UNIVERZA V LJUBLJANI
EKONOMSKA FAKULTETA

MAGISTRSKO DELO

PRIMERJAVA DIMENZIJSKEGA MODELA IN MODELA PODATKOVNEGA TREZORJA PRI IZGRADNJI PODATKOVNEGA SKLADIŠČA

Ljubljana, december 2023 BRANKA KOJIĆ

IZJAVA O AVTORSTVU

Podpisana Branka Kojić, študentka Ekonomske fakultete Univerze v Ljubljani, avtorica predloženega dela z naslovom Primerjava dimenzijskega modela in modela podatkovnega trezorja pri izgradnji podatkovnega skladišča, pripravljenega v sodelovanju s svetovalcem doc. dr. Urošem Godnovom

I Z J A V L J A M

1. da sem predloženo delo pripravila 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 poskrbela, 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 pridobila vsa potrebna dovoljenja za uporabo podatkov in avtorskih del v predloženem delu in jih v njem jasno označila;

7. da sem pri pripravi predloženega dela ravnala v skladu z etičnimi načeli in, kjer je to potrebno, za raziskavo pridobila 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;

11. da sem preverila verodostojnost informacij, ki izhajajo iz zapisov na podlagi uporabe orodij umetne inteligence.

V Ljubljani, dne _____ Podpis študentke: _____

KAZALO

KAZALO TABEL

KAZALO SLIK

SEZNAM KRATIC

angl. – angleško
BI – (angl. business intelligence); poslovna inteligenca
CRM – (angl. customer relationship management); upravljanje odnosov s strankami
ERP – (angl. enterprise resource planning); celovit poslovno informacijski sistem
ETL – (angl. extract, transform, load); pridobivanje, preoblikovanje, nalaganje
OLAP – (angl. online analytical processing); sprotna analitična obdelava podatkov
SCD – (angl. slowly changing dimensions); počasi se spreminjajoče dimenzije
SQL – (angl. structured query language); strukturirani povpraševalni jezik za delo s podatkovnimi bazami

1 UVOD

Živimo v dobi tehnologije, ki nam omogoča zbiranje velikih količin podatkov. Zbiranje podatkov je postalo razmeroma enostavno, mnoga podjetja imajo na voljo celo več podatkov, kot jih lahko obdelajo. Vendar so podatki običajno brez pomena, dokler jih ne analiziramo in iz njih pridobimo koristne informacije, ki se večinoma uporabljajo za operativno poslovanje in za sprejemanje poslovnih odločitev na podlagi analitike (Albright in Winston, 2014).

Podatki celotnega podjetja se pogosto hranijo v ločenih sistemih, saj podjetja uporabljajo tudi različne aplikacije za podporo poslovanju. Takšni sistemi podatkovnih baz so optimizirani za hitro obdelavo transakcij in omogočajo različne operativne naloge za podporo poslovnim procesom. Za zanesljive informacije o poslovanju pa je pogosto treba poiskati in zbrati podatke iz več operativnih sistemov, kar je lahko zamudno. Za informacije o trendih oz. analiziranje podatkov v daljšem časovnem obdobju, ki bi jih lahko uporabili pri sprejemanju strateških odločitev, se pojavijo težave tudi pri iskanju podatkov o preteklih dogodkih, saj ti pogosto niso več dostopni. Vprašanja uporabnikov pogosto zahtevajo, da se poizveduje po velikem številu transakcij, za kar operativni sistemi niso najbolj primerni. Poleg tega se podatki, ki služijo operativnim potrebam, fizično razlikujejo od tistih, ki služijo analitičnim potrebam, zato je smiselno ločevati tudi podporno tehnologijo za obdelavo podatkov in posledično podatkovne baze. Poizvedbe v operativnih sistemih v splošnem zapisujejo podatke v tabele, ki so povezane s preprostimi relacijami. Poizvedbe, ki omogočajo večdimenzionalne analize, pa morajo pregledati ogromno količino zapisov. Za reševanje opisanih težav in za podporo analitičnim procesom so se razvila podatkovna skladišča, s pomočjo katerih so podjetja začela dosegati konkurenčno prednost (Inmon, 2005; Kimball in Ross, 2013; Laudon in Laudon, 2003; Ponniah, 2001).

Orodja za vizualizacijo podatkov so se v zadnjih letih nadgrajevala in podprla določene funkcionalnosti procesov pridobivanja, preoblikovanja, nalaganja (angl. extract, transform, load, v nadaljevanju ETL) ter s tem omogočila, da podatke zajamemo, preoblikujemo in analiziramo v istem orodju. To je lahko preprost način za hitro doseganje ciljev in pripravo analiz, zlasti v manjših podjetjih. V primerih večje količine podatkov in cilja, da zagotovimo vsem uporabnikom isto podatkovno plast ne glede na izbrano orodje za uporabo podatkov pa je podatkovno skladišče primernejša izbira.

Podatkovno skladišče je definirano kot entitetno usmerjena, združljiva, časovno odvisna in nespremenljiva zbirka podatkov za podporo poročilnim in odločitvenim procesom v podjetju (Inmon, 2005).

Različni pristopi k modeliranju podatkovnih skladišč so prisotni že vsaj od zgodnjih devetdesetih let prejšnjega stoletja. V zadnjih letih se priljubljenost upravljanja podatkov in zagotavljanja kakovosti podatkov večata, poleg tega se viša tudi zavedanje pomembnosti

podatkov za organizacijo, s tem pa tudi pomembnosti dobro zastavljenega modela podatkovnega skladišča (Reis in Housley, 2022; Sherman, 2014).

O izbiri ustreznega podatkovnega modela za podatkovna skladišča se je intenzivno razpravljalo (Gluchowski, 2021). Največkrat je bilo govora o dimenzijskem modelu, ki je ga predstavil Ralph Kimball, in normaliziranega modela podatkovnega skladišča, katerega začetnik je Bill Inmon, in temelji na teoriji relacijskih baz podatkov. Odkar so bile objavljene prve knjige o dimenzijskem modeliranju, je to postal splošno sprejet in prevladujoč pristop k modeliranju podatkovnih skladišč. Temelji na dveh vrstah tabel – tabele dejstev, ki shranjujejo poslovne dogodke oz. transakcije, in dimenzijske tabele za shranjevanje opisnih informacij in kategorij, ki razširijo pomen tabel dejstev. V devetdesetih letih se je na področju podatkovnih skladišč pojavil nov model, imenovan podatkovni trezor (angl. data vault), ki ga je izumil Daniel Linstedt. Temelji na kompleksnih omrežjih iz naravnega okolja, ki so sestavljena iz vozlišč in povezav med njimi. Podatkovni trezor je poslovno usmerjen model za skladiščenje podatkov, njegov cilj je čim bolj natančno predstaviti poslovanje. Sestavljen je iz treh osnovnih tipov objektov, vsak od njih pa ima svoj namen. V vozliščih so shranjeni poslovni ključi, v povezavah so shranjeni odnosi med posameznimi vozlišči oz. poslovnimi ključi, v satelitih pa je shranjena ostala vsebina, ki dodatno opisuje vozlišča ali povezave. V zadnjih letih se je model podatkovnega trezorja uveljavil kot alternativa prej omenjenih (Boddu, 2023; Hultgren, 2012; Linstedt in Olschimke, 2015).

Z razvojem in uveljavljanjem novih pristopov k modeliranju podatkovnih skladišč se pojavi vprašanje, v čem se pristopi razlikujejo, kakšne so njihove slabosti in prednosti ter predvsem kako se odločiti, kateri pristop izbrati.

Namen magistrskega dela je prispevati k razumevanju razlik med dimenzijskim modeliranjem, ki velja za prevladujoč pristop k modeliranju, in modeliranjem podatkovnega trezorja, ki se v zadnjih letih uveljavlja kot alternativni pristop, ter olajšati odločitev pri izbiri.

Cilji magistrskega dela so pripraviti pregled literature o obeh pristopih k modeliranju podatkovnih skladišč in ugotoviti, kateri so ključni dejavniki, ki jih je smiselno upoštevati pri izbiri pristopa k modeliranju podatkovnih skladišč, ter tako podati smernice vsem, ki se soočajo s sprejemanjem takšnih odločitev.

V magistrskem delu bomo povzeli literaturo, na podlagi katere bomo s pomočjo javno dostopnega orodja za modeliranje izdelali tabelarični dimenzijski model (v nadaljevanju dimenzijski model) in model podatkovnega trezorja za poslovni problem iz izbranega podjetja ter ju med seboj primerjali in analizirali glede na dejavnike, ki bodo prej tudi teoretično obravnavani, ter glede na delovne izkušnje, ki smo jih pridobili v zadnjih letih pri razvoju podatkovnih skladišč za različna podjetja. Povzeli bomo tudi rezultate drugih statističnih in komparativnih raziskav. S tem želimo določiti ključne dejavnike, ki so pomembni pri odločitvi glede pristopa k modeliranju podatkovnega skladišča.

Magistrsko delo bo v osrednjem delu razdeljeno na štiri poglavja. V prvem poglavju bomo definirali podatkovno analitiko in predstavili življenjski cikel podatkov v organizacijah. Razložili bomo pomen in delovni proces podatkovnega modeliranja ter opisali namen podatkovnih skladišč. Nadaljevali bomo s poglavjem o modeliranju podatkovnih skladišč, kjer bomo opisali ključne dejavnike, ki so pomembni pri modeliranju, ter podrobno predstavili dva pristopa – dimenzijski model in model podatkovnega trezorja. V naslednjem poglavju bomo predstavili poslovni problem, ki ga bomo na podlagi znanja, pridobljenega s pregledom literature, pretvorili v obe vrsti predstavljenih modelov. Sledila bo analiza prej opredeljenih dejavnikov za oba pristopa k modeliranju. Na podlagi analize in primerjave obeh pristopov bomo v naslednjem poglavju zapisali predloge in smernice za lažjo izbiro pristopa k modeliranju podatkovnih skladišč.

2 PODATKOVNA ANALITIKA

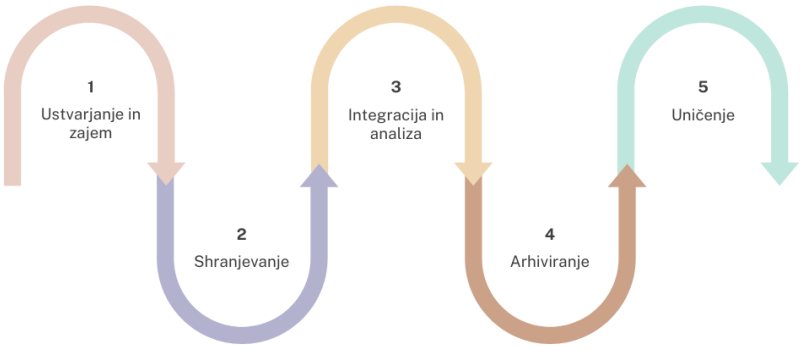
Podatkovna analitika je opredeljena kot uporaba računalniških sistemov za analizo velikih količin podatkov, katere cilj je podpora pri odločitvenih procesih. Je zelo interdisciplinarno področje, ki združuje vidike številnih drugih znanstvenih disciplin, kot so statistika, prepoznavanje vzorcev, strojno učenje in teorija signalov (Runkler, 2020).

Uporablja se tako v panogah, kjer si prizadevajo sprejemati bolj informirane odločitve, kot tudi na akademskih področjih, kjer raziskovalci preverjajo hipoteze in teorije. Podatkovna analitika je tesno povezana s poslovno analitiko, ki je osredotočena na uporabo podatkov za izboljšanje poslovnega odločanja (Balali in drugi, 2020).

2.1 Življenjski cikel podatkov

Življenjski cikel podatkov sestavljajo različne faze, skozi katere gredo podatki od nastanka do izbrisa. Velik del podatkov v podjetju ima predvidljiv življenjski cikel, ki je prikazan na sliki 1.

Slika 1: Življenjski cikel podatkov



Vir: prirejeno po Inmon in drugi (2019).

Življenjski cikel podatkov se začne z njihovim ustvarjanjem in zajemom. Podatki so lahko generirani s pomočjo naprav ali aplikacij, ki se uporabljajo za različne procese znotraj podjetja, lahko jih ustvarimo z ročnimi vnosi ali pa pridobivamo podatke, ki so bili proizvedeni izven podjetja. Čeprav je podatke mogoče ustvariti na različne načine, zbiranje vseh razpoložljivih podatkov ni nujno za uspeh podjetja. Sledi shranjevanje podatkov na varnem in dostopnem mestu, v večini primerov v sistemih za upravljanje baz podatkov, lahko pa tudi v datotečnih sistemih, prostoru v oblaku ali drugih oblikah. Preden so shranjeni podatki primerni za analizo, običajno poteka še postopek integracije. Namen tega postopka je strukturiranje podatkov na način, da so primerni za kombiniranje tudi z drugimi vrstami podatkov. Sledi proces analize podatkov, kjer je cilj identificirati vzorce in trende ali zgolj pridobiti uporabne vpoglede v podatke ter jih predstaviti v različnih oblikah, kot so nadzorne plošče in poročila. Po določenem času podatki niso več uporabni, zato jih arhiviramo, saj s tem ohranjamo kopije, ki jih je po potrebi mogoče obnoviti. V zadnji fazi življenjskega cikla podatke izbrišemo ali uničimo (IBM, brez datuma a; Inmon in drugi, 2019).

V magistrskem delu se bomo osredotočili na modeliranje podatkovnih skladišč, kar spada v proces integracije in analize.

2.2 Podatkovno modeliranje

V splošnem se modeliranje uporablja za podporo procesom načrtovanja in oblikovanja rešitve. Ne glede na to, ali oblikujemo nov izdelek, poslovni proces ali podatkovno bazo, najprej začnemo z izdelavo koncepta in vizije ter naborom zahtev, ki jim želimo zadostiti. Sledi modeliranje naših idej, ker pa je oblikovanje rešitve iterativen proces, je tudi postopek modeliranja iterativen. Uporabimo lahko različne pristope za modeliranje rešitev (Hultgren, 2012).

Pri delu s podatki se ukvarjamo z načrtovanjem tabel za shranjevanje podatkov. Za reševanje teh potreb se že dolgo uporabljajo različne tehnike in orodja za modeliranje podatkov. Podatkovno modeliranje je strukturiran pristop za analizo poslovnih zahtev in njihovo pretvorbo v podatkovni model. Proces po navadi vključuje tesno sodelovanje med poslovnimi analitiki in podatkovnimi inženirji. Modeliranje podatkov vključuje izbiro skladne strukture za podatke in predstavlja ključni korak pri ustvarjanju podatkov, ki so uporabni za sprejemanje odločitev in boljše poslovanje. Prednosti, ki jih omogoča modeliranje podatkov, so standardizacija definicij in terminologije podatkov, večja vključenost poslovnih uporabnikov v upravljanje podatkov ter boljša uporaba razpoložljivih podatkov (Craig, 2021). Z upoštevanjem pravil za modeliranje podatkov nastanejo modeli, ki predstavljajo zasnovo v obliki tabel in razmerij med njimi (Reis in Housley, 2022).

Podatkovni model predstavlja način, kako so podatki povezani z resničnim svetom in odraža, kako morajo biti podatki strukturirani in standardizirani, da najbolje predstavljajo poslovne zahteve (Reis in Housley, 2022). Podatkovni modeli so živi in se razvijajo skupaj s spreminjajočimi se poslovnimi potrebami (IBM, brez datuma b). Zagotavljajo tudi skupno

podlago za komunikacijo in sodelovanje ter s tem pomagajo zagotoviti, da si vsi deležniki prizadevajo za iste cilje in uporabljajo podatke na enoten način. Vizualno jih lahko prikažemo s pomočjo orodij za modeliranje podatkov, kar je zelo priporočljivo, saj lahko model sestavlja veliko število tabel in relacij med njimi (Sherman, 2014).

Najprej bodo predstavljeni pojmi, vezani na podatkovno modeliranje, ki nam bodo omogočili lažje razumevanje konceptov v nadaljevanju.

Začnimo s kratko razlago nivojev abstrakcije modeliranja. Modeliranje se začne na visoki ravni abstrakcije in postaja vedno bolj konkretno in specifično. Tipe podatkovnih modelov v splošnem delimo v tri kategorije po stopnjah abstrakcije: konceptualni, logični in fizični model. Konceptualni model zagotavlja splošen pogled na strukture podatkov na najvišjem nivoju in je neodvisen od katerega koli sistema ali tehnologije za upravljanje baze podatkov. Z njim si pomagamo pri definiranju obsega zahtev. Logični model je podrobna predstavitev podatkovnih zahtev organizacije, vključno z odnosi med entitetami. Je prav tako neodvisen od tehnologije, opisuje pa poslovne entitete, atribute in odnose, vendar ne določa, kako bodo podatki fizično shranjeni. Zadnji nivo je fizični, ki določa način shranjevanja podatkov v sistemu, vključno s tabelami, stolpci, indeksi in drugimi fizičnimi vidiki baz podatkov. Fizični nivo zahteva razumevanje značilnosti in omejitev sistema baz podatkov (Coronel in drugi, 2011; Sherman, 2014).

