodaja po letih/mesecih, po poslovnih enotah, skupinah artiklov/artiklih, po prodajnih referentih, po dobaviteljih, primerjave med leti in podobno.

## **4.2 Razvoj sistema poslovne inteligence**

Pri razvoju smo se oprli na Kimballovo metodologijo, zaradi izbire Microsoftovih orodij predvsem na delo *The Microsoft Data Warehouse Toolkit With SQL Server 2005 and the Microsoft Business Intelligence Toolset*. Razvoj je potekal v naslednjih korakih:

- Zbiranje zahtev
- Analiza zahtev
- Izdelava BUS matrike
- Določitev dimenzij in dejstev
- Analiza virov in podatkov
- Razvoj in polnjenje podatkovnega skladišča
- Izdelava OLAP
- Izdelava poročil
- Namestitev
- Izobraževanje uporabnikov in podpora

### **4.2.1 Zbiranje in analiza zahtev**

Zahteve, ki so narekovale uvedbo BI sistema, so v veliki meri izhajale iz težav z obstoječim sistemom. S stanjem smo bili seznanjeni že pred začetkom projekta. To so že zgoraj omenjene težave z razdrobljenostjo podatkov, pomanjkanjem primernih poročil in počasnostjo sistema. Skozi intervjuje smo pridobili zahteve po procesih, ki morajo biti podprti z analizami in objekti, ki morajo nastopati v analizah. Na podlagi tega je bila izdelana spodnja BUS matrika.

Tabela 2 BUS matrika podatkovnega skladišča

Dimenzije Procesi	Stranke	Artikli	Čas	Poslovne enote	Klasifikacije artiklov	Knjige računov	Operaterji	Vrste prometa	Računi	Dokumenti	Scenariji	Vrste zneskov
Prodaja	X	X	X	X	X	X	X		X	X		
Skladišče	X	X	X	X	X	X	X	X	X			
Planiranje	X		X	X	X						X	
Terjatve	X		X									X

BUS matrika podatkovnega skladišča je tabela, katere stolpci predstavljajo poslovne procese, stolpci pa objekte, ki nastopajo v teh procesih. V večdimenzijskem modelu so ti objekti dimenzije, vsak poslovni proces pa je predstavljen z merami. Vsaka dimenzija nastopa v enem ali več procesih, kar je v BUS matriki označeno z X. Tako je zelo nazorno prikazano, katere dimenzije je treba implementirati za analizo posameznega poslovnega procesa (Mundy, Thornthwaite, & Kimball, 2006, str. 53). V BUS matriki lahko tudi prikažemo poslovne procese v določenem pomenljivem zaporedju, v našem primeru glede na prioriteto implementacije – proces prodaje torej mora biti podprt prvi.

## 4.2.2 Razvoj rešitve

### 4.2.2.1 Fizični načrt

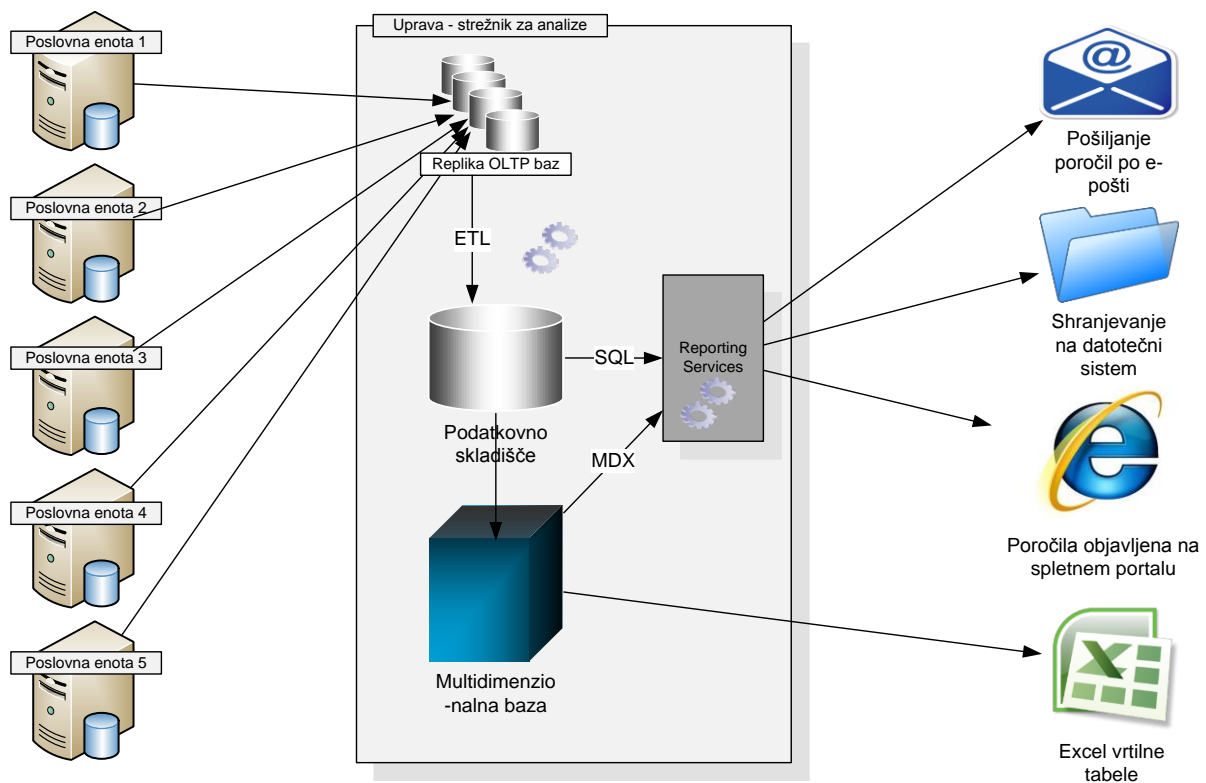
Ena izmed večjih odločitev je kakšno fizično namestitev DW/BI sistema narediti. Ali naj bo kar na strežniku, kjer že gostuje ERP sistem, ali uporabiti namenski strežnika za DW/BI ali celo več strežnikov? SQL Server namreč omogoča vse zgoraj opisane namestitve, ki jih uporabimo glede na predvidene količine podatkov, zahtevnost uporabe in število sočasnih uporabnikov.

Pri ocenjevanju količine podatkov si pomagamo s skupnim številom vrstic v tabelah dejstev, dimenzijske tabele večinoma lahko zanemarimo, saj navadno ne prispevajo bistveno k velikosti. Število vrstic pomnožimo z ocenjeno velikostjo ene vrstice. Začetno stanje je bilo ocenjeno na 24 milijonov vrstic, ob približni velikosti vrstice 200 B je začetna velikost podatkovnega skladišča približno 5 GB, vsako leto pa bi narastlo za 1 - 1,5 GB. Zahtevnost uporabe ocenimo glede na predvidene vzorce uporabe DW/BI sistema. V našem primeru, je bilo predvideno, da bo večina uporabe padla na pregledovanje OLAP kocke z Microsoft Excelom, kar se šteje za srednje zahtevno uporabo. Seveda ob pogoju preudarne rabe brskanja po OLAP kockah, večdimenzijske baze namreč niso primerne pri vračanju zelo velikega števila vrstic. Uporabljala se bodo tudi klasična poročila iz relacijske in večdimenzijske baze, kar štejemo pod manj zahtevno uporabo. Večina poročil se bo izvedla v nočnem času, kar pomeni še manjše breme. Do zelo zahtevne rabe, kot so ad hoc poizvedbe in rudarjenja podatkov pa predvidoma ne bo prišlo.

