

UNIVERZA V LJUBLJANI
EKONOMSKA FAKULTETA

MAGISTRSKO DELO

**PRIPRAVA IN INTEGRACIJA PODATKOV ZA SISTEME
POSLOVNE INTELIGENCE**

Ljubljana, 20. december 2018

ROK ZALOŽNIK

IZJAVA O AVTORSTVU

Podpisani Rok Založnik, študent Ekonomske fakultete Univerze v Ljubljani, avtor predloženega dela z naslovom Priprava in integracija podatkov za sisteme poslovne inteligence, pripravljenega v sodelovanju s svetovalcem prof. dr. Jurij Jakličem

IZJAVLJAM

1. da sem predloženo delo pripravil samostojno;
2. da je tiskana oblika predloženega dela istovetna njegovi elektronski obliki;
3. da je besedilo predloženega dela jezikovno korektno in tehnično pripravljeno v skladu z Navodili za izdelavo zaključnih nalog Ekonomske fakultete Univerze v Ljubljani, kar pomeni, da sem poskrbel/-a, da so dela in mnenja drugih avtorjev oziroma avtoric, ki jih uporabljam oziroma navajam v besedilu, citirana oziroma povzeta v skladu z Navodili za izdelavo zaključnih nalog Ekonomske fakultete Univerze v Ljubljani;
4. da se zavedam, da je plagiatorstvo – predstavljanje tujih del (v pisni ali grafični obliki) kot mojih lastnih – kaznivo po Kazenskem zakoniku Republike Slovenije;
5. da se zavedam posledic, ki bi jih na osnovi predloženega dela dokazano plagiatorstvo lahko predstavljalo za moj status na Ekonomski fakulteti Univerze v Ljubljani v skladu z relevantnim pravilnikom;
6. da sem pridobil/-a vsa potrebna dovoljenja za uporabo podatkov in avtorskih del v predloženem delu in jih v njem jasno označil/-a;
7. da sem pri pripravi predloženega dela ravnal/-a v skladu z etičnimi načeli in, kjer je to potrebno, za raziskavo pridobil/-a soglasje etične komisije;
8. da soglašam, da se elektronska oblika predloženega dela uporabi za preverjanje podobnosti vsebine z drugimi deli s programsko opremo za preverjanje podobnosti vsebine, ki je povezana s študijskim informacijskim sistemom članice;
9. da na Univerzo v Ljubljani neodplačno, neizključno, prostorsko in časovno neomejeno prenašam pravico shranitve predloženega dela v elektronski obliki, pravico reproduciranja ter pravico dajanja predloženega dela na voljo javnosti na svetovnem spletu preko Repozitorija Univerze v Ljubljani;
10. da hkrati z objavo predloženega dela dovoljujem objavo svojih osebnih podatkov, ki so navedeni v njem in v tej izjavi.

V Ljubljani, dne 20.12.2018

Podpis študenta: _____

KAZALO

UVOD	1
1 POSLOVNA INTELIGENCA	3
1.1 Opredelitev in cilji	3
1.2 Poslovna vrednost in koristi.....	4
1.3 Okolje poslovne inteligence.....	5
1.3.1 Operativni izvorni sistemi	5
1.3.2 Podatkovno skladišče	6
1.3.3 Področno podatkovno skladišče	7
1.3.4 Okolje za pripravo podatkov	8
1.3.5 Pridobivanje, preoblikovanje in nalaganje podatkov	8
1.3.6 Večdimenzionalna analitika	8
1.3.7 Podatkovno rudarjenje	9
1.3.8 Samopostrežna analitika	11
1.4 Trendi poslovne inteligence.....	12
2 PRIPRAVA IN INTEGRACIJA PODATKOV	12
2.1 Značilnosti podatkovnega skladiščenja	13
2.1.1 Znatost.....	14
2.1.2 Predstavitev podatkov	15
2.1.3 Meta podatki	16
2.2 pristopi k integraciji podatkov	16
2.2.1 Ročna integracija podatkov	17
2.2.2 Skupni uporabniški vmesnik	18
2.2.3 Integracija s pomočjo programske opreme.....	18
2.2.4 Integracija s pomočjo programskih vmesnikov	18
2.2.5 Enotni dostop do podatkov	18
2.2.6 Skupna shramba podatkov	19
2.3 Vzorci integracije podatkov	19
2.4 Tipične težave in izzivi.....	20
2.4.1 Heterogenost podatkov	20
2.4.2 »Umazani« podatki.....	21

2.4.3	Količina podatkov	21
2.4.4	Nepričakovani stroški.....	21
2.4.5	Pomanjkljivo znanje osebja.....	22
2.4.6	Pomanjkanje sodelovanja osebja.....	22
2.4.7	Napake zaradi staranja podatkov.....	23
2.5	Pridobivanje podatkov	23
2.5.1	Podatkovni viri	24
2.5.2	Profiliranje podatkov	24
2.5.3	Priprava podatkov	27
2.5.4	Osveževanje podatkovnega skladišča	27
2.6	Preoblikovanje podatkov	29
2.6.1	Izbira atributov	29
2.6.2	Pretvorba podatkovnih tipov	30
2.6.3	Čiščenje podatkov in zagotavljanje kakovosti	30
2.6.4	Povezovanje podatkov.....	35
2.6.5	Izpeljanke	35
2.6.6	Agregacija	35
2.6.7	Revizija informacij in skladnost sistema.....	36
2.6.8	Upravljanje praznih vrednosti	36
2.7	Nalaganje podatkov	36
2.7.1	Učinkovito nalaganje podatkov.....	37
2.7.2	Spreminjanje podatkov skozi zgodovino	37
3	PRIPRAVA IN INTEGRACIJA PODATKOV NA PRIMERU IZBRANEGA PODJETJA	38
3.1	Metodologija.....	39
3.2	Poslovne potrebe	39
3.3	Analiza informacijske arhitekture	40
3.4	Analiza podatkovnih virov in podatkov	42
3.4.1	Matični podatki strank.....	42
3.4.2	Podatki o bančnih računih	44
3.4.3	Podatki o kreditnih poslih	44
3.4.4	Podatki o varčevanjih	45
3.4.5	Podporni šifranti	45

3.4.6	Podatki o elektronskem bančništvu	45
3.5	Predlog rešitve.....	46
3.6	Pridobivanje podatkov	46
3.6.1	Pridobivanje podatkov iz Oracle podatkovnih baz.....	47
3.6.2	Pridobivanje podatkov iz zunanjih virov z uporabo spletnih storitev	48
3.6.3	Osveževanje podatkovnega skladišča.....	49
3.7	Preoblikovanje podatkov	50
3.7.1	Izbira atributov	50
3.7.2	Pretvorba podatkovnih tipov	50
3.7.3	Čiščenje podatkov in zagotavljanje kakovosti.....	52
3.7.4	Oblikovanje podatkovne sheme.....	56
3.7.5	Revizija informacij in skladnost sistema	58
3.8	Nalaganje podatkov	58
4	DISKUSIJA IN PREDLOGI.....	59
4.1	Ocena integracije podatkov	59
4.2	Predlogi in možnosti nadgradnje.....	61
SKLEP		61
LITERATURA IN VIRI		63
PRILOGE		67

KAZALO TABEL

Tabela 1: 6 osnovnih dimenzij kakovosti podatkov	31
--	----

KAZALO SLIK

Slika 1: Okolje poslovne inteligence	6
Slika 2: Primer OLAP kocke	10
Slika 3: Proces ETL	13
Slika 4: Pristopi integracije podatkov na različnih arhitekturnih ravneh	17
Slika 5: Informacijska arhitektura izbrane banke	41
Slika 6: Potek integracije podatkov na primeru izbranega podjetja	47
Slika 7: Logični podatkovni model pridobljenih podatkov	51
Slika 8: Izpis podatkov iz tabele »STRANKE«	52

Slika 9: Razpon vrednosti atributa »MATICNA« iz tabele »STRANKE«.....	53
Slika 10: Analiza atributa »DATUM_ROJSTVA« iz tabele »STRANKE«.....	55

SEZNAM KRATIC

ang. - angleško

ETL – (ang. Extract – Transform – Load); Proces pridobivanja, preoblikovanja in nalaganja podatkov

TDWI – (ang. The Data Warehousing Institute); Inštitut za podatkovno skladiščenje

OLAP – (ang. On-Line Analytical Processing); Sprotno analitično procesiranje

PoS – (ang. Point of Sale); Točka prodaje

SCD – (ang. Slowly Changing Dimension); Počasi spreminjajoče se dimenzije

ERP – (ang. Enterprise Resource Planning); Celovita informacijska rešitev

UVOD

Poslovanje podjetij se je v zadnjih letih zaradi nestabilnosti in nenehnih sprememb v poslovnem okolju močno spremenilo. Na vse težje pogoje poslovanja vpliva tudi vse večja konkurenčnost trga, ki je med drugim posledica globalizacije. Podjetja, ki se tega zavedajo in želijo preživeti ter prosperirati na dinamičnem trgu, se morajo pravočasno prilagajati trendom. Pri tem sta pomembna tudi sama strategija in vizija podjetja, ključne pa so poslovne odločitve, ki jih vsakodnevno sprejema management. »Dobre odločitve«, ki so potrebne za obstoj in razvoj podjetja, temeljijo na znanju in izkušnjah odločevalca, predvsem pa so za njih potrebne kakovostne in točne informacije (Sauter, 2011, str. 3).

Hiter razvoj informacijske tehnologije (ang. Information Technology) v zadnjih dveh desetletjih podjetjem omogoča boljšo podporo poslovnim procesom na različnih ravneh, hkrati pa odpira vrata novim poslovnim modelom. Podjetja, ki se zavedajo pomembnosti informacijske tehnologije, vanjo intenzivno vlagajo. Po navedbah Gartnerja (2017b), je največjo rast investicij v informacijsko tehnologijo opaziti na področju poslovne inteligence (ang. Business Intelligence) in poslovne analitike, pri čemer je pričakovati, da se bo trend v prihodnjih letih še stopnjeval.

Namen poslovne inteligence je iz osnovnih podatkov izluščiti uporabne informacije in pomagati managerjem pri zahtevnih odločitvah. Podjetja izvajajo takšne poslovne analize s pomočjo različnih tehnik, orodij in sistemov. Razvoj in uvedba takšnih sistemov poslovne inteligence predstavljata zahteven projekt, ki zahteva jasno vizijo, sredstva in potrpljenje. Čeprav imajo podjetja na voljo vse več različnih podatkov, so ti brez učinkovitega načina pridobivanja informacij in znanja iz njih praktično neuporabni. Podatkom lahko rečemo poslovna inteligenca, ko so v rokah odločevalcev, ki vedo, kaj z njimi narediti (Thierauf, 2001, str. 6). Uporabna vrednost informacij in znanje, ki ju podjetja lahko pridobijo iz množice podatkov, pa sta v veliki meri odvisna od kakovosti le-teh. Podatki, ki se uporabljajo za poslovno analitiko največkrat izvirajo iz različnih virov. To je eden izmed najpogostejših razlogov za vpeljavo podatkovnega skladišča, ki integrira podatke iz različnih sistemov in skrbi za njihovo konsistentnost ter veljavnost (Laursen & Thorlund, 2010, str. 138).

Proces priprave in integracije podatkov v podatkovno skladišče za namene poslovne inteligence v angleščini imenujemo Extract – Transform – Load (v nadaljevanju: ETL). Proces ETL je eden izmed najpomembnejših in najzahtevnejših delov projekta poslovne inteligence, kar potrjuje tudi dejstvo, da tipično zahteva do 80 % vseh razvojnih virov in sredstev (Inmon, 2005, str. 295). Razlog za to so številne ovire in izzivi, s katerimi se podjetja srečujejo pri izvedbi tega procesa. Ta je sestavljen iz 3 korakov. V prvem koraku je treba pridobiti podatke iz heterogenih virov. To so lahko razni sistemi in aplikacije, ki uporabljajo različne platforme, podatkovne baze, operacijske sisteme in komunikacijske

protokole, zaradi česar so podatki različne kakovosti in strukture. V drugem koraku, ki je tudi najzahtevnejši, moramo zato podatke s pomočjo različnih tehnik pretvoriti v obliko, ki bo zadoščala poslovnim potrebam. Pogosti problemi, ki se pojavljajo pri tem so povezani s: nekonsistentnimi ključi, kodiranjem, preoblikovanjem, preračunavanjem, izbiranjem najboljšega podatkovnega vira, razvrščanjem, združevanjem, izračunom sumarnih podatkov, ipd. Izziv z vidika učinkovitosti in razširljivosti podatkovnega skladišča pogosto predstavlja tudi velika količina podatkov. Tretji in zadnji korak procesa pa je integracija podatkov v podatkovno skladišče. Vsak projekt integracije podatkov je edinstven in prinaša s seboj specifične težave. Kljub temu pa lahko najdemo neke skupne vzporednice problemom, za katere obstajajo rešitve o tem, kako se z njimi spoprijeti.

Namen magistrskega dela je prispevati k razumevanju problematike priprave in integracije podatkov za sisteme poslovne inteligence. Raziskati želimo, kaj so ključni, oziroma najpogostejši problemi in izzivi, s katerimi se pri tem srečujemo, ter kaj so vzroki za njih. Predvsem pa želimo priti do spoznanj, kako se lahko s takšnimi težavami spopadamo in jih rešujemo. Namen dela je priti do uporabnih smernic in predlogov, ki bodo lahko v pomoč podjetjem pri vpeljevanju sistema poslovne inteligence.

Poglavitni cilj magistrskega dela je proučiti in analizirati najzahtevnejšo fazo v projektu razvoja sistema poslovne inteligence, to je priprava in integracija podatkov. S pomočjo obstoječe literature in teoretičnih izhodišč drugih avtorjev ter na podlagi lastne raziskave, bomo pregledali tipične izzive in probleme, s katerimi se podjetja srečujejo v omenjenem procesu. Cilj je tudi proučiti različne prijeme in tehnike za odpravo le-teh ter predlagati smernice za reševanje. V drugem, empiričnem delu, bomo na podlagi praktičnega primera pripravili in integrirali podatke v podatkovno skladišče. Pri tem bomo poizkušali ugotoviti, s čim smo imeli največ težav, zakaj, in kako smo jih reševali. Cilj tega dela je na praktičnem primeru identificirati probleme, na katere lahko naletijo podjetja pri vzpostavljanju sistema poslovne inteligence.

Magistrsko delo bo sestavljeno iz teoretičnega in empiričnega dela. V prvem delu bo vsebovalo poglobljen teoretično-analitičen pregled strokovne in znanstvene literature s področja priprave in integracije podatkov za sisteme poslovne inteligence. Poleg samega pregleda in analize bodo zaradi lažjega razumevanja predstavljeni tudi praktični primeri iz literature. V tem delu bodo za analizo uporabljene metode deskripcije in kompilacije, s slednjo bomo združili ugotovitve.

V drugem delu bo na podlagi pridobljenih spoznanj iz teoretičnega dela raziskave praktično predstavljena problematika na primeru izbranega podjetja. Tu bomo izluščili podatke iz internih informacijskih sistemov podjetja in dodatnega zunanega vira ter jih pripravili in integrirali v podatkovno skladišče za namene celostnega pregleda in segmentacije strank. Pri tem se bomo najverjetneje srečali s številnimi napakami in pomanjkljivostmi v podatkih, kot so: napake pri vnosu, prazne vrednosti, napake pri skladiščenju, napake zaradi staranja podatkov, ipd. Zato bomo v okviru priprave podatkov izvajali množico aktivnosti za

izboljšanje kakovosti, kot so: čiščenje in odstranjevanje duplikatov, popolnjevanje, preoblikovanje, upravljanje meta podatkov, idr. S pomočjo profiliranja bomo lažje predhodno zaznali napake, ki jih moramo popraviti pred integracijo. Pri integraciji bomo upoštevali tudi druge pomembne vidike, kot je recimo učinkovitost nalaganja.

Magistrsko delo bomo zaključili z diskusijo in s sklepnimi ugotovitvami.

1 POSLOVNA INTELIGENCA

Osnova za racionalne in strateško usmerjene odločitve v podjetju so informacije. Osnova za informacije pa so podatki. S procesom pridobivanja uporabnih informacij in znanja iz podatkov se ukvarja poslovna inteligenca.

1.1 Opredelitev in cilji

Kaj torej je poslovna inteligenca? Po Forresterju je poslovna inteligenca skupek metodologij, procesov, arhitekture in tehnologije, s pomočjo katerih pretvorimo podatke v smiselne in koristne informacije za analizo, poročanje in učinkovitejše odločanje (2018).

Podobno, vendar nekoliko bolj razširjeno, je poslovno inteligenco opredelil tudi The Data Warehousing Institute (v nadaljevanju: TDWI): »Poslovna inteligenca so procesi, tehnologije in orodja za pretvorbo podatkov v informacije, informacije v znanje in znanje v dobičkonosne poslovne odločitve. Poslovna inteligenca vključuje podatkovno skladiščenje, poslovno-analična orodja in upravljanje znanja« (Eckerson, 2002, str. 6). Slednja opredelitev je po našem mnenju zelo ustrezna, saj povzame vse bistvene elemente. Ja, poslovna inteligenca so procesi, tehnologije in orodja, poslovna inteligenca je integracija podatkov, je tudi poročanje in poslovna analitika, toda ni le to. Poslovna inteligenca je več kot le zbirka aplikacij in vizualizacij - to so le sredstva. Bistveno je znanje, ki ga lahko izluščimo iz razpoložljivih podatkov. Znanje je koncept razumevanja informacij na podlagi prepoznanih vzorcev (Loshin, 2012, str. 8). To je tisto, na podlagi česar se lahko sprejemajo boljše poslovne odločitve.

Zanimiva, nekoliko bolj splošna opredelitev poslovne inteligence je tudi: »Poslovna inteligenca so poslovne informacije in poslovne analize v okviru ključnih poslovnih procesov, ki vodijo do odločitev in ukrepov, ki povečujejo dobičkonosnost« (Williams & Williams, 2006, str. 2).

Podoben izraz, katerega pogosto pomotoma zamenjujemo z zgornjim, vendar nima enakega pomena je: sistem poslovne inteligence (ang. Business Intelligence System). Ta označuje sistem oziroma aplikacijo za poslovno poročanje in analitiko (Chaffey & Wood, 2005, str. 420). Nekatera podjetja imajo lahko takšne sisteme kar v obliki preprostih Excelovih preglednic ali pred-pripravljenih poročil. Druga, sodobnejša in tehnološko naprednejša podjetja imajo lahko kompleksnejše in bolj sofisticirane sisteme, ki odkrivajo vzorce ter

povezave med podatki z uporabo tehnologij podatkovnega rudarjenja in umetne inteligence. Predhodniki sistemov poslovne inteligence se imenujejo sistemi za podporo odločanja (ang. Decision Support Systems).

Na podlagi obstoječih definicij poslovne inteligence drugih avtorjev opazimo, da ni enotne opredelitve pojma. Različni strokovnjaki iz različnih področij jo opisujejo malce drugače, iz svojega zornega kota. Pa vendar lahko iz prebranega povzamemo večino opredelitev, da poslovna inteligenca ni le tehnologija ali metodologija, temveč je širši pojem, ki združuje produkte, tehnologijo in metode za pridobitev in organizacijo informacij, ki managementu pomagajo sprejemati »dobre« odločitve. Zaradi že zgoraj omenjenih razlogov, nam je od navedenih opredelitev najbližja tista od TDWI. Zato jo bomo imeli v mislih, ko bomo v nadaljevanju tega dela omenjali poslovno inteligenco.

Cilji poslovne inteligence so jasni: izkoristiti informacije, ki so na voljo za izboljšanje procesa odločanja in poslovanja podjetja. Drugače povedano, cilj je pomagati managerjem oziroma drugim odločevalcem sprejemati boljše poslovne odločitve.

Po Sauter-ju je cilj poslovne inteligence odločevalcu zagotoviti prave informacije ob pravem času, za sprejem pomembnih odločitev (2011, str. 55).

Loshin, pa je primarni cilj poslovne inteligence opredelil kot zmožnost upravljanja dostopa do potrebnih informacij za ocenitev poslovnih potreb, prepoznavo potencialnih virov podatkov in upravljanja pretoka podatkov v informacijske sisteme primerne za poročanje ter analize (Loshin, 2012, str. 5).

1.2 Poslovna vrednost in koristi

V poslovnem svetu je primaren cilj podjetij dobiček, zato se trudijo, da bi čimbolj zvišala prihodke in znižala stroške. Zaradi tega morajo biti pri vlaganjih še posebno previdna, da je vsaka investicija poslovno upravičena. Pozitiven trend investicij v podatkovno analitiko in poslovno inteligenco kaže na to, da podjetja v tem vidijo določene koristi, dodano vrednost. Zavedajo se pomembnosti podatkov, ki jih imajo na voljo. Z uporabo poslovne inteligence poizkušajo iz surovih podatkov, ki so sami po sebi neuporabni, pridobiti informacije, ki pa predstavljajo strateški vir in v pravih rokah tudi konkurenčno prednost podjetja.

Pridobljenim informacijam in znanju težko določimo denarno vrednost. Lahko pa na podlagi izkušenj številnih podjetij povzamemo področja, na katerih poslovna inteligenca prinaša dodano poslovno vrednost. Loshin v svoji knjigi *Business Intelligence: The Savvy Manager's Guide* navaja 4 glavna področja dodanih vrednosti poslovne inteligence (2012, str. 17):

- **Finančne koristi** izboljšana dobičkonosnost na podlagi višjih prihodkov in nižjih stroškov.

- **Povečanje produktivnosti:** pospešitev izvedbe poslovnih procesov, zmanjšan del ročnih intervencij in višji volumen proizvodnje / storitev.
- **Povečanje zaupanja:** več zaupanja s strani strank, trga in zaposlenih. Posledično tudi večje zadovoljstvo strank in partnerjev.
- **Zmanjšanje tveganja:** večja transparentnost do strank in trga, znižano tveganje kapitala, zakonska skladnost in manjša verjetnost uhajanja informacij in prevar.

Te predpostavke so seveda posplošene, v resnici podjetja opažajo različne prednosti uporabe poslovne inteligence. Te se lahko razlikujejo od primera do primera in so odvisne od načina poslovanja podjetja, organizacijske strukture, zaposlenih, uporabljenih tehnologij, ipd.

1.3 Okolje poslovne inteligence

Sedaj, ko razumemo cilje poslovne inteligence, se lahko osredotočimo na njene komponente. Podjetja se pri analizah in poročanju zanašajo na ustrezno infrastrukturo. Arhitektura le-te pa je pomemben faktor uspešnosti projekta poslovne inteligence (Anand, 2014). Velika količina obravnavanih podatkov je namreč neuporabna brez primerne pristopa in tehnoloških rešitev za obdelavo. Zaradi tega je pri vpeljavi rešitve poslovne inteligence ključno vzpostaviti primerno okolje.

Različne tehnološke komponente se uporabljajo za zajem, pripravo in integracijo podatkov, ter izvedbo samih analiz in končnega prikaza informacij poslovnim uporabnikom. Okolje poslovne inteligence vključuje spekter zmogljivosti, od operativnih postopkov, kot so ekstrakcija in integracija podatkov, standardizacija, čiščenje in validacija, pa do analitičnih komponent, kot so podatkovna skladišča, analitična orodja za prikaz informacij, nadzorne plošče, OLAP orodja, podatkovno rudarjenje, samopostrežna analitika (ang. Self-Service BI), ipd. Slika 1 prikazuje najpogostejše komponente, ki skupaj tvorijo poslovno inteligenco.

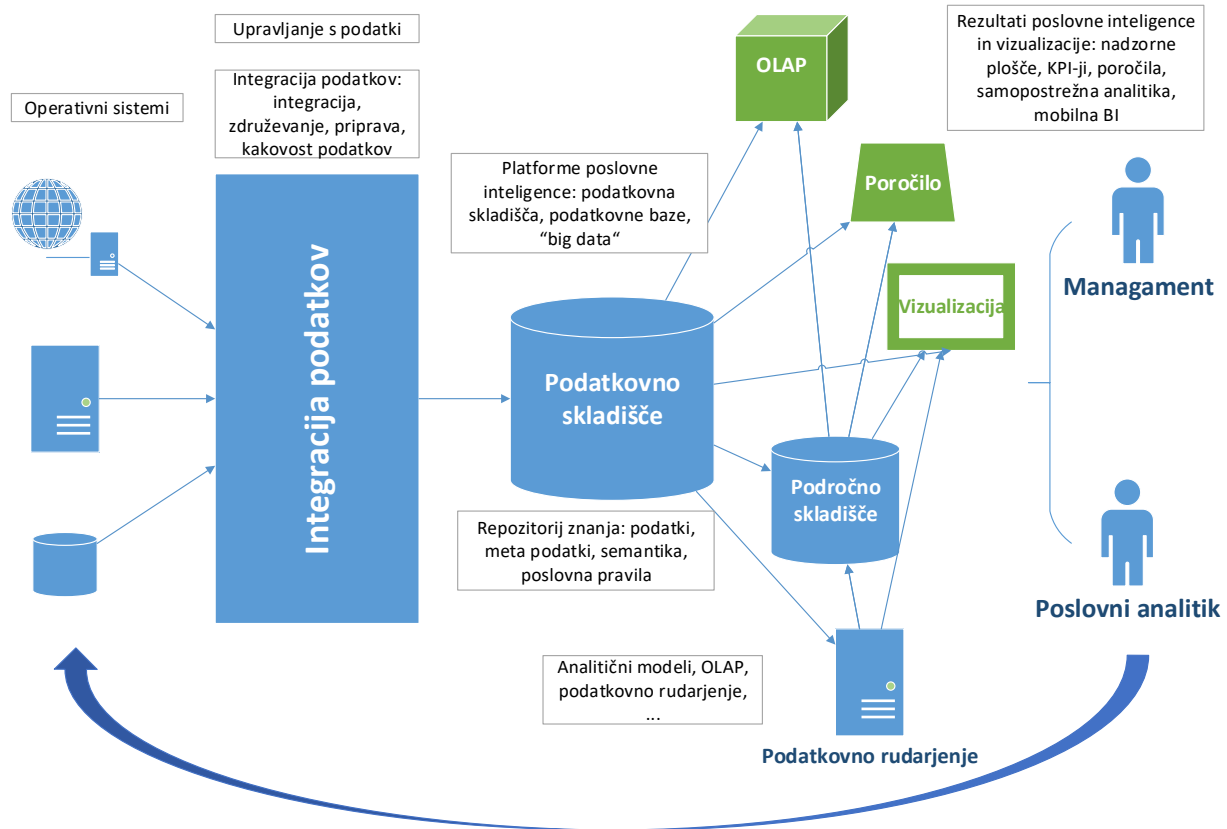
V nadaljevanju tega podpoglavja se bomo le-teh tudi nekoliko podrobneje dotaknili, pri čemer se bomo osredotočili na začetno fazo projekta poslovne inteligence, to je priprava in integracija podatkov.