Naslednja pojma sta normalizacija in denormalizacija. Normalizacija je postopek zagotavljanja, da podatkovni model izpolnjuje lastnosti, kot so natančnost, doslednost, enostavnost, nepodvajanje in stabilnost. Razvil jo je Edgar F. Codd leta 1972. Normalizacija zagotavlja doslednost tako, da so vsi podatki shranjeni samo enkrat, s čimer se odpravi možnost nasprotujočih si podatkov. Hkrati zmanjšuje anomalije pri vstavljanju, posodabljanju in brisanju podatkov (Imhoff in drugi, 2003). Normalizacija lahko povzroči tudi negativne posledice, zlasti pri velikih sistemih, saj postane podatkovni model zapleten in zahteva kompleksno združevanje tabel pri poizvedbah. Poznamo več stopenj normalizacije, pri čemer je tretja normalna oblika najpogostejša, saj zagotavlja integriteto podatkov in hkrati uravnovesi kompleksnost in učinkovitost sistema. Večina operativnih virov je izvedenih v tretji normalni obliki, ki jo dosežemo z odpravljanjem ponavljajočih se atributov in ločevanjem atributov, ki niso vezani na primarni ključ tabele. V splošnem velja, da višja ko je normalna oblika, več operacij združevanja je potrebnih pri poizvedovanju. Namen normalizacije je odstranjevanje anomalij iz transakcijskih baz, zato takšne strukture ni treba ohranjati v bazah, ki služijo izključno za analitične namene. Denormalizacija je obraten postopek, pri katerem nekatere podatke podvojimo z namenom optimizacije poizvedb. Možni koraki denormalizacije vključujejo ponovno združevanje tabel, ki so bile razdeljene zaradi pravil normalizacije, shranjevanje podvojenih podatkov v tabelah in shranjevanje agregiranih podatkov v tabelah (Date, 2019; Oppel, 2009; Sherman, 2014).

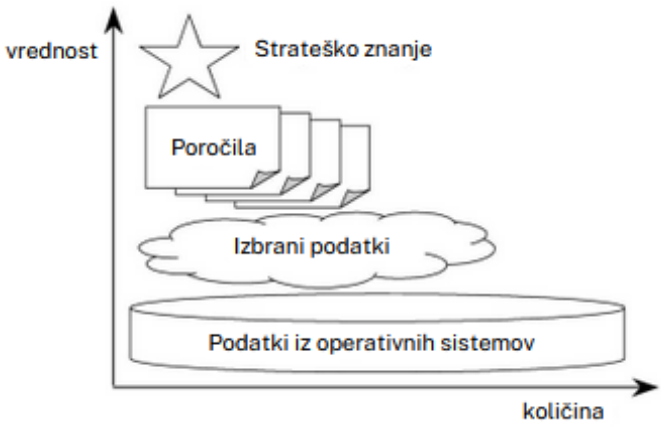
Pri modeliranju je pomembno razumevanje kardinalnosti povezave, ki določa kako so različne entitete povezane med seboj. Poznamo štiri osnovne kardinalnosti (Bagui in Earp, 2003):

- ena proti ena (1:1): ena entiteta je povezana z drugo entiteto in obratno, npr. en študent ima določeno vpisno številko, ki pripada samo njemu,
- ena proti mnogo (1:N): ena entiteta je povezana z več entitetami, npr. en profesor je nosilec več predmetov,
- mnogo proti ena (N:1): več entitet pripada eni entiteti, npr. več predmetov se izvaja pod okriljem enega profesorja,
- mnogo proti mnogo (N:M): več entitet je povezanih z več entitetami, npr. študent lahko opravlja več izbirnih predmetov in hkrati lahko izbirni predmet izbere več študentov.

2.3 Podatkovno skladišče

Do sredine osemdesetih let prejšnjega stoletja so podjetja v podatkovnih bazah shranjevala samo operativne podatke, ki so bili ustvarjeni s poslovnimi dogodki, kot so opravljanje nakupov, izdajanje računov ipd. Procesi odločanja pa zahtevajo hiter in celovit dostop do strateških informacij, ki so pridobljene predvsem iz ogromne količine operativnih podatkov s postopnim izbiranjem in združevanjem, kot je prikazano na sliki 2.

Slika 2: Vrednost informacije glede na količino podatkov



Vir: prirejeno po Golfarelli in Rizzi (2009).

Eksponentno povečanje operativnih podatkov je močno vplivalo na vlogo podatkovnih baz podjetij in spodbudilo uvedbo sistemov za podporo odločanju. Podatkovna skladišča so že od devetdesetih let prejšnjega stoletja namenjena podpori sistemov za odločanje. Zasnovana so tako, da združujejo ogromne količine podatkov iz različnih sistemov, kar omogoča boljše informiranje in odločanje v podjetjih, saj lahko uporabniki izvajajo kompleksne analitične poizvedbe, ne da bi pri tem upočasnili delovanje operativnih sistemov (Golfarelli in Rizzi, 2009).

2.3.1 Opredelitev podatkovnega skladišča

Podatkovno skladišče je entitetno usmerjena, integrirana, časovno odvisna in nespremenljiva zbirka podatkov za podporo poročilnim in odločitvenim procesom (Inmon, 2005).

V nadaljevanju razložimo pomen vsake od navedenih lastnosti. V operativnih sistemih so podatki shranjeni po posameznih aplikacijah, ki podpirajo specifične operativne dejavnosti. V nasprotju s tem so podatki v podatkovnem skladišču shranjeni po poslovnih entitetah, zato pravimo, da je entitetno usmerjeno. Za zagotovitev vseh podatkov za odločanje je treba zbrati podatke iz različnih aplikacij in posledično operativnih sistemov, ki jih je treba integrirati. Lastnost časovne odvisnosti izhaja iz tega, da so podatki v podatkovnem skladišču namenjeni analizi in odločanju, torej potrebujemo ne le trenutne podatke, temveč tudi zgodovinske, česar operativni sistemi ne zagotavljajo. Razlogi za to so različni - v nekaterih industrijah veljajo regulativne zahteve, ki zahtevajo arhiviranje podatkov po določenem času, pogosto pa izbris podatkov iz operativnih sistemov pomaga ohraniti optimalno delovanje in prostorsko učinkovitost sistema, s čimer je povezan tudi finančni vidik, kot so stroški shranjevanja podatkov. Ker podatki v podatkovnem skladišču niso namenjeni vsakodnevnem poslovanju, ni potrebe po posodabljanju podatkov v realnem času ob vsakem poslovnem dogodku. To pomeni, da so podatki v podatkovnih skladiščih manj spremenljivi kot v operativnih sistemih (Ponniah, 2001).

Prednosti podatkovnih skladišč vključujejo boljše razumevanje uspešnosti podjetja in podporo dejavnostim poslovne inteligence, kot so poročanje, analitika in odločanje. Z zagotavljanjem centraliziranega in integriranega pogleda lahko podatkovno skladišče pomaga podjetju prepoznati trende in vzorce, slediti ključnim indikatorjem uspešnosti in sprejemati informirane odločitve na podlagi podatkov (Inmon, 2005; Kimball &in Ross, 2013).

Skladiščenje podatkov je torej zbirka metod, tehnik in orodij, ki se uporabljajo za podporo uporabnikom – po navadi so to direktorji, analitiki – za izvajanje analiz podatkov, ki jim pomagajo pri odločitvenih procesih (Golfarelli in Rizzi, 2009).

Hultgren (2012) kot glavni cilj podatkovnega skladišča izpostavlja integracijo podatkov iz različnih virov in kreiranje enotnega pogleda na podatke oziroma kreiranje ene skupne resnice. Integracija je hkrati najpomembnejši in najtežji cilj pri izvedbi podatkovnega skladišča, saj podatki prihajajo iz različnih sistemov, ki so med seboj nepovezani in nestandardizirani. Zato je ob zajemu zahtev zelo pomembno določiti tudi nivo integracije, ki bo omogočen.

2.3.2 Vidiki modeliranja podatkovnih skladišč

V tem podpoglavju bomo obravnavali različne vidike, ki jih je treba upoštevati pri modeliranju in lahko vplivajo na izbiro pristopa k modeliranju podatkovnih skladišč.

Predstavili bomo predvsem dejavnike, ki so pomembni za oblikovanje prilagodljivega in razširljivega podatkovnega skladišča, ki omogoča učinkovito pridobivanje podatkov. V praktičnem delu magistrskega dela bomo na podlagi teorije ovrednotili izbrana modela podatkovnih skladišč.

2.3.2.1 Strategija modeliranja

Pred začetkom modeliranja je pomembno, da se odločimo katera strategija modeliranja je najbolj primerna v našem primeru. Strategije delimo na modeliranje na podlagi poročanja, podatkovno usmerjeno modeliranje in poslovno usmerjeno modeliranje.

Glavni cilj modeliranja na podlagi poročanja je izgradnja podatkovnega skladišča za podporo poročanju in analizi podatkov ter hitro posredovanje poročil končnim uporabnikom. Hoberman (2009) pravi, da je ta pristop najbolj primeren za podjetja, ki jasno razumejo svoje zahteve glede poročanja in si želijo zagotoviti vpoglede v podatke v kratkem času.

Podatkovno usmerjeno modeliranje temelji na identifikaciji ključnih virov podatkov v podjetju in odnosov med njimi ter izgradnjo podatkovnega modela, ki podpira analizo teh podatkov. Ciljna arhitektura je optimizirana za zajem in shranjevanje vseh podatkov, ki bi lahko bili uporabni v prihodnosti. Ta pristop je uporaben v primerih ko imamo na voljo veliko količino podatkov, saj omogoča prepoznavanje vzorcev, ki morda niso takoj očitni, pa tudi v primerih, ko je cilj izboljšanje kakovosti podatkov, saj lahko hitro prepoznamo težave (Inmon in Hackathorn, 1994; Kimball in Ross, 2013).

Strategija poslovno usmerjenega modeliranja je pristop, ki usklajuje podatkovno skladišče s poslovnimi cilji podjetja. Cilj je uporaba podatkov za podporo strateškemu odločanju, ki podpira splošne poslovne cilje podjetja. Po Hobermanu (2009) je poslovno usmerjen pristop najprimernejši za podjetja, ki jasno razumejo svoje poslovne cilje in potrebujejo podatkovni model, ki se bo prilagajal spreminjajočim se poslovnim potrebam.

2.3.2.2 Integracija podatkov in proces ETL

Proces ETL je postopek integracije podatkov, ki vključuje ekstrakcijo podatkov iz izvornih sistemov, transformacijo podatkov in nalaganje v podatkovno skladišče (Informatica, brez datuma). Prikazan je na sliki 3.

V zadnjih letih se velikokrat omenja tudi proces, pri katerem se podatki najprej naložijo v ciljni model, šele nato se izvedejo transformacije (angl. extract, load, transform – ELT). Razlog za to je povečanje zmogljivosti shranjevanja sistemov podatkovnih skladišč in pojavom podatkovnih skladišč v oblaku, kot so Snowflake, Microsoft Azure in Amazon Redshift, zato je postalo nekaj povsem običajnega najprej naložiti neobdelane podatke ter jih nato preoblikovati neposredno v sistemu podatkovnega skladišča (Reis in Housley, 2022).

Slika 3: Proces ETL

Vir: prirejeno po Informatica (brez datuma).

Na kompleksnost procesa ETL vplivajo obseg in pogostost posodabljanja podatkov, število virov podatkov, kakovost podatkov, poslovne zahteve in kompleksnost podatkovnega modela (Anandarajan in drugi, 2004). Kompleksnost integracije podatkov iz različnih izvornih sistemov je odvisna tudi od pristopa k modeliranju podatkovnega skladišča (Schnider in drugi, 2014). Po mnenju Caserte in Kimballa (2004) lahko zapleten podatkovni model vpliva na kompleksnost transformacij in posledično daljšega razvoja ter na kompleksnost testiranja, saj je na koncu vedno potrebno testiranje za zagotovitev ustrezno preoblikovanih in naloženih podatkov v podatkovno skladišče. Posledično se poveča tudi kompleksnost vzdrževanja modela. V izogib tem težavam je priporočljivo poenostaviti podatkovni model in ga razčleniti na obvladljive dele, kjer je to mogoče.

2.3.2.3 Nove analitične zahteve

Razvoj podatkovnega skladišča je iterativen proces in na začetku razvoja velikokrat nimamo znanih vseh zahtev, kar pomeni da lahko pričakujemo nove analitične zahteve oziroma se lahko pojavijo zahteve po umestitvi novih vsebin v model. Podjetje se lahko odloči za uvažanje novih virov podatkov, kot so podatki iz novih operativnih sistemov, družbenih omrežij, senzorjev, ali pa se spremenijo vprašanja, na katera bi želeli odgovoriti. Vse to lahko vpliva na podatkovni model, zato je pomembno, da je prilagodljiv spreminjajočim se poslovnih potrebam (Ponniah, 2001).

2.3.2.4 Spremembe podatkovnih struktur

V podjetjih se najpogosteje spreminjajo procesi, aplikacije in tehnologija, zato je pomembno, da ustvarimo podatkovni model, ki ni odvisen od teh treh dejavnikov, hkrati pa zajema vsa pomembna poslovna pravila podjetja. Ker pa so nekatere spremembe vseeno neizogibne, je treba vzpostaviti način za vključitev sprememb s čim manjšim vplivom na obstoječe objekte v podatkovnem skladišču.

Spremembe podatkovnih struktur v izvornih sistemih, kot so nove tabele, dodatni atributi v tabelah ter sprememba tipa relacij, so pričakovan del vsakega agilnega projekta za razvoj podatkovnega skladišča. V takih primerih je treba ustrezno prilagoditi tudi model podatkovnega skladišča ter poskrbeti, da to ne vpliva na obstoječe podatke. Izbira pristopa k modeliranju ima lahko velik vpliv na enostavnost obravnavanja strukturnih sprememb v podatkovnem skladišču (Ponniah, 2001; Schnider in drugi, 2014).

2.3.2.5 Obvladovanje zgodovinskih zapisov

Ena izmed ključnih razlik med transakcijskim sistemom in podatkovnim skladiščem je sposobnost shranjevanja zgodovine. Po izkušnjah Rainardija (2008) transakcijski sistemi večinoma hranijo podatke do tri leta nazaj, ostale podatke, za katere predpisi zahtevajo daljše shranjevanje, arhivirajo na različne načine. Shranjevanje zgodovine je pomemben vidik skladiščenja podatkov, saj podjetjem omogoča analizo podatkov skozi čas in sprejemanje odločitev na podlagi trendov. Loshin (2013) in Inmon (2005) se strinjata, da je podpora za analizo zgodovine ena izmed ključnih prednosti, ki jih omogočajo podatkovna skladišča. Poleg tega za številne industrije veljajo zahteve za hrambo zgodovinskih podatkov.

2.3.2.6 Učinkovitost poizvedb

Hitrost delovanja poizvedb v podatkovnem skladišču je pomembna za podjetja, ki s pomočjo analitike sprejemajo odločitve na podlagi podatkov. Hitrost in učinkovitost lahko vplivata na sposobnost pravočasne analize podatkov. Na delovanje poizvedb lahko vplivajo različni dejavniki, kot so kompleksnost podatkov, podatkovni model, programska oprema baze podatkov in optimizacija poizvedb.

Način, kako so podatki organizirani in shranjeni v podatkovnem skladišču, lahko vpliva na hitrost poizvedb. Dobro zasnovan podatkovni model lahko zmanjša redundanco podatkov in kompleksnost združevanja tabel (Imhoff in drugi, 2003).

2.3.2.7 Enostavnost dostopa

Podatki v podatkovnem skladišču morajo biti dostopni končnim uporabnikom s pomočjo intuitivnih vmesnikov in orodij za poizvedovanje in analizo podatkov. Inmon (2005) izpostavlja enostavnost dostopa kot eno izmed ključnih lastnosti podatkovnih skladišč.

V večini primerov poslovni uporabniki dostopajo do podatkov, shranjenih v podatkovnih skladiščih, preko poročil in nadzornih plošč. Orodja, ki to omogočajo, so enostavna za uporabo, vendar ne omogočajo toliko fleksibilnosti kot poizvedbe, ki jih kreiramo z uporabo strukturiranega povpraševalnega jezika za delo s podatkovnimi bazami (angl. structured query language, v nadaljevanju SQL). Poleg podatkovnih analitikov, znanstvenikov in administratorjev baz, tudi napredni poslovni uporabniki, ki potrebujejo več fleksibilnosti in

kontrolo nad izvajanjem, pogosto uporabljajo SQL poizvedbe, zato je pomembno, da razumejo model podatkovnega skladišča in ga znajo pravilno uporabljati (Golfarelli in Rizzi, 2009).

3 MODELI PODATKOVNIH SKLADIŠČ

Prvi pristopi k modeliranju podatkovnih skladišč obstajajo že od devetdesetih let prejšnjega stoletja, v zadnjih letih pa je postala pomembna skrb za kakovost podatkov ter prepoznavanje pomembnosti dobrega modela podatkovnega skladišča za podjetja (Reis in Housley, 2022; Sherman, 2014).

Bill Inmon je predstavil pristop k modeliranju podatkovnih skladišč leta 1990. Pred tem se je analiza pogosto izvajala v izvornih sistemih, kar je povzročalo počasnejše delovanje transakcijskih baz, zato je bil cilj ločiti izvorni sistem od analitičnega. Ob pojavu Inmonove teorije podatkovnih skladišč, se je uporabilo obstoječe tehnike za modeliranje podatkov, torej uporaba entitetno-relacijskega modela, ki je v tretji normalni obliki. Prva generacija podatkovnih skladišč je bila tesno vezana na relacijski model. Zahteva po takšni normalizaciji je zagotavljala manj podvajanja podatkov. Podatki za poročila so bili pripravljeni preko dodatne predstavitvene plasti (angl. data marts), ki pa je lahko bila tudi denormalizirana (Hultgren, 2012; Reis in Housley, 2022).

Leta 1996 je Ralph Kimball objavil knjigo, v kateri je predstavil dimenzijski model podatkovnega skladišča, ki se manj osredotoča na normalizacijo in omogoča hitrejše iteracije in modeliranje v primerjavi z Inmonovim pristopom, vendar potencialno večjo redundanco podatkov. Prvotni namen dimenzijskega modela je torej prilagodljiva plast v podatkovnem skladišču, ki je namenjena za predstavitev in hitro analitiko podatkov (Hultgren, 2012; Kimball in Ross, 2013).