Ocenjeno je bilo tudi povprečno število sočasnih uporabnikov DW/Bi sistema, približno 20. Za osveževanje podatkov v DW/BI sistemu pa je na voljo približno 5 ur dnevno. Vse to uvršča sistem med manjše: manj kot 500 milijonov vrstic v tabelah dejstev, večinoma srednje in manj zahtevna uporaba, približno ducat sočasnih uporabnikov ter več ur dnevno za vzdrževanje sistema (Mundy, Thornthwaite, & Kimball, 2006, str. 138). Za tak sistem se priporoča en strežnik, na katerem bodo nameščene vse strežniške komponente Microsoftovega sistema poslovne inteligence SQL Server:

- Database Engine, kjer bodo gostovale replike OLTP baz, podatkovno skladišče in za izvajanje replikacij podatkov.
- Integration Services za izvedbo ETL postopka
- Analysis Services, kjer bo gostovala večdimenzijska baza.
- Reporting Services za pregledovanje in distribucijo klasičnih poročil.
- 

Slika 20 Fizični načrt rešitve



#### 4.2.2.2 Razvoj postopka ETL

V prvem koraku je bilo treba poskrbeti za dnevni prenos podatkov iz poslovalnic na upravo podjetja. V ta namen smo uporabili tehnologijo replikacij baz podatkov, ki je vgrajena v SQL Server. Replikacije baz podatkov nam omogočajo kopiranje in distribuiranje podatkov in objektov baze podatkov v eno ali več baz podatkov in za ohranjanje baz usklajenih. Sinhronizacija podatkov je popolnoma samodejna in se navadno izvaja po vnaprej določenem

urniku. S tem smo zagotovili, da imamo vsak dan pred začetkom ETL postopka podatke vseh poslovalnic na centralnem strežniku na upravi.

Odločili smo se za vsakokratni prenos vseh podatkov iz replik OLTP baz v podatkovno skladišče. Razlogi je relativna enostavnost ETL postopka v primerjavi z drugimi načini (opisani v poglavju ETL). Seveda morajo razmere in zahteve to dovoljevati:

- Zahtevana frekvenca osveževanje podatkov je enkrat dnevno.
- Na voljo je dovolj veliko časovno okno za izvedbo ETL.
- Količina podatkov ni tako velika, stopnja rasti podatkov pa naj se predvidoma ne bi bistveno spreminjala.
- Ni bilo zahtev po sledenju spremembam v podatkih dimenzij skozi čas in smo se lahko izognili obravnavanju problema počasi spreminjajočih se dimenzij.

Nalaganje podatkov v podatkovno skladišče pa je bilo treba optimizirati za čim hitrejšo delovanje. Nalaganje podatkov v prazne tabele je že samo po sebi hitrejšo. Še boljše, da so tudi brez indeksov, primarnih in tujih ključev. Te ustvarimo na koncu ETL postopka.

ETL postopek je bil izdelan v obliki enega SSIS paketa. Ta je organiziran na več delov:

- Ustvarjanje baze podatkovnega skladišča in tabel.
- Nalaganje podatkov v dimenzijske tabele: Vir podatkov za dimenzije so različni šifrant - strank, artiklov, klasifikacij artiklov, referentov. Ker so bili šifrant že v ERP sistemu enotni, je preoblikovanje predstavljalo pretežno nadomeščanje primarnih ključev.
- Nalaganje podatkov v tabele dejstev je predstavljalo predvsem združevanje podatkov iz vseh poslovalnic. Podatke o prodaji je bilo nadaljnje združiti še iz podatkov POS blagajn in veleprodaje.
- Ustvarjanje primarnih, tujih ključev in indeksov na dimenzijskih in tabelah dejstev.

### **4.2.2.3 Večdimenzijska baza**

Poleg podatkovnega skladišča v relacijski bazi podatkov je bila uvedena še večdimenzijska baza za namene OLAP in kot vir podatkov za nekatera klasična poročila. Edini vir podatkov je podatkovno skladišče, ki ima že določene relacije med tabelami, te pa se ohranijo tudi v večdimenzijski bazi. Izdelana je bila OLAP kocka za vse tabele dejstev in dimenzij:

- Za vsako tabelo dejstev je bila ustvarjena svoja skupina mer (Prodaja, Zaloge, Planiranje, Terjatve) z nekaterimi merami neposredno iz podatkovnega skladišča (prodana količina, neto prodana vrednost) in dodatnimi merami izračunanimi v večdimenzijski bazi, kot so: razlika v ceni, indeksi vrednosti prodaje v primerjavi s predhodnim obdobjem, primerjave planske in uresničene vrednosti in podobno. Določene so bile tudi akcije za vrtnanje do podrobnosti, na primer Podrobnosti prodaje, ki omogoča vpogled v vrstice računa ob OLAP analizah.

- Za vsako dimenzijsko tabelo ali skupino tabel je bila izdelana dimenzija z več atributi in hierarhijami. Na primer dimenzija Artikli po hierarhijah Klasifikacija, Proizvajalci, Referenčni dobavitelji.
- Treba je bilo določiti relacije med dimenzijami in skupinami mer. Te sicer izhajajo iz relacij med tabelami dimenzij in dejstev in so v primeru zvezdne sheme enostavne. V primeru snežinkaste sheme moramo določiti še referenčne dimenzije, na primer dimenzija Klasifikacija artiklov uporablja referenčno dimenzijo Artikli za povezavo do skupine dejstev Prodaje. Možne so tudi že omenjene mnogo proti mnogo relacije.
- Procesiranje večdimenzijske baze je bilo izvedeno neposredno po končanem postopku ETL. Tako pripravljena večdimenzijska baza je bila pripravljena na analize z OLAP odjemalci.

#### **4.2.2.4 Poročila**

Zahteve po klasičnih poročilih so bile uresničene s SSRS. Večinoma so bila to prodajna poročila s sumarnimi podatki in zato je bil kot vir uporabljena večdimenzijska baza. Za podrobnejša poročila je bil kot uporabljeno podatkovno skladišče. Poročila so bila objavljena na intranetnem portalu, za večino pa je bilo omogočeno tudi samodejno periodično pošiljanje na elektronske naslove.