1.3.1 Operativni izvorni sistemi

To so sistemi, ki jih podjetja uporabljajo za podporo vsakodnevnim poslovnim procesom. Zajemajo in hranijo operativne podatke oziroma transakcije tekočega poslovanja. Njihova glavna naloga je velika učinkovitost in razpoložljivost, zaradi česar niso najbolj primerni za izvajanje zahtevnejših analitičnih poizvedb.

Operativni sistemi zaradi zgoraj omenjenih lastnosti navadno hranijo le sveže podatke, zgodovinski podatki pa se prenašajo v arhivske shrambe. Zato so eni izmed glavnih virov podatkov za polnjenje podatkovnega skladišča.

Slika 1: Okolje poslovne inteligence



Vir: Loshin (2012).

1.3.2 Podatkovno skladišče

Podatkovno skladišče (ang. Data Warehouse) je pomemben del infrastrukture poslovne inteligence. Inmon, znan tudi kot »oče podatkovnega skladišča« je podatkovno skladišče opredelil kot zbirko podatkovnih baz, ki so (2005, str. 389):

- **Predmetno usmerjene:** organizacija skladišča temelji na glavnih entitetah podjetja in ne na funkcionalnih področjih.
- **Integrirane:** podatki prihajajo iz več aplikativnih sistemov, ki so med seboj pogosto nekonsistentni.
- **Vsebujejo časovno dimenzijo:** za podatek v podatkovnem skladišču mora biti poznan čas oziroma časovni interval veljavnosti, kar omogoča opazovanje trendov.
- **So nesprejemljive:** podatki niso podvrženi spremembam v realnem času s strani aplikacij, vendar se osvežujejo z določeno frekvenco.

Bolj splošna opredelitev opisuje podatkovno skladišče kot zbirko poslovnih informacij in podatkov pridobljenih iz operativnih sistemov in drugih zunanjih podatkovnih virov.

Namenjeno je podpori poslovnih odločitev, saj omogoča analize in poročanje na različno podrobnih nivojih (Techopedia).

Na prvi pogled se podatkovno skladišče ne razlikuje bistveno od klasičnih, operativnih podatkovnih baz. Zato se pojavlja vprašanje, zakaj sploh obstajajo podatkovna skladišča, če imamo na voljo podatkovne baze. Razlogi za ločitev analitičnih podatkovnih skladišč od operativnih podatkovnih baz so številni, najbolj očitni pa so sledeči (Inmon, 2005):

- Podatki za analitične namene zahtevajo drugačno obliko in strukturo kot podatki iz operativnih podatkovnih baz.
- Tehnologije za obdelavo operativnih podatkov so bistveno drugačne od tehnologij za podporo analitičnih potreb.
- Uporabniki operativnih in analitičnih podatkov so različni.
- Značilnosti obdelave operativnih podatkov so bistveno drugačne od analitičnih.

Ključna prednost podatkovnega skladišča je, da zagotavlja vse potrebne informacije na vseh ravneh v organizaciji. Posledično odločanje ni več nujno omejeno le na vodstveni kader (Chu, 2003).

Prednost podatkovnega skladišča je tudi integracija podatkov pridobljenih iz različnih virov. To pomeni, da so le-ti v podatkovnem skladišču že ustrezno združeni, prečiščeni in preoblikovani. To analitiku močno olajša delo, ker so vsi podatki na enem mestu, zato so lažje dostopni in v predpisani obliki omogočajo lažje izvajanje analiz. Pomemben dejavnik, ki ga ne smemo pozabiti omeniti je učinkovitost. Operativne podatkovne baze so v osnovi namenjene shranjevanju podatkov tekočega poslovanja. Z analizami in poročanjem neposredno iz iste baze bi lahko preobremenili in upočasnili delovanje glavnega informacijskega sistema ter posledično celo ovirali poslovanje podjetja, kar pa je nedopustno. Za razliko od operativnih podatkovnih baz hranijo podatkovna skladišča tudi zgodovinske podatke. Zaradi zgoraj naštetih razlogov je podatkovno skladišče nepogrešljiv sestavni del poslovne inteligence.

1.3.3 Področno podatkovno skladišče

Pogosto se v zvezi s podatkovnimi skladišči uporablja tudi izraz področno podatkovno skladišče (ang. Data Mart). To je del podatkovnega skladišča, ki pokriva zahteve določenega oddelka. Področno podatkovno skladišče se torej osredotoča na bolj specifične zahteve, kar mu omogoča optimizirano analizo in poročanje.

Pomanjkljivost področnih skladišč je, da so podatkovni modeli različnih področnih podatkovnih skladišč med seboj nekompatibilni in neuporabni za druge poslovne uporabnike. To lahko pomeni težjo vzpostavitev celovitega sistema poslovne inteligence in za podjetje predstavlja še večji izziv.

1.3.4 Okolje za pripravo podatkov

Izraz »staging area«, ki v angleščini pomeni okolje za pripravo podatkov označuje kratkotrajno shrambo podatkov, ki je namenjena preoblikovanju podatkov iz izvornih sistemov v obliko primerno za ciljno podatkovno strukturo, najpogosteje podatkovno skladišče. Namen tega okolja je razbremenitev podatkovnega skladišča.

1.3.5 Pridobivanje, preoblikovanje in nalaganje podatkov

Množico procesov, s pomočjo katerih pridobimo in naložimo podatke v podatkovno skladišče pogosto imenujemo extract – transform – load (ETL), kar v angleščini pomeni pridobivanje, preoblikovanje in nalaganje podatkov.

Pridobivanje je torej prvi korak zahtevnega procesa polnjenja podatkovnega skladišča. Podatke je treba prebrati iz izvornih sistemov in jih prenesti v okolje za pripravo podatkov. Izzivi v tem delu so posledica različne strukture podatkov, saj le-ti prihajajo iz heterogenih sistemov.

Tako pridobljene podatke je potrebno preoblikovati in prestrukturirati v obliko, ki jo zahteva podatkovno skladišče. Zato se v tem koraku izvede množico aktivnosti prečiščevanja, združevanja in seštevavanja podatkov, na podlagi česar se podatki obogatijo s potrebnimi poslovnimi informacijami. Ustrezno preoblikovani podatki so pripravljeni za nalaganje v podatkovno skladišče.

Zadnji korak ETL procesa je nalaganje podatkov v podatkovno skladišče. V tem delu je predvsem izziv zagotavljanje učinkovitosti, sploh če je količina podatkov velika.

Obstajajo različne programske rešitve, ki deloma avtomatizirajo ETL proces in uporabnikom pomagajo pri njegovi izvedbi. Vendar ti še zdaleč ne rešujejo vseh težav, na katere lahko naletijo.

1.3.6 Večdimenzionalna analitika

Večdimenzionalna analitika (ang. On-Line Analytical Processing, v nadaljevanju: OLAP) je tehnologija, ki analitikom omogoča vpogled v podatke glede na različne dimenzije hkrati (Loshin, 2012, str. 83). Ta razširjena tehnologija poslovne inteligence se uporablja za poslovno poročanje, kompleksne analitične izračune, spremljanje učinkovitosti, finančne analize, napovedovanje, itd. Vir podatkov je največkrat podatkovno ali področno skladišče, kajti »živo« transakcijsko bazo bi s kompleksnimi operacijami hitro preobremenili.

S pristopom večdimenzionalnega modela, ki se ga poslužuje OLAP, si lahko strukturo podatkov predstavljamo kot kocko in na podatke gledamo iz različnih zornih kotov. Podatki so organizirani v več dimenzij in meritev. Dimenzije kategorizirajo podatke in lahko

vsebujejo več ravni podrobnosti, čemur pravimo hierarhija. V t.i. OLAP kocki posamezne koordinatne osi predstavljajo dimenzije, stolpci pa meritve. Slednje so združeni podatki, ki jih skušamo analizirati, največkrat so numerični. Za kocke obstaja več shem, pri čemer je vsaka sestavljena iz tabele dejstev in tabel(e) dimenzij. Tabela dejstev je osrednja, najbolj obsežna tabela, ki vsebuje podrobnosti o poslovnih dogodkih, medtem ko so dimenzijske tabele bistveno manjše in vsebujejo več ravni podrobnosti. Najbolj razširjeni sta zvezdna in snežinkasta shema.

Pogosta poslovna vprašanja, na katera s pomočjo OLAP-a lažje odgovorijo analitiki v smislu so lahko v smislu (Jaklič, 2017, str. 172):

- kolikšna je bila skupna,
- povprečna,
- največja,
- ali najmanjša prodaja,

glede na kategorijo izdelkov in regijo v prvem četrtletju?

Za lažjo predstavo večdimenzionalnega modela si pogledajmo še sliko 2. Recimo, da nas zanima koliko izdelka A smo prodali na lokaciji LJ v drugem četrtletju letošnjega leta. S pomočjo vizualizacije kocke na vprašanje odgovorimo enostavno tako, da pogledamo posamezne dimenzije in najdemo željen podatek.

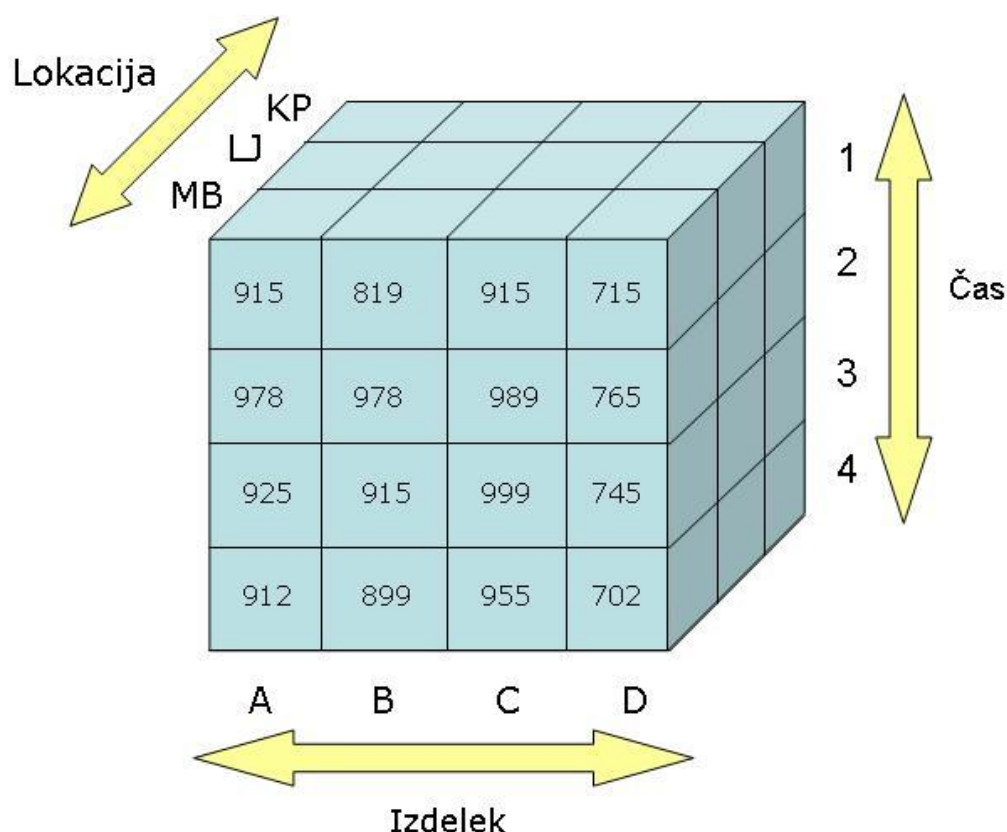
Podatki v OLAP-u so za razliko od podatkov v operativnih relacijskih bazah torej že združeni, kar pomeni da ni potrebe po dodatnih, računsko in časovno zahtevnih operacijah. Zaradi tega lahko analitiki do poslovnih izračunov dostopajo zanesljivo in v realnem času. Slabost pa je posledično to, da je potrebnega veliko časa za predpripravo podatkov, le-ti pa zasedajo veliko prostora na disku.

Poslovna vrednost OLAP-a je torej zmožnost hitre analize podatkov iz različnih perspektiv, ne glede na količino in kompleksnost. Analitikom in managerjem pomaga odgovarjati na pomembna poslovna vprašanja.

1.3.7 Podatkovno rudarjenje

Podatkovno rudarjenje je izraz, ki se je v zadnjih letih zelo razširil in za katerega pogosto slišimo, predvsem na področju poslovne inteligence. Skupina Gartner je pojem opredelila kot proces odkrivanja uporabnih povezav, vzorcev in trendov v velikih količinah podatkov s pomočjo statističnih in drugih matematičnih tehnik (Gartner, Inc., 2017a). Lahko bi rekli, da je podatkovno rudarjenje eden izmed najbolj naprednih tehnologij poslovne inteligence.

Slika 2: Primer OLAP kocke



Vir: FMF Ljubljana (brez datuma).

Bistvena razlika podatkovnega rudarjenja v primerjavi s klasičnimi statističnimi analizami je, da le-ta s pomočjo preiskovanja in učenja na podatkih odkrije skrito znanje, tudi »tisto, česar ne iščemo«. Z drugimi besedami, omogoča nam avtomatizirano iskanje informacij oziroma znanja v kopici podatkov, ki bi jih s t.i. »človeško gnanimi« (ang. human driven) metodami težko odkrili.

Posledično podatkovno rudarjenje prinaša več poslovne vrednosti kot katerakoli druga tehnologija ali orodje poslovne inteligence. Tehnologija se je izkazala kot zelo uporabna na različnih področjih uporabe, tako za poslovne, kot tudi druge namene. Najbolj tipični primeri uporabe so (LeBlanc, M.Moss, Sarka & Ryan, 2005, str. 176):

- **Navzkrižna prodaja (ang. Cross-selling):** nakup drugih izdelkov istega podjetja, zelo priljubljena in razširjena strategija spletnih trgovcev.
- **Odkrivanje prevar:** zelo pomembno opravilo za banke in izdajalce kreditnih kartic, ki želijo omejiti škodo.
- **Zadrževanje strank:** različni ponudniki storitev, vključno s telekomunikacijskimi podjetji, bankami in zavarovalnicami želijo odkriti vzroke za odhod strank in jih preprečiti.

- **Upravljanje odnosov s strankami (CRM):** koncept temelji na dobrem poznavanju strank. Pomemben je celoten proces, od pridobivanja stranke, razvoja in zadrževanja.
- **Optimizacija spletnih strani:** izboljšave za končne uporabnike.
- **Napovedovanje:** praktično vsako podjetje potrebuje neko obliko napovedovanja, s pomočjo katerega bi lahko bolje planiralo poslovanje.

Obstaja več tehnik rudarjenja, ki pa se v grobem delijo na dve kategoriji uporabe. Prva se imenuje prediktivna (nadzorovan pristop) in temelji na napovedovanju vrednosti neznanih primerov na podlagi sklepanja iz obstoječih primerov. Druga, deskriptivna (nenadzorovan pristop) tehnika, pa se poslužuje odkrivanja splošnih značilnosti iz obstoječih podatkov na podlagi določenih pravil in vzorcev (LeBlanc, M.Moss, Sarka & Ryan, 2005, str. 174).

Podatkovno rudarjenje je torej izrazito zmogljiva in uporabna tehnika, vendar sta za njeno uporabo potrebna tudi zmogljiva računalniška oprema in znanje analitika, ki mu v žargonu pravimo tudi »rudar«.

1.3.8 Samopostrežna analitika

Naprednejši uporabniki imajo dandanes na voljo sodobnejša analitična orodja, ki prinašajo nove analitične priložnosti. Tehnološki razvoj na področju poročanja in »in-memory« aplikacij, oziroma aplikacij, ki tečejo v hitrem pomnilniku, sta omogočila, da je uporaba t.i. samopostrežne (ang. Self-Service / Ad Hoc BI) analitike bolj dostopna, kot kadarkoli doslej.

Kadar standardna in pred-pripravljena analitična poročila ne zadostujejo več, si lahko zahtevnejši uporabniki potrebne odgovore na poslovna vprašanja poiščejo sami s pomočjo samopostrežne analitike. Bistvo samopostrežne analitike lahko opišemo s tremi ključnimi lastnostmi (LeBlanc, M.Moss, Sarka & Ryan, 2005, str. 41):

- **Željene informacije:** podatki in lastnosti, ki jih analitik želi.
- **Kakorkoli:** informacije se lahko prikazujejo ali analizirajo v različnih oblikah in formatih.
- **Uporabnik:** bistvo samopostrežne analitike je končni uporabnik - analitik.

Samopostrežna analitika je lahko preprosta kot poročilo, ki omogoča vnos določenih parametrov ali pa zapletena kot interaktiven vmesnik, ki uporabniku omogoča, da si sam »na klika« svojo vizualizacijo in pregled nad podatki.

Podjetja, ki se poslužujejo analitike lahko uporabljajo tako tradicionalna analitična orodja, kot tudi samopostrežno analitiko, nekatera pa hibridne pristope obojega. Ključna prednost samopostrežnih orodij je hitrost in prilagodljivost končnemu uporabniku, s katero se tradicionalna orodja ne morajo kosati. Neprestano večja količina podatkov in naraščajoče poslovne potrebe so vzrok, da morajo analitiki vse večji del poslovne inteligence opraviti

sami. Zato jim hitre odgovore na poslovna vprašanja lahko omogoči le primerno orodje, brez dodatne pomoči IT strokovnjakov, ki lahko med tem opravljajo pomembnejša dela.

1.4 Trendi poslovne inteligence

Hiter tempo tehnološkega razvoja je opazen na različnih področjih industrije, pri čemer področje poslovne inteligence ni nobena izjema. Vsako leto se pojavijo nove tehnologije, ki podjetjem zagotavljajo hitrejši in boljši vpogled v podatke. Opaznejši trendi razvoja, kot so: digitalizacija in varnost podatkov, računalništvo v oblaku, umetna inteligenca, mobilna poslovna inteligenca, ipd., odpirajo podjetjem vrata v svet praktično neomejenih analitičnih priložnosti. Pri tem pa ne gre pozabiti, da nove tehnologije niso zagotovilo za dobro in učinkovito poslovno inteligenco, če so podatki slabo integrirani ali slabše kakovosti.

Novejša raziskava, v kateri je sodelovalo več kot 2770 poslovnih uporabnikov, svetovalcev in prodajalcev orodij poslovne inteligence, na prvo mesto po pomembnosti prihajajočih trendov področja poslovne inteligence uvršča upravljanje in kakovost matičnih podatkov (ang. Master Data Management). Na drugem mestu sledi odkrivanje in vizualizacija podatkov, na tretjem samopostrežna analitika, na četrtem in petem mestu upravljanje ter integracija podatkov (Bange, in drugi, 2018).

Na podlagi rezultatov raziskave lahko zaznamo enotno sporočilo: poslovna inteligenca brez celovite integracije podatkov in iniciativ za zagotavljanje njihove kakovosti ne more biti učinkovita. Prav tako lahko sklepamo, da poslovni uporabniki po mnenju sodelujočih v raziskavi potrebujejo več avtonomije (samopostrežna analitika), kar pa lahko predstavlja dodaten izziv pri integraciji podatkov, saj mora biti le-ta še bolj učinkovita in preprosta.

2 PRIPRAVA IN INTEGRACIJA PODATKOV

Neustrezno integrirani podatki iz operativnega okolja sistemom poslovne inteligence ne omogočajo izvedbe koristnih in merodajnih analiz, zato so iz poslovnega vidika neuporabni, uvedba novega analitičnega sistema pa ni upravičena (Inmon, 2005, str. 19). Dejstvo je, da so operativni podatki pogosto zelo kompleksni, zaradi česar je proces integracije zapleten in neprijeten. Nič čudnega ni, da podjetja pri projektu poslovne inteligence največ časa in sredstev izgubijo prav s pripravo ter integracijo podatkov v podatkovno skladišče. Proces dodatno otežijo številni dejavniki, kot so recimo: poslovne zahteve, viri podatkov, omejena sredstva, pomanjkljivo znanje osebja, itd. Ker se omejitve, podatki in potrebe razlikujejo od primera do primera, ni proces integracije podatkov nikoli povsem enak. Čeprav obstajajo različne programske rešitve, ki pomagajo avtomatizirati določen del tega procesa, opravilo še zdaleč ni preprosto.

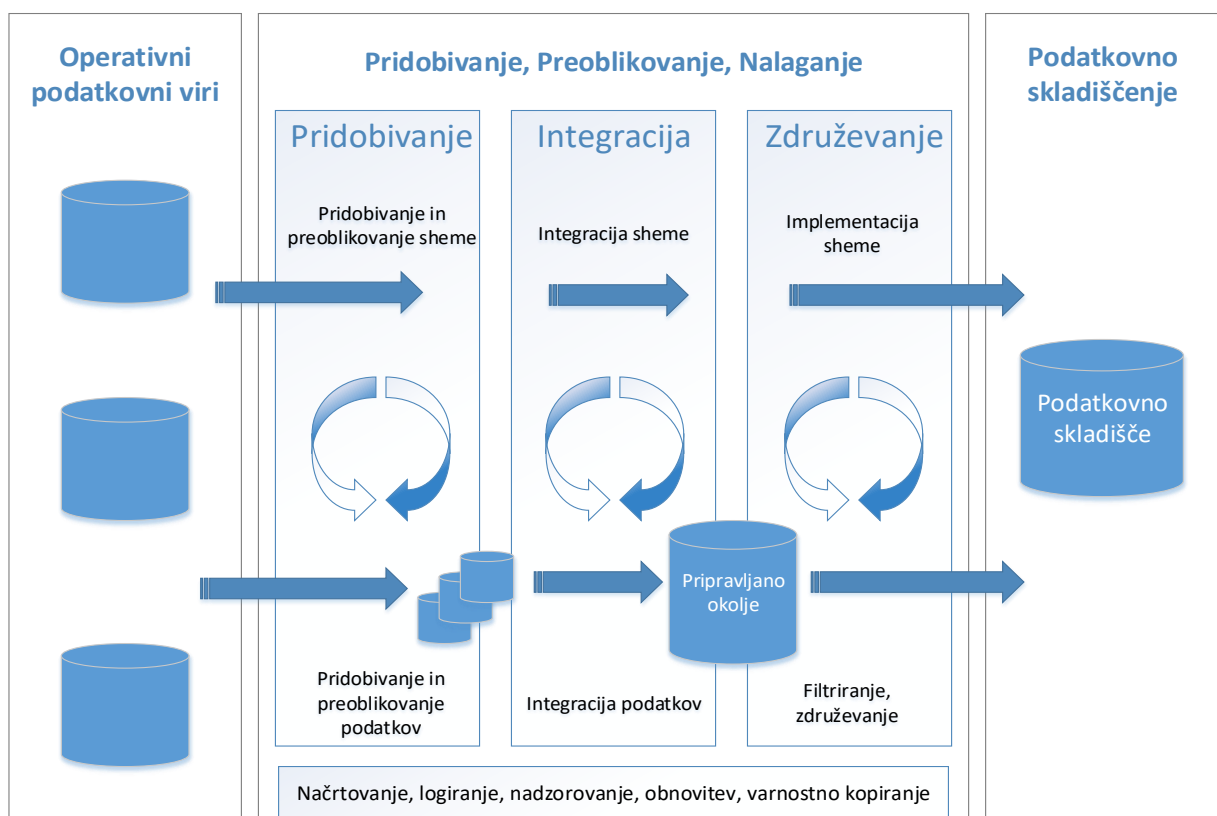
V tem poglavju bomo podrobneje proučili dobre prakse procesa priprave in integracije podatkov, ki temeljijo na izkušnjah številnih uspešnih projektov poslovne inteligence.

Pogledali si bomo različne pristope integracije podatkov, njihove vzorce in najpogostejše izzive, probleme, ter napake s katerimi se podjetja pri tem soočajo. Proučili bomo vzroke za njihov nastanek in analizirali pristope, s katerimi jih lahko odpravljamo. Upoštevali bomo najpomembnejše vidike, kot so: kakovost podatkov, učinkovitost delovanja in razširljivost. Analizo samega procesa bomo razdelili na tri logične sklope (slika 3):

- pridobivanje podatkov,
- preoblikovanje in čiščenje podatkov,
- nalaganje podatkov.

Pred tem si bomo pogledali še nekatere pomembne značilnosti podatkovnega skladiščenja in opisali nekaj splošnih vzorcev integracije podatkov.

Slika 3: Proces ETL



Vir: Rahm & Dos (2000).