Medtem ko se Kimball in Inmon osredotočata na strukturo poslovne logike v podatkovnem skladišču, ponuja podatkovni trezor drugačen pristop k modeliranju. Model podatkovnega trezorja je v devetdesetih letih prejšnjega stoletja ustvaril Dan Linstedt. Ta pristop ločuje poslovne ključe od vsebine, kar bomo podrobno predstavili v nadaljevanju, več pozornosti pa je dobil leta 2013, ko je Linstedt v novi knjigi predstavil izboljšan model podatkovnega trezorja (Linstedt in Olschimke, 2015). Eden od zagovornikov tega pristopa je tudi Hultgren (2012), ki pravi, da so namen in cilji za različne plasti shranjevanja podatkov – operacijski sistem, podatkovno skladišče in predstavitvena plast – različni, zato je smiselno za vsako izmed njih uporabiti optimalni pristop in razviti tehnike, ki so namenjene izključno eni plasti. Kot glavni težavi pri gradnji podatkovnega skladišča izpostavlja integracijo različnih virov in vpliv sprememb na model skozi čas, zato kot glavno merilo za pristop k modeliranju izpostavlja agilnost – zmožnost enostavnega prilagajanja spremembam.

V nadaljevanju bomo podrobneje predstavili model podatkovnega trezorja in dimenzijski model, kar nam bo služilo kot izhodišče za modeliranje izbranega primera.

3.1 Model podatkovnega trezorja

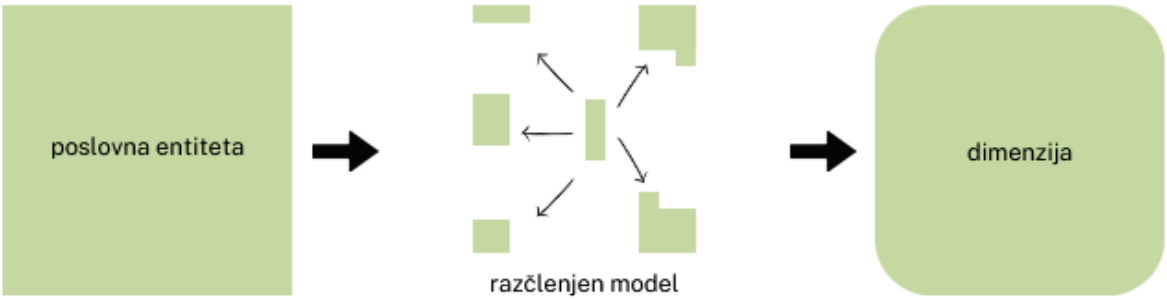
Model podatkovnega trezorja je izumil Dan Linstedt v devetdesetih letih. Temelji na kompleksnih omrežjih, podobnih tistim, ki jih najdemo v naravi, kot so možgani, organizacije in gospodarstvo (Linstedt in Olschimke, 2015).

Če ni omenjeno drugače, je vsebina tega poglavja povzeta po avtorjih Linstedt in Olschimke (2015) in Hultgren (2012).

Podatkovni trezor je poslovno usmerjen model za skladiščenje podatkov, katerega cilj je čim bolj natančno predstaviti poslovne procese. Poslovni ključi so temelj podatkovnega trezorja, saj so pomembni za integracijo, povezovanje in dostop do informacij tudi pri vsakodnevnem poslovanju.

Temeljna ideja podatkovnega trezorja je poenotena razčlenitev, kar pomeni, da poslovne entitete razdelimo na več enostavnih objektov, ki imajo jasno opredeljeno strukturo in vlogo v modelu. Kot smo že omenili v prejšnjem poglavju, je namen podatkovnih skladišč združevanje vsebine iz različnih virov na isti imenovalec, kar lahko zveni kontradiktorno ideji razčlenitve. Vendar pa prav razčlenitev in poenotenje pravil za izvedbo te omogoča večjo fleksibilnost in agilnost. Ideja je vidna na sliki 4, kjer na levi strani vidimo poslovno entiteto, v sredini pa je razčlenjen model, ki je del podatkovnega skladišča. Podatki v takšni obliki pogosto niso primerni za končnega uporabnika, zato jih v zadnjem koraku združimo.

Slika 4: Temeljna ideja modela podatkovnega trezorja



Vir: prirejeno po Hultgren (2012).

3.1.1 Tipi objektov

Podatkovni trezor temelji na treh osnovnih tipih objektov. V vozliščih so shranjeni poslovni ključi in so na ta način ločeni od preostalega modela, v povezavah so shranjeni odnosi med vozlišči, v satelitih pa je shranjena ostala vsebina, ki dodatno opisuje vozlišče ali povezavo. V naslednjih podpoglavjih bodo koncepti razloženi bolj podrobno.

3.1.1.1 Vozlišče

Namen vozlišč (angl. hub) je shranjevanje poslovnih ključev. Poslovni ključi morajo biti unikatni, lahko pa so sestavljeni tudi iz več polj. Vozlišče je sestavljeno iz poslovnega ključa, zgoščenega ključa, vira in datuma nalaganja poslovnega ključa. Vozlišče ne vsebuje opisnih informacij in tujih ključev.

Poslovni ključ enolično identificira en primerek poslovne entitete. Poleg enoličnosti je pomembno tudi, da se vrednost poslovnega ključa nikoli ne spremeni. Primer poslovnega ključa je šifra kupca, številka računa, šifra artikla … Včasih pravila za določanje poslovnih ključev ne omogočajo več enoličnosti, zato podjetja uvedejo sistemsko generirane ključe, ki jih uporabljajo za prepoznavanje poslovnih entitet.

Model podatkovnega trezorja je treba pripraviti na način, da povečamo hitrost obdelave združevanja tabel, saj poizvedbe zahtevajo veliko več združevanj kot pri ostalih modelih podatkovnih skladišč. To je razlog za uvedbo **zgoščenega ključa** (angl. hash key), ki je kreiran na podlagi poslovnega ključa in predstavlja primarni ključ vozlišča ter se uporablja kot tuji ključ v povezavah in satelitih. Za izračun zgoščenih ključev se pogosto uporablja zgoščevalni algoritem MD5. Pomembna lastnost teh ključev je ponovljivost, kar pomeni da se isti poslovni ključ vedno preračuna v isti zgoščeni ključ. Ta lastnost je precej pomembna pri polnjenju podatkovnega skladišča, saj vrstni red polnjenja ni več tako pomemben, čeprav velja da je združevanje tabel preko takih ključev lahko počasnejše v primerjavi s ključi na osnovi celih števil.

Datum nalaganja (angl. load date) označuje datum, ko se poslovni ključ prvič pojavi v podatkovnem skladišču. Datum je sistemsko kreiran skozi proces ETL. Vsi podatki, ki se naložijo znotraj istega cikla polnjenja, morajo imeti nastavljen isti datum. To nam omogoča tudi lažje iskanje tehničnih napak, ki se lahko pojavijo ob nalaganju podatkov.

Za vsak poslovni ključ določimo tudi **vir** (angl. record source), ki določa iz katerega izvornega sistema je ključ prišel in nam omogoča sledljivost do tega sistema. V kolikor se poslovni ključ pojavi v več izvornih sistemih, se kot vir zapiše glavni vir oziroma vir, ki se je prvi prenesel. Linstedt svetuje, da se izogibamo posplošenem opisu vira, kot je »SAP«, ki označuje celoten poslovno informacijski sistem (angl. enterprise resource planning – ERP), in uporabimo bolj podroben opis, ki zajema tudi posamezne module izvornega sistema, npr. »SAP.FINANCE.GL«. Na tak način imamo natančnejši podatek in lažje sledimo podatkom do izvornega sistema.

Priporočljivo je, da v vozliščih uporabljamo isti vrstni red polj za celoten model, saj je s tem model bolj pregleden in omogoča lažje vzdrževanje. Linstedt predlaga, da je na prvem mestu zgoščeni ključ, ki se uporablja tudi za sklicevanje na poslovno entiteto v satelitih in povezavah, sledi datum nalaganja, vir in nazadnje še poslovni ključ, ki ima ime določeno glede na entiteto, ki jo identificira.

3.1.1.2 Povezava

Povezava (angl. link) se uporablja za predstavitev vseh odnosov v modelu podatkovnega trezorja. Namenjena je modeliranju transakcij, poslovnih odnosov in hierarhij. Povezuje dve ali več vozlišč oz. poslovnih ključev. Čeprav je lahko povezava med ključi aktualna le določen čas, se časovna komponenta ne shranjuje, saj cilj povezav ni shranjevanje vsebine.

Poleg vseh **tujih zgoščenih ključev** oz. referenc na vozlišča, ki jih povezuje, se v povezavah shranjuje tudi **zgoščeni ključ**, ki je izračunan iz kombinacije vseh tujih poslovnih ključev in predstavlja primarni ključ tabele. Ta ključ je uporabljen tudi v satelitih, ki lahko povezavam dodajo vsebino. Poleg teh polj je treba tudi tukaj dodati še informacijo o **datumu nalaganja** in **viru**.

V splošnem ločimo tri vrste povezav glede na njihov namen. Hierarhične povezave (angl. hierarchical links) se uporabljajo za modeliranje nadrejenih-podrejenih odnosov. Namesto da bi modelirali vsako raven hierarhije kot ločeno vozlišče, je priporočljivo uporabiti eno vozlišče in hierarhično povezavo. Tipičen primer hierarhije je organizacijska struktura v podjetju. Nezgodovinske povezave (angl. nonhistorized links), ki so znane tudi kot transakcijske povezave, so tiste, ki vsebujejo informacije o dogodkih, ki so se zgodili in se nikoli ne bodo spremenili. Tipična primera sta shranjevanje senzoričnih podatkov in knjiženje računov, ki jih je mogoče spremeniti ali preklicati samo z drugim knjiženjem. Ostalim navadnim povezavam, ki opisujejo odnose med dvema ali več vozlišči, pravimo relacijske povezave (angl. relational link) (Dratwa, 2023).

Vse povezave imajo določeno kardinalnost mnogo proti mnogo (M:N) in s tem zagotavljajo prilagodljivost, saj spremembe poslovnih pravil ne vplivajo na model. Z vidika modeliranja se lahko modelar osredotoči samo na prepoznavanje povezav in se ne ukvarja s kardinalnostjo le-teh. Nivo granulacije je določen s številom vozlišč, ki jih tabela povezuje.

Ob morebitnih spremembah poslovnih zahtev v podatkovnem trezorju ni sprejemljivo spreminjanje obstoječih struktur. Ob spremembi nivoja granulacije v povezavah torej ne spreminjamo obstoječe tabele, temveč ustvarimo novo tabelo povezave, ki zadošča novim poslovnih zahtevam. Pomembno je, da se od tistega trenutka naprej novi zapisi nalagajo v novo tabelo in ne več v prejšnjo verzijo, ter da v nadaljnjih analizah uporabljamo obe tabeli.

3.1.1.3 Satelit

Sateliti (angl. satellite) so tabele, v katerih shranjujemo vse opisne podatke o poslovnih entitetah, katerih ključi so shranjeni ločeno v vozliščih, in o povezavah in transakcijah, katerih ključi so shranjeni ločeno v povezavah. En satelit je lahko povezan na natanko eno vozlišče ali povezavo.

Ker se opisni atributi v izvornih sistemih lahko pogosto spreminjajo, je namen satelitov tudi sledenje tem spremembam. Primarni ključ satelita ne predstavlja le zgoščeni ključ vozlišča ali povezave, temveč tudi datum nalaganja. Ob vsaki spremembi enega od stolpcev se torej v satelit zapiše nova vrstica, ki vsebuje enak zgoščeni ključ in nov datum nalaganja. Na tak način lahko sledimo spremembam, ki so se zgodile v izvornih sistemih. V satelitih je vsa vsebina, ki se lahko spreminja v času.

Linstedt in Olschimke (2015) priporočata, da se opisni atributi, ki so vezani na eno poslovno entiteto ali povezavo, ne shranjujejo vsi v istem satelitu. Namesto tega predlagata dva načina za ločevanje satelitov. Prva razdelitev je glede na izvorni sistem, kar pomeni, da za vsak izvorni sistem kreiramo svoj satelit. Prednost tega pristopa je predvsem enostavno dodajanje novih virov podatkov brez spreminjanja obstoječega modela. Poleg tega ni treba prilagajati podatkovnih tipov novih vhodnih podatkov, da bi ustrezali obstoječem modelu. Takšen način omogoča tudi paralelnost nalaganja podatkov, saj se lahko hkrati nalagajo iz različnih izvornih sistemov v različne satelite. Druga razdelitev na satelite pa je glede na pogostost sprememb atributov. Ideja je, da v enem satelitu shranjujemo atribute, ki se redko spreminjajo, v drugem pa tiste, ki se spreminjajo pogosteje. Velikokrat se namreč dogaja, da se le nekateri izmed stolpcev pogosto spreminjajo, ker pa se ob vsaki spremembi doda nova vrstica v model, lahko to čez čas zasede veliko prostora.

Vsak satelit vsebuje **datum nalaganja**, **vir**, **tuji zgoščeni ključ**, ki se poveže na vozlišče ali povezavo, opcijsko pa še podatek o datumu izvlečka (angl. extract date), ki predstavlja datum in čas, ko je bil podatek zajet na izvornem sistemu, in podatek o zgoščeni vrednosti (angl. hash difference). Ta predstavlja zgoščeno vrednost vseh opisnih podatkov v satelitu. Namen tega atributa je hitro primerjanje obstoječih vrstic z novimi in prepoznavanje spremenjenih vrstic. V prvih verzijah modela podatkovnega trezorja se je kot obvezen podatek omenjal tudi končni datum veljavnosti (angl. load end date), ki je predstavljal datum, ko je vrstica v satelitu postala neveljavna, torej ob spremembi vsaj enega izmed atributov in posledično dodane nove vrstice v satelit. Tak atribut pride prav ob poizvedovanju iz satelitov in hitro pridobitev stanja na določen datum. V dopolnjeni verziji modela podatkovnega trezorja je izpostavljen pristop, ki dovoljuje le dodajanje novih zapisov (angl. insert-only), zato uporaba tega atributa ni več priporočljiva. Namesto tega končne datume simuliramo z uporabo funkcije LEAD pri poizvedovanju podatkov.

Predstavljeni tipi objektov tvorijo osnovni podatkovni trezor (angl. raw data vault), ki predstavlja model podatkovnega skladišča v podjetju. V literaturi se večkrat pojavlja koncept poslovnega trezorja (angl. business vault), ki je dodatna neobvezna plast v podatkovnem skladišču in izhaja iz osnovnega modela. Uporablja se v primerih kompleksnih poslovnih pravil in omogoča lažjo uporabo pri tvorjenju končnih struktur, namenjenih uporabnikom. Struktura modela sledi principom modeliranja podatkovnega trezorja, od osnovnega modela pa se razlikuje v tem, da so v poslovnem trezorju upoštevana tudi poslovna pravila. Pogost primer uporabe je konsolidacija poslovnih ključev iz različnih virov, vpeljava agregacij in

izračunov posnetkov stanj v določenem trenutku ter vpeljava pravil za kakovost podatkov (Cuba, 2020; Linstedt in Olschimke, 2015).

3.1.2 Proces modeliranja

Cuba (2020) zagovarja proces t. i. mob podatkovnega modeliranja (angl. mob modelling), kjer sodelujejo poslovni analitiki, ki prevajajo poslovne zahteve, podatkovni inženirji, ki se osredotočajo na modeliranje po pravilih podatkovnega trezorja, strokovnjaki za izvorne sisteme, ki poznajo izvorne podatke ter operativni izvajalci, ki so odgovorni za izvedbo dogovorjenih nalog. Gre za vloge, ki jih lahko opravlja ena ali več oseb. Pomembno je, da so vsi udeleženci prisotni na skupnih delavnicah, kjer rešujejo težave in zagotavljajo skladnost s standardi modeliranja podatkov. Delavnice zahtevajo pripravljenost vseh udeležencev, njihov namen pa je reševati konkretna vprašanja in zagotoviti napredek pri modeliranju podatkov.

Proces modeliranja se začne z opredelitvijo osnovnih poslovnih entitet in njihovih poslovnih ključev, ki se bodo shranjevali v vozlišča. Sledi identifikacija odnosov med ključi, ki jih modeliramo v povezave. Zadnji korak pa je določanje vsebine poslovnim entitetam in povezavam.

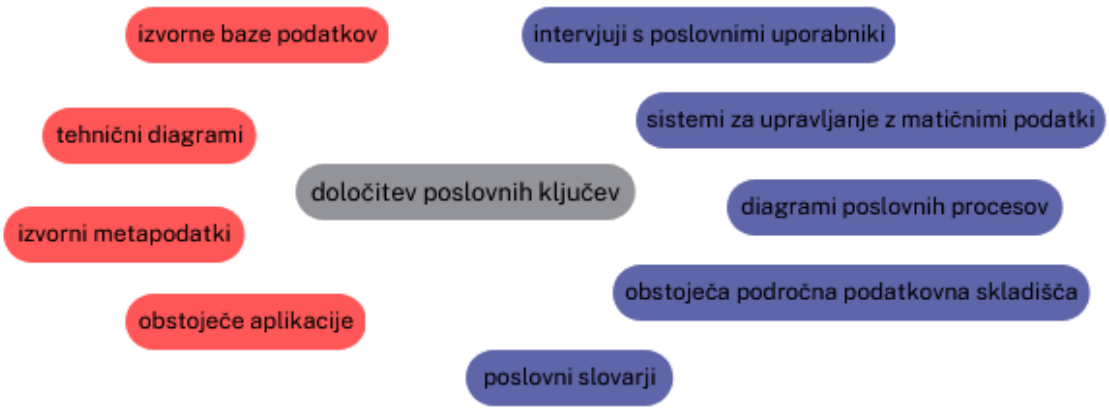
V nadaljevanju bomo omenjene korake predstavili bolj podrobno.