Kljub temu da obstaja orodje za izdelavo poročil, namenjeno poslovnim uporabnikom (Report Builder), je to še vedno razmeroma zapleteno. Ti so omejeni na uporabo Excel vrtilnih tabel za raziskovanje OLAP kocke in izdelavo enostavnih poročil. Za zahtevnejša poročila, ki zahtevajo interaktivnost, nadzorne plošče in objavo teh so potrebni strokovnjaki s področja IT, kar pa nujno pomeni tudi porabo časa za trivialna opravila glede oblikovanja poročil. To vrzel dobro rešujejo samopostrežna orodja, saj s svojo enostavnostjo omogočajo poslovnemu uporabniku tudi pripravo interaktivnih poročil in nazornih plošč ter enostavno objavo.

### **4.3 Možnosti uporabe samopostrežne poslovne inteligence**

Orodja in pristopi tradicionalne poslovne inteligence so brez težav omogočili izvedbo. Zaradi razmeroma majhne zahtevnosti in velikosti obstoječega sistema se postavlja vprašanje, ali bi lahko uporabili orodja samopostrežne poslovne inteligence in prihranili na račun cenejših orodij, opreme in morebiti hitrejšega razvoja rešitve. Analizirali bomo tudi prednosti in slabosti, ki ji prinaša pri razvoju, postavitvi in končni uporabi rešitve.

Če bi sistem poslovne inteligence implementirali izključno s Power BI, bi se ta precej razlikoval od obstoječega. Podatkovnega skladišča ob obliki relacijske baze ne bi bilo več. ETL postopek bi se odvil v storitvi Power BI, rezultati tega pa bi se prenesli neposredno v podatkovni model v Power BI. Z vidika uporabnika bi delo z vrtilnimi tabelami v Excelu ostalo enako, vendar bi namesto tega pričakovali bolj intenzivno rabo poročanja v storitvi Power BI, kjer bi uporabniki raziskovali podatke, izdelovali svoja poročila in nadzorne plošče



ter jih delili s sodelavci. Zato bi se lahko izognili izdelavi vnaprej določenih poročil v večini primerov.

#### **4.3.1 Stroški opreme**

PowerBI je v osnovni izvedbi celo brezplačen, vendar ima preveč omejitev in zato v našem primeru neprimeren. Izpostavim samo odsotnost samodejnega periodičnega osveževanja podatkov iz lokalnih (angl. on-premise) podatkovnih virov. Plačljiva različica ima manj omejitev in je trenutno na voljo za 10 USD mesečne naročnine na uporabnika. Samo za primerjavo: najem virtualnega strežnika z MS SQL Server Standard Edition v storitvi Microsoft Azure se začne pri 300 USD na mesec, prišteti pa bi morali še MS Excel kot odjemalec za OLAP. Tudi v našem primeru je bil uporabljen SQL Server Standard Edition, toda na lokalnem strežniku. Vendar je zelo pomembno dejstvo, da je takšna oprema potrebna že zaradi gostitve relacijskih baz ERP in druge programske opreme v primeru približno več kot petih sočasnih uporabnikov. Za manjše sisteme je na voljo brezplačna omejena različica SQL Server Express Edition, ki ne nudi podpore poslovni inteligenci. V takem primeru bi bila odločitev za Power BI lahka. V prid je tudi sistem najema, saj smo vedno deležni novih različic, v primeru nakupa je treba tudi nadgradnje na nove različice kupiti. Še posebej pomembno zaradi novosti in trendov pri predstavitvah podatkov, ker ravno to končni uporabnik vidi in želi.

Razvojno okolje je v obeh primerih brezplačno. S to razliko, da lahko Power BI Desktop uporabimo v produkciji, saj z njim lahko tudi pregledujemo poročila, vendar je fizično omejeno na enega uporabnika – tudi pri razvoju. Uporaba SQL Server Developer Edition pa za produkcijske namene ni dovoljena. Omogoča pa sočasno delo več razvijalcev na projektu in vključitev v sistem nadzora različic (angl. source control).

#### **4.3.2 Postopek ETL**

Postopek ETL v Power BI Desktop v celoti podprt z grafičnim uporabniškim vmesnikom, postopek načeloma lahko opravimo brez pisanja programske kode. V pomoč je tudi interaktivnost, ob izbiranju podatkovnega vira takoj dobimo predogled podatkov, tudi vsaka uporabljena transformacija takoj prikaže rezultate na omejenem naboru podatkov. Vsi koraki od izbire vira in vse transformacije se beležijo v dnevnik, ki omogoča pregled, urejanje in brisanje korakov. To omogoča, da ETL opravimo postopoma tudi s poizkusi in napakami, saj se lahko hitro vrnemo korak nazaj. To je prednost pred orodjem SSIS, kjer lahko preizkus delovanje opravimo šele, ko je narejen celoten tok podatkov od vira preko transformacij do cilja. Navadno to pomeni, da moramo pridobiti omejen nabor izvornih podatkov in določiti cilj, na primer tabelo v podatkovnem skladišču, kar seveda zahteva, da najprej izdelamo shemo podatkovnega skladišča. Potem lahko tudi postopoma vrivamo transformacije in na različne načine pregledujemo rezultate transformacij, vendar je to bolj zamudno kot v Power BI.

Delo z GUI je seveda lažje, vendar lahko postane počasno in duhamorno pri ponavljajočih se opravilih. V našem primeru je tabela dejstev Prodaje sestavljena iz podatkov POS blagajn in veleprodaje, po dve tabeli za vsako (glava računa, vrstice računa), vse pa se ponovi za vse poslovalnice, kar pomeni precej klikov z miško. ETL za vsako poslovalnico predstavlja:

- Določanje povezave na podatkovni vir
- Izbira tabel
- Izbira stolpcev
- Določanje podatkovnih tipov
- Določanje imen nekaterih stolpcev in tabel
- Združevanje stolpcev tabel glava računa, vrstice računa
- Izločanje stolpcev
- Končno združevanje vrstic vseh tabel

SSIS za avtomatizacijo ponuja veliko možnosti in tudi v Power BI jih najdemo nekaj. Če je vir podatkov relacijska baza, lahko namesto izbire tabel napišemo poizvedbo v SQL, kjer lahko opravimo določene »transformacije«, na primer združevanje stolpcev in vrstic tabel, izbiro stolpcev in podobno. Takšno poizvedbo lahko potem uporabimo na vseh virih podatkov z enako strukturo – v našem primeru na vsaki bazi podatkov posamezne poslovalnice. Druga možnost je napredni urejevalnik korakov poizvedb. Vsi koraki od določitve povezave na podatkovni vir do vseh transformacij so tu prikazani v obliki programske kode v posebnem poizvedovalnem jeziku imenovanem Power Query Formula Language. Napredni urejevalnik lahko uporabimo za urejanje obstoječih korakov transformacij ali pa začnemo z novo poizvedbo. Na primer kodo obstoječe transformacije lahko kopiramo in uporabimo na novih virih z enako strukturo podatkov. V povezavi z naprednim urejevalnikom lahko uporabimo tudi parametre. Na primer v razvojnem okolju smo izdelali rešitev, ki jo želimo namestiti na različna produkcijska okolja. Ker se imena strežnikov med okolji razlikujejo, bi to zahtevalo popraviljanje povezav na podatkovne vire. Lahko pa določimo parameter Ime strežnika, in ga v vsakem okolju določimo samo enkrat.