2.1 Značilnosti podatkovnega skladiščenja

Preden se spustimo v podrobnejšo analizo procesa integracije podatkov bi bilo smiselno, da se vrnemo nazaj k podatkovnemu skladiščanju. Kaj podatkovno skladišče je, in čemu je namenjeno smo zapisali že v prejšnjem poglavju. V tem poglavju, pa bomo pregledali še

nekaj podrobnosti podatkovnih skladišč, ki nam bodo služile pri razumevanju problematike v nadaljevanju tega dela.

2.1.1 Zrnatost

Zrnatost ali granularnost (ang. *granularity*) je pomembna lastnost podatkovnih skladiščih, ki nam pove kako podrobni so podatki. Manjša kot je zrnatost, bolj združeni in manj podrobni podatki nastopajo v podatkovnem skladišču ter obratno. Zrnatost predstavlja enega izmed večjih izzivov oblikovanja podatkovnega skladišča, saj vpliva na zaseden prostor shranjenih podatkov in na vrsto ter natančnost poizvedb (Inmon, 2005, str. 44).

V primeru, ko so izvorni podatki zelo podrobni oziroma zrnati, jih je potrebno pred integracijo ustrezno urediti, prefiltrirati in združiti. V nasprotnem primeru, pa razdružiti v manjše, bolj obvladljive dele. V obeh primerih, podjetja porabijo veliko časa in sredstev, da zagotovijo ustrezno stopnjo zrnatosti. Ta je odvisna od poslovnih potreb in tehničnih zmogljivosti analitične infrastrukture podjetja.

Bolj zrnati podatki omogočajo podrobnejše in posledično točnejše informacije. Največja prednost je fleksibilnost, saj lahko uporabniki izkoristijo isto zbirko podatkov za različne analitične namene (Kimball, Ross, Thornthwaite, Mundy & Becker, 2013, str. 75). Četudi se pojavi potreba po drugačnem vpogledu v podatke, so podatki v podatkovnem skladišču že pripravljeni, zato podjetju ni potrebno izgubljati dragocenega časa s ponovno integracijo.

Višja stopnja zrnatosti ima tudi svoje slabosti. Zamislimo si primer zbirke podatkov o nakupih strank na POS terminalu (ang. *Point Of Sale*) v neki trgovini, v obdobju enega meseca. V primeru višje zrnatosti, bi v podatkovnem skladišču hranili po en zapis z vsemi podrobnostmi za posamezen nakup (transakcijo). Predpostavimo povprečno 100 nakupov dnevno in 30 dni v mesecu, torej skupno 3000 zapisov. V primeru nizke zrnatosti, pa bi recimo potrebovali le en zapis za en mesec. Ta bi vseboval združene, sumirane podatke o nakupih v tem obdobju, atributi bi bili: mesec, skupno število nakupov, povprečen znesek nakupa in skupen znesek vseh nakupov.

Iz primera vidimo, da je število zapisov v drugem primeru bistveno manjše (razmerje 1:3000). S takšnim pristopom prihranimo kapacitete diskovnega prostora, ob enem pa tudi močno razbremenimo procesorske vire in kompleksnost baznih poizvedb, izgubimo pa seveda podrobnosti.

Če nam tako združeni podatki še vedno zagotavljajo sprejemljiv nivo podrobnosti, se tak način združevanja podatkov iz zgoraj omenjenih razlogov zagotovo izplača. Odločitev je odvisna od poslovnih potreb in se razlikuje od primera do primera.

2.1.2 Predstavitev podatkov

Podatkovna shema oziroma podatkovni model je bistvena lastnost podatkovnih baz, ki opredeljuje strukturo, v kateri so shranjeni podatki. Podatkovni model v klasičnih, operativnih podatkovnih bazah je namenjen predstavitvi osnovnih podatkov in učinkovitosti delovanja, zato mora biti ustrezno normaliziran. To pomeni, da so posamezne entitete ločene in povezane med seboj s tujimi ključi, kar omogoča hitrejše poizvedbe in manjšo podvojenost podatkov (redundanca).

Podatki v podatkovnem skladišču so lahko predstavljeni v različnih oblikah, pri čemer prevladujeta dva pristopa. Prvi, morda najbolj razširjen je dimenzijski pristop, ki ga zagovarja Ralph Kimball (Kimball, Ross, Thornthwaite, Mundy & Becker, 2013, str. 10). Bistvo tega je, da so podatki shranjeni v denormalizirani obliki, namesto entitetno-relacijskih modelov se uporabljajo dimenzijski modeli. Z osnovnimi koncepti dimenzijskega modeliranja, kot so zvezdne sheme, tabele dejstev in dimenzije, smo se seznanili že v prejšnjem poglavju, pri večdimenzionalnem modelu OLAP kocke. Ključna prednost tega pristopa je, da uporabniki tako strukturirane podatke lažje razumejo in uporabljajo. Slabost pa, da je podatke v takšno podatkovno skladišče težje integrirati in pri tem ohranjati integriteto podatkov.

Drugi pogost pristop, katerega je opisal Bill Inmon zavzema stališče, da morajo biti tudi podatki v podatkovnem skladišču normalizirani. Glede na predmetnost, so podatki porazdeljeni v različne entitete - tabele v podatkovnem skladišču. Glavni razlogi, ki jih navaja avtor, zakaj je normaliziran pristop optimalen za podatkovno skladišče so (Inmon, 2005, str. 137):

- zagotavljanje fleksibilnosti,
- ustrezna struktura za zelo podrobne (granularne) podatke,
- ni optimiziran za nobeno od zahtev obdelovanja podatkov,
- se dobro prilagaja podatkovnemu modelu.

Obstaja pa tudi relacijsko-dimenzijski pristop, ki je nekakšen hibrid med obema. Prednost tega pristopa je večja prilagodljivost poslovnim potrebam (Tupper, 2011, str. 330) in praviloma lažja integracija podatkov iz operativnih sistemov, saj je struktura že v izhodišču podobna.

Ne glede na vrsto podatkovnega skladiščenja, pa je podatke pri prehodu v podatkovno skladišče potrebno preoblikovati. Inmon navaja, da je pri tem najprej potrebno odstraniti povsem operativne podatke, preostalim pa dodati časovno dimenzijo, če ta še ni prisotna. Potem je potrebno združiti podatke s podobnimi značilnostmi, ki lahko prihajajo iz različnih tabel (2005, str. 90). Poleg tega, se po potrebi spremeni nivo podrobnosti (zrnatost) in doda dodatne izpeljane podatke (Jaklič, 2017, str. 104).

2.1.3 Meta podatki

Meta podatki so podatki, ki opisujejo druge podatke. Predstavljajo pomemben element v okolju poslovne inteligence in podatkovnih skladiščih. V splošnem poznamo več tipov meta podatkov, ki jih v lahko kategoriziramo takole:

- **Tehnični meta podatki:** opisujejo fizične objekte, kot so: tabele in polja v podatkovnem skladišču.
- **Poslovni meta podatki:** rečemo jim tudi semantični meta podatki, saj opisujejo vsebino podatkov, recimo: dejstva, dimenzije, logična razmerja, ipd.
- **Procesni meta podatki:** opisujejo operacije oziroma procese. Primer: opisa procesa integracije podatkov v podatkovno skladišče ali opis bazne poizvedbe.

Ralph Kimball je v eni izmed svojih knjig zapisal, da so meta podatki kot enciklopedija podatkovnega skladišča (2013, str. 173), ker prikazuje pomembnost meta podatkov, ki jih sicer najdemo skoraj povsod. Lahko gre za preproste tehnične meta podatke, kot je recimo število stolpcev v določeni tabeli, ali pa za kompleksnejše, kot je opis procesa transformacije podatkov.

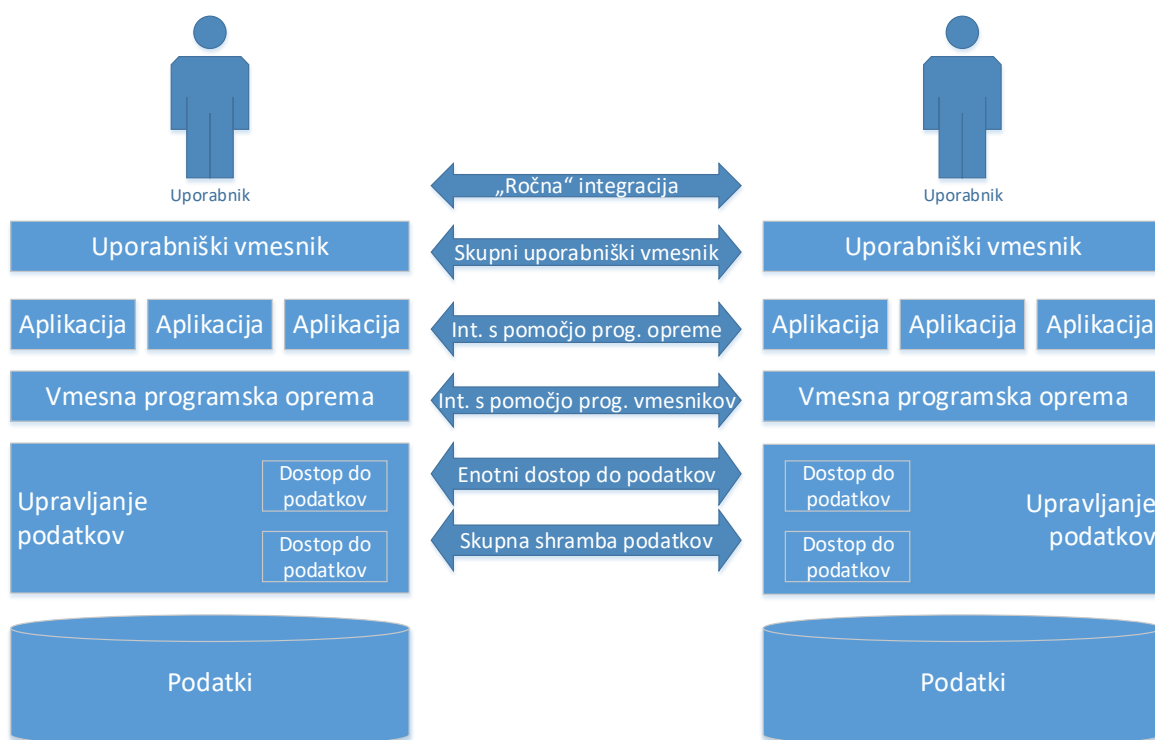
Dobro obvladovanje meta podatkov (ang. Metadata management) je eden izmed pomembnejših dejavnikov uspešnega projekta poslovne inteligence. Razlogov za to je več, glavni pa so naslednji (Loshin, 2012, str. 121):

- Metapodatki opisujejo konceptualni, logični in fizični model za pretvorbo surovih podatkov v ustrezne modele za analizo.
- Meta podatki opisujejo strukturo podatkov za podatkovno skladišče in poslovno inteligenco.
- Procesni meta podatki lahko zagotavljajo revizijsko sled.
- Vsebinski meta podatki pomagajo analitiku pravilno interpretirati podatke pridobljene iz analiz.
- S pomočjo meta podatkov lahko sledimo spremembam podatkov skozi čas.

2.2 Pristopi k integraciji podatkov

Integracijo podatkov lahko pojasnimo kot kombinacijo tehničnih in poslovnih procesov, s pomočjo katerih pridobimo in združimo podatke iz različnih virov (kot so podatkovne baze, datoteke in drugi sistemi), pri čemer dobimo enoten pogled na podatke za poslovno inteligenco. Obstaja več načinov, s katerimi lahko dosežemo enoten pogled na podatke, kot prikazuje slika 4.

Slika 4: Pristopi integracije podatkov na različnih arhitekturnih ravneh



Vir: Ziegler & Dittrich (2007).

Pristope lahko torej ločimo glede na raven abstrakcije, v katerem se integracija podatkov izvaja. S tem imamo v mislih večplastno arhitekturo informacijskih sistemov, ki na najvišji ravni vključuje uporabnike aplikacij, na nižjih ravneh so programska oprema in programski vmesniki ter na najnižji ravni podatkovna baza in podatki.

Spodaj bomo na kratko opisali 6 pristopov integracije podatkov na različnih ravneh (Ziegler & Dittrich, 2007, str. 4).

2.2.1 Ročna integracija podatkov

Najprej omenimo ročno integracijo podatkov, ki tehnično gledano sploh ni integracija podatkov. Pa vendar, gre za pristop, kjer uporabniki neposredno dostopajo do relevantnih podatkov v izvornih informacijskih sistemih in drugih virih. Uporabniki morajo podatke in različne vmesnike za dostop do njih dobro poznati, da jih lahko uporabljajo. V vsakem primeru, pa gre za neučinkovit in neprimeren pristop za upravljanje z večjimi količinami podatkov ali za kakršnokoli naprednejšo analitiko.

2.2.2 Skupni uporabniški vmesnik

Pristop integracije podatkov na najvišji ravni abstrakcije deluje tako, da uporabnik s pomočjo skupnega uporabniškega vmesnika dostopa do podatkov, ki so shranjeni v različnih, izvornih sistemih. Vmesnik zagotavlja enoten občutek dostopa do podatkov, vendar mora uporabnik podatke kasneje še vedno sam integrirati. Tipičen primer takšnega uporabniškega vmesnika je lahko spletni brskalnik (npr. Internet Explorer, Google Chrome, ipd.) in aplikacija, ki omogoča dostop do podatkov, na primer spletni iskalnik (npr. Google, Yahoo, Najdi.si, ipd.). Ta uporabniku omogoča vnos poizvedbe, nato pa mu zadetke ponudi na isti način, ne glede na to, iz katerega spletnega strežnika oziroma spletne strani izvirajo.

2.2.3 Integracija s pomočjo programske opreme

Naslednja možnost je uporaba namenskih aplikacij za integracijo podatkov. Takšne aplikacije dostopajo do podatkov iz različnih virov, vendar za razliko od zgornjega pristopa, vračajo že integrirane podatke. Pristop je v praksi uporaben, če je izvornih sistemov relativno malo in prav tako količina podatkov ni pretirana. V nasprotnem primeru lahko rešitev zaradi prevelike heterogenosti podatkov postane preveč kompleksna in neučinkovita.

2.2.4 Integracija s pomočjo programskih vmesnikov

Programski vmesnik oziroma v angleščini »middleware«, se v splošnem uporablja za povezovanje različnih programskih rešitev in sistemov med seboj. Rešuje premostitvene težave med sistemi in podatki. Ta pristop integracijo prenaša na nižjo raven, raven programskih vmesnikov, vendar nekatere dele integracije še vedno opravljajo aplikacije. Primer je recimo SQL programski vmesnik, ki zagotavlja »skupno točko« za izvedbo SQL poizvedb na povezanih sistemih. Vendar rezultati poizvedb še vedno niso integrirani, saj ne vračajo homogenih rezultatov. Zaradi tega ta pristop vseeno zahteva pomoč druge programske opreme oziroma aplikacij.

2.2.5 Enotni dostop do podatkov

Enotni dostop do podatkov (ang. uniform data access) imenujemo tudi navidezna integracija. Podatki ostajajo v izvornih sistemih, do njih pa se dostopa s t.i. pogledi, ki zagotavljajo enoten pogled nad podatki. Gre za podoben pristop, kot zgoraj omenjeni enoten uporabniški vmesnik, vendar je v tem primeru funkcionalnost dosežena na nižji abstraktni ravni. Prednost tega pristopa je, da so podatki vedno »sveži«, saj prihajajo neposredno iz izvornih sistemov. Slabost pa je dodatno breme, ki ga poizvedbe povzročajo na izvornih sistemih in slabša učinkovitost, saj se integracija ter poenotenje podatkov dogajata v realnem času. Tipičen primer tega pristopa je integracija podatkov s pomočjo spletnih storitev (recimo SOAP).

2.2.6 Skupna shramba podatkov

Zadnji pristop skupne shrambe podatkov imenujemo tudi fizična integracija podatkov in je za nas najpomembnejši. Sem spada tudi ETL proces, ki mu bomo v nadaljevanju dela posvetili največ pozornosti. Gre za integracijo podatkov v pravem pomenu besede, saj se v tem primeru podatki iz izvornih operativnih sistemov prenesejo v namenska skladišča, najpogosteje podatkovna skladišča, lahko pa tudi v operativne podatkovne shrambe. V prvem primeru je potrebno ustrezno načrtovati in realizirati tudi periodično osveževanje podatkov.

2.3 Vzorci integracije podatkov

Obstajajo različni pristopi integracije podatkov, ki se razlikujejo glede na način izvedbe, podjetja pa jih uporabljajo v skladu s strategijo in poslovnimi potrebami. Zaradi tega je nekakšne splošne smernice, kako naj integracija poteka, težko podati.

Na prvi pogled bi lahko rekli, da je pri integraciji ključen tehnološki vidik. Da je možnost uspeha projekta integracije podatkov in učinkovitosti nastalega podatkovnega skladišča najbolj odvisna od tehnologije. Toda v resnici ni najpomembnejša tehnologija, pomembnejša je zasnova, načela in vzorci integracije (Reeve, 2013, str. 79). Dejstvo je, da je lahko kljub uporabi najzmogljivejših in najbolj naprednih tehnologij integracija neuspešna in posledično podatkovno skladišče neuporabno, če se je lotimo napačno.

Zavedati se je potrebno, da je ključen cilj integracije podatkov v podatkovno skladišče in vzpostavitev sistema poslovne inteligence zagotavljanje točnih, konsistentnih in zanesljivih podatkov za podporo poslovne analitike. Za izpolnitev tega cilja mora ETL proces izpolnjevati tri osnovne zahteve (Kimball, Ross, Thornthwaite, Mundy & Becker, 2013, str. 483):

- **Zanesljivost:** ETL proces ni enkraten dogodek, saj se v določenih intervalih neprestano ponavlja. Brezkompromisno mora zagotavljati podatke, ki so vredni zaupanja na vseh ravneh podrobnosti.
- **Dosegljivost:** Podatkovno skladišče mora zadostiti potrebam po dosegljivosti (ang. Service Level Agreement).
- **Upravljaljivost:** Podatkovno skladišče ni nikoli končno. Neprestano raste, se osvežuje in spreminja v skladu s poslovnimi potrebami. Zato je pomembno, da omogoča enostavno upravljanje v skladu s potrebami v danem trenutku.

Izpolnitev zgornjih treh zahtev je začetni korak na poti izgradnje učinkovitega podatkovnega skladišča.

2.4 Tipične težave in izzivi

V zadnjih nekaj letih se je zaradi velikega tehnološkega razvoja veliko spremenilo tudi na področju integracije podatkov v sisteme poslovne inteligence. Zaradi novejših tehnologij, kot je na primer internet stvari (ang. Internet of Things), imajo podjetja na voljo vse več podatkov, ki pa jih je vse težje obvladovati. Zato se podjetja pri tem srečujejo s številnimi novimi problemi in izzivi.

Ker želimo z integracijo podatkov analitikom omogočiti enoten vpogled v podatke, je pomembno, da se pri tem držimo istih načel abstrakcije na vseh ravneh: pri strukturiranju podatkov, pri podatkovnih shemah in modelih, pri reševanju konfliktov v podatkih, ipd. Zastaviti si moramo realne in uresničljive cilje ter zahteve, saj v osnovi operativni informacijski sistemi niso namenjeni integraciji, zato je le-ta še bolj zahtevna. Težavnost procesa je v praksi odvisna od številnih dejavnikov, najbolj izraziti pa so sledeči (Ziegler & Dittrich, 2007, str. 3):

- arhitektura informacijskih sistemov,
- vsebina in funkcionalnost komponent sistema,
- vrsta in tipi podatkov (alfa-numerični podatki, multimedijske vsebine, strukturirani podatki, nestrukturirani podatki),
- zahteve glede systemske avtonomije,
- zahteve glede učinkovitosti,
- razpoložljivost virov (čas, denar, kadri, znanje, itd.),
- razpoložljiva programska oprema za upravljanje s podatki (ang. data management software),
- razpoložljivi podatkovni modeli sheme, semantika podatkov,
- razpoložljivi programski vmesniki,
- poslovna pravila in integriteta podatkov.

Izzivi, s katerimi se podjetja pri tem srečujejo so raznoliki, predvsem na račun večje kompleksnosti, količine in raznolikosti podatkov, pa tudi vse večjih zahtev, ki jih diktira konkurenca. Spodaj bomo opisali nekaj najpogostejših.

2.4.1 Heterogenost podatkov

Heterogenost podatkov je ena izmed prvih težav na katero naletimo pri integraciji podatkov iz več virov v podatkovno skladišče. Vsak sistem je razvit za specifične namene in potrebe, zato obdeluje ter hrani podatke v različnih formatih in oblikah, ki so posledica različnih poslovnih procesov. Integracija mora torej vsebovati različne ukrepe za poenotenje podatkov na skupno osnovo.

V praksi se heterogenost podatkov rešuje z ustreznimi preslikavami na izbrano podatkovno shemo, pri čemer se jih ustrezno preoblikuje. Nekaj več o tem bomo zapisali v naslednjem poglavju, v podpoglavju Preoblikovanje podatkov.

2.4.2 »Umazani« podatki

Kakovost podatkov je ključna skrb v sistemih poslovne inteligence. Problem je prisoten v vseh fazah integracije, kritičen pa postane, če ga ne odpravimo do zadnje faze. To je analiza podatkov, kjer lahko »umazani« podatki v negativni luči vplivajo na njihovo interpretacijo in posledično povzročijo napačne ali iracionalne poslovne odločitve. Težave s kakovostjo podatkov lahko povzročijo višje stroške in manjše zadovoljstvo strank (Yang W. Lee, 2009, str. 14).

Zaradi tega je potrebno kakovost podatkov redno spremljati in odpravljati. Izkaže se, da je za zagotavljanje kakovosti najbolje, da razvijalci in poslovni uporabniki sodelujejo ter s tem zagotavljajo kakovost podatkov v vseh fazah procesa, pri čemer je priporočljivo, da se določijo naloge in odgovornosti že takoj na začetku procesa (U.S. Department of Transportation, 2010, str. 31).

2.4.3 Količina podatkov

Značilen izziv integracije podatkov, s katerim se srečujejo podjetja, je nepričakovana potreba po dodatnih zmogljivostih in kapacitetah pri obdelavi podatkov v okolju podatkovnega skladiščenja, pri čemer ključno vlogo igra razširljivost (U.S. Department of Transportation, 2010, str. 32). Zgodi se namreč lahko, da stroški zaradi potrebe po razširitvi podatkovnega skladišča razvijalcem hitro uidejo izpod nadzora, kar pa celotno dodano vrednost podatkovnega skladišča izniči.

Podjetja ta problem rešujejo tako, da pri razvoju podatkovnega skladiščenja že načrtujejo t.i. alternativne shrambe v primerih, kjer lahko pride do velike rasti v velikosti, s čimer lahko znižajo stroške. Dobra plat tega problema pa je, da cena shrambe (cena prostora na GB) z leti vztrajno pada zaradi razvoja tehnologije, tako se ta strošek shrambe postopoma znižuje.

2.4.4 Nepričakovani stroški

Nepredvidljivi stroški so vedno prisotni pri zahtevnejših in dolgotrajnejših projektih. Pri projektu integracije podatkov v podatkovno skladišče to še kako drži, saj je težko točno predvideti, kako se bo projekt odvijal. Kaj hitro se namreč lahko projekt zaplete in zahteva več sredstev, kot je bilo prvotno načrtovano. Nepričakovani stroški se lahko pojavijo na različnih področjih (U.S. Department of Transportation, 2010, str. 32):

- stroški dela v vseh fazah projekta (načrtovanje, razvoj, pridobivanje podatkov, ...),

- nakup programske in strojne opreme,
- nepričakovane spremembe in razvoj tehnologije,
- direktni stroški hrambe podatkov in vzdrževanja.

Do težav lahko pride tudi zaradi potreb po dodatnem osebju ali zaradi večjih zahtev in omejitev projekta, kot je bilo sprva predvideno. Tudi samo poslovno okolje je danes vse bolj nepredvidljivo, kar lahko botruje k nepričakovanim spremembam in posredno povzroči dodatne stroške.

Da bi se nepričakovanim stroškom čim bolj izognili, moramo imeti realističen pristop pri načrtovanju in ocenjevanju stroškov projekta, kar zahteva izkušeno vodjo projekta. Pri načrtovanju je potrebno upoštevati vse dejavniki, ki bi lahko imeli večji vpliv na celotne stroške.

Pomembno pa je gledati tudi na dolgi rok, ne samo na trenutne stroške pri izvedbi začetnega procesa. Dobro zasnovan in razvit sistem, ki bo ustrezal končnim uporabnikom bo dlje časa učinkovit in bo potreboval manj vzdrževanja. Med tem bo slabše zasnovan sistem potreboval bolj redne popravke in izboljšave, kar lahko na dolgi rok prinese večje stroške, kot če bi nekaj več sredstev porabili za kakovosten razvoj že na začetku.

2.4.5 Pomanjkljivo znanje osebja

Različna podjetja vse bolj integrirajo podatke za namene analitike, zaradi česar je na voljo vse več kadra, ki ima izkušnje na tem področju. Kljub temu, pa tudi tem pogosto primanjkuje ustreznega znanja in veščin na področju celovitega vodenja projekta integracije podatkov (U.S. Department of Transportation, 2010, str. 35). Poleg strokovnega znanja in izkušenj so pogosti problemi tudi, kadar izvajalci dobro ne poznajo podatkov, njihove strukture ali sistemov, s katerimi imajo opravka.

Na podlagi slednjih dejstev lahko sklepamo, da je za uspešno izvedbo projekta integracije podatkov priporočljivo, če ne skoraj nujno, da imata vsaj projektni vodja in vodja razvoja izkušnje iz tega področja ter ustrezna znanja. Le tako bo končna rešitev modularna in razširljiva, hkrati pa učinkovita za dlje časa.