1. Opredelitev poslovnih entitet in njihovih ključev

Za identificiranje poslovnih entitet in njihovih ključev je potrebna analiza poslovnih procesov podjetja. Pomembno je, da se pogovarjamo s poslovnimi uporabniki, ki dobro poznajo poslovne procese in razumejo širši pomen njihovega poslovanja. Pogosto zadošča pogovor s poslovnimi uporabniki, saj za iskanje v operativnih sistemih uporabljajo šifre, ki enolično določajo vsebino, zato jih lahko uporabimo tudi kot poslovne ključe. Pomagamo si lahko tudi z analizo obstoječih operativnih podatkovnih baz, pregledom dokumentacije, obstoječih OLAP kock in podobno. Treba se je zavedati, da v tem koraku oblikujemo glavne komponente za podatkovni trezor, zato je priporočljivo vložiti veliko truda in ne hiteti. Stremeti moramo k prepoznavanju naravnih poslovnih ključev, ki temeljijo na poslovanju in niso odvisni od izvornih sistemov. Na sliki 5 so modro označene primarne metode za določitev vozlišč, rdeče pa metode, ki so nam lahko v pomoč, če modro označene ne zadoščajo.

Rezultat prvega koraka je nabor jasno definiranih poslovnih entitet ter njihovih poslovnih ključev, ki bodo sestavljali vozlišča.

Slika 5: Osnova za določitev vozlišč



Vir: prirejeno po Hultgren (2012).

2. Opredelitev povezav med poslovnimi entitetami

Naslednji korak je zaznava poslovnih odnosov med entitetami. Povezave lahko povezujejo dve ali več entitet. Za prepoznavanje povezav velja enako kot za prepoznavanje vozlišč, torej izhajati morajo iz potreb poslovnih procesov. Ker so v tem koraku vozlišča že identificirana, je naslednji korak razmišljanje o odnosih med obstoječimi vozlišči. Tako je rezultat drugega koraka nabor jasno definiranih povezav med poslovnimi entitetami, ki bodo v modelu zastopane v tabelah povezav.

3. Opredelitev vsebine

Zadnji korak je opredelitev vsebinskih atributov, ki opisujejo poslovno entiteto ali povezavo. Za določitev atributov je smiselno pregledati izvorne podatkovne baze v izogib izpustitvi kakšnega atributa v podatkovnem skladišču. Ko so atributi določeni, je smiselno razmisliti tudi o njihovi morebitni ločitvi v modelu. Kot smo že omenili, obstaja več dejavnikov, ki lahko vplivajo na razdelitev atributov na več satelitov.

3.2 Dimenzijski model

Cilj dimenzijskega modeliranja je ustvariti model, ki je enostaven, razumljiv ter omogoča hiter in učinkovit dostop do podatkov za analitiko. Velja za prednostni pristop k modeliranju podatkovnih skladišč.

3.2.1 Tipi objektov

Ključna objekta, ki se pojavljata v dimenzijskem modelu sta dimenzija in tabela dejstev, skupaj pa tvorita zvezdno shemo. V nadaljevanju bodo koncepti podrobneje razloženi.

3.2.1.1 Dimenzija

Dimenzijske tabele (angl. dimension tables) shranjujejo vsebino, ki je povezana s poslovnimi dogodki. Ko jih združimo s tabelo dejstev, lahko podajo odgovor na vprašanja kdo, kaj, kje, kdaj, kako in zakaj, v povezavi z dogodkom. Po navadi vsebujejo več atributov in manj vrstic kot tabele dejstev (Kimball in Ross, 2013).

Dimenzija je definirana s **primarnim ključem**, ki služi kot osnova za referenčno integriteto s tabelami dejstev. Primarnemu ključu pravimo surogatni ali nadomestni ključ, ki ga pogosto ustvarimo kar v podatkovnem skladišču in predstavlja celo število, ki nima vsebinskega pomena. Poleg tega dimenzija vsebuje tudi **naravni ključ** izvornega sistema (Sherman, 2014).

Atributi dimenzijskih tabel imajo ključno vlogo v podatkovnem skladišču, saj so vir za skoraj vse omejitve, ki se uporabljajo v poizvedbah za poročila. Kimball in Ross (2013) izpostavljata, da je pomembno zagotoviti, da naziv atributov sestavljajo cele besede in ne zgolj okrajšave, ki jih morda ne poznajo vsi uporabniki. Atributi sicer niso zgolj besedilne oblike, lahko so tudi numerični. Za sledenje sprememb skozi čas se uporabljata **datum začetka veljavnosti** (angl. effective date from) in **datum konca veljavnosti zapisa** (angl. expiration date).

Dimenzijske tabele pogosto predstavljajo hierarhične odnose. V takšnih primerih so hierarhične opisne informacije redundantno shranjene z namenom enostavne uporabe in zmogljivosti poizvedb, kar je ena izmed ključnih razlik v primerjavi s tretjo normalno obliko. (Reis in Housley, 2022).

3.2.1.2 Tabela dejstev

Tabele dejstev (angl. fact table) vsebuje dejanske, kvantitativne podatke in podatke, povezane s poslovnimi dogodki ali transakcijami. Podatki v tabeli dejstev so nespremenljivi, ker se nanašajo na dogodke (Reis in Housley, 2022).

Tabela dejstev je sestavljena iz treh vrst stolpcev. Prvi so **tuji ključi**, ki kažejo na primarne ključe dimenzijskih tabel, ki so povezane s tabelo dejstev z namenom omogočanja poslovne analize. Tuji ključi ne smejo vsebovati ničelnih vrednosti, razlog za to pa je enostavnejše poizvedovanje, saj v primeru združevanja tabel preko ničelnih stolpcev lahko izgubimo določene vrstice, kar lahko privede do zavajajočih rezultatov analize. Množica tujih ključev v večini primerov enolično definira vrstico v tabeli dejstev. V izvornih tabelah velikokrat zasledimo enolične identifikatorje transakcij, kot so številka prodajnega naloga, številka računa ipd. Ker so ti podatki vezani izključno na poslovno transakcijo, jih modeliramo v tabele dejstev, imenujemo pa jih **degenerirane dimenzije**. Tretji tip stolpcev so dejanske **mere**, vezane na poslovni dogodek ali transakcijo, kot sta znesek ali količina naročila. Pomembna lastnost je aditivnost, saj milijone vrstic iz tabel dejstev po navadi seštevamo za

namene analitike. Poznamo tri vrste mer, to so aditivne, poladitivne in neaditivne. Aditivne se lahko enostavno seštevajo po vseh dimenzijah, npr. količina kupljenih izdelkov se sešteje po strankah, trgovinah, izdelkih in datumih. Poladitivne se seštevajo le po nekaterih dimenzijah, npr. stanje na bančnem računu konec meseca, ki se sešteva po strankah, ne seštevamo pa jih za več mesecev, saj bi v tem primeru bilo bolj logično gledati povprečje stanj. Neaditivne pa se ne seštevajo po nobenih dimenzijah, to so npr. cene na enoto, razmerja, temperature. Glede na vrste mer delimo tudi tabele dejstev po istem vrstnem redu na transakcijske, periodične in zbirne posnetke stanj.

V tabelah dejstev se izogibamo shranjevanju besedilnih informacij, razen v primeru ko je besedilo enolično za vsako vrstico, kar se redko pojavi (Sherman, 2014).

Kimball in Ross (2013) kot eno izmed temeljnih načel dimenzijskega modeliranja navajata zagotavljanje enotnega nivoja granulacije za celotno tabelo dejstev, kar določa najnižji nivo podrobnosti poslovnega dogodka ali transakcije.

3.2.1.3 Zvezdna shema

Zvezdna shema predstavlja podatkovni model podjetja. Sestavlja jo tabela dejstev, ki je obdana z več dimenzijami. Razmerje med dimenzijo in tabelo dejstev je ena proti mnogo. Avtorji Reis in Housley ter Sherman (2022; 2014) navajajo, da je zvezdna shema primerna za poizvedovanje in analitiko ter je enostavna, zato jo uporabniki lažje razumejo in uporabljajo. Prav tako pa so orodja za prikaz vizualizacij v večini zgrajena na način, da dobro podpirajo uporabo zvezdnih shem. Večinoma je končni rezultat model, ki vsebuje več zvezdnih shem, saj podjetja želijo analizirati več različnih vsebin. Pomembno je, da se dimenzije, ki se lahko uporabi v več zvezdnih shemah, ponovno uporabi. Na tak način skrbimo za integriteto podatkov in se izogibamo več različnim poslovnim definicijam.

V klasičnih primerih ima vsaka dimenzija samo eno vrednost, ki ustreza eni vrstici v tabeli dejstev. Vendar obstajajo primeri, ko lahko eni vrstici v tabeli dejstev ustreza več vrednosti v dimenziji, npr. pacient lahko pri eni zdravstveni obravnavi dobi več različnih diagnoz. To rešujemo z modeliranjem tabel, ki se obnašajo kot **most** (angl. bridge) med tabelo dejstev in dimenzijo (Kimball in Ross, 2013).

Zvezdna shema predstavlja kompromis med normaliziranim in denormaliziranim modelom. Večina podatkov se nahaja v tabelah dejstev, ki so normalizirane in s tem zagotavljajo enostavno vzdrževanje in minimalno redundanco. Po drugi strani pa so dimenzije denormalizirane tabele, ki združujejo atribute, ki so v izvornih sistemih shranjeni v različnih tabelah (Sherman, 2014).

3.2.2 Proces modeliranja

Proces dimenzionalnega modeliranja po Kimballu in Rossu (2013) vključuje štiri korake: določitev poslovnega procesa, določitev nivoja granulacije, identifikacija dimenzij in identifikacija tabel dejstev. Celoten model se razvija preko več interaktivnih delavnic s poslovnimi uporabniki, pomembno pa je, da pri postopkih modeliranja sodelujejo ljudje, ki poznajo poslovanje in potrebe podjetja.

1. Določitev poslovnih procesov

Poslovni procesi so osnovne aktivnosti, ki jih podjetje izvaja, na primer naročila, nabava, plačila. Večinoma so podprte z operativnimi sistemi. Ekipa modelarjev mora že na začetku razumeti potrebe in cilje podjetja. Na skupnih delavnicah s poslovnimi uporabniki je treba ugotoviti, kakšni procesi odločanja se trenutno izvajajo in kakšne so njihove analitične potrebe. Hkrati je pomembno, da se vključijo tudi strokovnjaki za izvorne sisteme, ki pomagajo pri razumevanju realnih obstoječih podatkov. Cilj prvega koraka torej zajema razumevanje poslovnih potreb in analitičnih zahtev ter identifikacijo procesov, ki jih je treba modelirati.

2. Določitev nivoja granulacije

Eden izmed ključnih korakov pri dimenzijskem modeliranju je določanje nivoja granulacije. S tem natančno določimo kaj predstavlja ena vrstica v tabeli dejstev. Nanaša se na najnižjo raven, ki jo zajema opredeljen poslovni proces in je usklajena s tem, kar je podprto v izvornih operativnih sistemih, ki beležijo poslovne dogodke. Razumevanje in jasna opredelitev nivoja granulacija sta ključna za uspešno modeliranje.

3. Identifikacija dimenzij

V tem koraku je treba identificirati vsebino, ki obkroža poslovni dogodek, ter določiti atribute, ki bodo uporabljeni za filtriranje in združevanje. Dimenzije določajo atribute, ki odgovarjajo na vprašanja kdo, kaj, kje, kdaj in zakaj glede na poslovni dogodek v tabeli dejstev. Pri tem si lahko precej pomagamo z upoštevanjem že določenega nivoja granulacije, saj to hkrati določa vse potrebne dimenzije. Paziti je treba, da ustrezno določimo tudi primarni ključ dimenzij, saj mora biti povezava med dimenzijo in tabelo dejstev ena proti mnogo.

4. Identifikacija tabel dejstev

V tem koraku identificiramo vse mere, ki izhajajo iz poslovnega dogodka in so v večini primerov numerične. Včasih na prvi pogled ni popolnoma jasno, ali numerični atribut spada v dimenzijo ali tabelo dejstev. Pri določitvi so lahko v pomoč vprašanja, ali gre za konstantno številko, ali se pogosto spreminja, ali je vezana izključno na en poslovni dogodek ipd. Znotraj tabele dejstev so dovoljene samo mere, ki so skladne z določenim nivojem

granulacije. Vzpostavimo tudi povezave med tabelo dejstev in dimenzijskimi tabelami preko tujih ključev.

Za boljše razumevanje lahko rezultate vseh štirih točk prikažemo v bus matriki (angl. bus matrix), ki je orodje za načrtovanje in komuniciranje modela podatkovnega skladišča. Vsaka vrstica v bus matriki predstavlja poslovni proces, medtem ko vsak stolpec ustreza določeni dimenziji. Na preseku vrstice in stolpca označimo, ali je dimenzija povezana z določenim poslovnim procesom. Na ta način je mogoče hitro ugotoviti, da različni poslovni procesi uporabljajo skupne dimenzije, ki jih je smiselno standardizirati (Kimball, 2008).

4 MODELIRANJE IZBRANEGA PRIMERA

V tem poglavju bomo predstavili poslovni problem podjetja, ki je ponudnik turističnih storitev. Podjetje se ne želi javno izpostavljati, zato ga bomo označili kot izbrano podjetje.

Na podlagi poslovnega primera bomo pripravili dimenzijski model in model podatkovnega trezorja, nato pa bomo analizirali vidike modeliranja podatkovnih skladišč. Oba podatkovna modela uvrščamo med logične modele glede na nivo abstrakcije modeliranja.

Za risanje podatkovnih modelov bomo uporabili orodje dbdiagram.io, ki je brezplačno orodje za risanje diagramov, z izjemo modela podatkovnega trezorja. Za slednjega bomo uporabili jezik za modeliranje Visual Data Vault. Gre za jezik, ki uporablja standardizirane simbole za modeliranje podatkovnega trezorja (Scalefree, 2010).

4.1 Predstavitev poslovnega problema

Izbrano podjetje se ukvarja z izvajanjem turističnih dejavnosti, prevladujeta predvsem nastanitvena in gostinska dejavnost.

Soočajo se z nerazumevanjem finančnih podatkov zaradi pomanjkanja tehnične podpore. Finančni izkazi, kot sta izkaz poslovnega izida in bilanca stanja, se preračunavajo iz glavne knjige in omogočajo podatke na najnižjem nivoju, ki ga omogoča sistem za podporo računovodskim procesom. Zaposleni v oddelkih računovodstva in kontrolinga morajo pogosto raziskovati izvor posameznih postavk v izkazih. V primeru podrobnih analiz jim ne zadošča nivo granulacije, ki je podprt v izvornem sistemu za računovodstvo. To povzroča velike časovne izgube, saj se morajo pogosto v raziskovanje vključiti tudi informatiki, ki s pomočjo algoritmov in primerjav lahko pridejo do bolj podrobnih nivojev granulacije.

Za lažje razumevanje bomo v nadaljevanju opisali trenuten tok podatkov. Ob poslovnem dogodku, npr. prodaji, nastanejo zapisi v sistemu za podporo prodaji, kjer se zabeležijo vse podrobnosti transakcije, kot so številka prodajnega dokumenta, prodajalec, produkt in prodajni kanal. Nato se zapisi v agregirani obliki avtomatsko prenesejo v sistem za

računovodstvo oz. knjiženje. Ob tem se lahko več transakcij združi v eno knjižbo, zato podatki niso več na voljo na tako nizkem nivoju, kot so v prodajnem sistemu.

Poglejmo si primer, ki je prikazan v tabelah 1 in 2. Ob obisku gostinskega objekta kupec zjutraj naroči kavo in čaj, popoldne pa še eno kavo z mlekom. Ob tem nastaneta dva računa v sistemu za podporo prodaji. Kljub temu, da je vsaka izmed treh postavk uvrščena v konto za prihodke od gostinstva (pijače), se razlikujejo v podrobnostih, kot so produkt, prodajni kanal in podobno. Ob prenosu v sistem za knjiženje se zapisi združujejo na najnižjem nivoju računovodskega sistema, ki je sestavljen iz konta in stroškovnega mesta. To pomeni, da bodo omenjene tri postavke v računovodskem sistemu zabeležene agregirano v eni knjižbi oz. zapisu. Ob analizi podatkov iz računovodskega sistema torej ne dostopamo več do informacij o produktih, prodajnih kanalih in podobno. To povzroča, da uporabniki iz oddelkov računovodstva in kontrolinga ob analizi podatkov iz glavne knjige ne razumejo povsem izvora nekaterih številk. V takih primerih anomalije ugotavljajo predvsem informatiki, ki razumejo celoten tok podatkov in dostopajo do vseh procesov toka podatkov. Ob tem se poraja vprašanje, ali lahko enostavno analiziramo podatke direktno iz sistema za podporo prodaji – odgovor je ne, saj se zaradi različnih poslovnih pravil ne prenašajo vsi zapisi v sistem za knjiženje in hkrati ne obstaja enolično pravilo za prenos, torej se nabor zapisov razlikuje. Sistem za podporo prodaji namreč vsebuje tudi dobavnice in še veliko ostalih izjem, ki še niso del glavne knjige in jih ni potrebno upoštevati v analitiki, obenem pa glavna knjiga predstavlja dejansko stanje in je osnova za vse nadaljnja poročanja.

Tabela 1: Zapisi v sistemu za podporo prodaji

Datum dogodka	**Številka dokumenta**	**Številka dokumenta NAV**	**Šifra produkta**	**Neto vrednost**	**Količina**
2022-01-03	477352	P202203011562	33 (čaj)	1,25	1
2022-01-03	477352	P202203011562	35 (kava)	1,74	1
2022-01-03	477357	P202203011562	41 (kava z mlekom)	1,98	1

Vir: lastno delo.