Preglednost nad celotnim postopkom ETL in posameznimi toki podatkov je v SSIS veliko boljša. Zapleteni postopki ETL, ki vsebujejo veliko virov in transformacij, lahko postanejo v Power BI težko obvladljivi. SSIS ima še veliko več funkcionalnosti, ki omogočajo zapletenejše postopke ETL.

### **4.3.3 Oblikovanje podatkovnega modela**

Orodje za oblikovanje podatkovnega modela v Power BI je podobno tistemu v SSAS v tabelarnem načinu, vendar različno od SSAS v večdimenzijskem načinu, ki je bil uporabljen v našem primeru. Orodje omogoča pregledovanje podatkov, pridobljenih iz postopka ETL, urejanje relacij, določanje in formatiranje podatkovnih tipov in dodajanje novih tabel, stolpcev in mer, izvedenih iz obstoječih podatkov.

Podatkovni model se ustvari samodejno s samodejnim zaznavanjem relacij in podatkovnih tipov, ročno moramo dodati ali urediti nepravilno zaznane. Če je relacij med tabelama več, je dovoljena samo ena aktivna relacija. To je pogosto pri »role-playing« dimenzijah, pogost primer je časovna dimenzija, ki se s tabelo dejstev povezuje večkrat, na primer z datumom računa in datumom odpošiljanja blaga v tabeli dejstev Prodaje. Podatkovni model lahko vsebuje več tabel dejstev, omogočene so tudi mnogo proti mnogo relacije kot v SSAS večdimenzijskem modelu.

Novost je določanje obojestranske smeri filtriranja na relaciji med dimenzijsko tabelo in tabelo dejstev, ki v bistvu kar nekoliko zabriše razliko med dimenzijami in dejstvi. Smer filtriranja pri tradicionalnih orodjih je vedno od dimenzijske tabele proti tabelam dejstev. Na primer mero Prodana vrednost iz tabele dejstev filtriramo po Ime prodajalca iz dimenzije. Z uporabo dvosmernega filtriranja pa lahko na primer preštejemo število prodajalcev iz dimenzije, tako da filtriramo po datumih računa iz table dejstev. Dvosmerno filtriranje deluje samo v primeru enostavne sheme (zvezdne ali snežinkaste). V primeru več tabel dejstev, ki so povezane z enakimi dimenzijami, obojestransko filtriranje povzroči nepravilne rezultate.

Zahtevnejše izračune nad podatki opravimo z vpeljavo novih stolpcev ali mer. Praktična razlika je, da se vrednosti novih stolpcev shranjujejo, mere se pa izračunajo sproti ob analizah. Izračune opravimo z jezikom DAX, znanim iz tabelarnega modela v SSAS. V našem primeru bi na tak način izračunali mere, kot so: razlika v ceni, tekoče vsote prodaje v letu, indekse vrednosti prodaje v primerjavi s predhodnim obdobjem, primerjave planske in uresničene vrednosti in podobno. V SSAS večdimenzijskem modelu so bile te mere izračunane z MDX.

Olajšano je tudi delo z dimenzijami. Skoraj vedno in povsod prisotne časovne dimenzije ni treba izdelovati. Ob analizi podatkov za filter preprosto uporabimo datumsko polje iz table dejstev in ustvari se navidezna dimenzija s hierarhijo leto – četrletje – mesec – dan. Podobno je s filtriranjem po drugih poljih iz tabele dejstev, na primer Številka računa iz tabele Prodaja. Pri tradicionalnem pristopu je bilo treba ustvariti posebno degenerirano dimenzijo. Tudi akcije vrtnanja do izvornih podatkov so omogočene brez posebnega načrtovanja v podatkovnem modelu za razliko od večdimenzijskega modela v SSAS.

V našem primeru bi s Power BI uresničili skoraj vse relacije in izračune. Izjema so otrok – starš relacije, v našem primeru v dimenziji Klasifikacija artiklov, v kateri nastopa relacija znotraj tabele med klasifikacijo in nadrejeno klasifikacijo. Taka relacija nam omogoča rekurzivne hierarhije v dimenziji, ki nimajo vnaprej znane globine. Power BI takih relacij ne omogoča, vendar je možno narediti približek s funkcijami v DAX jeziku, ki pa ima določeno končno globino.

Ena večjih omejitev Power BI je zagotovo omejitev velikosti podatkovnega modela na 1 GB. Kljub veliki stopnji stiskanja podatkov (približno 1:10) to pomeni približno 10 GB izvornih

podatkov. V našem primeru bi to na začetku zadostovalo, vendar bi po nekaj letih omejitve dosegli. Seveda bi bila možna delitev v smislu več področnih podatkovnih skladišč. Na primer delitev na področje prodaje in materialnega poslovanja, kjer je največ podatkov, vendar tudi to na dolgi rok ne bi zadostovalo. Bolj smiselna bi bila omejitev na zajem samo novjših podatkov, na primer zadnjih nekaj let, primerjava tekočih obdobj z lanskimi je namreč najbolj pogosta. Vendar bi tako izgubili možnost vrtnje počez v analizah, na primer spremljanje večjih strank, ki nastopajo kot kupci in dobavitelji, glede prodaje, nabave, planiranja in plačilne discipline. In seveda izgubili bi možnost analize starejših podatkov.

Upoštevati je treba tudi nalaganje podatkov v storitev Power BI, torej na Microsoftov podatkovni center na Irskem in dejstvo, da inkrementalno nalaganje ni omogočeno. Na testu smo dosegali hitrosti okoli 35 MBit/s, s katero se 10 GB izvornih podatkov naloži v 40 minutah, kar je ob uporabi samodejnega periodičnega osveževanja sprejemljivo. Omogočeno je dnevno ali tedensko osveževanje, kar zadošča zahtevam.

#### **4.3.4 Poročanje**

Podpora poročanju je odlična in je lahko dober razlog za uporabo Power BI. V našem primeru bi bila lahko to priložnost, da se obstoječa poročila v obliki tabel vizualno izboljšajo. Zaradi enostavnosti uporabe se lahko oblikovanje prepusti končnim uporabnikom. To je lahko nevhvaležno in časovno potratno opravilo, če uporabniki niso zadovoljni s trivialnimi stvarmi (za razvijalca), kot so na primer privzete barve, pisave in podobno. Le zahteve po samodejni dostavi poročil po elektronski pošti v našem primeru ne bi mogli uresničiti.

Oblikovanje poročil lahko izvedemo s Power BI Desktop, ali pozneje v storitvi Power BI, ki ponuja skoraj enak GUI za izdelavo poročil. Omogoča pa še hitri vpogled nad podatki. Ta funkcionalnost z enim klikom s pomočjo nabora algoritmov v podatkih poskuša najti in prikazati korelacije, izstopajoče podatke, osamelce, trende, sezonske vplive in podobno. Dodatna funkcionalnost je tudi poizvedovanje v naravnem jeziku, s katero hitro poiščemo zelene podatke z navodilom, kot je na primer »prodaja po kategorijah izdelkov«. No žal slovenski jezik ni podprt, vendar sama imena entitet algoritem poišče v podatkovnem modelu. Rezultati obeh funkcionalnosti so lahko v obliki številke ali pa v obliki poročila. Oboje lahko poleg navadnih poročil uporabimo kot gradnike nadzornih plošč, kar omogoča izdelavo poročil veliko hitreje in lažje.