2.4.6 Pomanjkanje sodelovanja osebja

Pomanjkanje sodelovanja med zaposlenimi je pereč problem na skoraj vseh področjih, nič drugače ni tudi pri procesu integracije podatkov. Predvsem je problem izrazit, kadar razvijalci ne sodelujejo s poslovnimi uporabniki pri zajemu zahtev in obratno. Posledica tega je rešitev, ki slabo izpolnjuje potrebe poslovnih uporabnikov, torej je neučinkovita in nesmiselna iz poslovnega vidika.

Razlogi za slabo sodelovanje zaposlenih so lahko različni, pogosto je so posledica konfliktnih medsebojnih odnosov, pomanjkanja zaupanja, slabe komunikacije ali neprimernih delovnih pogojev v podjetju.

Obstaja več priporočil, ki obravnavajo kadrovske problematiko integracije projekta (U.S. Department of Transportation, 2010, str. 34):

- Vrhnji management naj bo aktivno vključen v projekt, pri čemer naj bo njegova primarna skrb strateški načrt integracije in zagotavljanje sredstev za nemoteno izvedbo projekta.
- Vsi zaposleni, ne glede na fazo razvoja in oddelek, morajo sodelovati med seboj v ključnih fazah, kot sta zajem zahtev in načrtovanje.
- Sprotno praktično učenje udeležencev je priporočljivo in bolj enostavno, kot kasnejše izobraževanje, saj neposredno upošteva zmožnosti in omejitve sistema.
- Velik del možnosti za uspešno izvedbo projekta je odvisen od razumevanja zaposlenih, zato so dobrodošli komunikativni povezovalci, ki znajo ljudem prisluhniti, jih razumeti in motivirati.

2.4.7 Napake zaradi staranja podatkov

Velik izziv pri integraciji podatkov predstavljajo podatki, ki se s časom spreminjajo. Problem je še posebno izrazit, če podatki prihajajo iz različnih virov in tako ni jasno, kateri so verodostojni. Kritičnost zaznavanja sprememb v podatkih je odvisna od tega za kakšne spremembe gre. V določenih primerih je lahko sprememb malo ali pa niso zanimive, ponekod pa jih je veliko in jih je treba nujno zaznati. Nesprejemljivo je recimo, če bi za podatke o določeni stranki ne-zazledili spremembe podatkov (npr. naslov) in tako upoštevali stare.

Nekatera podatkovna skladišča morajo slediti spremembam, za nekatera pa to ni nujno potrebno. Možnost sledenj spremembam je več, izbira primerne pristopa pa je odvisna od potreb.

2.5 Pridobivanje podatkov

Predpostavimo projekt poslovne inteligence in vzpostavitev podatkovnega skladišča, pri čemer so cilji in zahteve jasno opredeljene. V prvem koraku procesu integracije podatkov je torej potrebno identificirati potencialne vire in analizirati podatke, ali so primerni za uporabo. Če so, jih je v nadaljevanju potrebno izluščiti iz izvornih sistemov in jih pripeljati do okolja za pripravo podatkov, kjer se nadaljuje drugi korak procesa ETL.

2.5.1 Podatkovni viri

Včasih so skoraj vsi podatki v podatkovno skladišče prihajali le iz operativnih sistemov. Danes pa podjetja integrirajo tudi podatke iz drugih virov. To so lahko različni dokumenti in datoteke (npr. CSV, JSON, XML), spletne strani, socialna omrežja, zvočne in video vsebine, ipd. Podatki izvirajo iz različnih sistemov in okolij, pri čemer ločimo podatke, ki prihajajo iz internih podatkovnih baz ter podatke, ki prihajajo iz drugih, zunanjih virov.

Bistvena razlika med njimi je, da so prvi že strukturirani, saj prihajajo iz poslovnih sistemov in imajo neko predpisano obliko, četudi se ta še ne ujema z našimi poslovnimi potrebami. Zunanji podatki pa so nestrukturirani (ang. unstructured data), zato veliko vlogo pri njihovem razumevanju igrajo meta podatki. Pri uporabi in shranjevanju zunanjih nestrukturiranih podatkih podjetja pogosto naletijo na probleme. Najpogostejši so (Inmon, 2005, str. 268):

- **Razpoložljivost:** za razliko od podatkov pridobljenih znotraj podjetja, za zunanje podatke ne moremo vedeti, kdaj se bodo pojavili. Zaradi tega je potrebno takšne vire nadzorovati, da se lahko podatke pravočasno zajame.
- **Nekonsistentnost:** pred uvozom v podatkovno skladišče, je kot že omenjeno, potrebno podatke ustrezno preoblikovati. Nestrukturirani podatki ta postopek dodatno otežujejo, saj so različnih formatov in oblik, njihove domene pa so nepoznane.
- **Nepredvidljivost:** zunanji podatki lahko izvirajo praktično iz kateregakoli sistema, ob kateremkoli času.

Zvočne in video vsebine so tipične predstavnice nestrukturiranih podatkov. Ti se lahko hranijo v podatkovnih skladiščih, pogosto pa to zaradi velike količine predstavlja prevelik strošek. Podjetja jih zato hranijo v namenskih podatkovnih bazah. Najpogostejše gre za t.i. »near-line« podatkovne baze, ki predstavljajo kompromis med »on-line« podatkovnimi bazami, ki omogočajo neposreden in hiter dostop ter arhivskimi bazami. Taka arhitektura je lahko stroškovno bolj učinkovita, a hkrati še vedno zagotavlja zadostno fleksibilnost za potrebe poslovne analitike. Služi lahko kot dodatek podatkovnemu skladišču.

2.5.2 Profiliranje podatkov

Tehnično analizo, s katero opišemo vsebino, konsistentnost in strukturo podatkov imenujemo profiliranje (Kimball, Ross, Thornthwaite, Mundy & Becker, 2013, str. 450).

Nekoliko širše je pojem opisal Soares, in sicer kot: »proces razumevanja podatkov v sistemu kjer se nahajajo in v povezavi z drugimi sistemi« (2015, str. 56). Gre za enega izmed ključnih korakov v procesu izboljševanja kvalitete podatkov v sistemih poslovne inteligence. Poslučuje se opisnih statističnih metod, s pomočjo katerih zagotavlja pomembne informacije, kot so: podatkovni tip, dolžina, razpon vrednosti, frekvenca, varianca, edinstvene vrednosti, prazne t.i. »null« vrednosti, ponavljajoči se vzorci v besedilu, in druge.

Profiliranje podatkov se uporablja za različne namene, na primer: za optimizacijo baznih poizvedb, čiščenje podatkov, podatkovno analitiko, povratni inženiringa podatkovnih baz in integracijo podatkov. Kimball navaja, da (uspešna) podjetja profiliranje podatkov v zahtevnem procesu integracije podatkov izvajajo ne le enkrat, temveč večkrat (2013, str. 451). Prvič je profiliranje priporočeno izvesti že zelo na začetku razvojnega procesa, in sicer na podatkih, ki jih nameravamo uporabiti kot vir. S takšno analizo lahko dovolj zgodaj ugotovimo, ali so ti sploh primerni za nadaljnjo integracijo.

Poglobljeno profiliranje se pogosto izvaja tudi v nadaljevanju procesa, in sicer pri razvoju (več)dimenzionalnega modela podatkovnega skladišča. Na podlagi rezultatov profiliranja se je lažje odločiti, kateri podatki so primerni za uporabo in kako jih ustrezno preoblikovati. Poleg tega, pa se s pomočjo profiliranja preverja ustreznost že integriranih podatkov v podatkovnem skladišču.

Dejstvo je, da težave povezane z izvornimi podatki negativno vplivajo na izvedbo projekta, zato je pomembno, da se jih čimprej odkrije in razreši (Reeve, 2013, str. 30). Prednosti uporabe profiliranja podatkov so torej: višja kvaliteta podatkov, boljše razumevanje podatkov in skrajšana izvedba projekta.

Obstajajo napredna orodja, ki so namenjena profiliranju podatkov in omogočajo učinkovite analize ter so sposobna avtomatizirati številna opravila v tem procesu. Toda v številnih primerih ta niso potrebna, saj si lahko pomagamo že s preprostimi baznimi poizvedbami. Primer enostavnega profiliranja podatkov je SQL stavek: »SELECT DISTINCT«. Ta nam vrne razpon edinstvenih vrednosti določenega polja v izbrani tabeli.

V nadaljevanju si bomo pogledali nekaj najpogostejših aktivnosti in vidikov profiliranja podatkov.

Analiza stolpcev

Najbolj osnovna oblika profiliranja podatkov je analiza stolpcev oziroma atributov, kjer se analizirajo njihove vrednosti. Cilj je odkriti prave meta podatke in morebitne probleme s kvaliteto. Rezultati takšne analize zagotavljajo boljše razumevanje pomena posameznih stolpcev in podatkov. Tipične aktivnosti te analize zadevajo sledeče vidike podatkov (Loshin, 2012, str. 113):

- **Razpon vrednosti:** na podlagi tega se sklepa ali vrednosti ustrezajo vrsti podatka oziroma podatkovnemu tipu.
- **Redkost (ang. sparseness):** stopnja redkosti kaže na pomembnost atributa. Če ta vsebuje malo vrednosti, ali pa so skoraj vse med seboj enake, lahko to pomeni da atribut nima posebnega pomena, ali pa je zelo pomemben. To odločitev nosi analitik.
- **Vrednotenje formata:** uporabno je poiskati morebitne vzorce podatkov, na podlagi katerih lahko določimo poslovna pravila. Primer je recimo telefonska številka, davčna številka, poštna številka, ipd.

- **Kardinalnost:** nam pove število edinstvenih vrednostih atributa, kar je zelo uporabno iz stališča pravilnosti podatkov.
- **Distribucija frekvence:** nam pove kolikokrat se posamezna vrednost atributa pojavi v zbirki podatkov. Na podlagi tega lahko sklepamo, da imajo nekatere vrednosti večji pomen kot druge. Prav tako jo lahko uporabimo za zaznavo napak v podatkih, kjer bi pričakovali enakomerno razporejenost podatkov, pa vidimo da temu ni tako.
- **Manjkajoče vrednosti:** so manjkajoči podatki in prazne t.i. »null« vrednosti. Analitiku je pomembna informacija, ali je prav, da so podatki ničelni, ali gre morebiti res za manjkajoče vrednosti, katere bi bilo potrebno dopolniti.
- **Prepoznavna abstraktnih tipov:** zaznava podatkovnih tipov na podlagi vrednosti atributa. Podobno, kot pri vrednotenju formata, le da gre v tem primeru za kompleksnejše podatke.
- **Preobremenitev:** na podlagi preverjanja vrednosti atributa v različnih domenah, poizkuša analiza ugotoviti, ali posamezen atribut hrani več kot le en podatek. V primeru, da je temu tako, je to lahko znak za uvedbo novega atributa.

Analiza relacij

Analiza relacij, ki je v angleščini znana tudi kot »Cross-domain analysis«, se osredotoča na povezave med podatki. Cilj analize je prepoznati povezave med podatki v različnih tabelah, in odvisnosti med različnimi atributi. Analizo relacij lahko razdelimo na analizo domene, analizo funkcionalnih odvisnosti in analizo ključev (Loshin, 2012, str. 120).

Analiza domene, kot že ime pove, išče domene vrednosti atributov med različnimi tabelami in prepoznavna reference na njih. O domeni vrednosti lahko sklepamo recimo takrat, ko ima atribut relativno majhno zalogo vrednosti, ki jih lahko zavzame, ali pa so vrednosti atributa intuitivno porazdeljene v kontekstu vsebine. Na podlagi analize vseh domen lahko sklepamo, ali vrednosti posameznih atributov pripadajo znani domeni, več domenam, ali nobeni.

Analiza funkcionalne odvisnosti analizira stopnjo medsebojne odvisnosti dveh ali več atributov. To najlažje ponazorimo s primerom: atribut »Posta«, ki mora biti funkcionalno odvisen od Atributa »Postna_številka«. Torej, če ima slednji vrednost »1000«, mora imeti atribut »Posta« vrednost »Ljubljana«, ipd. Ta analiza torej pomaga odkriti morebitne nepravilne povezave in odvisnosti v podatkih. Odkrite nepravilnosti je potem potrebno seveda popraviti.

Uporabna je tudi analiza ključev. Ključ je v kontekstu podatkovnih baz atribut, lahko tudi več atributov, s katerimi lahko identificiramo posamezen zapis v tabeli. Primarni ključ je takšen, ki enolično identificira posamezen zapis in je edinstven v posamezni tabeli. Tuji ključ pa je ključ, ki se uporablja za identifikacijo zapisa v drugi tabeli.

Bistvo te analize je, da prepozna obstoječe povezave s ključi, še bolj pa, da zaznava neveljavne povezave, ki kršijo integriteto podatkov. Na primer, kadar šifra izdelka v tabeli prodaje ne obstaja več v aktivni tabeli izdelkov, se vrednost tujega ključa ne pojavi v zapisih ciljne tabele.

2.5.3 Priprava podatkov

Vmesnemu okolju integracije podatkov, ki se nahaja med izvornimi operativnimi sistemi in ciljnim sistemi v novem okolju (največkrat podatkovno ali področno skladišče) rečemo pripravljalno okolje oziroma »staging area«. Gre torej za začasno, prehodno okolje, v katerem se hranijo in obdelujejo podatki med ETL procesom.

Pripravljalno okolje je lahko realizirano v obliki tabel relacijske podatkovne baze, tekstovnih ali drugih datotek (kot je XML). Primarni cilj okolja je povečanje učinkovitosti ETL procesa s tem, da zagotavlja primerno okolje za izvedbo operacij za preoblikovanje podatkov. »Staging« okolje omogoča tudi sledenje spremembam, kateri podatki so bili pridobljeni in kateri posredovani naprej. Prednost je tudi, da za integracijo podatkov izvorni in ciljni sistem ne rabita obdelati podatkov istočasno, temveč se lahko koraka opravita ločeno (Reeve, 2013, str. 31).

2.5.4 Osveževanje podatkovnega skladišča

Pri uvajanju podatkovnega skladišča se vanj nalagajo vsi zajeti podatki od določenega časovnega obdobja naprej. Zaradi običajno velike količine podatkov je postopek drag in zahteven. Zaradi zgoraj omenjenih razlogov je smiselno, da se pri nadaljnjem osveževanju podatkov v podatkovno skladišče prenašajo le še relevantne spremembe.

Ta koncept prenosa spremenjenih podatkov se v angleščini imenuje »change data capture«. Ideja, ki stoji za tem je preprosta: prenesejo naj se le podatki, ki so se spremenili od zadnjega polnjenja. Čeprav se to sliši enostavno, v resnici ni. Ker so podatkovni viri najpogostejše heterogeni, je potrebno za vsak vir posebej pripraviti strategijo zajema podatkov. Postopoma se je razvilo več načinov, s pomočjo katerih lahko zajamemo spremenjene podatke. Izbira najprimernejšega pa je odvisna od specifične situacije in zahtev podjetja (Kimball, Ross, Thornthwaite, Mundy & Becker, 2013, str. 452).

Časovni izvlečki

Nekateri operativni informacijski sistemi imajo v podatkovnih modelih predvidena polja, ki hranijo podatke o datumu nastanka in spremembe posameznega zapisa v podatkovni bazi. S pomočjo teh dveh podatkov si lahko pomagamo pri zajemanju sprememb. Čeprav je pristop enostaven za izvedbo, pa ni vedno primeren.

Problem lahko nastane, če do podatkov v podatkovni bazi dostopajo tudi druge programi in skripte, ki lahko spreminjajo podatke in pri tem zaobidejo sprožilce (ang. triggers), ki posodablajo datumska atributa. Preveriti je priporočljivo tudi, kaj se zgodi v primeru brisanja zapisov, saj različni sistemi dogodek različno obravnavajo. Tretja negativna lastnost pristopa je, da ni uporaben, če obstajajo pomanjkljivi podatki. Torej če datumska polja vsebujejo prazne vrednosti (NULL).

Drug možen pristop zajemanja sprememb se prav tako zanaša na časovno komponento podatkov. Vzemimo primer, da želimo implementirati dnevno osveževanje podatkovnega skladišča. Pripravimo bazno poizvedbo, v kateri zajamemo podatke, ki imajo datum spremembe oziroma nastanka zapisa mlajši od enega dneva.

Postopek je preprost, vendar pomanjkljiv in nezanesljiv. Kimball navaja, da je omenjeni pristop ena izmed najpogostejših napak, ki jo podjetja naredijo v ETL procesu (2013, str. 452). Pristop je nezanesljiv iz preprostega razloga. Če recimo odpove sistem, ki je vir podatkov in so le-ti v času osveževanja nedostopni, to pomeni da jih v naslednjem ciklu osveževanja (recimo naslednjo noč) poizvedba ne bo zajela. Posledično nekateri podatki ne bodo nikoli prispeli v podatkovno skladišče.

Možna izboljšava tega pristopa, ki ta problem rešuje je, da se pri zajemu podatkov namesto datuma upošteva primarni ključ. Tako se prenesejo le tisti podatke, ki jih v podatkovnem skladišču še ni. Primer takšne SQL poizvedbe:

```
SELECT *  
FROM izvorna_tabela  
WHERE id NOT IN (SELECT id FROM ciljna_tabela)
```

Celovito preverjanje razlik

Celovito preverjanje razlik je drug pristop zajemanja sprememb, ki naivno preverja vse »včerajšnje« zapise v podatkovnem skladišču, ena po ena, z »današnjimi« podatki, da bi odkril spremembe. Pozitivna lastnost takšnega pristopa je zanesljivost in temeljitost. Slabost pa je seveda učinkovitost, saj je takšna primerjava lahko zelo zahtevna in dolgotrajna.

V literaturi zasledimo, da je priporočljivo ta postopek izvesti, ko so podatki še v izvornem sistemu. S tem se izognemo nepotrebnim prenosom prevelike količine podatkov.

Preverjanje baznega dnevnika

Pogosto uporabljena je tudi analiza preverjanja baznega dnevnika oziroma v angleščini »log scraping«. Gre za bolj zapleteno tehnično analizo, pri kateri podatkovna baza na podlagi dnevnških transakcij išče spremembe v podatkih. Pri tem se vedno upošteva stanje podatkovne baze (ang. snapshot) v določeni časovni točki.

Kimball navaja, da ta tehnika običajno služi kot izhod v sili (2013, str. 453), saj lahko analiza zaradi procesorsko zahtevnih procesov upočasni celotno delovanje podatkovne baze, posledično pa tudi tekoče poslovne procese.

2.6 Preoblikovanje podatkov

Proces preoblikovanja izvornih podatkov v ustrezno obliko za ciljne sisteme je lahko zelo zapleten, saj zahteva dobro poznavanje poslovnih zahtev. Kimball je ETL proces nazorno ponazoril s primerom restavracije (Kimball, Ross, Thornthwaite, Mundy & Becker, 2013, str. 23). Podproces preoblikovanja podatkov primerja s kuhinjo, kjer mojstri v ozadju s primerno opremo, znanjem in izkušnjami surove sestavine čarobno pretvarjajo v končne produkte (jedi). Sestavine so v tem primeru seveda surovi - izvorni podatki, pripravljene jedi pa so ustrezno preoblikovani podatki.

Uporabnost profiliranja podatkov, ki smo ga obravnavali v prejšnjem podpoglavju pride tu še posebej do izraza, saj z njegovo pomočjo odkrijemo pomembne meta podatke podatkov in pa napake, ki jih moramo odpraviti pred prenosom podatkov v podatkovno skladišče. Da bi podatke ustrezno preoblikovali, se nad njimi (po potrebi) izvajajo različne operacije, ki lahko vključujejo: čiščenje, filtriranje, potrjevanje in izvajanje drugih poslovnih pravil nad podatki.

Pomembno pri tem je tudi, da se zavedamo, da podjetja želijo preoblikovati podatke čim bolj učinkovito in pri tem minimizirati nepotrebna opravila. Glavna prioriteta mora ostati kvaliteta, integriteta in konsistentnost podatkov. Postopek preoblikovanja se navadno izvaja v pripravljalnem (»stage«) okolju.

Spodaj bomo opisali glavne operacije preoblikovanja podatkov, pri čemer so si številne med seboj podobne in se tudi v praksi prepletajo.

2.6.1 Izbira atributov

Prvo opravilo, s katerim se soočimo še pred preoblikovanjem podatkov, je izbira potrebnih podatkov. Na začetku namreč največkrat pridobimo celotno zbirko podatkov iz izvornih sistemov, tudi tiste attribute, ki jih ne potrebujemo. Z omejitvijo na le potrebne podatke, lahko pospešimo integracijo in povečamo učinkovitosti poizvedb v podatkovnem skladišču, hkrati pa privarčujemo diskovni prostor. Zato menimo, da je smiselno temu koraku dati prednost, in ga izvesti takoj na začetku.

Naprednejši programi za integracijo podatkov omogočajo, da uporabnik sam označi attribute podatkov, ki jih želi integrirati. Če pa integracijo opravljamo ročno, recimo s pomočjo SQL stavkov, pa lahko željene attribute preprosto naštejemo v poizvedbi.

2.6.2 Pretvorba podatkovnih tipov

Podatki so v različnih sistemih različno predstavljeni in shranjeni. Zato jih je treba poenotiti oziroma drugače povedano prevesti na skupno osnovo, pri čemer imamo v mislih tehnični vidik. Primer je pretvorba tekstovnih (»string«) atributov, ki predstavljajo števila v numerična (»integer«) attribute. To je pomembno, saj nad tekstovnimi atributi ne moremo izvajati matematičnih operacij.

Drug primer težav, s katerimi se pogosto srečujemo, so napačno kodirani zapisi zaradi uporabe različnih kodnih tabel med sistemi. Nek sistem lahko recimo uporablja kodiranje Windows-1256, drug pa UTF-8. Če ustreznih pretvorb med njimi pri integraciji podatkov ne opravimo, se lahko zgodi, da se določeni znaki ne shranijo in posledično ne prikažejo pravilno. To je značilen problem integracije podatkov, še posebno če gre za podatke, ki vsebujejo nestandardne ali posebne znake. Primer je lahko kar slovenščina, ki vsebuje šumnike, ki pa jih nekatere kodne tabele ne poznajo.

2.6.3 Čiščenje podatkov in zagotavljanje kakovosti

Čiščenje podatkov (ang. data cleaning ali data cleansing ali scrubbing) zaznava in odpravlja napake ter nekonsistentnosti med podatki z namenom izboljšanja kvalitete. Problemi s kakovostjo podatkov so prisotni že v posameznih podatkovnih virih, kot so podatkovne baze in datoteke. Zato se ob integraciji podatkov iz več virov potreba po čiščenju le-teh drastično poveča. Podatkom, ki vsebujejo napake in so posledično slabše kakovosti v žargonu rečemo »umazani« podatki.

Proces izboljševanja kakovosti podatkov je zahteven, predvsem pa stalen v poslovnem okolju, kjer je podatkov veliko in prihajajo iz različnih virov. Za čim boljše reševanje tega problema se mora podjetje soočati z njim na različnih ravneh poslovanja, pri čemer je za uspeh ključno, da (Kimball, Ross, Thornthwaite, Mundy & Becker, 2013, str. 456):

- se podjetje zaveže h kulturi kakovostnih podatkov,
- izvaja prenovo in optimizacijo procesov na izvedbeni ravni,
- vlaga v okolje pridobivanja podatkov,
- vlaga v spremembe kritičnih poslovnih procesov iz vidika kakovosti podatkov,
- promovira zavednost o kakovosti podatkov zaposlenih,
- promovira sodelovanje med oddelki,
- javno proslavlja uspešnost na področju kakovosti podatkov,
- neprekinjeno meri in izboljšuje kakovost podatkov.

Bolj tehnično gledano je proces sestavljen iz dveh delov. Prvi del se osredotoča na čiščenje obstoječih podatkov in popraviljanje napak v njih, drugi pa na preprečevanje napak v bodoče (LeBlanc, M.Moss, Sarka & Ryan, 2005, str. 333). Na drugi korak ne smemo pozabiti, sicer bomo morali vedno znova popravljati iste napake. Preden spoznamo konkretne prijeme, s

pomočjo katerih lahko izboljšamo kakovost podatkov je pomembno, da razumemo, kako lahko kakovost podatkov sploh zmerimo in ovrednotimo.

Dimenzije kakovosti podatkov

Izraz dimenzija kakovosti podatkov se na področju upravljanja s podatki pogosto uporablja za opis merljivih karakteristik in atributov podatkov, s pomočjo katerih lahko ocenimo njihovo kakovost (DAMA UK Ltd, 2013, str. 3). Ne pozabimo, da izraza ne smemo enačiti z dimenzijo, ki smo jo omenili pri podatkovnem skladiščenju in OLAP analitiki, saj ima drugačen pomen.

Če želimo kakovost podatkov oceniti, moramo torej določiti dimenzije kakovosti in jih ovrednotiti na naših podatkih. Kljub temu, da ne obstajajo splošno opredeljene dimenzije kakovosti, si lahko pri razumevanju težav v podatkih pomagamo z določenimi smernicami. Spodnja tabela (tabela 1) prikazuje 6 osnovnih dimenzij kakovosti, ki nam lahko pomagajo oceniti v kakšnem stanju so podatki.