Tabela 2: Zapisi v računovodskem sistemu

Datum knjiženja	**Številka dokumenta**	**Šifra konta**	**Dimenzija**	**Znesek**
2022-01-03	P202203011562	760211	S10551 (Kavarna pritličje)	4,97

Vir: lastno delo.

Cilji, ki so si jih zastavili v izbranem podjetju, so vzpostavitev podatkovnega skladišča, ki bo združevalo podatke iz različnih virov in bo služilo kot osnova za pripravo poročil in nadzornih plošč. Prva prioriteta je zagotovitev finančnih podatkov iz glavne knjige, ki služijo

kot osnova za pripravo finančnih izkazov in zagotovitev prodajnih podatkov, ki bodo omogočili globje razumevanje finančnih podatkov. Cilj je torej združitev dveh trenutno najbolj pomembnih virov za analitiko in priprava podatkovnega modela na način, da bo enostavno razširljiv tudi za ostale vire in vsebine. S tem bi dosegli večjo samostojnost poslovnih uporabnikov, ki bi sami lažje razumeli kaj se dogaja v ozadju številk brez pomoči informatikov. Poleg naštetih dejavnikov, si v izbranem podjetju želijo, da bo nov model podatkovnega skladišča enostaven, da bodo poleg poslovnih uporabnikov, ki ga bodo uporabljali v analitične namene, lahko zagotovili tudi kader, ki bo v prihodnje skrbel za razširitve in dopolnitve modela čim bolj samostojno. Želja po modernizaciji in po razširljivem modelu podatkovnega skladišča je tudi posledica strateške odločitve širjenja poslovanja v tujino v naslednjih petih letih.

4.2 Podatkovni model v izvornem sistemu

V tem podpoglavju bomo opisali podatkovna modela v obeh izvornih sistemih, ki smo ju zaznali kot pomembna za reševanje poslovnega problema.

4.2.1 Finance

Izbrano podjetje uporablja Microsoft Dynamics NAV za podporo računovodskim procesom.

Tabela 3: Opis izvornih tabel – finance

Naziv tabele	**Opis**
Glavna_knjiga	Glavna knjiga (knjižbe)
Konto	Konti
Tip_konta	Tipi kontov
Dimenzijska_mnozica	Vrednosti osnovnih dimenzij (stroškovno mesto, profitni center, destinacija), ki tvorijo organizacijsko hierarhijo
Dimenzijske_vrednosti	Opis vrednosti osnovnih dimenzij
Kontakt	Kontakti (kupci in dobavitelji)
Kontakt_povezava	Povezava med kontaktom in kupcem oz. dobaviteljem
Kupec	Kupci
Knjizna_skupina_kupca	Knjižne skupine kupcev
Dobavitelj	Dobavitelji
Knjizna_skupina_dobavitelja	Knjižne skupine dobaviteljev
Osnovno_sredstvo	Osnovna sredstva
Razred_OS	Razredi osnovnih sredstev
Podrazred_OS	Podrazredi osnovnih sredstev
Lokacija_OS	Lokacije osnovnih sredstev

Vir: lastno delo.

Tabela 3 prikazuje seznam in kratek opis izvornih tabel, ki smo ga pripravili s pomočjo informatikov v izbranem podjetju.

Slika 6 prikazuje diagram izvornega podatkovnega modela za finance. Tipov atributov v nadaljevanju ne bomo posebej obravnavali, na sliki pa so prikazani zaradi omejitev orodja za izris modela.

Slika 6: Izvorni podatkovni model – finance

Vir: lastno delo.

4.2.2 Prodaja

Dostop do osnovnih tabel in podatkovnega modela sistema za podporo prodajnim procesom nam izbrano podjetje zaradi težav pri omejitvah pravic, kompleksnosti in nepoznavanja, ni zagotovilo. Pridobili smo vpogled v podatkovni model, ki je bil v preteklosti pripravljen kot osnova za analitične poizvedbe, saj so se uporabniki na tabele povezali prek Excela. Ta model bo služil kot vir za razvoj podatkovnega skladišča. Model vsebuje tabele, ki so opisane v tabeli 4.

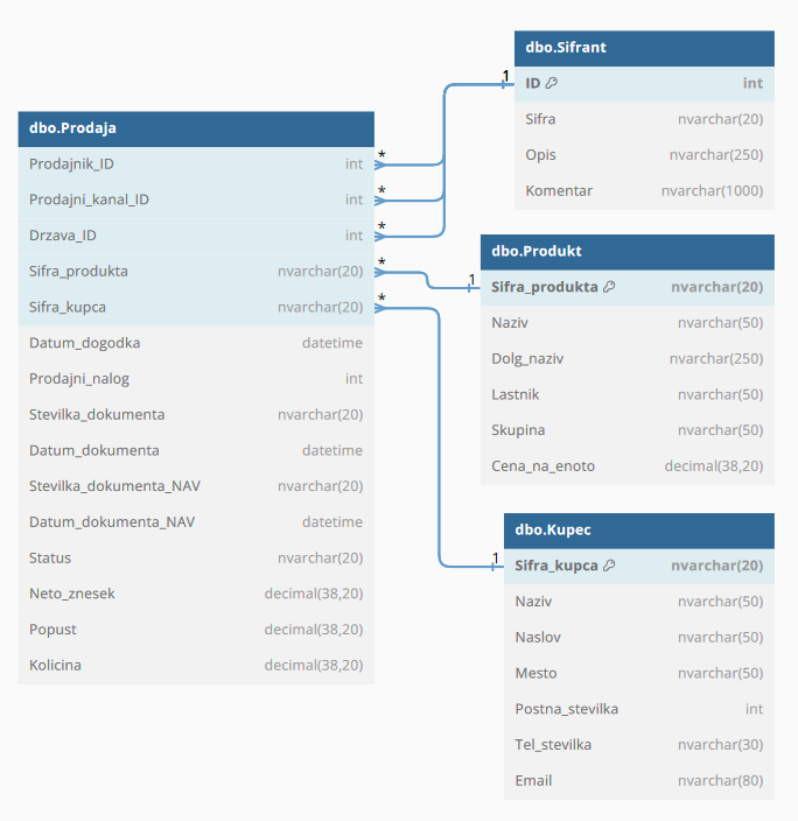
Tabela 4: Opis izvornih tabel – prodaja

Naziv tabele	**Opis**
Prodaja	Tabela prodajnih dogodkov oz. transakcij
Sifrant	Tabela vseh šifrantov, ki vsebuje šifre, nazive in opise za prodajalce, prodajne kanale in države.
Produkt	Produkti
Kupec	Kupci

Vir: lastno delo.

Slika 7 prikazuje diagram izvornega podatkovnega modela za prodajo.

Slika 7: Izvorni podatkovni model – prodaja



Vir: lastno delo.

4.3 Model podatkovnega trezorja

Modeliranja podatkovnega trezorja se lotimo na način, ki smo ga opisali v teoretičnem delu, torej v treh korakih.

4.3.1 Opredelitev poslovnih entitet in njihovih ključev

Za opredelitev poslovnih entitet smo izvedli delavnice s ključnimi uporabniki iz oddelka računovodstva in kontrolinga. Na delavnicah so bili poleg ključnih uporabnikov prisotni še podatkovni modelarji, poslovni analitiki in informatiki, ki poznajo izvorne podatke. Iz pogovora smo hitro razbrali osnovne entitete, ki so del ključnih poslovnih procesov in so pomembne za analizo in poročanje. Skupaj smo pregledali obstoječa poročila iz področja financ in prodaje ter se pogovorili o namenu poročil, pri čemer so uporabniki jasno izpostavili, katere entitete so za njih pomembne in katere še manjkajo. Zapisali smo seznam poslovnih entitet ter se dogovorili za poimenovanja. Ob tem se je pojavilo nekaj dvomov, saj oddelka uporabljata več različnih nazivov za enako vsebino, zato smo poimenovanja poenotili.

Opredelitev poslovnih ključev za entitete iz finančnega sistema ni predstavljala težav, saj uporabniki iz oddelka računovodstva dnevno uporabljajo sistem Microsoft Dynamics NAV, kjer za iskanje zapisov v glavni knjigi uporabljajo predvsem enolične poslovne ključe. Za opredelitev poslovnih ključev entitet, ki so izključno del prodajnega sistema, smo si pomagali z analizo obstoječih poročil v excel-tabelah in dodatnimi delavnicami, ki smo jih naknadno izvedli z informatiki.

Kljub temu, da so bili v večini primerov poslovni ključi tudi fizično zapisani kot primarni ključi tabel v izvornih bazah, se v splošnem velikokrat uporabljajo tako imenovane sekvence, ki avtomatsko generirajo enolične vrednosti primarnih ključev. Zagovorniki modeliranja podatkovnega trezorja v literaturi izpostavljajo pomembnost upoštevanja poslovnih ključev in ne avtomatsko generiranih ključev v izvornih tabelah. Glede na izkušnje, lahko rečemo, da se v izvornih sistemih pojavijo tudi nepojasnjeni zapisi, ki jih nihče od poslovnih uporabnikov ne zna razložiti, npr. kupec brez šifre, ker se je uporabnik zmotil ob vnosu v sistem. To se lahko zgodi vedno, kadar ne obstajajo tehnične omejitve, kot so primarni ključi, ki to lahko preprečijo, zato predlagamo, da se po končani opredelitvi poslovnih ključev preveri tudi njihova tehnična ustreznost.

Nabor poslovnih entitet, ki smo jih zaznali so knjižba, prodaja, kupec, dobavitelj, kontakt, konto, organizacijska struktura, osnovno sredstvo, prodajalec, produkt, prodajni kanal in država.

Pri tem je pomembno poudariti, da so transakcije, kot sta knjižba in prodaja, dejansko poslovni koncepti in jih zato vedno obravnavamo kot samostojne poslovne entitete (Hultgren, 2012).

4.3.2 Opredelitev povezav med poslovnimi entitetami

Po končani opredelitvi poslovnih entitet lahko razmislimo o odnosih med njimi. Pogovorili smo se tudi o poslovnih odnosih med entitetami, izhajali smo iz potreb poslovnih procesov in kasneje tudi preverili, kako so te povezave informacijsko podprte v izvornih sistemih. Nabor povezav med poslovnimi entitetami, ki smo jih zaznali so:

- Pravna oseba je lahko hkrati kupec in dobavitelj. Čeprav se njena šifra kupca in šifra dobavitelja razlikujeta, je pravna oseba vedno zapisana kot en kontakt. Kontakt torej določa enolično šifro za pravno osebo, hkrati določa tudi povezavo do dobavitelja in kupca. Gre za relacijsko povezavo.
- Knjižba je transakcija, ki zabeleži podatke o kupcu, dobavitelju, kontu, osnovnem sredstvu in najnižjem nivoju organizacijske strukture. Glede na namen jo uvrščamo med nezgodovinske povezave.
- Prodaja je transakcija, ki zajema podatke o prodajalcu, produktu, prodajnem kanalu, in državi. Podobno kot pri knjižbah gre za nezgodovinsko povezavo.
- Vsaka knjižba nastane na podlagi ene ali več prodajnih transakcij. Njuna medsebojna povezava spada med relacijske.
- Organizacijska struktura je hierarhično sestavljena iz več nivojev (stroškovno mesto, profitni center, destinacija), zato bo imela povezavo sama nase. Uvrščamo jo med hierarhične povezave. Pri tem ne pozabimo upoštevati še možnost reorganizacije, kar pomeni, da moramo za sledenje časovnim veljavnostim hierarhije dodati satelit na to povezavo.

4.3.3 Opredelitev vsebine

Pri opredelitvi vsebinskih atributov smo skupaj s poslovnimi uporabniki pregledali atribute na izvornih podatkovnih bazah in se pogovorili o njihovem pomenu. To je bilo izredno pomembno predvsem za sistem Microsoft Dynamics NAV, ki je generičen sistem za računovodstvo in vsebuje veliko število atributov, ki jih izbrano podjetje ne potrebuje in ne uporablja. Skupaj smo se pogovorili tudi o smiselnosti ločitve atributov v modelu.

Vsaka izmed poslovnih entitet in povezav ima atribute, ki jo podrobno opisujejo. V nadaljevanju bomo opisali tiste, kjer smo opazili posebnosti.

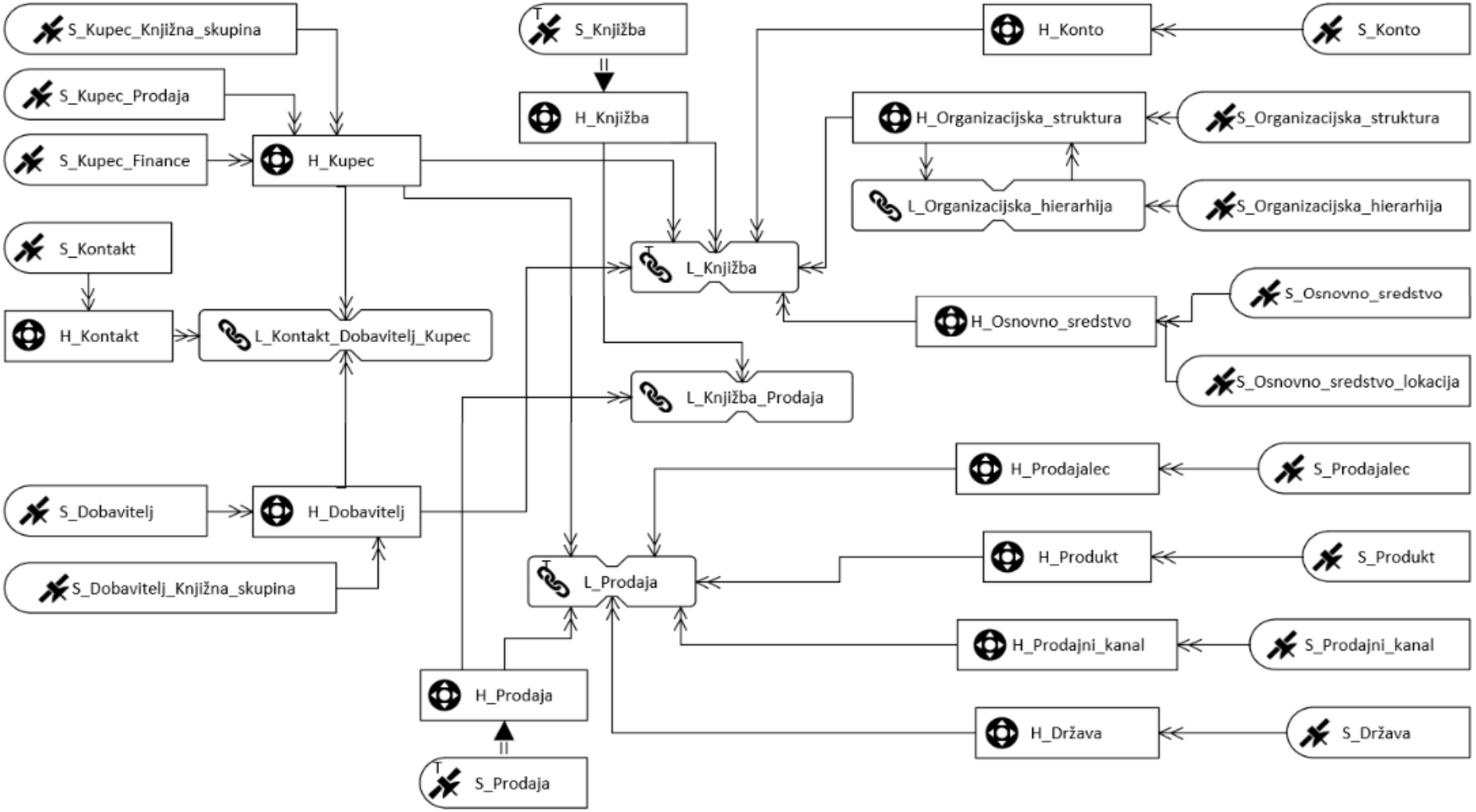
- Kupec je poslovna entiteta, ki obstaja v obeh izvornih sistemih. Po pogovorih z uporabniki in pregledu izvornih tabel, smo ugotovili, da se nekateri podatkovni tipi ne ujemajo (npr. v računovodskem je kraj določen kot »1000 Ljubljana«, v prodajnem sistemu pa sta ločeni polji za poštno številko in kraj). Atribute bomo zato razdelili glede na izvorni sistem. Poleg tega smo zaznali, da se pogosto spreminja atribut knjižna skupina kupca, medtem ko se ostali atributi spreminjajo redko (npr. naslov, kraj, telefonska številka), zato jih bomo dodatno razdelili glede na pogostost sprememb. To pomeni, da bomo atribute kupca razdelili v tri sklope, kar bodo v modelu predstavljali trije sateliti.

- Dobavitelj obstaja samo v računovodskem sistemu. Podobno kot pri kupcu, se tudi pri dobavitelju lahko pogosto spreminja atribut knjižna skupina dobavitelja, ostali atributi so bolj statični. Razdelili jih bomo torej v dva sklopa glede na pogostost sprememb.
- Atribute osnovnega sredstva bomo prav tako razdelili v dva sklopa glede na pogostost sprememb, saj se lokacije po mnenju poslovnih uporabnikov precej bolj pogosto spreminjajo kot ostali atributi.

Na sliki 8 je prikazan osnovni model podatkovnega trezorja, ki zajema vse zgoraj navedene elemente. V prihodnjem poglavju o primerjavi modelov bomo izhajali iz tega modela.