#### **4.3.5 Postavitev rešitve in varnost podatkov**

Postavitev rešitve je hitra enostavna, v celoti jo objavimo v storitev Power BI, dobessedno s klikom na en gumb. Skupaj se prenesejo metapodatki o postopku ETL, nabor podatkov in definicije poročil. Nastaviti je potrebno še osveževanje podatkov. V primeru uporabe lokalnih virov podatkov je potrebna namestitev tako imenovanega prehoda, ki predstavlja povezavo med lokalnim strežnikom in storitvijo Power BI. Vnesti je potrebno še ustrezne poverilnice za

dostop do virov podatkov in nastaviti pogostost osveževanja. V našem primeru, torej ob uporabi SQL Serverja, je postavitve bolj zahtevna. Strojna in strežniška programska oprema je bila že nameščena, potrebna pa je ločena namestitev postopka ETL in nastavitev izvajanja, namestitev večdimenzijske baze in poročil. Vse skupaj je zahtevnejše in dlje trajajoče v primerjavi s Power BI. Prednost pa je možnost določanja ločenih nastavitev za razvojno in produkcijsko okolje, kot so na primer povezave na izvore podatkov, ki je v Power BI malo bolj zapletena.

Zagotavljanje varnosti in dodeljevanje dostopa do podatkov je v tradicionalnem poslovno inteligenčnem sistemu lahko zapleteno v primeru uporabe različnih komponent in porazdelitvah na več strežnikov. Naš primer je bil razmeroma enostaven – en strežnik z operacijskim sistemom Windows Server in SQL Serverjem. Upravljanje identitet in overovljanje je bilo že zagotovljeno s storitvijo Active Directory. Vsi uporabniki so imeli enake pravice dostopa do poslovne inteligence: dostop do OLAP kocke in poročil (ki zahteva tudi dostop do podatkovnega skladišča). V ta namen je bila ustvarjena skupina uporabnikov v Active Directory. Skupina je bila preslikana v varnostni sistem SQL Server, kjer je bil dodeljen dostop za branje v bazi podatkovnega skladišča, varnostni SSAS – dodeljen dostop do OLAP kocke in SSRS – dodeljen dostop do poročil. Nadzor dostopa do podatkov se je torej vršil centralno.

Tudi na področju varnosti je v Power BI poudarek na enostavnosti in neodvisnosti od IT oddelka. Privzeto lahko vsak posameznik samostojno pridobi dostop do storitve Power BI, posredovati mora le elektronski naslov, dodeljen s strani podjetja oziroma organizacije. Elektronski naslovi komercialnih ponudnikov, na primer Gmail-a, niso dovoljeni. Za upravljanje identitet in overjanje ob prijavi v Power BI skrbi storitev Azure Active Directory, ki te poverilnice uporabi, ko uporabnik želi dostopati virov, ki zahtevajo overjanje. Za organizacije, ki uporabljajo elektronske naslove v obliki, kot je ime.priimek@podjetje.si, je preslikava z lokalnimi poverilnicami v Active Directory samodejna (Iseminger, 2016, str. 17). Na tak način lahko uporabnik Power BI enostavno pridobi in objavi podatke iz lokalnih virov, če ima dovoljene za dostop in jih deli z drugimi uporabniki PowerBI. Uporabnik ima dosti večjo svobodo, vendar tudi odgovornost, podatke namreč lahko deli tudi z uporabniki Power BI, ki sicer nimajo poverilnic za dostop do izvornih podatkov. Če so uporabniki znotraj iste domene, lahko prejeta nadzorne plošče tudi delijo naprej. Omogočena je tudi delitev z uporabniki Power BI izven enake domene in celo javna objava na spletu.

Predmet delitve so lahko nadzorne plošče ali paketi vsebin, ki vsebujejo nabor podatkov, poročila in nadzorne plošče. V primeru delitve nadzornih plošč le te naslovniki samo pregledujejo in delijo naprej. Paketi vsebin omogočajo naslovnikom pregledovanje, ustvarjanje novih in tudi popravljanje obstoječih poročil. V slednjem primeru se uporabniku ustvari lastna kopija, originalno poročilo ostane nespremenjeno.

V našem primeru lahko do sistema poslovne inteligence dostopajo samo določeni uporabniki v podjetju, zato bi morali samostojno prijavljanje v Power BI preprečiti. To lahko stori administrator podjetja v storitvi Azure Active Directory. Tu se lahko določi tudi naprednejše večstopenjsko overjanje, privzeto je namreč samo z uporabniškim imenom v obliki službenega elektronskega naslova in gesla.

S Power BI je postavitve rešitve lažja in hitrejša. To velja za vse korake od postavitve razvojnega okolja do postavitve v produkciji in objavljanja poročil. Lahko pa ima za posledico slabši nadzor nad podatki in nekontrolirano širjenje kopij in različic poročil.

## Sklep

Uporabnost poslovne inteligence se je v preteklih desetletjih že dobro izkazala, vendar je bilo vse usmerjeno k večjim organizacijam. To je veljalo za miselnost, metodologije ter orodja in storitve za podporo sistemom poslovne inteligence. Ves razvoj na tem področju je bil usmerjen k zagotavljanju rešitev za najtežje primere uporabe in obvladovanju ogromnih količin podatkov. Sama zahtevnost metodologij in orodij zahteva posebej za to področje namenjene strokovnjake. Vse skupaj običajno predstavlja prevelike stroške za večino majhnih in srednjih organizacij. Tudi v večjih organizacijah se lahko pojavlja nezadovoljstvo zaradi togosti in počasnost razvoja takšnega sistema.

Predvsem v zadnjih letih smo priča spremembam tem področju s pojavom samopostrežne poslovne inteligence. Razvoj na tem področju je intenziven, žene ga predvsem povpraševanje uporabnikov po enostavnejšem in hitrejšem razvoju rešitev in uporaba agilnih metodologij, ki so se pred tem že izkazale pri razvoju programske opreme. Ključen del je odigral tudi razvoj pomnilniških in stolpčno usmerjenih baz. Rezultat so orodja, ki pohitrijo in poenostavijo tako razvoj kot postavitve sistema poslovne inteligence. Slednje je omogočil tudi nedavni razmah računalništva v oblaku. To omogoča poslovnemu uporabniku veliko neodvisnost od IT strokovnjakov pri analizah podatkov, kar skupaj s poslovnimi modeli najema programske opreme predstavlja nizke vstopne stroške. Zaradi tega je samopostrežna poslovna inteligenca idealna za organizacije, ki v takšen sistem še niso investirale in lahko v celoti zadosti potrebe po analizah.