Tabela 1: 6 osnovnih dimenzij kakovosti podatkov

Dimenzija	Opredelitev	Mera	Primer
1. Popolnost	Stopnja zajetih / hranjenih podatkov glede na vse potencialne podatke. So vsi pomembni podatki na voljo?	Mera manjkajočih praznih vrednosti ali prisotnih ne-praznih vrednosti.	294 od 300 zapisov, ki vsebujejo podatke o strankah, ima vneseno telefonsko številko. Popolnost je $294/300 = 98\%$.
2. Edinstvenost	Noben zapis ne sme imeti duplikatov glede na identifikator - primarni ključ.	Primerjava števila zapisov v podatkih in števila zapisov v resnici.	V šoli je 500 učencev, medtem, ko je v podatkovni bazi 520 zapisov o učencih. Edinstvenost je $500/520 = 96,2\%$.
3. Pravočasnost	Stopnja, do katere podatki predstavljajo dogodek iz resničnega sveta ob določenem času.	Razlika v času.	Študent zagotovi študentskemu referatu svojo posodobljeno telefonsko številko v ponedeljek. Uslužbenec jo v sistem vnese šele v četrtek. Razlika je torej 3 dni.

se nadaljuje

nadaljevanje

Dimenzija	Opredelitev	Mera	Primer
4. Veljavnost	Podatki so pravilni, če zadoščajo opredeljeni strukturi (format, tip podatka, razpon vrednosti).	Primerjava med podatki in metapodatki oziroma dokumentacijo podatka.	Nek spletni obrazec zahteva vnos davčne številke. Če uporabnik vnese neštevilski znak ali dolžina znakov ni enaka 8, obrazec javi napako.
5. Točnost	Stopnja do katere podatki pravilno oziroma točno opisujejo stvar ali dogodek iz resničnega sveta.	Stopnja do katere podatki odsevajo lastnosti stvari ali dogodka iz resničnega sveta.	Podatek o datumu rojstva vnesen v napačnem formatu. Namesto v evropskem »DD/MM/YYYY« v ameriškem formatu »MM/DD/YY«, posledično podatek ni točen.
6. Doslednost	Odsotnost razlike med dvema ali več predstavitevami podatka v primerjavi z opredelitvijo.	Analiza vzorca in / ali frekvenca vrednosti.	Podatek o datumu rojstva študenta je isti in v istem formatu v dveh različnih podatkovnih bazah.

Vir: DAMA UK, 2013, str. 8-12

Razumevanje ključnih dimenzij kakovosti je prvi korak v procesu izboljševanja kakovosti podatkov, saj analitiku in razvijalcu omogoča, da razlikuje pomanjkljivosti v podatkih in jih na ta način s pomočjo različnih tehnik odpravi.

Čiščenje podatkov

»Umazani« podatki so torej heterogene strukture in formata, lahko so pomanjkljivi in nepopolni, pojavljajo se podvojene vrednosti, ipd. Da bi problem rešili in zagotovili točne, dosledne in ne-pomanjkljive podatke je potrebno različne predstavitve in formate prestrukturirati in poenotiti (standardizirati), napačne in odvečne podatke odstraniti, pomanjkljive vrednosti pa ustrezno dopolniti. Pomembno je, da se pred začetkom opredeli jasna pravila preoblikovanja, ki pa naj temeljijo na poslovnih potrebah, ne na tehnologiji (Sarsfield, 2009, str. 112).

V splošnem bi proces čiščenja podatkov lahko razdelili v več faz (Rahm & Dos, 2000, str. 5):

- **Analiza podatkov:** sami meta podatki, ki jih lahko najdemo v shemah so običajno nezadostni za odkritje lastnosti in pomanjkljivosti podatkov, zato je treba podatke dodatno analizirati, da bi ocenili njihovo kakovost. Obstajata dva pristopa analize podatkov, to sta že omenjena podatkovno profiliranje in podatkovno rudarjenje.
- **Opredelitev pravil preoblikovanja:** zapletenost čiščenja podatkov je odvisna od heterogenosti podatkovnih virov in kakovosti podatkov. V tej fazi se identificira korake za preoblikovanje podatkov v ustrezno obliko. Na začetku se praviloma zazna in opredeli ustrezno shemo podatkov za ciljni sistem (npr. podatkovno skladišče), v nadaljevanju pa se odpravlja konkretne težave, kot so recimo podvojenost podatkov, ipd.
- **Verifikacija podatkov:** ustreznost in učinkovitost preoblikovanja podatkov je priporočljivo preveriti in oceniti, da lahko po potrebi proces izboljšamo. Navadno je potrebno več ponovitev cikla, ki zajema analizo, opredelitev pravil in verifikacijo podatkov, saj nekatere napake postajajo s tem vse bolj očitne.
- **Preoblikovanje:** izvedba preoblikovanja podatkov pri nalaganju v podatkovno skladišče ali pri vračanju rezultatov baznih poizvedb.
- **Povratni tok prečiščenih podatkov:** z očiščenimi podatki si lahko pomagamo tudi na operativnem nivoju, zato jih je priporočljivo zamenjati z »umazanimi« podatki v izvornih sistemih. S tem lahko zagotovimo točne podatke tudi v operativnih sistemih, hkrati pa se pri naslednjem osveževanju podatkovnega skladišča izognemo vnovičnemu popravljanju istih napak.

Standardizacija podatkov

Poenotenje oziroma standardizacija različnih formatov na isto osnovo je ključen korak, ki doprinese k večji veljavnosti in točnosti podatkov. Potrebno je izbrati skupen format in nato željene podatke s pomočjo ustreznih preslikav pretvoriti na isto osnovo.

Primer tipične standardizacije je pretvorba različnih datumskih atributov v isti format, recimo: »DD.MM.YYYY«. Drugi značilen primer je odstranjevanje in dodajanje, oziroma poenotenje pripon ali določenih besed (ang. stop words) nizom v tekstovnih polj. Za demonstracijo si zamislimo podatke, ki vsebujejo naslove strank: ulico in hišno številko. Nekateri zapisi vsebujejo na dolgo zapisan, pravilno formiran naslov, nekateri pa krajšano obliko, npr. Dunajska cesta 122 ali Dunajska c. 150. Takšne neskladnosti se lahko odpravi s pomočjo naprednih namenskih orodij, v preprostejših primerih pa lahko kar s SQL poizvedbami. Primer takšne preproste SQL poizvedbe je:

```
UPDATE naslovi
SET ulica = REPLACE(REPLACE(ulica, 'c.', 'cesta')
```

S kompleksnejšimi izrazi, kot so regularni izrazi, je mogoče opraviti tudi kompleksnejše standardizacije podatkov.

Odstranjevanje duplikatov

Duplikati oziroma podvojeni podatki so pogosto posledica združevanja podatkov iz več virov. Vodijo v redundanco podatkov, ki je lahko vzrok za številne težave v podatkovnem skladišču. Če duplikatov pred združitvijo ne zaznamo in odpravimo jih je potrebno odstraniti po združitvi podatkov. Odstranjevanje duplikatov najlažje ponazorimo s preprostim primerom.

Predpostavimo, da imamo tabelo podatkov o potencialnih strankah, ki so bili pridobljeni iz različnih virov. Podatki so precej pomanjkljivi, vendar pa obstajajo naslednja polna polja pri vseh zapisih: »ime«, »priimek«, in »telefonska_stevilka«. Ker vemo, da ima lahko več oseb isto ime in priimek, podatka kot sta EMŠO in davčna številka pa nimamo, si lahko pri identificiranju oseb pomagamo kar s telefonsko številko, ki je dober približek enoličnega identifikatorja osebe. Obstoj duplikatov lahko hitro preverimo s sledečo SQL poizvedbo:

```
SELECT telefonska_stevilka, COUNT(*)  
FROM potencialne_stranke  
GROUP BY telefonska_stevilka  
HAVING COUNT(*) > 1  
ORDER BY 2 DESC
```

Odstranimo pa jih lahko s sledečo:

```
DELETE FROM potencialne_stranke a  
WHERE ROWID > (SELECT MIN (ROWID)  
FROM potencialne_stranke b  
WHERE b.telefonska_stevilka = a.telefonska_stevilka)
```

Izvajanju takšnih opravil je treba nameniti posebno pozornost, saj lahko površno profiliranje napak privede do ne-željenih posledic, ki jih je kasneje dosti težje in dražje odpraviti.

Preverjanje integritete

Tekom integracije je priporočljivo tudi preveriti, da omejitve integritete (ang. integrity constraints) med podatki niso prekršene, saj lahko v nasprotnem primeru kaj kmalu naletimo na težave pri povezovanju podatkov.

Če se na primer osredotočimo na primarni ključ neke tabele podatkov, mora veljati, da v njej ne obstaja več kot en zapis, ki bi imel enako vrednost primarnega ključa. Načeloma to omejitev uveljavlja že sam sistem za upravljanje s podatkovno bazo (SUPB), vendar ni vedno nujno tako (izjeme so lahko recimo starejši ali pomanjkljivo razviti sistemi). Drug podoben problem, s katerim se pogosto srečamo pa je, da recimo v neki tabeli (ali več tabelah) obstaja zapis s tujim ključem, ki pa kot primarni ključ v svoji tabeli ne obstaja. Torej se zapis sklicuje na neobstoječ podatek, kar pomeni da imamo pomanjkljive podatke, ki jih ne moremo uporabiti.

2.6.4 Povezovanje podatkov

Pred uvozom podatkov v podatkovno skladišče se navadno normalizirane podatke iz relacijskih baz do določene mere združi. Upoštevajoč relacije se tabele poveže, po potrebi pa se poleg vključijo tudi podatki iz podpornih šifrantov. Temu postopku povezovanja in združevanja relacijskih tabel pravimo denormalizacija.

Pri povezovanju podatkov si pomagamo s tujimi ključi, pogosto pa moramo generirati tudi nove, t.i. umetne ključe, ki povezujejo zapise med seboj na podlagi poslovnih pravil in domen atributov. Pri tem so zelo pomembni meta podatki odkriti s pomočjo profiliranja podatkov. Umetne ključe je največkrat treba generirati zaradi konflikta primarnih ključev, če so si le-ti med seboj v različnih tabelah preveč podobni.

2.6.5 Izpeljanke

Pogosto pri pripravi podatkov za nalaganje podatkovnega skladišča v skladu s poslovnimi potrebami ustvarimo tudi nove attribute. To so pogosto izpeljanke obstoječih atributov do katerih pridemo s pomočjo matematičnih operacij. V kontekstu OLAP-a tem izpeljankam pravimo tudi mere. Primer izpeljanke, ki bi jo lahko implementirali v tabeli naročil je atribut »zaslužek«. Dobimo ga kot preprosto zmnožek atributov »cena_enote« in »količina«.

2.6.6 Agregacija

Oblika podatkov v podatkovnem skladišču je v veliki meri odvisna od poslovnih potreb in razpoložljivih podatkov. Včasih se podatki pogosto spreminjajo, spet drugič je potreba po zgodovinskih podatkih ali pa je problem velika količina podatkov, ipd. Vsi ti dejavniki vplivajo na končno obliko podatkov v podatkovnem skladišču.

Kadar so podatki za poslovne potrebe bolj podrobni, kot jih potrebujemo, jih lahko združimo in dobimo t.i. agregirane zapise. Temu postopku združevanja več zapisov s pomočjo pravimo agregacija. Agregacijo podatkov dosežemo z združevanjem posameznih zapisov, pri čemer so možnosti realizacije neomejene (Inmon, 2005, str. 124). Nekaj primerov agregacije operativnih podatkov v zapis v podatkovnem skladišču:

- Združenje vrednosti na podlagi matematičnih operacij, kot je seštevanje.
- Zapisi se obdelajo in združijo na podlagi minimalne, največje, povprečne, ipd. vrednosti.
- Zapisi se združijo na podlagi začetnih in končnih vrednosti.
- Združijo se najmlajši in najstarejši zapisi.

Z agregacijo torej dosežemo željen nivo granularnosti. Vzemimo za primer bančne podatke o transakcijah neke stranke. Za vsako transakcijo v operativnem sistemu obstaja po en zapis, z ustrezno agregacijo podatkov pa lahko ustvarimo sumiran zapis, ki povzame vse aktivnosti te stranke v določenem obdobju, recimo enem mesecu. S tem sicer, kot že omenjeno v

podpoglavju Granularnost, izgubimo določene podrobnosti, vendar če to zadošča poslovnim potrebam, lahko s tem prihranimo na prostoru in analitiku olajšamo delo, da mu ne bo potrebno procesirati tako velike količine podatkov.

2.6.7 Revizija informacij in skladnost sistema

Koristne informacije, s pomočjo katerih se lahko izognemo nepredvidljivim in morda na prvi pogled povsem neopaznim napakam, lahko pridobimo s sledljivostjo procesa. To lahko storimo tako, da pred in po preoblikovanju podatkov, izračunamo hitre statistike, kot je recimo število vrstic, stolpcev in tabel. S tem dobimo sliko ali je rezultat preoblikovanja v okviru naših pričakovanj. Pri tem lahko shranimo tudi druge uporabne meta podatke, kot so viri in čas preoblikovanja, ki nam bodo mogoče koristili kasneje.

2.6.8 Upravljanje praznih vrednosti

S praznimi vrednostmi se v podatkih pogosto srečujemo, zato je niso nič nenavadnega, je pa pomembno, da ločimo med vrednostmi, ki so prazne (manjka podatek) in vrednostmi, ki imajo vrednost nič (0). Ker se prazne vrednosti pojavljajo v različnih oblikah, lahko so to systemske »NULL« vrednosti, prazni nizi (»«), ničle (0) ali kaj drugega, je priporočljivo, da se jih pretvori na isto osnovo, saj lahko iz vsebinskega vidika bistveno vplivajo na interpretacijo.

2.7 Nalaganje podatkov

Zadnji korak v procesu same integracije podatkov je nalaganje podatkov v ciljni sistem. To je lahko preprosta datoteka ali večja podatkovna shramba recimo podatkovno skladišče. Vrste polnjenja podatkovnega skladišča so različne, lahko gre za začetno polnjenje, lahko gre za nalaganje le spremenjenih podatkov ali pa vseh podatkov hkrati (osveževanje). Vrsta polnjenja je odvisna od strategije in poslovnih potreb podjetja.

Ključna funkcionalnost tega koraka poleg dejanskega polnjenja podatkov, ki se je ne sme zanemariti, je validacija podatkov. Čeprav naj bi bili v večini primerov podatki ustrezno preoblikovani že pred prihodom v podatkovno skladišče, jih je priporočljivo preveriti. Podatkovno skladišče naj bi torej preverilo omejitve podatkov glede na ciljno podatkovno shemo, kot so: integriteta, obvezna polja, edinstvenost, ipd. To je pomembno zaradi kakovosti podatkov, morebitne napake pa lahko še vedno pravočasno zaznamo in jih odpravimo.

Nalaganje podatkov navadno poteka v določenih intervalih (npr. dnevno), zato mora biti ta del procesa čimbolj učinkovit. Poleg tega ima ta del še eno pomembno funkcijo, to je ohranjanje integritete podatkov. Sistem mora biti načrtovan in razvit tako, da je sposoben ob morebitni sistemski napaki ali celo izpadu elektrike ohraniti in povrniti stanje podatkov.

2.7.1 Učinkovito nalaganje podatkov

Učinkovitost nalaganja podatkov je pomembna zmožnost podatkovnega skladišča, še posebno, kadar je podatkov veliko. Iz tega vidika je nujna tudi dobra razširljivost oziroma skalabilnost podatkovnega skladišča, ki je pogost problem pri integraciji heterogenih podatkov (Kadadi, 2015, str. 110). Nalaganje podatkov lahko poteka tako, da se preko programskega vmesnika dodajajo posamezni zapisi, en po en ali pa vsi zapisi hkrati. Slednji način je običajno hitrejši.

Kadar je količina podatkov velika, je dobro podatke razdeliti v manjše, bolj obvladljive dele, ki se jih lahko obdeluje vzporedno. S takšnim načinom se lahko čas nalaganja podatkov občutno zmanjša. Drug pristop za povečanje učinkovitosti nalaganja pa je zmanjšanje količine podatkov. To se navadno naredi v prejšnjem koraku procesa ETL z združevanjem podatkov v pripravljalnem okolju. Ta korak je v vsakem primeru priporočljivo izvesti, saj ne le pohitri nalaganje, ampak tudi privarčuje diskovni prostor in omogoči hitrejšo obdelovanje podatkov v nadaljevanju.

Faza nalaganja podatkov se ne konča le z dodajanjem podatkov v podatkovno skladišče. Po tem se morajo kreirati še indeksi, kar se običajno izvede po nalaganju podatkov, lahko pa tudi kasneje, da se obremenitev sistema razporedi. Učinkovitost indeksiranja je v precejšni meri odvisna od tehnologije.

2.7.2 Spreminjanje podatkov skozi zgodovino

Pri poročanju in analizah načeloma obstajajo tri možnosti upoštevanja časovne dimenzije. Prva možnost je, da zajeti podatki predstavljajo trenutno stanje. Druga možnost je opis podatkov skozi čas (recimo letošnje leto v primerjavi z lanskim), tretja možnost pa je stanje podatkov v določeni točki v preteklosti (Hillard, 2010, str. 153).

Staranje podatkov je torej pomemben element, ki ga je potrebno upoštevati pri načrtovanju in realizaciji ETL procesa. Pri tem je treba določiti, kako se bodo obravnavale vrednosti atributov, ki se počasi spreminjajo skozi čas. Temu pravimo upravljanje počasi spreminjajočih se dimenzij (ang. Slowly Changing Dimension, v nadaljevanju: SCD). Upravljanje spreminjajočih se dimenzij je možno na različne načine, odvisno od vrste sprememb, ki so zanimive za poslovne uporabnike (Jaklič, 2017, str. 186).

Tip 1 – prepis starih vrednosti

Prvi tip SCD ne predvideva hranjenja sprememb. Novejši podatki pri prenosu v podatkovno skladišče enostavno prepišejo obstoječe zapise. Pri tem je pomembno, da se spremembe prenesejo tudi naprej v vse morebitne vire, ki hranijo star podatek. Pristop je primeren, kadar ni potrebe po ohranjanju zgodovine.

Tip 2 – nov zapis v dimenziji

Drugi tip je standardni pristop, ki zagotavlja zgodovino sprememb v dimenzijah podatkov. Ob vsaki spremembi atributa se obstoječi zapis dimenzije kopira, potem pa se posodobijo vrednosti atributov v novem zapisu. Zato sta pri vsakem novem zapisu potrebna vsaj še dva dodatna datumska atributa, ki opisujeta razpon veljavnosti zapisa in s tem omogočata sledljivost. Z njuno pomočjo lahko ugotovimo, kateri zapis je trenutno aktualen.

Tip 3 – nov atribut

Tretji tip SCD predvideva omejeno zgodovino sprememb. To zagotavlja tako, da hrani staro in novo vrednost v ločenih atributih istega zapisa. S tem se ohrani prejšnja in sedanja vrednost atributa. V praksi se ta pristop ravno zaradi te omejenosti redko uporablja.

Tip 4 – dodajanje mini dimenzije

Četrty tip je podoben tretjemu in temelji na ločitvi določenih atributov dimenzije v samostojne dimenzije. S tem se pridobi sledljivost zgodovine in poveča učinkovitost poizvedb.

Tip 5, 6 in 7

Obstajajo še drugi tipi SCD, ki pa so različne mešanice in hibridi prvih štirih tipov.

Z implementacijo različnih tipov SCD lahko podatkovnem skladišču zagotovimo določeno sledljivost v smislu zgodovine sprememb podatkov. Izbira pristopa je v veliki meri odvisna od poslovnih potreb in razpoložljivih sredstev ter jo je potrebno upoštevati že pri načrtovanju podatkovnega skladišča.

3 PRIPRAVA IN INTEGRACIJA PODATKOV NA PRIMERU IZBRANEGA PODJETJA

V tem poglavju bomo analizirali konkreten primer integracije podatkov na primeru izbrane banke, ki pa želi ostati neimenovana. Nekateri podatki, ki se bodo pojavljali v zapisih in tabelah bodo s tem namenom deloma prikriti in spremenjeni, vendar želimo poudariti, da bosta struktura podatkov in druge informacije, ki v praksi vplivajo na sam proces integracije, ostali nespremenjeni.

Področje bančništva je dober primer, kjer sta poslovna inteligenca in posledično integracija podatkov še kako pomembna. Banke nudijo in tržijo široko paleto produktov in storitev, od osebnih računov, varčevalnih knjižic, kreditov, depozitov, pa do elektronskega bančništva, zavarovanj, ipd. Posledično hranijo, obdelujejo in analizirajo velike količine različnih podatkov na dnevni ravni. Zato potrebujejo ustrezne poslovno inteligenčne rešitve in analitična orodja. Ker je obseg poslovanja zelo širok in raznolik, se bomo pri praktičnem

primeru integracije podatkov osredotočili na podmnožico celotne zbirke podatkov znotraj podjetja. Pod drobnogled bomo vzeli podatke o strankah in njihovih tekočih poslih, kot so: transakcijski računi, krediti, varčevanja, elektronsko bančništvo, ipd.

3.1 Metodologija

Na podlagi vsakodnevnega sodelovanja z bančnimi tržniki, analitiki in drugimi višjimi vodstvenimi kadri v izbrani banki smo zaznali potrebo po integriranih celovitih informacijah o strankah in njihovem poslovanju, ki bi jim pomagale pri analiziranju in segmentaciji strank ter pri sprejemanju pomembnih poslovnih odločitev. Potreba po vpeljavi učinkovitega sistema poslovne inteligence in podatkovnega skladiščenja je bila zaradi pomanjkljivih analitičnih zmožnosti v izbrani banki izražena že v različnih preteklih projektih.

Pri analizi zahtev obravnavanega problema in obstoječega stanja informacijske arhitekture izbrane banke smo si pomagali z razpoložljivimi internimi akti in interno dokumentacijo. Na podlagi dostopa do informacijskih sistemov, podatkovnih baz in samih podatkov smo prepoznali ustrezne podatkovne vire za nadaljnjo integracijo podatkov. Pri tem nam je bilo v veliko pomoč tudi lastno znanje, poznavanje informacijskih sistemov in pridobljene izkušnje z delom, ki so prispevale k boljšemu razumevanju problematike ter k lažji izvedbi same integracije.

3.2 Poslovne potrebe

Vsak projekt mora biti dobro načrtovan, opredeljen in mora imeti določen jasen namen. Izhajati mora iz poslovnih potreb. Pomemben strateški cilj praktično vsakega podjetja je dobra prodaja produktov, za kar je potrebno učinkovito trženje, pogoj za to pa so celovite informacije. Dobro integrirani podatki so namreč obvezni za pridobivanje zanesljivih poročil in izvajanje kakovostnih analiz, ki podjetju omogočajo boljše razumevanje potreb strank ter hkrati boljši pregled nad razpršenostjo posameznih produktov in storitev. Posledično lahko vodstvo bolje nadgrajuje obstoječo ponudbo in pravočasno zazna priložnosti za razvoj novih produktov. Boljša prilagoditev ponudbe in posledično večja konkurenčnost podjetja se odražata v povečanju tržnega deleža in prihodkov.

Potrebi po celovitih in kakovostnih informacijah o poslovanju svojih strank želi izbrana banka zadostiti z vzpostavitvijo sistema poslovne inteligence, ki bo poslovnim uporabnikom omogočala celostni, t.i. 360 stopinjski pogled na stranke. Stranke so povezane z bančnimi računi, ki imajo stanje, limit, plačilne kartice, poleg tega ima stranka lahko sklenjene kredite, varčevanja, ipd. Vsaka vrsta produkta ima posebne attribute, zaradi česar je podatke težko predstaviti na enem mestu. Omogočiti želimo predvsem analize poslovanja strank in njihovo segmentacijo, z integracijo osnovnih podatkov o strankah in njihovih poslih. Zagotovili bomo informacije, kot so: čas poslovanja v letih, število aktivnih bančnih računov, skupno stanje na vseh bančnih računih, skupna višina limitov na bančnih računih, število aktivnih

kreditov, skupen znesek aktivnih kreditov, število aktivnih varčevanj, skupen znesek aktivnih varčevanj, ali je uporabnik spletne oziroma mobilne banke, in druge.

Izziv, s katerim se torej sooča izbrana banka, tako kot tudi številna druga podjetja je, kako podatke kakovostno in učinkovito integrirati, da bodo pripravljeni na nadaljnje analiziranje. Cilj tega poglavja je prikazati in analizirati primer učinkovite priprave in integracije podatkov iz operativnih sistemov v analitično okolje. Poudarek ne bo na končni programski rešitvi in uporabljenih orodjih, temveč želimo analizirati ključne prijeme in izzive pri izvedbi tega procesa.

Rezultat integracije bo osnovna, na kateri bodo lahko bančni analitiki, tržniki in drugi zaposleni lažje segmentirali stranke in učinkovitejše tržili bančne produkte ter s tem prispevali k uresničitvi strateškega cilja podjetja, vodstvo pa bo imelo boljši pregled in nadzor nad trženjem ter prodajo produktov in storitev. Pri integraciji podatkov bomo uporabili smernice, ki smo jih spoznali v prvih dveh poglavjih, predvsem s področja pridobivanja, preoblikovanja in nalaganja podatkov ter zagotavljanja kakovosti in učinkovitosti. Podatke bomo v večji meri pridobili iz operativnih sistemov, polega pa bomo vključili tudi nekatere podatke iz zunanjih virov.