Vseeno pa razmislimo tudi o potencialnih izboljšavah, ki jih je mogoče vključiti v dodatno plast, t. i. poslovni trezor. Entitete, ki prihajajo iz različnih virov, bi lahko združili, kar bi olajšalo uporabo teh tabel. Glede na potrebe poročil bi bilo treba oceniti, ali je smiselna vpeljava agregiranih tabel, ki združujejo podatke iz prodaje in financ. V nekaterih primerih, zlasti tam, kjer je več satelitov povezanih z enim vozliščem, kot na primer pri dobaviteljih in osnovnih sredstvih, je smiselno razmisliti o izračunu posnetkov stanj. To je pomembno za pregled po časovnih obdobjih, kjer moramo upoštevati kombinacijo veljavnosti vseh satelitov. Prav tako bi se bilo smiselno posvetovati z uporabniki o uvedbi pravil za zagotavljanje kakovosti podatkov.

Slika 8: Model podatkovnega trezorja



Vir: lastno delo.

4.4 Dimenzijski model

Kot smo navedli v teoriji, Kimballov proces dimenzionalnega modeliranja vključuje štiri ključne korake.

4.4.1 Določitev poslovnih procesov

Ne glede na izbiro pristopa k modeliranju je ključnega pomena razumevanje poslovnih ciljev in procesov podjetja. V podpoglavjih 4.1 in 4.3 smo te vidike že podrobneje predstavili. Kot osrednja procesa smo identificirali knjiženje in prodajo.

4.4.2 Določitev nivoja granulacije

Naslednji korak je določitev nivoja granulacija za oba poslovna procesa, prodajo in knjiženje.

Zapis v glavni knjigi oz. knjižba nastane za natanko določen datum in uro knjiženja, konto, stroškovno mesto, ki je hkrati najnižji nivo organizacijske strukture in določa tudi pripadajoč profitni center in destinacijo, ter vir, ki lahko predstavlja dobavitelja, kupca ali osnovno sredstvo. Ob tem nastane enolična številka dokumenta v glavni knjigi.

Zapis ob prodajnem procesu zajema podatke o datumu dogodka, kupcu, prodajalcu, produktu, prodajnem kanalu, državi, prodajnem nalogu. Ob tem nastane enolična številka dokumenta v prodajnem sistemu. Ko je transakcija potrjena, se ji pripiše še podatek o pripadajoči številki dokumenta v glavni knjigi in datumu knjiženja, kar pa ne spremeni obstoječega nivoja granulacije.

4.4.3 Identifikacija dimenzij

Nivo granulacije nam določa tudi vse potrebne dimenzije. Za dogodek knjiženja so to datum, konto, stroškovno mesto kot najnižji nivo organizacijske hierarhije, dobavitelj, kupec in osnovno sredstvo. Za prodajni dogodek so to datum, kupec, prodajalec, produkt, prodajni kanal in država. Številka prodajnega naloga in dokumenta sta enolični za vsako transakcijo in hkrati ne določata dodatnih vsebinskih atributov, zato ju ne modeliramo v ločeni dimenziji.

Vsebinski atributi dimenzij bodo prikazani v nadaljevanju na sliki 9. Kljub temu, da v logičnem modelu ni treba določiti tipov polj, so na sliki prikazani zaradi omejitev orodja za izris modela.

4.4.4 Identifikacija tabel dejstev

Mere, ki se nanašajo na dogodek knjiženja, vključujejo znesek v dobro, znesek v breme in znesek. Na prodajni dogodek pa se nanašajo znesek, popust in količina. Kljub temu, da je tudi podatek o ceni na enoto produkta numeričen, ni vezan zgolj na en poslovni dogodek, temveč na točno določen produkt, zato bo to del dimenzije produktov.

Poleg mer in tujih ključev na prej določene dimenzije, so v tabelah dejstev tudi degenerirane dimenzije. Pri knjiženju to zajema številko vnosa v glavni knjigi in številka dokumenta v glavni knjigi. Potrebno je še polje, ki vsebuje komentar, in se nanaša na točno določeno vrstico. Pri prodaji pa to obsega številko prodajnega naloga, številko prodajnega dokumenta in številko dokumenta v glavni knjigi.

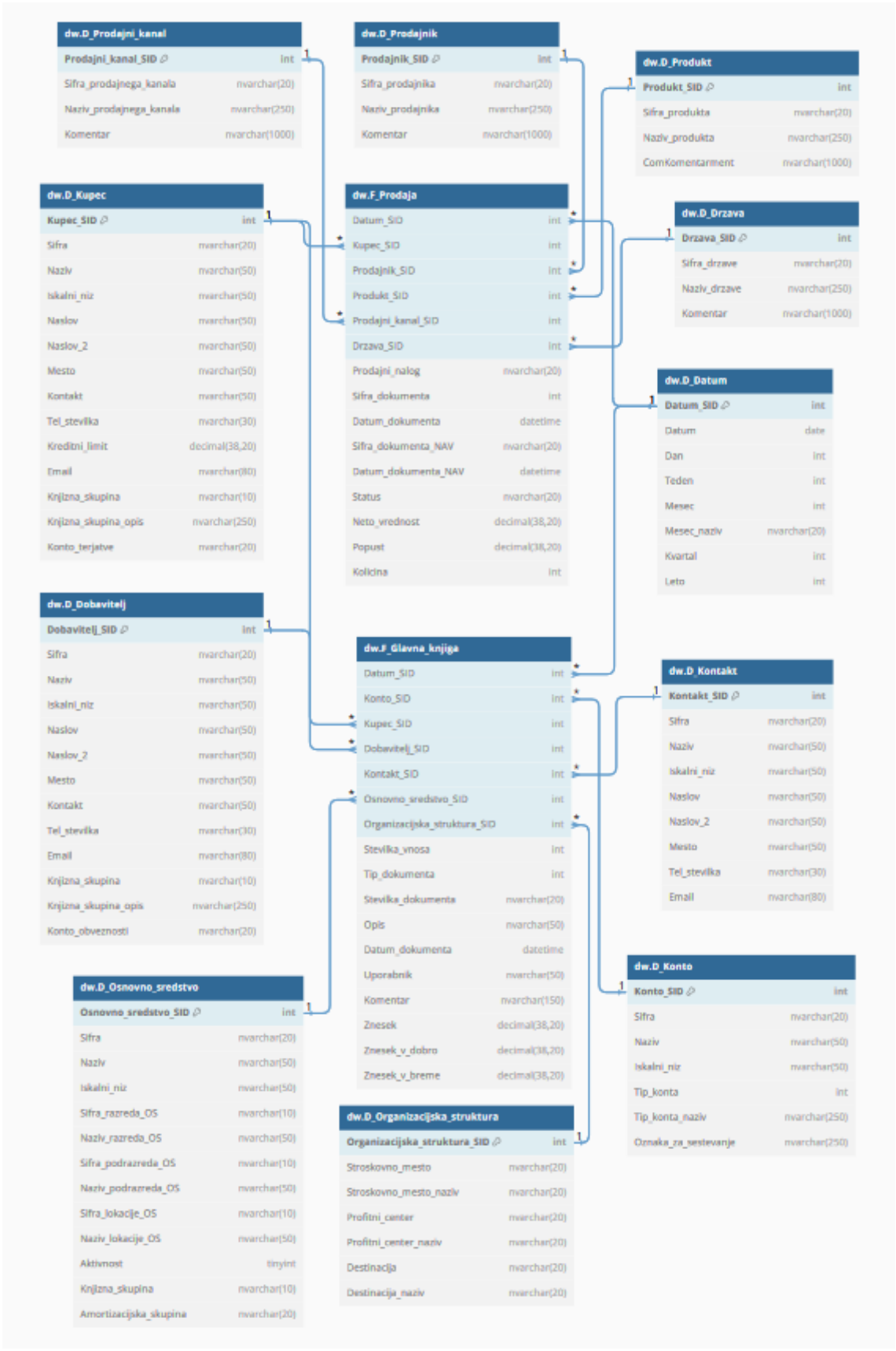
V tabeli 5 je prikazana analiza še z orodjem bus matrike. Opazimo lahko, da bosta v modelu dve skupni dimenziji: kupec in organizacijska struktura, zato morata biti pripravljeni na način, da ustrezno podpreta oba poslovna procesa.

Tabela 5: Bus matrika

Poslovni proces/ dimenzija	Prodajni kanal	Prodajalec	Kupec	Dobavitelj	Osnovno sredstvo	Konto	Organizacijska struktura	Kontakt	Datum	Država	Produkt
Knjiženje (glavna knjiga)			x	x	x	x	x	x	x		
Prodaja	x	x	x						x	x	x

Vir: lastno delo.

Slika 9: Dimenzijski model



Vir: lastno delo.

4.5 Analiza in primerjava modelov

V nadaljevanju bomo primerjali oba pripravljena modela glede na predstavljene vidike modeliranja.

4.5.1 Strategija modeliranja

Spoznali smo tri različne tipe strategij modeliranja. Glede na zastavljene cilje izbranega podjetja, ki vključujejo podporo finančnih in prodajnih procesov, razširljivost modela in samostojnost poslovnih uporabnikov, predlagamo uporabo poslovno usmerjenega modeliranja, ki v ospredje postavlja poslovne procese podjetja. V tem primeru je bolj kot izbor modeliranja pomemben sam proces analize, torej osredotočenost na poslovne procese in vključenost poslovnih uporabnikov. Tako dimenzijski kot tudi model podatkovnega trezorja lahko namreč zgradimo tako, da ustrezno podpirata vse procese. Kljub temu, da smo pri modeliranju primarno sledili poslovnim procesom, smo v naslednjih fazah uporabili tudi elemente modeliranja na podlagi poročanja, saj smo za lažje razumevanje procesov analizirali že obstoječa poročila.

Kljub temu poslovno usmerjeno modeliranje ni vedno mogoče in smiselno. Če si predstavljamo, da bi bil cilj podjetja hitra izdelava vnaprej določenih poročil s področja financ in prodaje, bi lahko izbrali strategijo modeliranja na podlagi poročanja. Iz vidika uporabe orodij za vizualizacije podatkov, bi v tem primeru bil bolj primeren dimenzijski model, ki je najbolj primeren za uporabo v orodjih za podporo poslovni inteligenci (angl. business intelligence, v nadaljevanju BI). Po navadi takšni modeli zajemajo samo podatke, potrebne za odgovore na aktualna analitična vprašanja, in jih je treba razširiti ob pojavu novih. Glede na izkušnje menimo, da je tak pristop ustrezen za hitre rešitve in ne za razvoj celovitega modela podatkovnega skladišča.

Ob pomanjkanju poznavanja poslovnih procesov, kar se rado zgodi ob fluktuaciji zaposlenih, je smiselna uporaba podatkovno usmerjenega modeliranja, saj temelji bolj na tehničnih kot vsebinskih faktorjih. V našem primeru tega pristopa ne bi izbrali, saj se cilji izbranega podjetja s tem ne ujemajo, hkrati v podjetju zagotavljajo intenzivno vključenost poslovnih uporabnikov. Poleg tega uporabljajo generični sistem za finance, kjer se shranjujejo podatki, ki jih ne uporabljajo in ne potrebujejo v podatkovnem skladišču.

4.5.2 Integracija podatkov in proces ETL

Pri dimenzijskem modeliranju zajema proces ETL celoten postopek prenosa podatkov iz izvornih sistemov do dimenzijskega modela, kar vključuje tudi transformacijo podatkov v želeno obliko in uporabo poslovnih pravil. Iz tega sledi, da morebitne spremembe poslovnih pravil povzročijo tudi spremembe v samem procesu ETL. Pri modeliranju podatkovnega trezorja pa proces ETL vključuje prenos podatkov iz izvornih sistemov in nalaganje v

osnovni podatkovni trezor, kjer se na podlagi enostavnih pravil podatki razvrstijo med vozlišča, povezave in satelite in kjer se shranjujejo nespremenjeni podatki (Hospodka, 2022; Yessad in Labiod, 2016). Izjema pri tem so t. i. trda poslovna pravila (angl. hard rules), ki vključujejo uskladitev podatkovnih tipov, odstranitev začetnih presledkov in podobno (Cuba, 2020).

Kljub temu pa so tudi v modelu podatkovnega trezorja nujne določene transformacije podatkov glede na poslovna pravila, vendar so del druge podatkovne plasti, ki ji pravimo poslovni trezor. Poglejmo primer za kupce. V dimenzijskem modelu imamo eno dimenzijo, ki je skupna za kupce iz vseh virov, kar pomeni, da moramo pri polnjenju zagotoviti ustrezno integracijo in upoštevanje poslovnih pravil za kupce. V modelu podatkovnega trezorja pa nam ni treba storiti nič, saj so vsebinski atributi ločeni v dveh satelitih. Ta del bi vseeno morali reševati v poslovnem trezorju, ki je po navadi kreiran s pomočjo pogledov (angl. views) nad podatkovnim skladiščem, kar sicer lahko kasneje vpliva na zmogljivost poizvedb. Proces ETL je torej v tem primeru skoraj popolnoma neodvisen od transformacij in glede na izkušnje lahko trdimo, da transformacije precej vplivajo na uspešnost prenosa podatkov, saj smo v praksi opazili, da se velikokrat prenos prekine ali pa vsaj izvaja dlje časa zaradi neupoštevanja poslovnih pravil.

Model podatkovnega trezorja zagotavlja enostavno integracijo različnih virov podatkov. To je še posebej pomembno v primerih, ko je treba združiti veliko različnih virov in jih jasno ločiti med seboj (Krneta D., 2020). Ker ima lahko poslovni objekt več satelitov, lahko en satelit predstavlja en vir podatkov (Naamane in Jovanovic, 2016).

Schalkwyk (2014) je v svoji raziskavi primerjal hitrost nalaganja podatkov v model podatkovnega trezorja in dimenzijski model. Eksperiment je razdelil na dva dela, najprej je preveril čas nalaganja vseh podatkov, nato pa je testiral prenos spremenjenih podatkov. V model podatkovnega trezorja so se vsi podatki naložili v 225 sekundah, v dimenzijski model pa v 852 sekundah. Prenos spremenjenih podatkov se je v modelu podatkovnega trezorja izvajal 155 sekund, v dimenzijskem pa 172 sekund. V tem primeru se je torej izkazalo, da je nalaganje podatkov hitrejše v modelu podatkovnega trezorja.

Model podatkovnega trezorja zagotavlja učinkovito vzporedno nalaganje podatkov zaradi neodvisnosti tabel in uporabe zgoščenih ključev, v dimenzijskem pa je mogoče vzporedno nalagati le tabele, ki niso odvisne med seboj (Giebler in drugi, 2019).

Tudi Naamane in Jovanovic (2016) poudarjata, da je ena izmed ključnih prednosti modela podatkovnega trezorja sposobnost vzporednega nalaganja podatkov. Ker so minimizirane odvisnosti med tabelami, se lahko večina tabel polni hkrati. Gluchowski (2021) je mnenja, da je vzporedno nalaganje velika prednost pred ostalimi modeli, hkrati pa ugotavlja, da nalaganje podatkov sledi preprostim in konsistentnim vzorcem, kar olajša avtomatizacijo tega postopka, čemur prispeva tudi dejstvo, da se podatki naložijo v model podatkovnega trezorja brez večjih transformacij, v dimenzijskem modelu pa se upoštevajo že poslovna

pravila. Subotic in drugi (2014) trdijo, da vzporedno nalaganje omogoča hitrejši skupni čas nalaganja, kar je še posebej pomembno pri zagotavljanju pravočasne osvežitve podatkov.

Pri integraciji podatkov velja omeniti tudi orodja, ki podpirajo posamezni pristop. Leta 2021 je podjetje Data Vault Alliance razvilo program certificiranja orodij, ki omogočajo avtomatizacijo pri modeliranju podatkovnega trezorja. Pridobljen certifikat pomeni, da orodje sledi standardom modeliranja podatkovnega trezorja in da izpolnjuje pogoje za avtomatizacijo generiranja kode. Trenutno je takih orodij sedem, med drugim so to WhereScape, Erwin by Quest, dFakto ipd. WhereScape podpira tudi dimenzijsko modeliranje, prav tako tudi Erwin v okviru produkta Erwin Data Modeler, fokus orodja dFakto pa je na modeliranju podatkovnega trezorja. V praksi smo pogosto opazili uporabo orodja dbt (angl. data build tool), ki je odprtokodno orodje za transformacijo podatkov. Omogoča tudi enostavno verzioniranje, testiranje in dokumentiranje podatkovnih modelov. Najbolj priljubljen odprtokodni paket v okviru dbt pa je AutomateDV, ki ponuja funkcije za lažji razvoj modela podatkovnega trezorja. Glede na omenjene funkcionalnosti orodja dbt, bi ga lahko uporabili tudi za ostale pristope k modeliranju, tudi dimenzijsko, vendar ne obstajajo dodatni paketi v takšni obliki kot za modeliranje podatkovnega trezorja. Glede na to, da je dimenzijsko modeliranje pogosto uporabljen pristop za podatkovna skladišča, obstaja veliko orodij, ki ga podpirajo. V splošnem se velikokrat pojavljata orodji SSAS (SQL Server Analysis Services) oziroma oblačna verzija AAS (Azure Analysis Services), ki podpirata izdelavo tabelaričnega dimenzijskega modela. Pri izbiri orodij za avtomatizacijo je poleg dejavnikov, kot so podpora izbranem pristopu k modeliranju, hitrost razvoja, dokumentacija in skupnost, pomemben seveda tudi stroškovni vidik (DataVaultAlliance, brez datuma; McIntyre, 2023).

4.5.3 Nove analitične zahteve

V nadaljevanju bomo pogledali, kakšne so posledice novih analitičnih zahtev za oba pripravljena modela podatkovnega skladišča.