Z možnostmi, ki jih ponuja samopostrežna poslovna inteligenca, prihaja tudi do odmika od nekaterih vrednot, ki jih zagovarja tradicionalni pristop, vendar to ne pomeni, da je ta zastarel. Oba pristopa imata svoje mesto in se lahko tudi dopolnjujeta, še posebej v primeru obstoječih investicij v tradicionalni sistem ali ko samopostrežna orodja ne zadostujejo več. Ta imajo omejitve, ob katere smo zadeli tudi v našem primeru, ob preučitvi orodja Power BI. Največja omejitev je dovoljena količina podatkov, ki se giblje v razredu velikosti več deset milijonov vrstic. Preverjeno pa se izkaže pri enostavnosti uporabe in hitrosti izvedbe tako pri postopku ETL kot pri oblikovanju podatkovnega modela. Še posebej pa se odlikuje pri poročanju, tudi z naprednimi možnostmi samodejnih hitrih vpogledov v podatke in poizvedovanja v naravnem

jeziku. Tudi postavitev sistema v storitvi Power BI in deljenje podatkov in poročil s sodelavci ali javnostjo je hitro in enostavno in lahko že kar skrb vzbujajoče glede varovanja podatkov.

V našem primeru bi se odločil za nadaljevanje uporabe obstoječe rešitve, saj ta izpolnjuje zahteve in omogoča obvladovanje takšne količine podatkov tudi z vidika omejevanja dostopa. Možnost uporabe samopostrežnih orodij vidim predvsem v uporabi naprednejših zmogljivosti poročanja z obstoječim podatkovnim skladiščem kot glavnim virom podatkov. Ti so že primerno urejeni in primerni za uporabo v analizah. Odločitev bi lahko posplošili tudi na druge organizacije, ki so že investirale v tradicionalni sistem poslovne inteligence.

## Literatura

1. Ali, A. (2012, 30. avgust). *Understanding New Column Store Index of SQL Server 2012*. Najdeno 3. maja 2016 na spletnem naslovu <http://www.databasejournal.com/features/mssql/understanding-new-column-store-index-of-sql-server-2012.html>
2. Gartner. (2016, 4. februar). *Magic Quadrant for Business Intelligence and Analytics Platforms*. Najdeno 3. maja 2016 na spletnem naslovu <https://www.gartner.com/doc/reprints?id=1-2XXET8P&ct=160204>
3. Harinath, S., Zare, R., Meenakshisundaram, S., Carroll, M., & Guang-Yeu Lee, D. (2009). *Professional Microsoft SQL Server Analysis Services 2008 with MDX*. Indianapolis: Wiley Publishing.
4. Imhoff, C., & White, C. (2011). *TDWI Best Practices Report: Self-Service Business Intelligence*. b.k.: TDWI.
5. Ingham, B. (2012, 26. maj). *The choice between Tabular or Multidimensional models in SQL Server Analysis Services 2012*. Najdeno 3. maja 2016 na spletnem naslovu <http://www.element61.be/e/resourc-detail.asp?ResourceId=485>
6. Inmon, W. H. (2005). *Building the Data Warehouse, Fourth Edition*. Indianapolis: Wiley Publishing.
7. Iseminger, D. (2016, april). *Power BI Security Whitepaper*. Najdeno 3. maja 2016 na spletnem naslovu <http://download.microsoft.com/download/4/8/C/48CFCF8A-2025-4B97-B249-7B505E26E7ED/Power%20BI%20Security%20Whitepaper.docx>
8. Jacobson, R., & Misner, S. (2006). *SQL Server Analysis Services Step by Step*. Redmond: Microsoft Press.
9. Kimball, R., & Ross, M. (2013). *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling, Third Edition*. Indianapolis: John Wiley & Sons.
10. Kimball, R., & Ross, M. (2002). *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling; Second Edition*. Indianapolis: Wiley Publishing.
11. Larson, B. (2009). *Delivering Business Intelligence with Microsoft SQL Server 2008*. New York: McGraw-Hill.
12. Moss, L. T., & Atre, S. (2003). *Business Intelligence Roadmap: The Complete Project Lifecycle for DecisionSupport Application*. Boston: Addison Wesley.

13. Mundy, J., Thornthwaite, W., & Kimball, R. (2006). *The Microsoft Data Warehouse Toolkit: With SQL Server 2005 and the Microsoft Business Intelligence Toolset*. Indianapolis: Wiley Publishing.
14. OLAP Council. (1995, januar). *OLAP AND OLAP Server Definitions*. Najdeno 3. maja 2016 na spletnem naslovu <http://www.olapcouncil.org/research/glossaryly.htm>
15. Rainardi, V. (2008). *Building a Data Warehouse: With Examples in SQL Server*. Berkeley: Apress.
16. Russo, M. (2011, 1. oktober). *The many-to-many revolution*. Najdeno 3. maja 2016 na spletnem naslovu <http://www.sqlbi.com/articles/many2many/>
17. Russo, M. (2013, 16. november). *Optimizing Data Models for Tabular*. Najdeno 3. maja 2016 na spletnem naslovu [http://www.sqlpass.org/Portals/333/Marco%20Russo\\_Optimizing%20Data%20Models%20for%20Tabular%20Solutions%20and%20PowerPivot.pdf?ver=2013-11-18-140112-790](http://www.sqlpass.org/Portals/333/Marco%20Russo_Optimizing%20Data%20Models%20for%20Tabular%20Solutions%20and%20PowerPivot.pdf?ver=2013-11-18-140112-790)
18. Russo, M. (2014, 29. avgust). *Whitepaper: SSAS Tabular as Analytical Engine*. Najdeno 3. maja 2016 na spletnem naslovu <http://www.sqlbi.com/articles/ssas-tabular-as-analytical-engine/>
19. SAM d.o.o.. (brez datuma). Najdeno 3. maja 2016 na spletnem naslovu [www.sam.si](http://www.sam.si)
20. Thompson, O. (2004). *Business Intelligence Success, Lessons Learned*. Najdeno 3. maja 2016 na spletnem naslovu <http://www.technologyevaluation.com/research/articles/business-intelligence-success-lessons-learned-17547/>
21. Veerman, E., Lachev, T., Sarka, D., & Loria, J. (2006). *SQL Server 2005 Business Intelligence – Implementation and Maintenance*. Redmond: Microsoft Press.