3.3 Analiza informacijske arhitekture

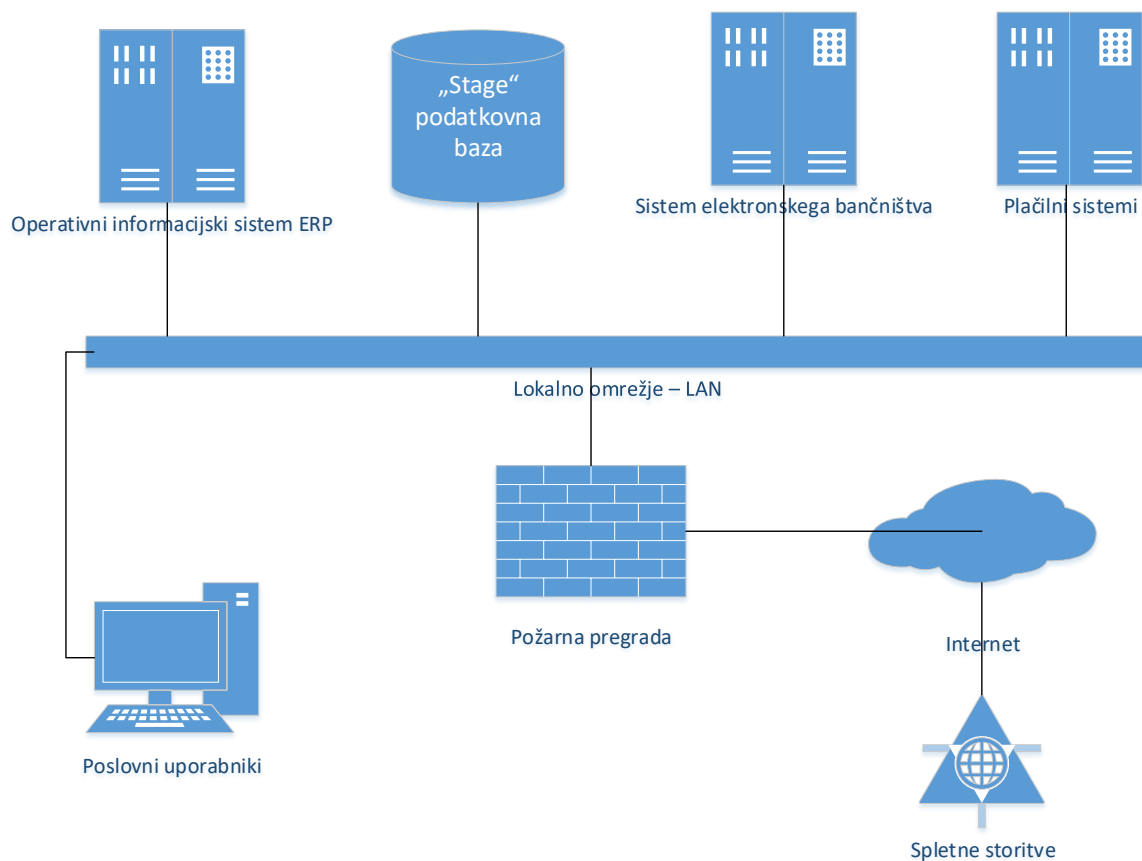
Obstoječa informacijska arhitektura izbrane banke je temelj, ki podpira glavne poslovne procese tekom katerih ob poslovnih dogodkih nastajajo podatki. Pomembno je, da naprednejši poslovni uporabniki, kot so recimo analitiki, dobro poznajo informacijske sisteme, ki jih uporabljajo. Poznavanje podatkov, tako iz strukturnega, kot iz vsebinskega vidika, je osnova za dobre analize. Ravno tako morajo vsebino poznati razvijalci, če želijo vzpostaviti funkcionalno podatkovno skladišče, pri čemer je pomembno tudi, da v veliki meri komunicirajo in sodelujejo z uporabniki sistema.

Informacijske potrebe banke so kompleksne, saj morajo podpirati številne poslovne procese in ob tem zagotavljati visoko stopnjo varnosti in zanesljivosti. Informacijska infrastruktura izbrane banke je sestavljena iz različnih računalniških virov in strojne opreme, ki skupaj zagotavljajo nemoteno delovanje informacijskega sistema kot celote. V okviru integracije podatkov nas zanimajo predvsem posamezni informacijski sistemi in aplikacije, ki predstavljajo morebiten vir podatkov. Slika 5 prikazuje glavne komponente informacijske arhitekture izbrane banke.

Glavni informacijski sistem je celovita bančna aplikacija oziroma ERP (ang. Enterprise Resource Planning, v nadaljevanju: ERP), kot ga bomo imenovali v nadaljevanju. Povezana je s produkcijsko Oracle relacijsko podatkovno bazo, v kateri se hranijo in obdelujejo podatki. Aplikacija obdeluje vse transakcijske podatke, ki so neposredno povezani z bančnim poslovanjem. To so torej podatki o strankah in njihovih računih, podatki o plačilnem prometu, podatki o kreditih, varčevanjih in ostalih bančnih poslih, ipd. Zato bo ERP predstavljal glavni podatkovni vir. Aplikacija je sicer sestavljena iz različnih

programskih modulov, ki se med seboj prepletajo in povezujejo, vsak pa podpira določeno področje poslovanja. Poleg glavnega produkcijskega okolja obstaja tudi ločeno testno okolje s testno aplikacijo in podatki.

Slika 5: Informacijska arhitektura izbrane banke



Vir: Izbrano podjetje (2017).

Druga pomembna komponenta informacijske arhitekture, ki je za potrebe celostnega pregleda strank ravno tako pomemben vir podatkov, je sistem namenjen elektronskem bančništvu. Gre za povsem ločeno Oracle podatkovno bazo in namenski spletni strežnik, na katerem teče spletna banka in storitve za mobilno banko. Ta informacijski (pod)sistem hrani in obdeluje podatke pridobljene v spletni in mobilni banki, ki se deloma sinhronizirajo z bančno aplikacijo.

Tretja komponenta informacijske arhitekture, ki nas zanima, je dodatna podatkovna baza, v katero se prepisujejo podatki iz produkcijske podatkovne baze. Imenujemo jo lahko kar pripravljalna oziroma »stage« podatkovna baza. Gre ravno tako za Oraclov sistem, ki pa je namenjen različnim obdelavam, poizvedbam in poročanju. S tem namenom je ločen od glavnega ERP sistema, ki bi ga v nasprotnem primeru utegnile kompleksne poizvedbe obremeniti do te mere, da bi ovirale delovanje in posledično poslovanje celotne poslovne

mreže. Ta podatkovna baza je primerna za pripravo in integracijo različnih podatkov, podatkovno skladiščenje in izvedbo analiz ter poročanje.

3.4 Analiza podatkovnih virov in podatkov

V skladu z namenom praktičnega dela se bomo pri integraciji podatkov osredotočili na podatkovne vire, ki so relevantni za analiziranje strank in njihovega poslovanja, da bi s tem omogočili integralni pregled in segmentacijo strank. Podatki, ki jih bomo integrirali, izvirajo iz treh virov:

- **Informacijski sistem ERP:** glavna operativna podatkovna baza predstavlja osnoven vir podatkov o strankah in njihovem poslovanju.
- **Informacijski sistem elektronskega bančništva:** drugi vir je namenska podatkovna baza sistema elektronskega bančništva.
- **Spletne storitve:** pri integraciji se bomo oprli na zunanji podatkovni vir, to so spletne storitve izbranega ponudnika, ki zagotavljajo ažurne javno dostopne podatke o podjetjih. Od matičnih podatkov podjetja, pa do bolj podrobnih podatkov, kot so finančni podatki, ipd. Ta vir bomo uporabili predvsem za dopolnjevanje, ažuriranje in preverjanje obstoječih podatkov strank pravnih oseb.

Relevantni podatki glede na opredeljene poslovne potrebe so shranjeni v internih relacijskih podatkovnih bazah, za dopolnitev nekaterih podatkov o podjetjih, pa si bomo pomagali tudi z zunanjim virom.

Večina podatkov se v relacijske podatkovne baze vnaša preko uporabniških aplikacij, kjer se pri shranjevanju s pomočjo programskih poslovnih pravil in omejitev (ang. constraints) na podatkovni tabeli v določeni meri preverjajo: format, obstoj obveznih atributov in druge omejitve. Kljub temu, pa se zaradi številnih dejavnikov, kot so: programski hrošči, pomanjkljivosti aplikacij, dograjevanje aplikacije skozi čas, površnosti uporabnikov in drugih, v podatkih vseeno lahko pojavijo napake, ki jih bomo kasneje analizirali s pomočjo profiliranja podatkov.

Spodaj bomo poizkusili po sklopih, strnjeno in razumljivo opisati, kje in kako so shranjeni posamezni podatki, da bi bolje razumeli osnovno podatkovno strukturo in najpomembnejše omejitve podatkov.

3.4.1 Matični podatki strank

Osnova za celostni pregled strank so matični podatki, zato jih bomo integrirali najprej. Pri tem bo potrebno tako kot pri večini drugih podatkov v nadaljevanju ločiti ali imamo opravka s fizično ali s pravno osebo. Iz obsežnega seznama obstoječih podatkov o strankah, nas v sklopu primera zanimajo osnovni podatki, ki so pomembni iz vidika trženja in segmentacije:

- id stranke,
- ime in priimek (če gre za fizično osebo) ali naziv (če gre za podjetje),
- EMŠO ali matična številka,
- davčna številka,
- datum rojstva oziroma datum vpisa v register poslovnih subjektov,
- ulica,
- kraj,
- pošta,
- elektronska pošta,
- telefonska številka,
- vrsta stranke (fizična ali pravna oseba),
- standardna klasifikacija dejavnosti (če gre za pravno osebo),
- status stranke,
- datum pridružitve
- in poslovna enota, v katero spada stranka.

Zgoraj omenjeni podatki se v osnovi nahajajo v operativni podatkovni bazi glavnega ERP sistema, v tabeli »STRANKE«. Zapis v tabeli predstavlja posamezno stranko, pri čemer je edinstveni primarni ključ atribut »ID«. Ker so v isti tabeli shranjene fizične in pravne osebe, primarni ključ ne more biti davčna številka, saj je lahko na primer ena stranka vnesena kot fizična oseba, hkrati pa posebej vnesena še kot pravna oseba (S.P.), z isto davčno številko.

Ostale pomembnejše tuje ključe bomo pri izgradnji dimenzijskih tabel nadomestili oziroma dopolnili s povezanimi opisnimi atributi iz tabele šifrantov. Primer: atribut vrsta stranke, je v bistvu številski tuji ključ, ki sam po sebi ne pove nič, zato ga moramo povezati s tabelo šifrantov, da dobimo več informacij.

Pri integraciji bomo morali upoštevati tudi dejstvo, da se nekateri podatki podvajajo, in da jih je mogoče zaslediti v različnih virih. Na primer, elektronska pošta pri vnosu stranke v ERP sistemu ni obvezen podatek, je pa zahtevan pri aktivaciji uporabnika spletne banke. Zato lahko v primeru, ko stranka nima vnesenega elektronskega naslova v tabeli »STRANKE«, hkrati pa ima omogočeno spletno banko, podatek pridobimo iz podatkovne baze elektronskega bančništva. Podobnih primerov je več, pomembnejše bomo izpostavili v nadaljevanju.

Pri dopolnjevanju in ažuriranju določenih podatkov, kot so ulica, pošta in naslov, si bomo pomagali z že prej omenjenim zunanjim podatkovnim virom. Uporabili bomo spletno storitev, ki vrača najnovejše matične podatke iskanega podjetja, s čimer bomo poizkusili zagotoviti večjo točnost in kakovost podatkov.

3.4.2 Podatki o bančnih računih

Transakcijski bančni račun je najosnovnejši bančni produkt, ki ga ima večina strank. Strankam omogoča sodobno poslovanje, od prejetanja dohodkov in drugih nakazil, brezgotovinskega plačevanja, dvigovanja gotovine na bankomatih, do uporabe trajnih nalogov, ipd. Posamezna stranka ima lahko nič ali več bančnih računov. Zaradi različne narave poslovanja so podatki o bančnih računih fizičnih in pravnih oseb shranjeni v ločenih tabelah v operativni podatkovni bazi: »RACUNI_FIZICNE« in »RACUNI_PRAVNE«.

Zapis v posamezni tabeli predstavlja bančni račun, ki je s stranko iz tabele »STRANKE« povezan preko tujega ključa »ID_STRANKE«. Pri integraciji bomo z namenom poenostavitve tabeli združili v eno, »RACUNI«. Ker imata obe tabeli isto strukturo, med vrstami računov (fizične / pravne) pa ločuje atribut »VRSTA_RACUNA«, združitev ne bi smela biti težavna. Iz omenjenih tabel bomo vzeli naslednje attribute:

- id računa,
- id stranke,
- vrsta računa,
- skupina računa,
- datum otvoritve računa,
- datum ukinitve računa,
- blokada računa.

Pri integraciji bomo vključili tudi nekatere druge podatke, ki se neposredno navezujejo na bančni račun in na podlagi katerih bomo lahko pridobili koristne informacije, kot so: stanje na računu, višina tekočega in izrednega limita na računu (če obstajata), povprečna višina mesečnih prilivov, ipd. Podatki o limitih bančnih računov se nahajajo v ločenih tabelah »TEKOCI_LIMITI« in »IZREDNI_LIMITI«.

Podatki o transakcijskem prometu vseh strank se nahajajo v tabeli »PROMET«, ki vsebuje po en zapis na posamezno transakcijo, kar pomeni, da je količina zapisov v tej tabeli zelo velika. Zato bomo v nadaljevanju pri integraciji te podatkovne tabele zapise združili oziroma grupirali po dnevih, torej zrno v tabeli ne bo več posamezna transakcija, temveč vsota transakcij v posameznem dnevu. S tem bomo občutno zmanjšali količino podatkov, hkrati pa nam bodo ti še vedno omogočali pridobitev zgoraj naštetih informacij.

3.4.3 Podatki o kreditnih poslih

Klasični bančni produkti, ki jih bomo tudi vključili k celostnemu pregledu strank so krediti. Osnoven vir podatkov bo tudi tu ERP podatkovna baza, vendar so zaradi kompleksne narave posla podatki razpršeni po različnih tabelah. Iz osnovne tabele »KREDITI« bomo vzeli naslednje attribute:

- id kredita,
- id stranke,
- vrsta kredita,
- namen kredita,
- znesek,
- datum sklenitve,
- datum zapadlosti,
- status.

Ostali podatki, kot so: obrestna mera, višina anuitete in zavarovanja kredita se nahajajo v ločenih tabelah, ki jih bo treba ustrezno povezati.

3.4.4 Podatki o varčevanjih

Podatki o varčevanjih strank se nahajajo v tabeli »VARCEVANJA«. Integrirali bomo podobne attribute, kot iz tabele kreditov.

3.4.5 Podporni šifranti

Večina izbranih podatkovnih tabel vsebuje tudi različne meta podatke o strankah in poslih, kot so: skupina, status, vrsta posla, ipd. Ti so v izvornih tabelah predstavljeni s tujimi ključi, ki predstavljajo posamezen zapis iz šifranta. Glavni informacijski sistem v izbranem podjetju šifrante shranjuje v skupni tabeli, imenovani »SIFRANTI«. Tabela vsebuje sledeče attribute:

- id,
- vrsta šifranta,
- šifra postavke,
- naziv postavke,
- aktiven.

Atribut »VRSTA« ločuje med različnimi šifranti, atribut »SIFRA« pa je v tuji ključ, na katerega se sklicujejo povezane tabele. Poleg tega bomo integrirali tudi tabelo »POSLOVNE_ENOTE«, ki hrani podatke o vseh poslovnih enotah izbranega podjetja.

3.4.6 Podatki o elektronskem bančništvu

Podatki o elektronskem poslovanju strank se nahajajo deloma v ERP podatkovni bazi in deloma v podatkovni bazi elektronskega bančništva. S stališča poslovnih potreb nas bo zanimalo katere stranke uporabljajo in katere ne uporabljajo spletne oziroma mobilne banke. Ti podatki so zanimivi predvsem iz vidika trženja. Iz podatkovne baze elektronskega bančništva bomo integrirali podatke iz tabele »EB_UPORABNIKI«, pri čemer bomo vzeli naslednje attribute:

- id uporabnika,
- id stranke,
- elektronski naslov,
- mobilna številka,
- tip storitve (mobilna ali spletna banka),
- datum aktivacije,
- datum ukinitve.

3.5 Predlog rešitve

Integracije podatkov iz identificiranih treh podatkovnih virov se bomo na primeru izbranega podjetja lotili sistematično. Uporabili bomo pristop, ki temelji na najnižji arhitekturni ravni, to je skupna podatkovna shramba oziroma fizična integracija. Ta pristop smo izbrali zato, ker se soočamo s starejšima informacijskima sistemoma in večjo količino podatkov, ki jih želimo iz operativnega okolja prenesti v podatkovno skladišče. S tem želimo zagotoviti ločeno okolje, ki bo vsebovalo že pripravljene podatke za analitike in bo omogočalo večjo učinkovitost analiz. Osveževanje podatkov bo periodično.

Podatke bomo integrirali v posameznih fazah na podlagi ETL procesa. Uporabili bomo »stage« podatkovno bazo, v kateri bomo pridobili in pripravili podatke. V isti podatkovni bazi bomo nato pripravili ločeno podatkovno shemo, ki bo omogočala vpogled v ključne podatke strank z orodji poslovne inteligence. Načrt integracije prikazuje slika 6 in je sestavljen iz štirih korakov:

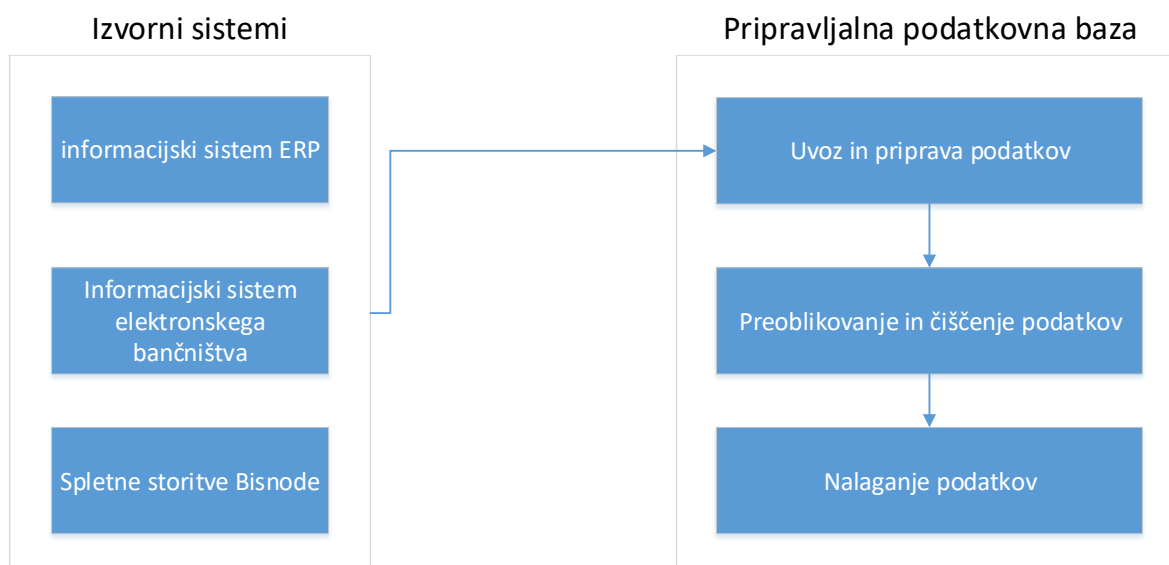
1. uvoz operativnih podatkov v pripravljeno okolje,
2. čiščenje in preoblikovanje podatkov v pripravljalnem okolju,
3. nalaganje podatkov v podatkovno skladišče (in razvoj podatkovne sheme ob prvi integraciji),
4. periodično ponavljanje prvih treh korakov.

3.6 Pridobivanje podatkov

V teoretičnem delu smo ugotovili, da se profiliranje podatkov najpogosteje izvaja večkrat v procesu integracije podatkov. Prvič lahko že na začetku, če želimo pred integracijo preveriti ustreznost podatkovnih virov. V našem primeru tega ne bomo storili, saj so glede na potrebe izbrani podatkovni viri edini možni. Bomo pa analizirali podatke s pomočjo profiliranja v nadaljevanju, da bomo odkrili pomanjkljivosti in napake, ki jih bomo poizkusili tudi odpraviti.

Na začetku želimo podatke pridobiti iz izvornih sistemov. Zato je potrebno vzpostaviti zanesljiv in varen sistem, ki bo sposoben zagotavljati standardiziran prenos podatkov iz različnih virov. Temu pravimo tudi vzpostavitev tehnične interoperabilnosti.

Slika 6: Potek integracije podatkov na primeru izbranega podjetja



Vir: lastno delo.

3.6.1 Pridobivanje podatkov iz Oracle podatkovnih baz

»Stage« podatkovna baza je do operativne ERP podatkovne baze povezana preko t.i. bazne povezave oziroma v angleščini »database linka«. Bazna povezava je enosmerna in končnemu uporabniku omogoča, da s pomočjo SQL poizvedb v lokalni podatkovni bazi enostavno dostopa do podatkov v oddaljeni podatkovni bazi. To je najbolj enostavna metoda za prenašanje podatkov med dvema Oracle podatkovnima bazama, ker omogoča združitev korakov pridobivanja in preoblikovanja podatkov v enega (Oracle, 2017).

Izkoristili jo bomo za prenos podatkov v pripravljalno okolje, tako nam ne bo potrebno izvažati podatkov v datoteke in jih potem ponovno uvažati v ciljno podatkovno bazo. Namesto tega bomo podatke neposredno prepisali iz operativne podatkovne baze. To lahko za posamezno podatkovno tabelo storimo s preprosto SQL poizvedbo, kot je:

```
CREATE TABLE stranke
AS (SELECT * FROM stranke@erp.world);
CREATE UNIQUE INDEX stranke_index1 ON stranke (id);
```

Zgornja SQL stavka naredita polno kopijo tabele »STRANKE« iz ERP podatkovne baze, pri čemer se kreira še indeks na primarnem ključu, ki skrbi za učinkovitejše poizvedbe. Opisani pristop pa ni optimalen, saj moramo, če želimo zagotoviti periodično osveževanje podatkov, razviti programsko skripto, ki bo sledeče prepise izvajala na določen interval.

Boljša alternativa, ki jo bomo uporabili temelji na Oraclovih materializiranih pogledih (ang. materialized views). Materializiran pogled je objekt, ki vsebuje rezultat vnaprej določene

poizvedbe. V primerjavi z ročnim kreiranjem tabel, kot smo ga opisali zgoraj, materializirani pogledi omogočajo tudi avtomatsko osveževanje podatkov. Primer materializiranega pogleda, ki ustvari kopijo tabele »STRANKE« je:

```
CREATE MATERIALIZED VIEW stranke_mv REFRESH START WITH SYSDATE NEXT
TRUNC(SYSDATE + 1) + 2/24 COMPLETE
AS
SELECT * FROM stranke@erp.world;
```

S parametroma: »START WITH SYSDATE« in »NEXT TRUNC(SYSDATE + 1) + 2/24 COMPLETE« sistemu povemo, da naj se podatki v celoti prepisujejo vsak dan, ob drugi uri ponoči.

Na podoben način naredimo materializirane poglede za vse zgoraj naštetе podatkovne tabele, ki jih bomo integrirali. Pri pripravi materializiranega pogleda za tabelo »PROMET«, pa podatke v poizvedbi, kot že opisano zgoraj, združimo po dnevih, s čimer zmanjšamo količino nepotrebnih podatkov. Tudi podatke iz podatkovne baze elektronskega bančništva bomo pridobili na podoben način, le da bomo pri tem uporabili drugo podatkovno povezavo, zato postopka ne bomo posebej opisovali.

3.6.2 Pridobivanje podatkov iz zunanjih virov z uporabo spletnih storitev

Podatke iz zunanjega podatkovnega vira bomo v pripravljalno okolje prenesli s klici spletnih storitev. Te vračajo odgovor v obliki XML datoteke, ki je strukturirana v skladu z opredeljeno WSDL shemo. Prejete podatke bo potrebno ustrezno razčleniti, formatirati in shraniti v podatkovne tabele. Interakcijo s spletnimi storitvami bo izvajala kar »stage« podatkovna baza.

Struktura vrnjenega odgovora je odvisna od storitve, ki jo bomo uporabili. Iz nabora ponujenih storitev bomo v skladu s poslovnimi potrebami uporabili storitev, ki vrača osnovne podatke podjetja, s pomočjo katerih bomo preverili in po potrebi popravili obstoječe podatke iz operativnega sistema. Integracijo teh podatkov bomo izvedli na sledeči način. Razvili bomo bazno proceduro, ki kot obvezen parameter sprejme davčno številko podjetja, s katero potem kliče spletno storitev. Vrnjen odgovor procedura nato shrani v tabelo »ZUNANJI_PODATKI«, ki ima pet atributov:

- vrsta storitve,
- davčna številka podjetja,
- vrnjeni podatki,
- datum poizvedbe,
- status storitve.

Opisan postopek omogoča pridobitev podatkov za posamezno podjetje, zato ga bo potrebno večkrat ponoviti, če bomo želeli pridobiti podatke o vseh pravnih osebah, ki so stranke

izbranega podjetja. To bomo storili s ponavljajočimi klici procedure za vse davčne številke podjetij, za katere želimo podatke.