V izbranem podjetju uporabljajo aplikacijo za upravljanje odnosov s strankami (angl. customer relationship management, v nadaljevanju CRM), kjer se nahajajo podrobne informacije o strankah. Predpostavimo lahko, da bodo zaradi potreb po boljšem razumevanju strank in kreiranju novih ponudb, želeli v prihodnosti dodati v podatkovno skladišče tudi podatke iz aplikacije CRM.

Razmislimo, kaj bi se zgodilo v primeru novega vira o kupcih. V aplikaciji CRM se med drugim nahajajo demografski podatki, ki so pomembni za razumevanje obnašanja kupcev in jih ni v drugih sistemih. To so npr. spol, raven izobrazbe, poklic in datum rojstva.

Če želimo razširiti dimenzijski model z omenjenimi atributi, je treba dodati nove atribute v dimenzijo kupca (D_Kupec), kar je vidno na sliki 10. Pri tem je treba paziti, da ne izgubimo

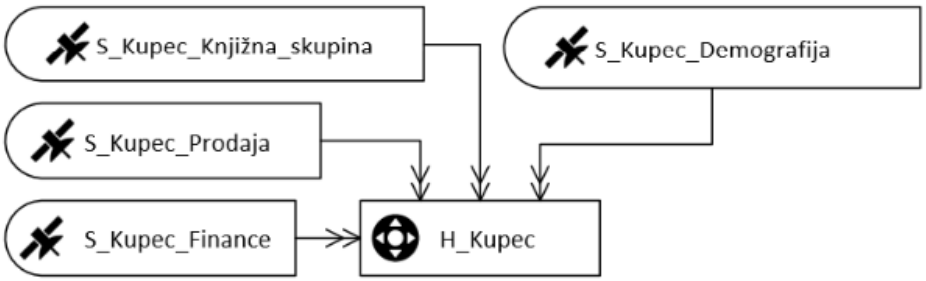
zgodovine zapisov in hkrati prilagoditi procese nalaganja v dimenzijo, tako da vključuje tudi nove atribute.

Slika 10: Končno stanje dimenzije kupca

Vir: lastno delo.

Razširitev modela podatkovnega trezorja poteka vedno na način, da se ne dotikamo obstoječih objektov, zato bi dodali nov satelit S_Kupec_Demografija, ki bi vseboval nove atribute, kot prikazuje slika 11.

Slika 11: Končno stanje vozlišča in satelitov za kupca



Vir: lastno delo.

V tem primeru nove analitične zahteve v modelu podatkovnega trezorja povzročijo dodaten objekt, v dimenzijskem modelu pa razširitev obstoječega objekta. Dodajanje novih objektov deluje enostavneje, saj ne vplivamo na obstoječe podatke, v primeru pogostega dodajanja

novih virov ali atributov tako narašča tudi število objektov, kar lahko poveča kompleksnost modela.

V literaturi nismo zaznali nedvoumnih stališč glede optimalnega števila satelitov v modelu podatkovnega trezorja. Menimo, da na to med drugim vplivajo tudi uporabljena tehnologija in orodja, količina podatkov in avtomatizacija procesov. Model je treba razumeti in ga uporabljati, pa tudi vzdrževati ter posodabljati dokumentacijo. Število satelitov mora zato predstavljati kompromis med izpolnjevanjem poslovnih zahtev in ohranjanjem obvladljivosti ter učinkovitosti.

4.5.4 Spremembe podatkovnih struktur

Omenili smo že, da se najpogostejše spremembe podatkovnih struktur pojavijo v obliki novih tabel, dodatnih atributov v tabelah in spremembah tipa relacije.

Za začetek poglejmo, kaj se zgodi v primeru spremembe tipa relacije v izvornem sistemu. Pri razvoju obeh modelov podatkovnih skladišč smo predpostavili, da šifra kontakta določa natanko eno šifro kupca in dobavitelja. Nismo pa predvideli, da je lahko en kupec ali dobavitelj zapisan pod več kontakti. Primer zapisov v izvorni tabeli je prikazan v tabeli 6. Kljub temu, da podatka o kontaktu nimamo zapisanega ob vsaki knjižbi, je pomemben za pregled terjatev in obveznosti hkrati.

Tabela 6: Zapisi v izvorni tabeli Kontakt_povezava

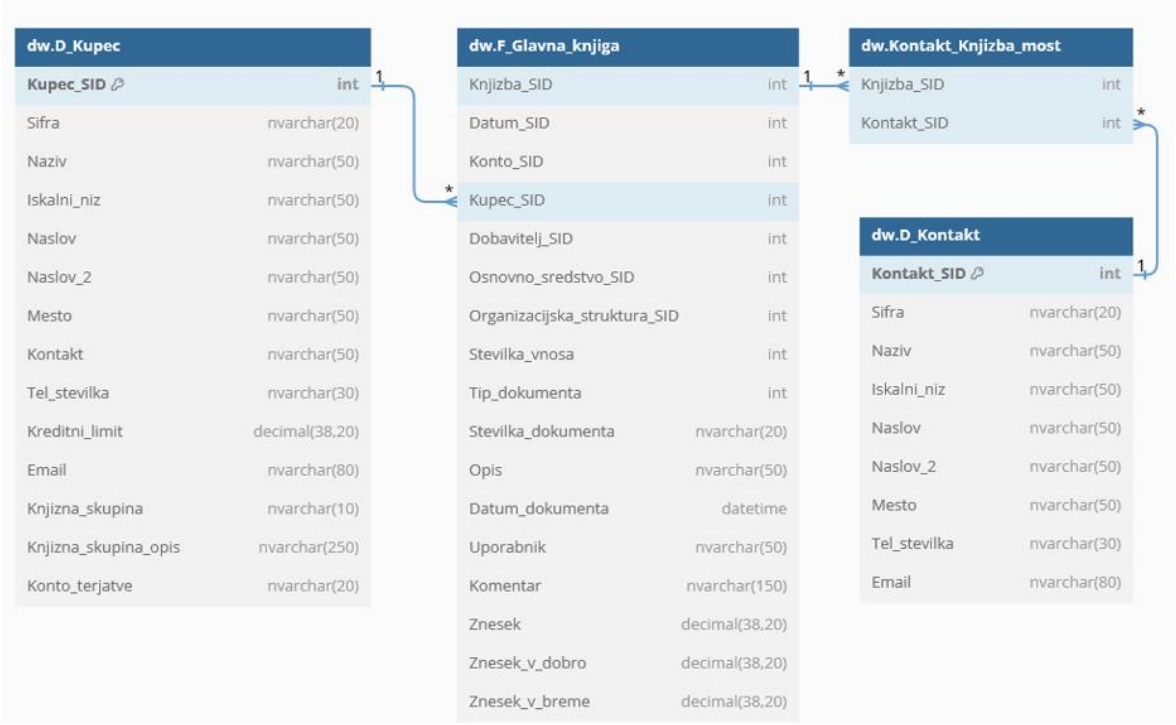
Šifra kontakta	Poslovna oznaka	Šifra
123	KUP	C12
123	DOB	D5
124	KUP	C12

Vir: lastno delo.

Poglejmo vpliv spremembe na dimenzijski model. V pripravljenem modelu smo tabelo dejstev povezali z dimenzijo kontaktov z uporabo kardinalnosti ena proti mnogo (1:N). Takšna povezava po spremembi tipa relacije ne bo več mogoča, saj lahko eno knjižbo pripišemo več kontaktom. Za reševanje takih relacij dodamo most oz. vmesno tabelo med tabelo dejstev in dimenzijo kontaktov, kot je prikazano na sliki 12.

Sprememba kardinalnosti ne vpliva na model podatkovnega trezorja, saj je treba že od začetka vse relacije obravnavati kot mnogo proti mnogo (M:N).

Slika 12: Dimenzijski model po spremembi tipa relacije



Vir: lastno delo.

Poglejmo še primer nove tabele oz. entitete. Predpostavimo, da je sistem za podporo prodaji uvedel oznako za tip plačila, ki se zabeleži ob vsaki prodajni transakciji.

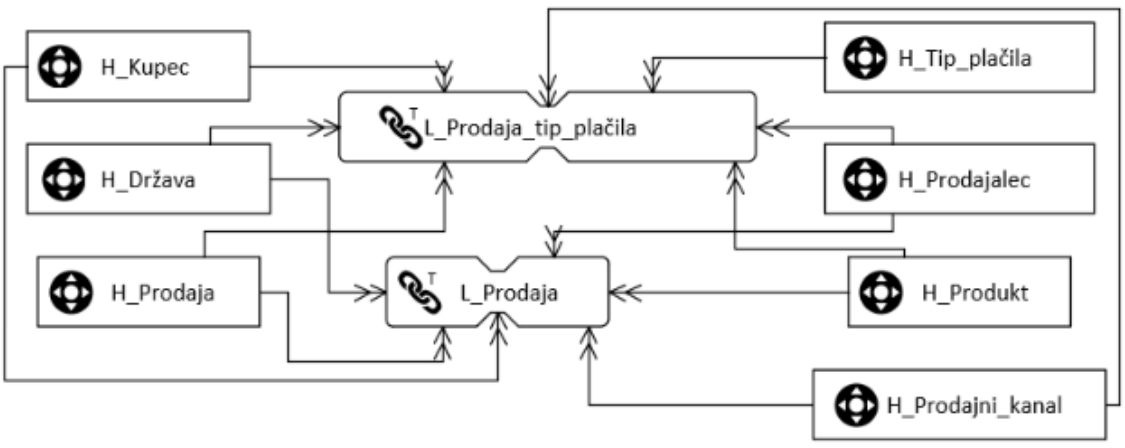
Razmislimo, na kakšen način bi to lahko vplivalo na dimenzijski model. Kreirati bi bilo treba novo dimenzijo, npr. D_Tip_placila, ki bi vsebovala seznam možnih tipov plačila in pripadajoče vsebinske atribute. Poleg tega bi dodali tudi tuji ključ v tabelo dejstev in ju med seboj povezali, kot je prikazano na sliki 13. V primeru, da gre samo za eno oznako, ki ne določa dodatnih atributov za filtriranje in združevanje, bi se lahko atribut obnašal kot degenerirana dimenzija v tabeli dejstev.

V modelu podatkovnega trezorja bi bilo smiselno pripraviti novo vozlišče, npr. H_Tip_plačila, in pripadajoč satelit, če obstajajo dodatni vsebinski atributi. Treba bi bilo tudi kreirati novo povezavo, ki bi od tistega trenutka dalje nadomestila obstoječo (L_Prodaja) in bi vsebovala dodaten tuji zgoščen ključ, kot kaže slika 14, kjer zaradi preglednosti niso vključeni sateliti. Pri tem ne pozabimo, da je treba v vseh obstoječih poizvedbah nad modelom dodati novo tabelo povezave, saj se bodo nove prodajne transakcije beležile samo v njej. Glede na literaturo in filozofijo »insert-only« je to edini pravilni način, v različnih člankih in blogih pa se pojavlja tudi pristop spremembe obstoječe povezave z ohranjanjem zgodovinskih zapisov.

Slika 13: Dimenzijski model po uvedbi nove entitete

Vir: lastno delo.

Slika 14: Model podatkovnega trezorja po uvedbi nove entitete



Vir: lastno delo.

4.5.5 Obvladovanje zgodovinskih zapisov

Kot že omenjeno, je ena izmed glavnih prednosti podatkovnih skladišč shranjevanje zgodovine. Atributi se lahko skozi čas spreminjajo in postopek modeliranja zajema tudi določitev strategije za ravnanje s spremembami.

Pri dimenzijskem modeliranju v ta namen uporabljamo koncept počasi se spreminjajočih dimenzij (angl. slowly changing dimensions – SCD), kjer poznamo več različnih tipov (Kimball in Ross, 2013):

Tip 0 deluje na način, da se vrednost atributa nikoli ne spremeni. Uporaben je za atribute, ki predstavljajo prvotne vrednosti.

Tip 1 uporabimo, kadar želimo, da se stara vrednost atributa zamenja z novo vrednostjo, kar pomeni, da izgubimo zgodovino sprememb atributa. Uporaba tipa 1 je najbolj enostavna za izvedbo, primerna je v primerih, kadar shranjevanje zgodovine ni smiselno.

Z uporabo tipa 2 zagotovimo, da se ob primeru spremembe vrednosti atributa doda nova vrstica. Ob tem je treba poskrbeti za ustrezne časovne atribute, ki opredeljujejo veljavnost posamezne vrstice.

Uporaba tipa 3 omogoča vpogled v podatke glede na trenutne in zgodovinske vrednosti atributa. Ob spremembi atributa dodamo v dimenzijo nov stolpec, kjer shranimo prejšnjo vrednost, v obstoječ stolpec pa trenutno vrednost. Tip 3 je primeren za uporabo v primerih, ko poslovni uporabniki želijo analizirati uspešnost določenih sprememb, kot so reorganizacije in podobno. Na tak način namreč lahko primerjajo mere glede na staro in novo vrednost atributa.

Tip 4 je priporočljiv za uporabo pri velikih večmilijonskih tabelah. Če se spremembe pogoste, to pomeni, da bo število vrstic hitro naraščalo, kar lahko negativno vpliva na učinkovitost poizvedb. Deluje na način, da atribute, ki se pogosto spreminjajo, ločimo v mini dimenzijo, ki ima svoj primarni ključ in je z njim povezana na tabelo dejstev.

Sledijo hibridni pristopi. Tip 5 je kombinacija tipa 4 in 1. Uporabljamo ga, kadar želimo omogočiti analizo trenutnega stanja v prvotni dimenziji brez uporabe tabele dejstev ali kadar želimo prikazati zgodovinske zapise v tabeli dejstev glede na trenutno stanje atributov.

Kombinacijo tipa 1, 2 in 3 združimo v tip 6, kjer imamo v dimenziji dva atributa, enega za trenutno stanje, ki se obnaša kot tip 1, in drugega za zgodovinsko stanje, ki se obnaša kot tip 2. To omogoča analizo glede na trenutno ali zgodovinsko vrednost.

Zadnji med tipi, tip 7, je nastal zaradi potreb po spremljanju trenutnih in zgodovinskih stanj, kot to omogoča tip 6, vendar s predpostavko, da želimo to spremljati za večje število atributov. V tem primeru vsebuje tabela dejstev dve povezavi – eno na dimenzijo trenutnih stanj, drugo na dimenzijo vseh stanj, ki se obnaša kot tip 2.

V modelu podatkovnega trezorja se zgodovina atributov nahaja v satelitih. Cuba (2020) izpostavlja, da k temu pripore pravilo, ki dovoljuje samo dodajanje novih zapisov (angl. insert-only) in ne dovoljuje kakršnih koli sprememb podatkov v satelitih, saj opredeljuje operacije posodabljanja kot drage. Velja omeniti, da je bila ta filozofija dodana naknadno in da se v starejših knjigah vseeno dovoljuje uporaba posodabljanja vrednosti datumov veljavnosti. Z verzijo modela podatkovnega trezorja 2.0 pa se tega ne priporoča več zaradi lažje prilagodljivosti na moderne podatkovne baze v oblaku, kjer se operacije vstavljanja izvajajo veliko hitreje.

Poglejmo primer spremljanja zgodovinskih zapisov za knjižno skupino kupca, ki je za oddelek računovodstva pomembna pri analizi terjatev. V izvornem sistemu se ob spremembi vrednosti le-ta prepiše z novo.

V dimenzijskem modelu bi teoretično lahko uporabili vse tipe razen 0 in 1, ki ne beležita zgodovine sprememb. V tabeli 7 je prikazano prvotno stanje v dimenziji kupca, v tabeli 8 stanje po spremembi knjižne skupine z uporabo tipa 2, v tabeli 9 pa z uporabo tipa 3. Zaradi preglednosti smo v tabelah prikazali le atribute, ki so pomembni za razumevanje beleženja zgodovine.

Tabela 7: Prvotno stanje v dimenziji D_Kupec

Kupec SID	**Šifra**	**Naziv**	**Knjižna skupina**	**Knjižna skupina opis**	**Datum začetka veljavnosti**	**Datum konca veljavnosti**
458886	305	X d.o.o.	7	SLO-P	2022-01-13	9999-12-31

Vir: lastno delo.

Tabela 8: Končno stanje v dimenziji D_ Kupec (tip 2)

Kupec SID	**Šifra**	**Naziv**	**Knjižna skupina**	**Knjižna skupina opis**	**Datum začetka veljavnosti**	**Datum konca veljavnosti**
458886	305	X d.o.o.	7	SLO-P	2022-01-13	2023-07-01
510021	305	X d.o.o.	11	SLO-O	2023-07-02	9999-12-31

Vir: lastno delo.

Glede na to, da se knjižna skupina kupca spreminja pogosteje kot ostali atributi, bi v nadaljevanju pričakovali dodatne vrstice za enega kupca v zgornji tabeli. Vsaka vrstica pridobi nov primarni ključ, ki se glede na čas nastale knjižbe tudi ustrezno poveže s tabelo dejstev.

Tabela 9: Končno stanje v dimenziji D_ Kupec (tip 3)

Kupec SID	Šifra	Naziv	Knjižna skupina opis	Prejšnja knjižna skupina opis	Datum začetka veljavnosti	Datum konca veljavnosti
458886	305	X d.o.o.	SLO-O	SLO-P	2022-01-13	9999-12-31

Vir: lastno delo.

V primeru naknadnih sprememb bi v tabeli 9 dodali nove stolpce, kar je lahko nepraktično ob pogostih spremembah. Poleg tega v tem primeru uporabniki ne potrebujejo primerjave med novo in staro vrednostjo.

Pri uporabi tipa 4 bi ločili atribute za knjižne skupine v mini dimenzijo, ki bi vsebovala vse možne vrednosti knjižnih skupin. Tabela dejstev bi poleg tujega ključa na dimenzijo kupca vsebovala tudi tuji ključ na dimenzijo knjižnih skupin. V primeru spremembe bi torej prvotna dimenzija kupca ostala enaka kot prej. Spremembo bi zaznali v tabeli dejstev, saj bi nove vrstice vsebovale drugačen tuji ključ na dimenzijo knjižnih skupin.