## **Priloge**

- 1 Seznam kratic
- 2 Terminološki slovar
- 3 Kontrola toka ETL v SSIS
- 4 Tok podatkov za dimenzijo artiklov v SSIS
- 5 Tok podatkov za tabelo dejstev prodaje v SSIS
- 6 Shema podatkovnega modela v SSAS
- 7 Določanje vrst relacij med dimenzijami in dejstvi v SSAS
- 8 Določanje izračunov novih mer v SSAS



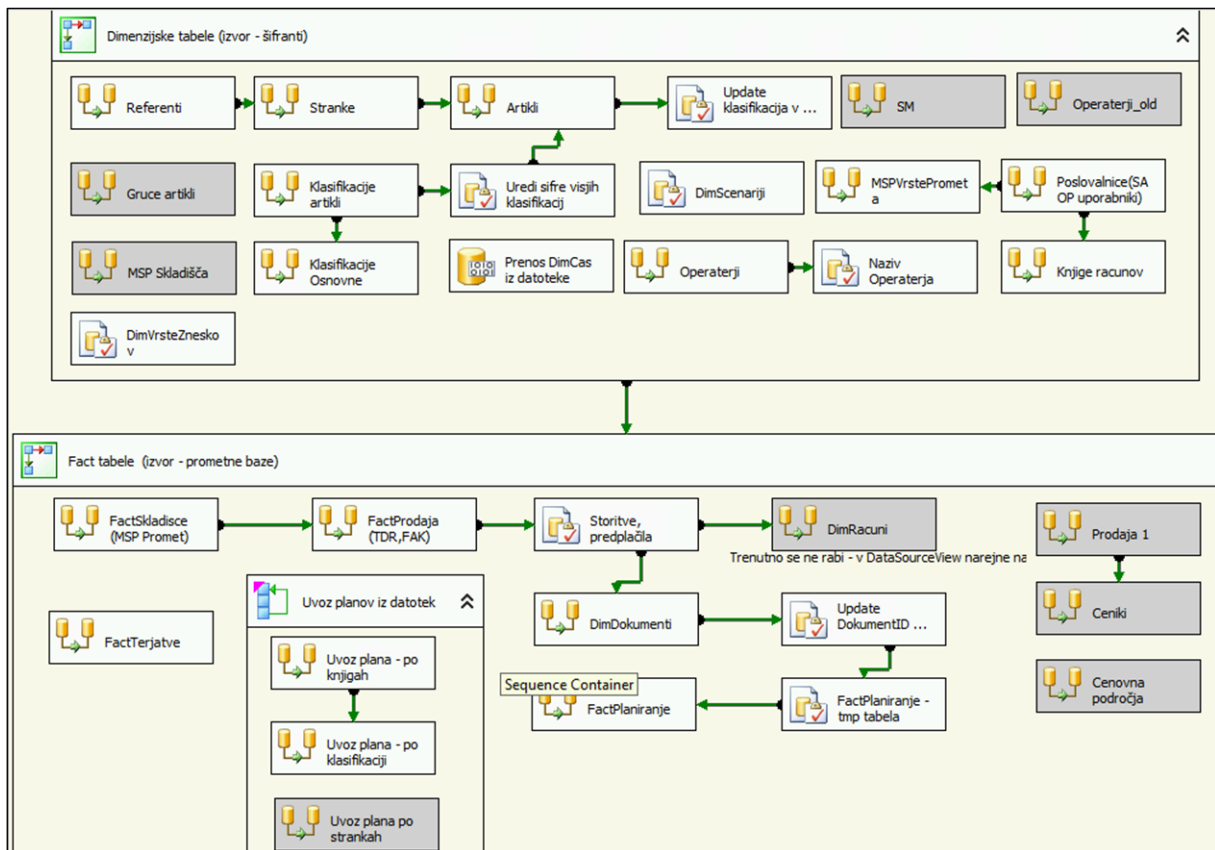
# 1 Seznam kratic

<i>Kratica</i>	<i>Opis</i>
BI	business intelligence
BISM	Business Intelligence Semantic Model
DAX	Data Analysis Expressions
DW	data warehouse
EIS	executive information system
ERP	enterprise resource planning
ETL	extract, transform, load
FTP	file transfer protocol,
GUI	grafični uporabniški vmesnik
HOLAP	hybrid OLAP
IT	informacijska tehnologija
KPI	key performance indicator
VBP	večdimenzijska baza podatkov
MDX	Multidimensional Expressions
MOLAP	multidimensional OLAP
MS	Microsoft
ODBC	Open DataBase Connectivity
OLAP	online analytical processing
OLTP	online transaction processing
POS	point of sale terminal
ROLAP	relational OLAP
SaaS	software as a service
SQL	structured query language
SSAS	SQL Server Analysis Services
SSIS	SQL Server Integration Services
SSMS	SQL Server Management Studio
SSRS	SQL Server Reporting Services
SUBP	sistem za upravljanje podatkovnih baz
XML	extensible markup language
XMLA	XML for Analysis

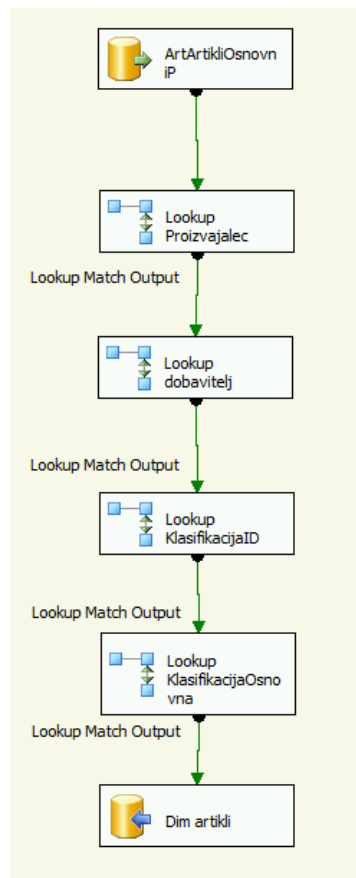
## 2 Terminološki slovar

<i>Angleško</i>	<i>Slovensko</i>
business intelligence	poslovna inteligenca
column oriented storage	stolpčno usmerjeno shranjevanje
columnstore index	stolpčno usmerjen indeks
data warehouse	podatkovno skladišče
dashboard	nadzorna plošča
decision support system	sistem za podporo odločanju
executive information system	direktorski sistem
extract, transform, load	ekstrakcija, preoblikovanje in nalaganje
fact table	tabela dejstev
hypercube	hiperkocka
in-memory database	pomnilniška baza podatkov
key performance indicators	ključni kazalniki poslovanja
measure	mera
multidimensional array	večdimenzijsko polje
online analytical processing	sprotna analitična obdelava podatkov
online transaction processing system	sistem za sprotno obdelavo transakcij
scalability	nadgradljivost
software as a service	programje kot storitev
solution	programsko rešitev
source control	nadzor različic
staging area	vmesno področje pri postopku ETL
trigger	prožilec
update	posodabljanje
web service	spletna storitev