Ključni korak integracije pa ni klic storitve, temveč razčlemba pridobljenih podatkov, ki so shranjeni v XML obliki v enem samem atributu. S pomočjo Oracle funkcije »extractValue«, ki zna izluščiti posamezne vrednosti iz XML podatkovne strukture, bomo pripravili pogled »MAT_PODATKI_V«. S tem bomo na enostaven način in brez dodatnega prepisovanja ter podvajanja podatkov lahko s pomočjo klasičnih poizvedb dostopali do pridobljenih podatkov. Spodaj je primer izvorne kode Oracle pogleda, ki ustrezno razčlenjuje matične podatke podjetja:

```
CREATE OR REPLACE VIEW mat_podatki_v AS
SELECT
    davcna,
    extractvalue(x.COLUMN_VALUE, '/Item/Maticna') AS maticna,
    extractvalue(x.COLUMN_VALUE, '/Item/Naziv') AS naziv,
    extractvalue(x.COLUMN_VALUE, '/Item/Naselje') AS naselje,
    extractvalue(x.COLUMN_VALUE, '/Item/Ulica') AS ulica,
    extractvalue(x.COLUMN_VALUE, '/Item/UlicaNaziv') AS ulica_naz,
    extractvalue(x.COLUMN_VALUE, '/Item/UlicaHS') AS ulica_hs,
    extractvalue(x.COLUMN_VALUE, '/Item/PostnaSt') AS postna_st,
    extractvalue(x.COLUMN_VALUE, '/Item/PostaNaziv') AS posta_naz,
    TO_DATE(substr(REPLACE(extractvalue(x.COLUMN_VALUE,
'/Item/DatumVpisa'), '-', ''), 1, 8), 'yyyymmdd') AS datum_vpisa,
    extractvalue(x.COLUMN_VALUE, '/Item/Oblika') AS oblika,
    extractvalue(x.COLUMN_VALUE, '/Item/SKIS') AS skis,
    extractvalue(x.COLUMN_VALUE, '/Item/Telefon') AS telefon,
    extractvalue(x.COLUMN_VALUE, '/Item/Email') AS email,
    extractvalue(x.COLUMN_VALUE, '/Item/URL') AS url
FROM zunanji_podatki s,
    TABLE(xmlsequence(
        EXTRACT (
            s.podatki,
            '//GetData_PodjetjeOsnovnoResponse/GetData_PodjetjeOsnovnoResult/
BisnodeWebServiceData/Data/PodjetjeOsnovno/Item',
            'xmlns:soap="http://www.w3.org/2003/05/soap-envelope",
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance",
            xmlns:xsd="http://www.w3.org/2001/XMLSchema',
            xmlns="http://servisi.gvin.com/BisnodeWebService" '
        )
    )) x
WHERE s.vrsta_storitve = 10
    AND status = 1;
```

3.6.3 Osveževanje podatkovnega skladišča

Osveževanje podatkov je zahtevnejše, kadar imamo opravka z večjimi količinami le-teh. V našem primeru to drži, zato je smiselno po začetnem nalaganju podatkovnega skladišča periodično osveževati le še spremenjene podatke. Na podlagi količine in vsebine podatkov

v posameznih tabelah bomo temu primerno prilagodili način in frekvenco osveževanja. V splošnem pa bi radi zadostili potrebi, da se podatki osvežijo vsaj enkrat dnevno.

Kot že opisano zgoraj, bomo podatke iz internih virov v pripravljeno okolje prenašali s pomočjo materializiranih pogledov. Osveževanje bo potekalo ponoči, da s tem ne bomo po nepotrebnem obremenjevali produkcijskih sistemov v delovnem času. Izkoristili bomo možnost t.i. hitrega osveževanja (ang. fast refresh), ki jo ponujajo materializirani pogledi. Gre za osveževanje le spremenjenih podatkov, ki temelji na preverjanju baznih logov. Kljub temu, da je lahko ta proces relativno zahteven za sistem, je kljub temu še vedno bistveno bolj učinkovit, kot če bi vsako noč celotne podatke zbrisali in jih na novo prepisali. To bi bilo smotno le v primeru, če bi dnevno prihajalo do bistvenih sprememb v podatkih, kar pa ne drži.

Bolj zahtevno in časovno potratno je osveževanje podatkov iz zunanjih virov. Zaporedni klici spletnih storitev so relativno počasni in lahko znatno obremenijo tudi omrežje. Če poleg tega upoštevamo še dejstvo, da ti podatki glede na poslovne potrebe izbranega podjetja niso tako pomembni in se ne spreminjajo zelo pogosto, ni potrebno, da je frekvenca osveževanja tako velika. V tem primeru bo zadoščalo osveževanje podatkov le enkrat tedensko. Za vsa podjetja iz tabele strank, se mora ob tem izvesti procedura, ki kliče spletne storitve in prepíše pridobljene podatke v podatkovno bazo. Za razčlenjevanje pridobljenih podatkov bo poskrbel že pripravljen pogled.

3.7 Preoblikovanje podatkov

Pridobljene podatke bomo najprej analizirali, s čimer bomo odkrili morebitne napake in pomanjkljivosti, ki jih bomo poizkusili odpraviti. Nato bomo upoštevajoč relacije, podatke ustrezno združili in jih drugače preoblikovali, da bomo dobili podatkovno shemo, ki ustreza danim poslovnim potrebam. Logični model nastale podatkovne sheme prikazuje slika 7.

3.7.1 Izbira atributov

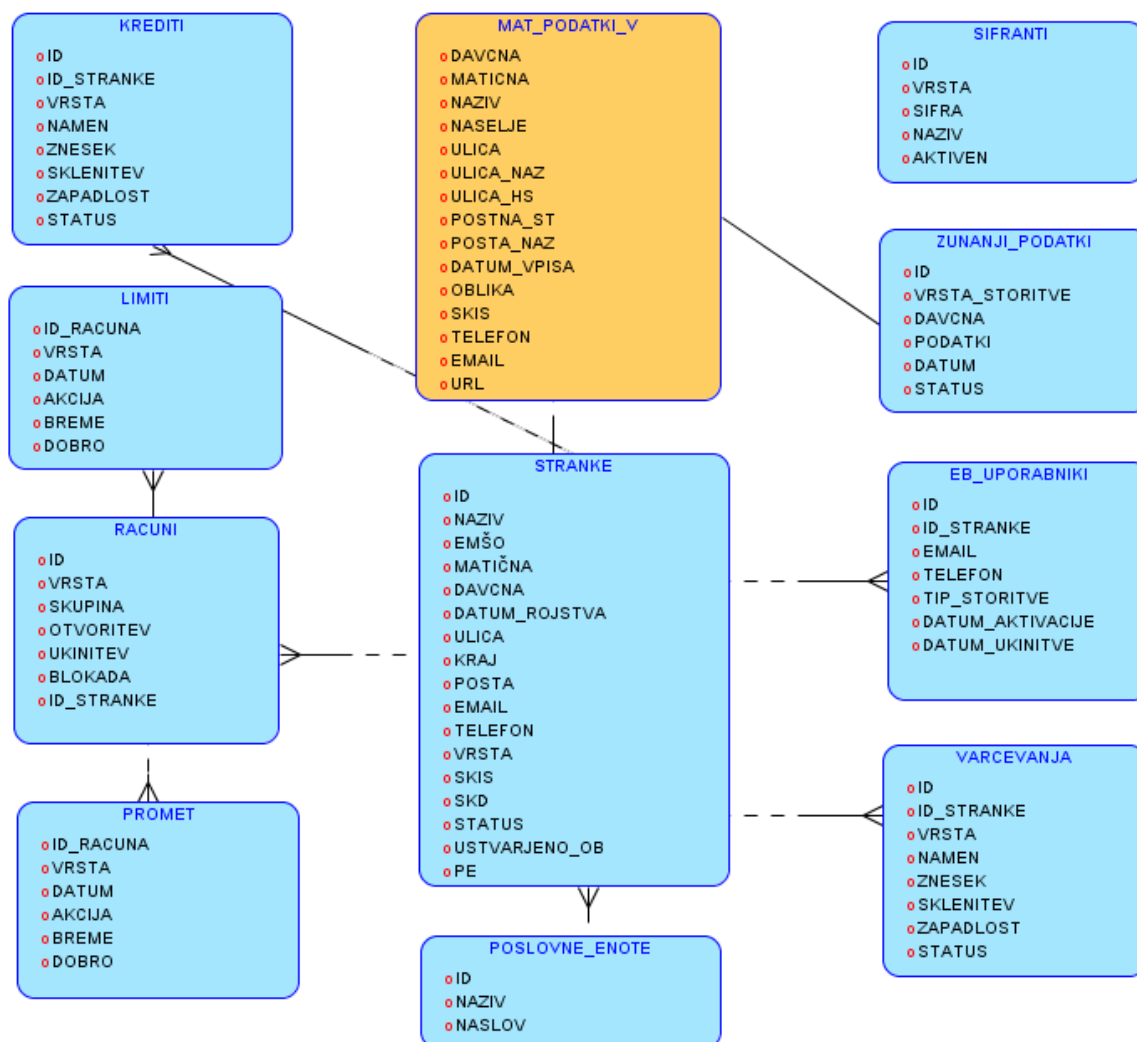
V prvi fazi procesa pridobimo podatke iz izbranih internih podatkovnih tabel in iz zunanjih virov. V skladu s poslovnimi potrebami pa vseh atributov iz zbrane množice podatkov ne bomo potrebovali, zato jih je smiselno iz integracije izločiti že pred nadaljevanjem postopka. Pri analiziranju in pri oblikovanju ciljne podatkovne sheme bomo upoštevali le pomembne attribute, ki smo jih navedli v analizi podatkovnih virov. Vključili bomo tudi različne podporne šifrantne in tvorili dodatne izpeljane attribute oziroma mere.

3.7.2 Pretvorba podatkovnih tipov

Večji del zajetih podatkov izvira iz obeh analiziranih internih podatkovnih virov. Te podatke v pripravljeno okolje pridobimo s pomočjo materializiranih pogledov, ki so sposobni

ohraniti izvirne podatkovne tipe posameznih atributov, zato tu dodatna pretvarjanja in usklajevanja ne bodo potrebna. Prav tako ne bi smelo biti težav s kodiranjem, saj oba izvorna sistema uporabljata isto različico podatkovne baze, kot ciljna podatkovna baza.

Slika 7: Logični podatkovni model pridobljenih podatkov



Vir: lastno delo.

Problem lahko nastane pri integraciji podatkov iz zunanjega podatkovnega vira. Tu moramo biti pozorni na dve podrobnosti. Prvič, kako bomo nastavili ustrezno kodiranje v Oracle proceduri, ki komunicira s spletno storitvijo, da se pri prenosu podatkov ne bodo »pokvarili« posebni znaki - šumniki. V izogib težavam bomo nastavili kodiranje UTF-8, s čimer povemo, da naj sistem pridobljene podatke dekodira s to kodno tabelo.

Drugič, želimo preprečiti napačno interpretiranje podatkovnih tipov atributov. Vzemimo za primer podatek o datumu vpisa podjetja, ki ga bomo integrirali iz spletne storitve. V osnovi gre za datumski atribut, ki pa je pri prenosu podatkov v XML obliki zapisan kot navaden niz (»string«). Ker vemo, da gre za datum, moramo podatkovni bazi povedati, da naj ga tako

tudi obravnava. To bomo storili s pomočjo Oracleve funkcije »TO_DATE«, pri čemer moramo kot drug parameter funkcije opredeliti tudi vreden format datuma (glej izvorno kodo pogleda v podpoglavju 3.6.2). Format, v katerem je podan datum, lahko izvemo iz meta podatkov spletne storitve (npr. WSDL datoteka).

3.7.3 Čiščenje podatkov in zagotavljanje kakovosti

Podatke bomo pred končnim preoblikovanjem analizirali in jih po potrebi prečistili z namenom zviševanja kakovosti. Pri tem je smiselno, da se osredotočimo le na tiste, ki so pomembni iz vidika poslovnih potreb. Glede na ugotovitve v teoretičnem delu lahko kakovost podatkov na dolgi rok zagotovimo le z upoštevanjem naslednjih dveh korakov.

Najprej moramo s pomočjo profiliranja analizirati podatke, ki jih želimo uporabiti, da s tem odkrijemo pomanjkljivosti in napake, ki jih moramo potem odpraviti. Drugi korak temelji na ugotovitvah do katerih so nas privedle analize v prvem koraku, da preprečimo oziroma omejimo nastanek napak pri vnosu podatkov v informacijskih sistemih v bodoče. To lahko storimo z implementacijo dodatnih omejitev v uporabniških aplikacijah, kot so: obvezna polja, preverjanje formata vnesenih podatkov in drugimi kontrolami.

Matični podatki strank

Odkrivanja napak se bomo lotili sistematično. Začeli bomo s tabelo, ki vsebuje matične podatke strank - »STRANKE« iz ERP podatkovne baze. Glede na pridobljene informacije iz literature lahko pričakujemo napake v smislu: pomanjkljivih podatkov, duplikatov, različnih podatkovnih formatov, pravopisnih napak, ipd.

Analizo začnemo s preprosto poizvedbo željenih stolpcev iz te tabele, s čimer dobimo hiter vpogled v strukturo in vsebino podatkov v tabeli (slika 8).

Slika 8: Izpis podatkov iz tabele »STRANKE«

ID	MATICNA	EMSO	DAVCNA	NAZIV	ULICA	ULICA2	KRAJ	POSTA	DATUM_ROJSTVA	DATUM_ROJSTVA2	VRSTA	SKIS	SKD	TELEFON	EMAIL	USTVARJE...	STATUS	PE
10000533	9999910		36599808	*****	PARTIZANSKA C. 1		SEŽANA	6210		3.10.1984		12	14500				2	99
20501004	5980003			*****	C. DVEH CESARJEV 393		LJUBLJANA	1000				2	14000	061 268-150			2	
90000027	9999900	2605927500163	35174153	*****	KERSNIKOVA ULICA 13		CELJE	3000		26.5.1927		3	14420	03/5483298			1	10
10000534				*****	MENGEŠKA CESTA 31		MENGEŠ	1234				1	11000				2	
31801001	1831275		32026218	*****	KERSNIKOVA 2		LJUBLJANA	1000				2	15000	94.200	426 40 67		2	99
90000014	9999900	2709948500392	63510634	*****	VRANSKO 100		VRANSKO	3305		27.9.1948		3	14420				1	10
10000131	5000491		56101252	*****	DUNAJSKA 370		ČRNUČE	1231				1	11002	438-6130			2	
60000020	5246440000		50706195	*****	MLADINSKA ULICA 1		BLED	4260				10	15000	94.120	04/5372-700		1	99
30506001				*****	GRAJSKA 44		BLED	4260				2	15000				2	
30513003				*****			ZAGORJE	1410				2	15000				2	
30101007				*****	C. NA BRDO 49		LJUBLJANA	1000				2	15000				2	

Vir: lastno delo.

Na prvi pogled rezultata poizvedbe opazimo veliko praznih atributov, kar je posledica dejstva, da so v isti tabeli shranjene fizične in pravne osebe, ki imajo različne specifične attribute. Za prvi atribut »ID« vemo, da je edinstven primarni ključ, zato ga ne bomo dodatno analizirali. Začnemo s preprosto analizo atributov: »MATICNA« »EMSO« in »DAVCNA«.

S pomočjo spodnje poizvedbe izvedemo analizo distribucije vrednosti atributa matične številke za vse stranke, ki nimajo EMSA (pravne osebe):

```
SELECT maticna, COUNT (*)
FROM stranke
WHERE emso is NULL
GROUP BY maticna
HAVING COUNT (*) > 1
ORDER BY 2 DESC;
```

Rezultati zgornje poizvedbe so vidni na spodnji sliki (slika 9).

Slika 9: Razpon vrednosti atributa »MATICNA« iz tabele »STRANKE«

MATICNA	COUNT(*)
9999900	266179
9898989898	1130
0	1101
	985
9999910	973
98989898	943
999	845
99	779

Vir: lastno delo.

Če izhajamo iz dejstva, da bi morala biti matična številka edinstvena za posamezno podjetje, nam rezultati pokažejo, da s podatki zagotovo nekaj ni v redu. Vidimo lahko, da se matične številke pri različnih zapisih (podjetjih) ponavljajo, le-te pa imajo določen vzorec. Na podlagi tega lahko sklepamo, da so zaposleni pri vnosu strank v informacijski sistem v primerih, ko niso imeli podatka ali pa se jim je mudilo, vnesli kar naključno vrednost ali pa celo nič. Enako analizo naredimo tudi na preostalih dveh atributih »EMSO« in »DAVCNA«, kjer pridemo do podobnih ugotovitev.

Iz vidika kakovosti in nepristranskosti je bolje imeti manjkajoče podatke, kot pa netočne. Zato bomo pri najdene nepravne vrednosti raje nadomestili s praznimi (»NULL«). To storimo za vse tri attribute s SQL poizvedbo:

```
UPDATE stranke
SET maticna = NULL
WHERE maticna IN ('9898989898',
                  '0',
                  '9999910',
                  '98989898',
                  '999',
                  '99');
```

Takšne manjkajoče vrednosti je v praksi brez primerne vira podatkov težko dopolniti. Ker pa imamo v našem primeru na voljo ustrezen zunanji podatkovni vir, ki med drugim vsebuje tudi podatek o davčni in matični številki podjetja, bomo poizkusili manjkajoče matične številke dopolniti. To lahko storimo le za tiste stranke, ki imajo vneseno davčno številko, s pomočjo katere povežemo podatke iz pogleda »MAT_PODATKI_V«. Uporabimo naslednjo poizvedbo:

```
UPDATE stranke a
  SET a.maticna = (SELECT maticna FROM mat_podatki_v WHERE davcna =
a.davcna) ;
```

Kljub temu ugotovimo, da še vedno ostajajo zapisi, ki imajo prazne vrednosti pri vseh treh atributih, davčna, matična in EMŠO hkrati. Zato se postavlja vprašanje, kako v tem primeru ločiti med fizičnimi osebami in podjetji? S pomočjo analize atributa »VRSTA« in podpornih šifrantov v ERP podatkovni bazi ugotovimo, da atribut ločuje med različnimi vrstami strank. Vrednosti 2 in 8 predstavljata fizične osebe, ostale vrednosti pa pripadajo različnim tipom pravnih oseb.

Nadaljujemo z analizo atributov: »ULICA«, »KRAJ«, »POSTA« in »SKD«. Ugotovimo, da ima veliko zapisov tudi te vrednosti prazne, zato jih poizkusimo dopolniti s podatki iz zunanjega vira. To lahko storimo samo za pravne osebe, pri čemer uporabimo podoben isti vir podatkov in pristop, kot pri dopolnjevanju matične številke.

Podobna primera sta tudi atributa »EMAIL« in »TELEFON«. Čeprav podatka nista relevantna za potrebe segmentacije strank, pa sta uporabna iz vidika trženja. Poizkusili ju bomo dopolniti iz drugega internega vira – podatkovne baze elektronskega bančništva, kjer sta prisotna v večji meri, tako za fizične kot za pravne osebe. Naredimo podobno poizvedbo kot zgoraj pri dopolnjevanju podatka o matični številki, le da kot izvirno tabelo v poizvedbi uporabimo tabelo »EB_UPORABNIKI«, podatke pa povežemo s pomočjo tujega ključa »ID_STRANKE«.

Pri analizi izvornih matičnih podatkov smo prav tako zaznali, da sta atributa »ULICA2« in »DATUM_ROJSTVA2« podvojena. Sklepamo, da sta nastala kot posledica številnih nadgrajenj informacijskega sistema, saj z analizo vrednosti v obeh atributih ugotovimo, da so bile stranke, ki imajo polna atributa »ULICA2« in »DATUM_ROJSTVA2«, dodane v podatkovno bazo kasneje, kot stranke, ki imajo polna atributa »ULICA« in »DATUM_ROJSTVA«. Podvojena atributa bomo torej združili, pri čemer bomo vzeli vrednost nepraznega atributa v paru. V primeru, ko pa sta polna oba atributa bomo vzeli vrednosti iz drugega atributa, saj gre za novejši podatek. Tu imamo opravka s problematiko staranja podatkov. Podatke združimo s spodnjimi SQL ukazi, na koncu pa odvečna atributa še odstranimo iz tabele:

```

UPDATE stranke
  SET ulica = (CASE WHEN ulica IS NULL THEN ulica2 ELSE ulica END) ,
     rojstni_dan = (CASE WHEN rojstni_dan IS NULL THEN rojstni_dan2
ELSE rojstni_dan END);

ALTER TABLE stranke DROP COLUMN ulica2;
ALTER TABLE stranke DROP COLUMN rojstni_dan2;

```

Preverimo še vrednosti v združenem atributu »ROJSTNI_DAN«. S poizvedbo želimo odkriti nesmiselne podatke, zato preverimo ali obstajajo stranke, ki so bile rojene pred letom 1890, kar bi bilo nenavadno:

```

SELECT id, datum_rojstva
  FROM stranke
 WHERE datum_rojstva NOT BETWEEN TO_DATE ('01.01.1890', 'dd.mm.yyyy'
AND SYSDATE
ORDER BY 2 DESC;

```

Rezultati poizvedbe (slika 10) potrdijo našo hipotezo, da obstajajo stranke z napačno vnesenimi rojstnimi dnevi. Da te vrednosti ne bi pokvarile celotnega nabora podatkov, jih bomo raje nadomestili s praznimi vrednostmi. Uporabimo isti postopek kot smo ga zgoraj, pri čiščenju matične številke.

Slika 10: Analiza atributa »DATUM_ROJSTVA« iz tabele »STRANKE«

ID	DATUM_ROJSTVA
90006688	16.11.1295
161723	31.10.1263
964695	26.07.1201
315982	07.01.1200
17895	25.08.1198
99992895	21.04.1198
955817	18.11.1197
99990433	27.10.1197
90024760	17.05.1197

Vir: lastno delo.

Ostanejo nam še trije nepreverjeni atributi: »STATUS«, »USTVARJENO_OB« in »PE«. Hitro profiliranje nam pove, da je prvi atribut pri vseh zapisih neprazen, medtem, ko drugi in tretji vsebujeta prazne vrednosti. Ker sta za segmentacijo strank pomembna oba, ju bomo poizkusili dopolniti.

Atribut »USTVARJENO_OB« hrani podatek o tem, kdaj je stranka postala komitent izbranega podjetja. Manjkajoče vrednosti bomo poizkusili dopolnili tako, da bomo poiskali datum sklenitve najstarejšega posla, ki ga ima stranka v podjetju. Preverili bomo podatke o

bančnih računih, kreditih in varčevanjih. Za lažje posodabljanje podatkov bomo ustvarili funkcijo, ki vrača datum najstarejšega posla podane stranke. Uporabili bomo sledečo poizvedbo:

```
SELECT MIN(NVL(datum_otvoritve, SYSDATE))
FROM (
    SELECT datum_otvoritve FROM racuni_fizicne WHERE id_stranke =
p_id_stranke
    UNION ALL
    SELECT datum_otvoritve FROM krediti WHERE id_stranke =
p_id_stranke
    UNION ALL
    SELECT datum_otvoritve FROM varcevanja WHERE id_stranke =
p_id_stranke
);
```

Na podoben princip bomo dopolnili tudi vrednosti atributa »PE«, ki pove, kateri poslovni enoti pripada stranka. Za razliko pri tem ne bomo upoštevali najstarejšega posla, ki ga ima stranka, temveč najmlajšega.

Ostali podatki

Tudi ostale pridobljene podatke bomo analizirali in po potrebi prečistili na podoben način, kot smo to storili v tabeli matičnih podatkov strank. Ker gre večinoma za podatke o bančnih poslih, ki so vezani na stranke ali bančne račune bomo obvezno preverili tudi integriteto povezav. To bomo storili tako, da bomo za vsak zapis, ki vsebuje atribut »ID_STRANKE« ali »ID_RACUNA«, recimo v tabeli kreditov, preverili, da za ta tuji ključ obstaja zapis v primarni tabeli (tabeli strank oziroma računov). Morebitne zapise, ki bi kršili integriteto bomo iz množice podatkov odstranili.

3.7.4 Oblikovanje podatkovne sheme

Pridobljene in prečiščene podatke je potrebno v skladu s ciljno podatkovno shemo ustrezno povezati, agregirati, tvoriti izpeljane attribute, ipd. Ti postopki so del preoblikovanja podatkov in se v praksi med seboj pogosto tudi prepletajo. Pridobljeni podatki so dobra osnova za izdelavo različnih podatkovnih shem za podporo analitičnih poslovnih potreb.

Za ponazoritev primera bomo razvili preprosto podatkovno shemo, ki bo razumljiva in enostavnejša za uporabo. Zajeli bomo bistvene podatke o stranki, da bi s tem poslovnim uporabnikom omogočili sledeče:

- Vsaka stranka je, oziroma je bila povezana, z vsaj enim bančnim produktom. Poslovne uporabnike zanimajo ključne metrike poslovanja stranke, kot so: število bančnih računov, število transakcij, znesek transakcij, višina limita, število kreditov, ipd.

- Poleg tega, so za segmentacijo in trženje zanimivi tudi drugi podatki stranke. Na primer demografski, kot so: starost, vrsta stranke, kraj in naslov stranke, ipd.

Shema bo torej omogočala enostaven celostni pregled strank in njihovo segmentacijo. Za bolj podroben pregled posameznih bančnih produktov, kot so krediti bi morali narediti ločene podatkovne sheme, v katere bi vključili vse specifične attribute tega produkta.

V shemi bomo izhodiščne matične podatke stranke dopolnili s podatki iz podpornih šifrantov ter dodali naslednja dejstva in mere:

- starost stranke,
- čas poslovanja stranke v letih,
- vrsta primarnega bančnega računa,
- trenutno stanje na vseh aktivnih bančnih računih,
- število transakcij na vseh aktivnih bančnih računih v tekočem mesecu,
- skupna višina limitov na vseh aktivnih bančnih računih,
- skupna višina izrednega limita na vseh aktivnih bančnih računih,
- število aktivnih kreditov,
- skupen znesek aktivnih kreditov,
- število aktivnih varčevanj,
- skupen znesek aktivnih varčevanj,
- ali je uporabnik spletne banke,
- ali je uporabnik mobilne banke.