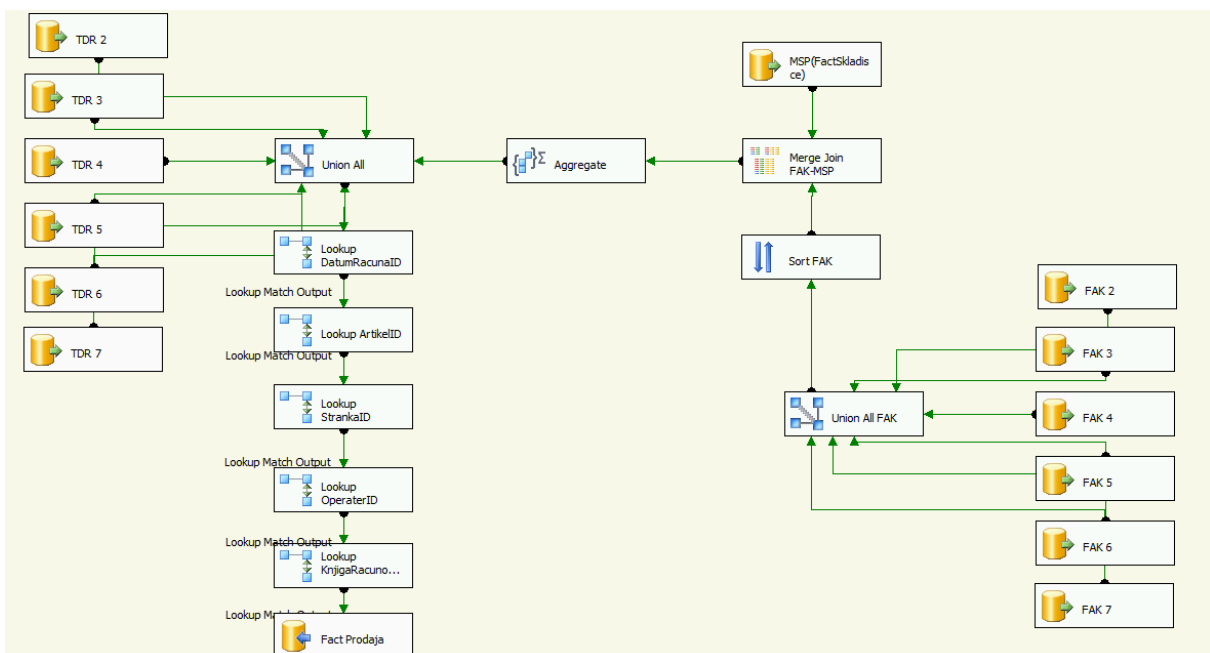
### 3 Kontrola toka ETL v SSIS



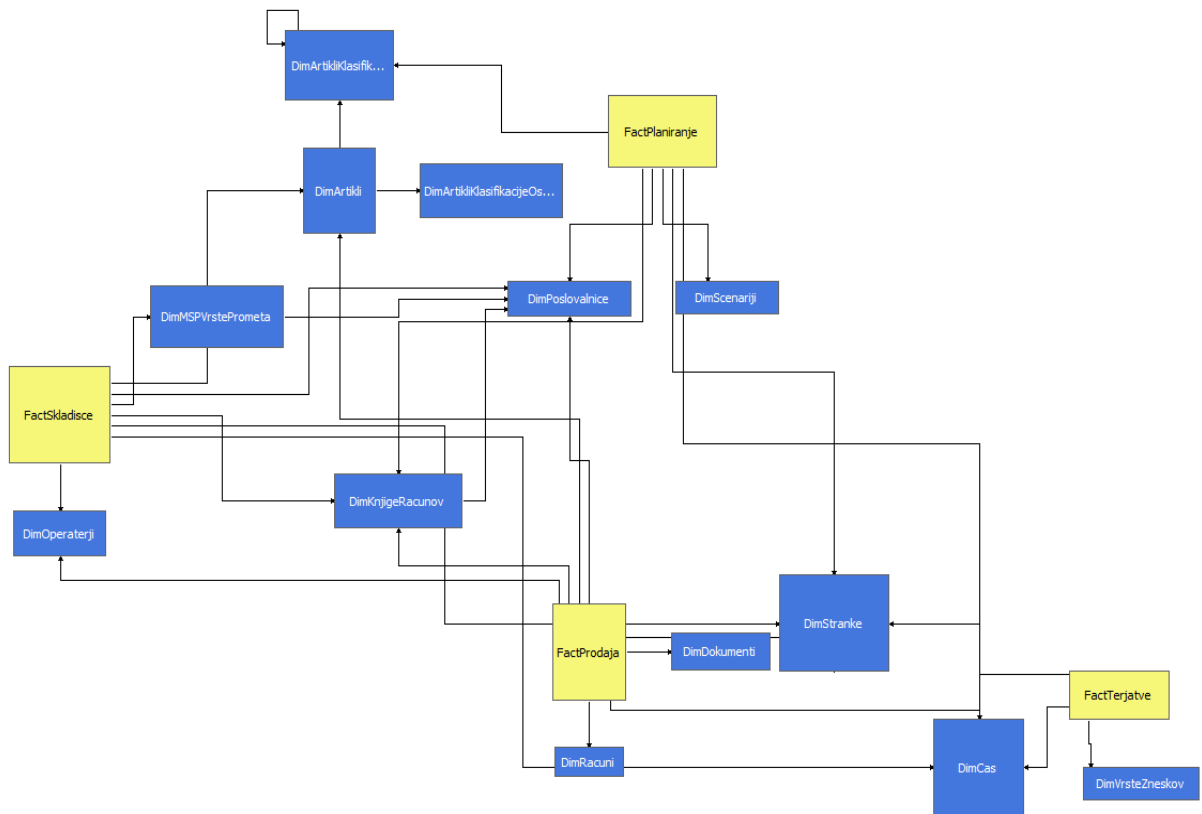
## 4 Tok podatkov za dimenzijo artiklov v SSIS



## 5 Tok podatkov za tabelo dejstev prodaje v SSIS



## 6 Shema podatkovnega modela v SSAS





## 7 Določanje vrst relacij med dimenzijami in dejstvi v SSAS

The screenshot displays the SSAS Dimensional Model Designer interface. On the left, a list of dimensions is shown, including Stranka, Poslovalnice, Čas, and others. The main area shows four measure groups: Prodaja, Zaloga, Planiranje, and Terjatve. A 'Define Relationship' dialog box is open, showing a 'Regular' relationship type. The dialog indicates that the dimension table is joined directly to the fact table. It shows the relationship between the 'DimStranke' dimension table and the 'FactProdaja' measure group table, with 'StrankaID' as the primary key in both. The dialog also shows the relationship between the 'StrankaID' dimension column and the 'StrankaID' measure group column.

## 8 Določanje izračunov novih mer v SSAS

The screenshot shows the SSAS Script Organizer window. The left pane displays a list of measures, with '[RVC lani]' selected. The right pane shows the properties for this measure. The 'Name' is '[RVC lani]'. The 'Parent hierarchy' is set to 'Measures'. The 'Expression' is defined as:

```
(PARALLELPERIOD ([Čas].[Leto-Mesec-Dan].[Leto]
, 1
, [Čas].[Leto-Mesec-Dan].CURRENTMEMBER)
, [Scenariji].[Scenariji].[Dejanski RVC], [Measures].[Vrednost])
```

The 'Additional Properties' section shows the 'Format string' set to '#,#.00', 'Visible' set to 'True', 'Non-empty behavior' set to 'Vrednost', and 'Associated measure group' set to '(Undefined)'. The 'Display folder' is also visible but empty.