Za opredeljeno podatkovno strukturo smo razvili kompleksno SQL poizvedbo nad integriranimi tabelami, sestavljeno iz številnih relacij in gnezdenih poizvedb. Zaradi večje učinkovitosti smo tabelo dejstev realizirali s pomočjo materializiranega pogleda »F_STRANKE_PREGLED«, ki podatke osvežuje enkrat dnevno (Priloga 1). Zrno poizvedbe je stranka, zato smo najprej vključili vse attribute iz tabele »STRANKE«, ki že vsebuje prečiščene in dopolnjene podatke. Nato smo vključili podatke iz povezanih šifrantov. Iz tabele poslovnih enot smo dopolnili poizvedbo z opisnim atributom poslovne enote »PE_NAZIV«, poleg tega pa smo dodali tudi attribute »SKD_NAZIV«, »VRSTA_NAZIV« in »STATUS_NAZIV«. Pri tem smo si pomagali s funkcijo, ki nam za podano vrsto šifranta in tuji ključ vrne naziv postavke.

Izpeljanki starost stranke in čas poslovanja lahko izračunamo preprosto, saj imamo podatka o datumu rojstva oz. vpisa v register in datumu, ko je bila stranka vnesena. Večji izziv pa predstavljajo dimenzije, ki opisujejo ključne metrike poslovanja stranke. Ker ima lahko posamezna stranka več računov, več limitov, več kreditov in več varčevanj, podatke iz tabel teh poslov ne moremo enostavno povezati. Zato smo tudi tu uporabili funkcije za posamezne izpeljanke.

Za primer pogledimo funkcijo »STANJE«, ki za podano stranko vrača trenutno stanje na računu. Skupno stanje na vseh bančnih računih stranke funkcija izračuna tako, da sešteje razliko prometa v dobro in v breme na vseh aktivnih računih stranke. Izvorna koda je sledeča:

```
FUNCTION stanje(p_id_stranke) RETURN NUMBER IS
    stanje NUMBER := 0;
BEGIN
    SELECT SUM(NVL(dobro, 0) - NVL(breme, 0)) INTO stanje
    FROM promet
    WHERE id_racuna IN (SELECT id FROM racuni WHERE id_stranke =
p_id_stranke)
        AND datum <= SYSDATE;
    RETURN stanje;
END stanje;
```

Podobne funkcije smo naredili tudi za ostale izpeljane attribute.

3.7.5 Revizija informacij in skladnost sistema

V fazi razvoja integracijskega procesa, ob prvem nalaganju podatkov je priporočljivo preveriti ali rezultati integracije ustrezajo našim pričakovanjem. Predvsem v smislu pravilnosti in kakovosti podatkov ter učinkovitosti procesa.

Oblikovana podatkovna shema »F_STRANKE_PREGLED« bi morala vsebovati podatke o vseh aktivnih strankah iz operativne podatkovne zbirke. Zato smo se s preprostim štetjem zapisov prepričali, da količina približno ustreza količini istega nabora podatkov neposredno iz operativne baze.

S preprostimi analizami ključnih atributov v novo nastali shemi smo preverili tudi smiselnost in format podatkov.

3.8 Nalaganje podatkov

Nalaganje preoblikovanih podatkov v končno podatkovno shrambo je zadnji del procesa integracije podatkov, kjer se dokončno napolni podatkovna shema, pri čemer ima pomembno vlogo učinkovitost nalaganja. V praksi imamo na razpolago različne tehnike polnjenja podatkov, kot smo ugotovili že v teoretičnem delu. Izbira ustrezne tehnike je odvisna od številnih dejavnikov, še najbolj pa od tehničnih omejitev razpoložljive infrastrukture.

V našem primeru podatke pridobimo, prečistimo in preoblikujemo v pripravljalni podatkovni bazi. Zaradi pomanjkanja ločenega podatkovnega skladišča podatki ostanejo v istem okolju, ki služi kot enostavno podatkovno skladišče. Ciljno podatkovno shemo smo pripravili s pomočjo materializiranega pogleda, s čimer smo zagotovili hitre poizvedbe za končne poslovne uporabnike. Le-te lahko dodatno optimiziramo s kreiranjem ustreznih indeksov na ključnih atributih.

Slabost pristopa, ki v celoti prepisuje podatke je večja poraba razpoložljivih sistemskih virov, zaradi česar bo prepis potekal v nočnih urah. Ker se ob osveževanju vsi podatki prepišejo, se zgodovina podatkov ne ohranja. Govorimo torej o spreminjajočih se dimenzijah (SCD) tipa 1, kjer imajo poslovni uporabniki na voljo le zadnje stanje.

Uvedba področnega podatkovnega skladišča, ki bi bilo bolj prilagojeno analitičnim potrebam, bi omogočala večjo učinkovitost analiz in uporabnikom ponujala več analitičnih zmožnosti. V tem primeru bi se soočali s še večjimi izzivi pri polnjenju skladišča, ker bi bilo potrebno podatke prenesti v novo, ločeno okolje. Poleg tega se podatkovne sheme v različnih poslovnih področjih razlikujejo, kar pomeni, da je potrebno proces prilagoditi vsakemu posebej.

4 DISKUSIJA IN PREDLOGI

Spomnimo se Inmonovega zapisa, da so neustrezno integrirani podatki iz poslovnega vidika neuporabni. Dejstvo je, da sam proces priprave in integracije podatkov v podatkovno skladišče predstavlja enega izmed najzahtevnejših korakov v projektu poslovne inteligence. K temu pripomorejo heterogeni podatkovni viri, slaba kakovost podatkov, velika količina podatkov in mnogi drugi dejavniki. V magistrskem delu smo prav zato na podlagi strokovne in znanstvene literature ter s pomočjo praktičnega primera podrobneje proučili omenjeno problematiko.

4.1 Ocena integracije podatkov

V empiričnem delu smo prikazali pripravo in integracijo podatkov na primeru izbrane banke. Poizkusil smo integrirati podatke iz treh različnih podatkovnih virov, da bi zadostili potrebam poslovnih uporabnikov ter jim omogočili celostni pregled strank in njihovega poslovanja. Najtežji del v samem procesu sta predstavljala zaznava in razrešitev morebitnih pomanjkljivosti v podatkih ter druge ovire, na katere smo naleteli.

V splošnem bi lahko omejitve v procesu integracije podatkov razdelili na kadrovske, organizacijske in tehnične ovire. V sklopu prvih je zelo pomembno ustrezno projektno vodenje in planiranje, pri čemer morajo biti zahteve, cilji in viri projekta (izvedbeni roki, proračun, kadri) jasno določeni in dosegljivi. Ker gre v primeru izbranega podjetja za projekt, ki se dejansko ne izvaja, so do izraza prišle bolj tehnične težave.

Te pa so tiste, ki lahko naredijo projekt integracije podatkov zares problematičen, saj neposredno vplivajo na kakovost podatkov shranjenih v podatkovnem skladišču. V teoretičnem delu smo predstavili številne ovire. V praktičnem, pa smo naleteli na največ težav iz naslova heterogenosti in vprašljive kakovosti podatkov, staranja podatkov in velike količine podatkov ter posledično slabe učinkovitosti nalaganja in osveževanja podatkovnega skladišča.

Prva težava, na katero smo naleteli na začetku ETL procesa so bili torej heterogeni in »umazani« podatki. Integracija več različnih podatkovnih virov običajno zahteva prestrukturiranje podatkovne sheme v enotno (Rahm & Dos, 2000, str. 7), kar smo z razvojem ustreznega pogleda uspešno premostili že v fazi pridobivanja podatkov. Proces čiščenja smo začeli s profiliranjem podatkov, s čimer smo zaznali vidnejše napake, kot so: duplikati, prazne vrednosti, nepravilne vrednosti in druge. Večino teh smo odkrili z najbolj osnovno obliko profiliranja, to je analiza stolpcev, odpravili pa smo jih »ročno«, s klasičnimi SQL poizvedbami.

V nadaljevanju so se potrdila spoznanja Sarsfielda, ki ugotavlja, da sta za zagotavljanje kakovosti, bolj kot uporabljene tehnologije in tehnike čiščenja, pomembna dobro poznavanje vsebine in poslovnih potreb, ki jim poizkušamo zadostiti (2009, str. 112). Iz analitičnega vidika namreč niso vsi podatki enakovredni, zato smo se v našem primeru osredotočili le na pomembne. Čiščenje in preoblikovanje podatkov sta se tudi v praksi izkazala kot ena izmed zahtevnejših elementov integracije.

Vpliv velike količine podatkov, ki negativno vpliva na učinkovitost oblikovanja končne podatkovne sheme smo poizkusili omiliti s tem, da smo bili pri pridobivanju podatkov še posebej pozorni na to, da smo zajeli le nujno potrebne. Poleg tega smo pri integraciji podatkovne tabele o transakcijskem prometu strank s pomočjo združevanja zmanjšali zrnatost podatkov. Obe strategiji sta se izkazali kot razmeroma preprosti za izvedbo in učinkoviti, saj smo privarčevali znatno količino podatkov in diskovni prostor.

Velika količina podatkov pa negativno vpliva tudi na učinkovitost osveževanja podatkovnega skladišča. Pomembna lastnost le-teh je namreč zmožnost osveževanja le spremenjenih podatkov, saj se lahko v nasprotnem primeru zgodi, da se podatki ne uspejo osvežiti v eni iteraciji osveževanja (Kimball, Ross, Thornthwaite, Mundy & Becker, 2013, str. 451). V primeru izbrane banke smo to uspeli zagotoviti z uporabo »hitrih« materializiranih pogledov. S tem smo pohitrili celoten proces osveževanja podatkov in razbremenili systemske vire, hkrati pa smo zagotovili tudi razširljivost podatkovnega skladišča.

Tudi staranje in spreminjanje podatkov skozi zgodovino je problematika podatkovnega skladiščenja, ki se ji v primeru izbranega podjetja nismo izognili. Ker je obravnava tega vidika odvisna predvsem od poslovnih potreb, smo ugotovili, da jo je modro upoštevati že v začetni fazi razvoja ETL procesa, ko je potrebno določiti strategijo obravnavanja sprememb. V prikazanem primeru potrebe po ohranjanju zgodovine sprememb v podatkih ni bilo, zato smo pri razvoju podatkovne sheme uporabili spreminjajočo se dimenzijo tipa 1. Gre za najenostavnejši pristop obravnavanja spreminjajočih se dimenzij, kar nam je olajšalo razvoj končne podatkovne sheme. V primeru potrebe po vpeljavi ohranjanja zgodovine bi morali prilagoditi sistem osveževanja, težave iz naslova učinkovitosti pa bi lahko zaradi hrambe dodatnih podatkov prišle še bolj do izraza.

4.2 Predlogi in možnosti nadgradnje

Opisan praktičen primer je ena izmed možnosti integracije podatkov v izbranem podjetju, ki poleg internih podatkovnih virov združuje tudi zunanji vir za dopolnjevanje matičnih podatkov strank. Integrirani podatki zadostujejo za osnovni celostni pregled strank in njihovo segmentacijo, je pa možno celovitost informacij z integracijo dodatnih virov še izboljšati. Uporaba podatkovne povezave za pridobivanje podatkov omogoča, da bi lahko enostavno pridobili tudi ostale podatke iz obeh internih podatkovnih baz. Ravno tako bi lahko k virom dodali tudi ostale spletne storitve ponudnika, pri čemer bi bilo potrebno dograditi proceduro in razviti dodatne poglede za razčlenjevanje novih podatkov. Z večjim naborom podatkov bi lahko razširili analitične možnosti podatkovnega skladišča.

Kot najzahtevnejši del integracije se, kot že omenjeno, ni izkazalo pridobivanje podatkov, temveč preoblikovanje in zagotavljanje njihove kakovosti. V našem primeru smo podatke analizirali in prečistili »ročno« v pripravljalnem okolju. Ta proces bi bilo priporočljivo v celoti avtomatizirati z razvojem ustrezne programske skripte oziroma z uporabo naprednejših orodij. Poleg tega, bi bilo priporočljivo odkrite napake odpraviti tudi v operativnih podatkovnih bazah. S tem bi preprečili pojavitev le-teh ob vnovičnem osveževanju podatkovnega skladišča.

Vzroke za določene vsebinske napake v podatkih lahko pripišemo površnosti poslovnih uporabnikov pri vnosu podatkov. Širše gledano, pa je za to kriv pomanjkljivo razvit informacijski sistem oziroma njegove kontrole, ki takšne vnose dopuščajo. Le-te bi bilo potrebno ustrezno dograditi in tako izboljšati pravilnost vnesenih podatkov.

Iz vidika učinkovitosti in analitičnih zmožnosti bi bilo smiselno uvesti tudi namensko področno skladišče. V njem bi lahko razvili ločene podatkovne sheme po posameznih bančnih produktih, ki bi omogočale podrobnejše analize. Podatkovnim shemam bi bilo smiselno dodati tudi časovno dimenzijo, s katero bi poslovnim uporabnikom omogočili zgodovino sprememb in analize po obdobjih.

SKLEP

Hitre spremembe, nestabilnost in globalna konkurenca od podjetij zahtevajo, da svojo strategijo in poslovne cilje dinamično prilagajajo potrebam in zahtevam trga. Da bi to lahko naredila, mora management sprejemati pravočasne in dobre poslovne odločitve, kar pa lahko naredi samo na podlagi celovitih, kakovostnih in točnih informacij.

Informacijsko tehnologijo na področju poslovne inteligence danes s pomočjo različnih tehnik, orodij in sistemov to že omogoča. Razvoj in uvedba takšnih sistemov, pa je za preživetje podjetij v trenutnih okoliščinah prioriteten projekt, ki zahteva močno kadrovsko in finančno podporo managementa.

Cilj projekta uvedbe poslovne inteligence je, kako množico različno kakovostnih in strukturiranih podatkov, ki se nahajajo v različnih informacijskih sistemih integrirati in osveževati v lastni podatkovni shrambi, iz katere je mogoče v realnem času pridobivati uporabne informacije, ki jih potrebujejo odločevalci.

V magistrskem delu smo proučili in analizirali proces priprave in integracije podatkov, ki podjetjem običajno predstavlja največji izziv v tem projektu. Osredotočili smo se predvsem na tri faze ETL procesa, to so: pridobivanje, preoblikovanje in nalaganje podatkov. Ugotovili smo, da se podjetja z različnimi težavami soočajo na organizacijski in tehnični ravni integracije.

Spoznanja iz teoretičnega dela smo v drugem delu preizkusili na praktičnem primeru priprave in integracije podatkov. Izhajali smo iz poslovnih potreb izbrane banke, na podlagi česar smo zaznali potrebo po integraciji podatkov, ki bodo podjetju omogočali celostni pregled strank in njihovo segmentacijo. Z analizo obstoječe infrastrukture podjetja smo identificirali tri podatkovne vire, dva interna in enega zunanega. Integracije smo se lotili s pristopom skupne podatkovne shrambe, da bi poslovnim uporabnikom zagotovili povsem ločeno analitično okolje.

Heterogenost pridobljenih podatkov in druge napake, ki smo jih odkrili s pomočjo podatkovnega profiliranja, smo v večji meri uspeli odpraviti s čiščenjem ter poenotenjem strukture in formata podatkov. Največji izziv so predstavljali podvojeni podatki, ki so imeli različne vrednosti, pri čemer pa ni bilo takoj razvidno katere so točne. Pri preoblikovanju in nalaganju podatkov v podatkovno skladišče smo imeli največ problemov zaradi velike količine podatkov. Z razvojem preproste podatkovne sheme smo želeli prikazati praktično uporabo integriranih podatkov.

Ugotovili smo, da je integracija podatkov ena najtežjih in tudi najpomembnejših aktivnosti v projektu uvedbe poslovne inteligence. Več kot je različnih podatkovnih virov, in večja kot je količina podatkov, bolj zapletena je lahko integracija in s tem večja verjetnost, da se bodo pojavile težave pri izvedbi projekta.

V pomoč je lahko tudi sodobna tehnologija, ki ponuja več možnih pristopov integracije, se je pa potrebno zavedati, da to ni zagotovilo za uspeh. Za kakovost integracije morajo poskrbeti vsi sodelujoči na projektu, od managementa, vodje projekta, razvijalcev, pa vse do končnih poslovnih uporabnikov, pri čemer morajo biti cilji projekta jasno opredeljeni in dosegljivi. Tako bo integracija vsebinsko pravilna, tehnične ovire pa lažje premostljive.

Menimo, da smo s pomočjo prikazanega primera uspeli prikazati poglobljeno problematiko priprave in integracije podatkov ter ključne prijeme za reševanje, s čimer smo dosegli zastavljen cilj raziskovalnega dela. Želimo pa opozoriti, da je ugotovitve le na podlagi tega primera, ki se v praksi ne izvaja, težko splošiti za različne druge primere, čeprav izhaja iz realnih potreb banke. Uporaba poslovne inteligence podjetjem daje pomembno prednost za hitrejšo in boljše prilagajanje zahtevam trga.

LITERATURA IN VIRI

1. Anand, N. (februar 2014). ETL and its impact on Business Intelligence. *International Journal of Scientific and Research Publications*, 4(2), 1–3.
2. Bange, C., Bloemen, J., Derwisch, S., Fuchs, C., Grosser, T., Iffert, L., Janoschek, N., Keller, P., Seidler, L. & Tischler, R. (2018). *BI Trend Monitor 2018*. Würzburg: BARC.
3. Chaffey, D. & Wood, S. (2005). *Improving Performance Using Information Systems*. London: Prentice Hall.
4. Chu, M. Y. (2003). *Blissful Data : Wisdom and Strategies for Providing Meaningful, Useful, and Accessible Data for All Employees*. Fullerton: AMACOM.
5. DAMA UK Ltd. (2013). The six primary dimensions for data quality assessment: defining data quality dimensions. Pridobljeno iz <https://www.dqglobal.com/wp-content/uploads/2013/11/DAMA-UK-DQ-Dimensions-White-Paper-R37.pdf>
6. Doan, A., Halevy, A. & Ives, Z. (2012). *Principles of Data Integration*. San Diego: Morgan Kaufmann.
7. Eckerson, W. (2002). The Rise of Analytics Applications: Build or Buy. In *TDWI report series* (str. 6). Seattle: The Data Warehousing Institute.
8. FMF - Fakulteta za matematiko in fiziko Ljubljana. (b.d.). *OLAP*. Pridobljeno 30. junij 2018 iz <http://wiki.fmf.uni-lj.si/wiki?title=OLAP>
9. Forrester Research, Inc. (2018). *Business Intelligence*. Pridobljeno 5. julij 2018 iz <https://www.forrester.com/Business-Intelligence>
10. Gartner, Inc. (2017a). *Data Mining*. Pridobljeno 24. maj 2018 iz <https://www.gartner.com/it-glossary/data-mining>
11. Gartner, Inc. (17. februar 2017b). *Gartner*. Pridobljeno 24. maj 2018 iz <https://www.gartner.com/newsroom/id/3612617>
12. Halevy, A. Y., Ashish, N., Bitton, D., Carey, M., Draper, D., Pollock, J., Rosenthal, A. & Sikka, V. (2005). Enterprise information integration: successes, challenges and controversies. *Proceedings of the 2005 ACM SIGMOD international conference on Management of data* (str. 778-787). Washington: University of Washington & Transformic Inc.
13. Hillard, R. (2010). *Information-Driven Business: How to Manage Data and Information for Maximum Advantage*. Indianapolis: Wiley.
14. Inmon, B. (2005). *Building the Data Warehouse* (4. Izd.). Indianapolis: Wiley.

15. Izbrano podjetje. (2017). Arhitektura informacijskega sistema. Pridobljeno 3. september 2018 iz Interni viri izbranega podjetja.
16. Jaklič, J. (2017). *Poslovna Inteligenca*. (prosojnice pri predmetu). Ljubljana: Ekonomska fakulteta.
17. Kadadi, A. (2015). *Challenges of data integration in big data*. Greensboro: University of Arkansas at Pine Bluff.
18. Kimball, R., Ross, M., Becker, B., Mundy, J. & Thornthwaite, W. (2010). *The Kimball Group Reader : Relentlessly Practical Tools for Data Warehousing and Business Intelligence*. Indianapolis: Wiley.
19. Kimball, R., Ross, M., Thornthwaite, W., Mundy, J. & Becker, B. (2013). *The Data Warehouse Lifecycle Toolkit: The Complete Guide to Dimensional Modeling* (3. Izd.). New York: Wiley.
20. Krishnan, K. (2013). *Data Warehousing in the Age of Big Data*. San Francisco: Morgan Kaufmann.
21. Lans, R. v. (2012). *Data Virtualization for Business Intelligence Systems: Revolutionizing Data Integration for Data Warehouses*. Burlington: Morgan Kaufmann.
22. Laursen, G. H. & Thorlund, J. (2010). *Business Analytics for Managers : Taking Business Intelligence Beyond Reporting*. Indianapolis: Wiley.
23. LeBlanc, P., M.Moss, J., Sarka, D. & Ryan, D. (2005). *Applied microsoft business intelligence*. Indianapolis: Wiley.
24. Loshin, D. (2012). *Business Intelligence: The Savvy Manager's Guide*. Silver Spring: Elsevier Science & Technology.
25. Loshin, D. (2013). *Big Data Analytics: From Strategic Planning to Enterprise Integration with Tools, Techniques, NoSQL, and Graph*. Silver Spring: Elsevier Science & Technology.
26. Oracle. (julij 2010). *Data Cleansing and Correction with Data Rules*. Pridobljeno 5. julij 2018 iz https://docs.oracle.com/cd/E18283_01/owb.112/e10935/data_cleansing.htm
27. Oracle. (2017). *Oracle*. Pridobljeno 10. junij 2018 iz <https://docs.oracle.com/database/121/DWHSG/extract.htm#DWHSG8296>
28. Rahm, E. & Dos, H. H. (2000). *Data Cleaning: Problems and Current Approaches*. Pridobljeno 5. julij 2018 iz http://www.betterevaluation.org/sites/default/files/data_cleaning.pdf

29. Reeve, A. (2013). *Managing Data in Motion: Data Integration Best Practice Techniques and Technologies*. San Francisco: Morgan Kaufmann.
30. Sarsfield, S. (2009). *Data Governance Imperative*. Ely: IT Governance Ltd.
31. Sauter, V. L. (2011). *Decision Support Systems for Business Intelligence*. Indianapolis: Wiley.
32. Sebastian-Coleman, L. (2013). *Measuring Data Quality for Ongoing Improvement : A Data Quality Assessment Framework*. San Francisco: Morgan Kaufmann.
33. Shaker H. Ali El-Sappagh; Abdeltawab M. Ahmed Hendawi; Ali HamedEl Bastawissy. (julij 2011). A proposed model for data warehouse ETL processes. *Journal of King Saud University – Computer and Information Sciences*, 23(2), 91–104.
34. Soares, S. (2015). *Data Governance Tools : Evaluation Criteria, Big Data Governance, and Alignment with Enterprise Data Management*. Boise: MC Press.
35. Techopedia. (b.l.). V *Techopedia*. Pridobljeno 7. julij 2018 na spletni strani: <https://www.techopedia.com/definition/1184/data-warehouse-dw>
36. Thierauf, R. J. (2001). *Effective Business Intelligence Systems*. Santa Barbara: Greenwood Publishing Group, Incorporated.
37. Tupper, C. (2011). *Data Architecture - From Zen to Reality*. San Francisco: Morgan Kaufman.
38. U.S. Department of Transportation. (avgust 2010). *Data Integration Primer*. Pridobljeno 8. avgust 2018 iz <https://www.fhwa.dot.gov/asset/dataintegration/if10019/if10019.pdf>
39. Williams, S. & Williams, N. (2006). *The Profit Impact of Business Intelligence*. San Francisco: Morgan Kaufmann.
40. Yang W. Lee, J. D. (2009). *Journey to Data Quality*. Cambridge: MIT Press.
41. Ziegler, P. & Dittrich, K. R. (2007). *Data Integration — Problems, Approaches, and Perspectives*. Zurich: Department of Informatics, University of Zurich.

PRILOGA

Priloga 1: Izvorna koda podatkovne sheme za celostni pregled strank

Spodaj je prikazana izvorna koda sheme, oziroma materializiranega pogleda »F_STRANKE_PREGLED«. V shemo so vključeni osnovni matični podatki in drugi pomembni kazalniki stranke.

```
CREATE MATERIALIZED VIEW f_stranke_pregled REFRESH START WITH SYSDATE
NEXT TRUNC(SYSDATE + 1) + 6/24 COMPLETE
AS
SELECT
    a.id,
    a.davcna,
    a.maticna,
    a.emso,
    a.naziv,
    a.ulica,
    a.posta,
    a.kraj naslov,
    a.email,
    a.telefon,
    sif_naziv ('skd', a.skd) skd_naziv,
    a.vrsta,
    sif_naziv('vrsta_stranke', a.vrsta) vrsta_naziv,
    a.pe,
    ag.nazl_agenc pe_naziv,
    a.status,
    sif_naziv('statusi_stranke', a.status) status_naziv,
    floor(months_between(SYSDATE, a.datum_rojstva)/12) starost,
    floor(months_between(SYSDATE, a.ustvarjeno_ob)/12)
poslovanje,
    vrsta_racuna(a.id_stranke) vrsta_racuna,
    stanje(a.id_stranke) trr_stanje,
    st_transakcij(a.id_stranke) st_transakcij,
    visina_limita(a.id_stranke, 'tekoci') tek_limit,
    visina_limita(a.id_stranke, 'izredni') izr_limit,
    st_kreditov(a.id_stranke) st_kreditov,
    znesek_kreditov(a.id_stranke) znesek_kreditov,
    st_varcevanj(a.id_stranke) st_varcevanj,
    znesek_varcevanj(a.id_stranke) znesek_varcevanj,
    ima_eb(a.id_stranke) uporabnik_eb,
    ima_meb(a.id_stranke) uporabnik_meb
FROM stranke a, poslovne_enote p
WHERE a.pe = p.pe (+);
```