Pomanjkljivost tipa 4 je predvsem ta, da ne podpira analize prvotne dimenzije kupca po atributih knjižne skupine, saj je povezava med njima le v tabeli dejstev, čemur se izognemo z uporabo tipa 5, kjer bi atribut knjižne skupine v dimenziji kupca spremljali glede na tip 1, kar pomeni, da bi vedno ohranili trenutno stanje atributa. Hkrati pa bi mini dimenzija ostala v obliki kot pri tipu 4. Na tak način lahko analiziramo tabelo dejstev vedno po trenutni knjižni skupini iz dimenzije kupca ali po dejanski knjižni skupini v času nastanka zapisa, ki je shranjena v dimenziji knjižnih skupin.

Če bi za sledenje zgodovinskih zapisov uporabili tip 6, bi dimenzija kupca izgledala kot kaže tabela 10. S pomočjo atributa, ki se spreminja po pravilih tipa 1, lahko izvajamo analize glede na trenutno oz. zadnjo vrednost. Z atributom, ki sledi pravilu tipa 2, so možne analize glede na vrednost, ki je bila veljavna v določenem obdobju.

Tabela 10: Končno stanje v dimenziji D_ Kupec (tip 6)

Kupec SID	Šifra	Naziv	Knjižna skupina opis	Zgodovinska knjižna skupina opis	Datum začetka veljavnosti	Datum konca veljavnosti
458886	305	X d.o.o.	SLO-O	SLO-P	2022-01-13	2023-07-01
510021	305	X d.o.o.	SLO-O	SLO-O	2023-07-02	9999-12-31

Vir: lastno delo.

Tip 7 bi bil videti tako, da bi obstajali dve dimenziji kupca. Prva bi delovala na način, kot je prikazano v tabeli 8 (tip 2), in druga, ki bi vsebovala trenutno stanje. Tabela dejstev bi vsebovala oba tuja ključa, kar bi omogočalo analizo po trenutnem stanju in po stanju, ki je bilo aktualno v času nastanka zapisa.

Poglejmo isti primer še v modelu podatkovnega trezorja. Vsebina je v satelitu S_Kupec_Knjižna_skupina. V tabeli 11 prikažemo začetno stanje, v tabeli 12 pa stanje po spremembi na viru, kjer lahko opazimo novo vrstico. Način shranjevanja je torej primerljiv s tipom 2 pri dimenzijskem modeliranju.

Tabela 11: Prvotno stanje v satelitu S_Kupec_knjižna_skupina

Kupec zgoščeni ključ	**Datum nalaganja**	**Vir**	**Zgoščena vrednost**	**Knjižna skupina**	**Knjižna skupina opis**
406b5a10	2022-01-31	NAV	8fabc68d	7	SLO-P

Vir: lastno delo.

Tabela 12: Končno stanje v satelitu S_Kupec_knjižna_skupina

Kupec zgoščeni ključ	**Datum nalaganja**	**Vir**	**Zgoščena vrednost**	**Knjižna skupina**	**Knjižna skupina opis**
406b5a10	2022-01-31	NAV	8fabc68d	7	SLO-P
406b5a10	2023-07-02	NAV	4afbl97e	11	SLO-O

Vir: lastno delo.

Kljub temu, da je spremljanje zgodovinskih zapisov v modelu podatkovnega trezorja in dimenzijskem modelu z uporabo tipa 2 primerljivo, se razlikuje vsebina stolpcev v končnih tabelah 8 in 12. Pri kreiranju SQL poizvedb je pogosto treba pridobiti aktualno oz. zadnje stanje, kar pomeni da uporabimo omejitve nad datumskim poljem. V tabeli 8 bi zadnji zapis lahko pridobili z omejitvijo na vrednost '9999-12-31' v stolpcu »Datum konca veljavnosti«. V tabeli 12 pa datuma konca veljavnosti zapisa nimamo, kar izhaja iz pravila »insert-only«. V tem primeru bi bila omejitev malenkost bolj zapletena, saj bi zajemala uporabo funkcije LEAD, ki na podlagi naslednjega »Datuma nalaganja« izračuna veljavnost trenutne vrstice.

V modelu podatkovnega trezorja smo atribute od kupca že na začetku razdelili glede na pogostost sprememb, kar nam zagotavlja, da ob spremembi knjižne skupine nove vrstice dodajo le v satelit S_Kupec_Knjižna_skupina. V dimenzijskem modelu pa se ob uporabi tipa 2 nova vrstica doda v dimenzijo D_Kupec. Pri dimenzijah z večjim številom opisnih polj lahko to vpliva na velikost tabele in velike količine podvojenih podatkov, saj se v novi vrstici beležijo tudi vsi ostali atributi, ki se niso spremenili. Nekaj podobnosti lahko opazimo tudi v primerjavi z uporabo tipa 4, saj v obeh primerih ločimo atribute knjižnih skupin v novo tabelo, kjer spremljamo zgodovino, originalna tabela pa se ne spreminja.

4.5.6 Učinkovitost poizvedb

V modelu podatkovnega trezorja lahko za določeno vsebino pričakujemo več tabel v primerjavi z dimenzijskim modelom (Schalkwyk, 2014), kar potrjuje tudi naš primer, kjer imamo za enako vsebino v dimenzijskem modelu 13 tabel, v modelu podatkovnega trezorja pa 34. Večje število tabel v modelu pomeni več operacij za združevanje tabel, kar v splošnem povzroča tudi počasnejše izvajanje poizvedb (Kimball, 2008).

Poglejmo primer poizvedbe, ki prikaže znesek po kupcih in njihovih knjižnih skupinah za leto 2022. Prva poizvedba je iz dimenzijskega modela, kjer lahko opazimo, da so povezave enostavne, uporabljeni sta 2 tabeli.

```
SELECT c.Sifra, c.Knjizna_skupina, c. Knjizna_skupina_opis,
SUM(f.Znesek)
FROM F_Glavna_knjiga f
INNER JOIN D_Kupec c on c.Kupec_SID = f.Kupec_SID
WHERE f.Datum_SID between 20220101 and 20221231
GROUP BY c.Sifra, c.Knjizna_skupina, c. Knjizna_skupina_opis
```

Naslednja poizvedba je pripravljena nad modelom podatkovnega trezorja. Za enak rezultat potrebujemo 4 tabele, povezave med njimi pa so kompleksnejše, saj je iz satelitov treba vzeti zadnji aktualen zapis. Pridobimo ga z omejitvijo na »Datum_konca_veljavnosti«, ki ga izračunamo s pomočjo funkcije LEAD, ki na podlagi naslednjega zapisa določi končni datum trenutnega zapisa. V primeru, da je zapis za določen »Kupec_Zgosceni_kljuc« zadnji, dobi atribut »Datum_konca_veljavnosti« vrednost null.

```
WITH kupec AS (
  SELECT Sifra, Kupec_zgosceni_kljuc, LEAD(Datum_nalaganja)-1 OVER
  (PARTITION BY Kupec_zgosceni_kljuc  ORDER BY Datum_nalaganja
  ASC) AS Datum_konca_veljavnosti
  FROM S_Kupec_Finance
),
knjizna_skupina AS (
  SELECT Knjizna_skupina, Knjizna_skupina_opis,
  Kupec_zgosceni_kljuc, LEAD(Datum_nalaganja)-1 OVER (PARTITION BY
  Kupec_zgosceni_kljuc ORDER BY Datum_nalaganja ASC) AS
  Datum_konca_veljavnosti
  FROM S_Kupec_Knjizna_skupina
)
SELECT c.Sifra, ps. Knjizna_skupina, ps.Knjizna_skupina_opis,
SUM(sk.Znesek)
FROM L_Knjizba k
INNER JOIN S_Knjizba sk on sk.Knjizba_zgosceni_kljuc = k.
Knjizba_zgosceni_kljuc
INNER JOIN kupec c on c.Kupec_zgosceni_kljuc =
k.Kupec_zgosceni_kljuc AND Datum_konca_veljavnosti is null
INNER JOIN knjizna_skupina ps on ps.Kupec_zgosceni_kljuc =
k.Kupec_zgosceni_kljuc AND Datum_konca_veljavnosti is null
WHERE sk.Datum_knjizenja between '2022-01-01' and '2022-12-31'
GROUP BY c.Sifra, ps. Knjizna_skupina, ps.Knjizna_skupina_opis
```

Kljub ne bistveni razliki v številu tabel v tem primeru lahko že na prvi pogled ugotovimo, da je prva poizvedba enostavnejša in bolj razumljiva od druge.

Poglejmo še rezultate statističnih in komparativnih raziskav.

Statistična raziskava Schalwyka (2014) potrjuje, da je povprečni čas izvajanja poizvedb v modelu podatkovnega trezorja večji od povprečnega izvajanja poizvedbe za podobne primere v dimenzijskem modelu. V raziskavi so bili upoštevani primeri, ki so najbolj podobni realnim primerom poizvedb v podatkovnih skladiščih.

V raziskavi, ki so jo izvedli Grigoriev in drugi (2021), so primerjali oba modela tako, da so izvedli več poizvedb na obeh modelih in pokazali, da je bila zmogljivost modela podatkovnega trezorja slabša pri kompleksnih poizvedbah.

S tem se strinja tudi Hospodka (2022), čigar meritve so prav tako pokazale, da so poizvedbe v modelu podatkovnega trezorja počasnejše v primerjavi s poizvedbami v dimenzijskem modelu, kar avtor smatra kot pričakovano zaradi normalizacije modela podatkovnega trezorja in združevanja številnih tabel.

Dimenzijski model je zasnovan tako, da so poizvedbe po tabelah dejstev in denormaliziranih tabelah učinkovite. V modelu podatkovnega trezorja so zaradi visoke standardizacije podatkov neposredne poizvedbe lahko zelo počasne, zato je za analitiko in poročanje priporočljiva izgradnja dimenzijskega modela nad podatkovnim trezorjem (Yessad in Labiod, 2016). Enako trdita tudi Naamane in Jovanovic (2016), ki menita da je ravno učinkovitost poizvedb razlog, da je treba nad modelom podatkovnega trezorja zagotoviti še dodatno plast za poročanje in dostopanje končnih uporabnikov, za kar priporočata dimenzijski model.

4.5.7 Enostavnost dostopa

Sherman (2014) pravi, da so prednosti zvezdne sheme njena intuitivnost za pisanje poizvedb, ki so zmogljive, in v splošnem dobro podpira analitiko. Podpira naloge, ki jih poslovni ljudje običajno izvajajo, ko uporabljajo vrtilne tabele v preglednicah, zato je zanje zelo intuitiven. Za poslovneže je dimenzijski model pogosto enostaven za uporabo in omogoča hitro izvedbo poizvedb, saj so orodja BI zgrajena tako, da izkoriščajo konstrukcijo zvezdne sheme. Schnider (2014) je mnenja, da dimenzijski model v splošnem ne potrebuje dodatne predstavitvene plasti, vseeno jo včasih pripravimo zaradi različnih omejitev orodij BI. Kljub temu tudi dodatna plast po navadi sledi enakim pravilom modeliranja.

Dostop do podatkov v modelu podatkovnega trezorja je lahko za poslovne uporabnike manj intuitiven, zato se pogosto priporoča vzpostavitev dodatne plasti, npr. poslovni podatkovni trezor ali področne zvezdne sheme (angl. data mart), zgrajene nad osnovnim modelom podatkovnega trezorja. V dodatni predstavitveni plasti je pogosto smiselno slediti pravilom,

da povezave postanejo tuji ključi v tabelah dejstev, sateliti postanejo dejstva ali atributi v dimenziji, vozlišča pa postanejo dimenzije ali pa določajo granulacijo tabel dejstev (Gribova, 2022; Schnider in drugi, 2014).

V predstavljenem primeru izbranega podjetja smo na začetku izpostavili, da je eden izmed ciljev zagotovitev čim enostavnejšega modela, saj želijo pridobiti tudi kader, ki bo samostojno skrbel za spremembe v modelu. Na podlagi izkušenj in prej omenjenih raziskav ocenjujemo, da bi proces predaje znanja in učenja novih veščin potekal hitreje v primeru dimenzijskega modela, ki je v splošnem znan večini razvijalcev na tem področju.

Pri izbiri je priporočljivo oceniti strokovnost obstoječega kadra, če že imajo izkušnje z modeliranjem, je smiselno uporabiti to znanje. Modeliranje podatkovnega trezorja lahko zahteva dodatna usposabljanja ali zunanje strokovno znanje (Boddu, 2023).

5 PRIPOROČILA IN SMERNICE ZA IZBIRO MODELA

V tem poglavju bomo naredili povzetek ugotovitev in primerjav med modelom podatkovnega trezorja in dimenzijskim modelom. V tabeli 13 so navedeni vsi dejavniki, ki smo jih preučili in prepoznali kot ključne pri odločanju o pristopu k modeliranju podatkovnih skladišč. Za vsak dejavnik so opisane ključne lastnosti za oba predstavljena modela.

Na podlagi te tabele je razvidno, da ni enoznačnega odgovora na dilemo o izbiri modela podatkovnega skladišča. Naslednje ugotovitve lahko služijo kot smernice pri določanju ustreznosti posameznega pristopa v različnih situacijah.

Kot ključne prednosti modela podatkovnega trezorja bi izpostavili enostavno razširljivost in prilagodljivost. Gre za učinkovit pristop za podatkovna skladišča, ki združujejo veliko število izvornih sistemov. Lastnosti modela omogočajo vzporedno nalaganje podatkov, kar lahko pozitivno vpliva na hitrost procesa ETL. Kot smo lahko opazili, je dodajanje novih podatkov v model precej preprosto in hitro, vendar pa to povzroča kompleksen podatkovni model, ki je zahtevnejši za razumevanje v primerjavi z dimenzijskim modelom. Kompleksnost modela vpliva tudi na hitrost izvedbe analitičnih poizvedb. Kljub temu, da je postopek ETL enostaven, se večji del postopka transformacije preloži na naslednjo fazo. Med slabostmi velja omeniti, da model ni najbolj primeren za neposredno analizo in poročanje, za ta namen se namreč priporoča kreiranje dodatne podatkovne plasti, prilagojene poslovnim uporabnikom.

Glavne prednosti dimenzijskega modela vključujejo zmogljivost poizvedb in razumljivost, saj uporablja zvezdne sheme, ki so poslovnim uporabnikom intuitivne. Je prilagodljiv iz vidika spremljanja zgodovinskih zapisov, saj omogoča več načinov za izvedbo. Proces ETL vključuje tudi upoštevanje poslovnih pravil, zato je bolj zapleten v primerjavi z modelom podatkovnega trezorja. Med slabostmi lahko omenimo izzive v smislu razširljivosti in prilagodljivosti ob spremembah modela.

Ne glede na izbran pristop je izrednega pomena tudi to, da je le-ta sprejet s strani razvojne skupine ter da vsi vključeni razumejo koncepte, prednosti in težave izbranega pristopa modeliranja.

Primerjave, ki smo jih predstavili, ponujajo osnovno podlago, ki lahko olajša odločanje glede izbire pristopa k modeliranju podatkovnih skladišč.

Tabela 13: Primerjava modelov

Vidik modeliranja	**Model podatkovnega trezorja**	**Dimenzijski model**
Strategija modeliranja	Najbolj primeren za poslovno usmerjeno modeliranje.	Primeren predvsem za modeliranje na podlagi poročanja, lahko tudi za poslovno usmerjeno modeliranje.
Integracija podatkov in proces ETL	Proces vključuje le uporabo trdih poslovnih pravil. Omogoča enostavno integracijo velikega števila izvornih sistemov zaradi jasnih pravil nalaganja v končni model. Neodvisnosti med tabelami in zgoščeni ključi omogočajo hitro nalaganje podatkov in možnost vzporednega nalaganja. Proces je skoraj popolnoma neodvisen od transformacij. Na voljo je sedem certificiranih orodij za avtomatizacijo modeliranja.	Proces vključuje uporabo poslovnih pravil, zato spremembe vplivajo tudi na proces ETL. Vzporedno nalaganje je mogoče za tabele, ki niso odvisne med seboj. Pri razvoju so nam lahko v pomoč orodja, ki podpirajo dimenzijsko modeliranje, kot je npr. SSAS.
Nove analitične zahteve	Treba je kreirati nove objekte in jih vključiti v poizvedbe, ki se uporabljajo nad modelom. Ni neposrednega vpliva na obstoječe objekte. Število objektov na tak način lahko hitro narašča, s tem se poveča tudi kompleksnost modela.	Treba je kreirati nove objekte ali pa razširiti obstoječe in ustrezno poskrbeti za integracijo ter pri tem paziti, da ne izgubimo zgodovinskih podatkov.
Spremembe podatkovnih struktur	Sprememba kardinalnosti relacije ne vpliva na model. Uvedba novih entitet povzroči nove objekte v modelu.	Sprememba kardinalnosti relacije lahko povzroči spremembe v modelu, kot je dodatna vmesna tabela. Uvedba novih entitet povzroča potrebo po kreiranju novih objektov in spremembo obstoječih za zagotovitev ustreznih povezav v modelu.
Obvladovanje zgodovinskih zapisov	Zgodovina se beleži v satelitih, časovno veljavnost določa podatek o datumu nalaganja vrstice. Pri pridobivanju aktualnih zapisov je potrebna uporaba funkcije LEAD za določitev konca veljavnosti vrstice.	Dimenzijski model podpira več pristopov za obvladovanje zgodovinskih zapisov. Pri uporabi najpogostejšega tipa 2 je logika precej podobna, a je pridobivanje aktualnih zapisov enostavnejše kot v modelu podatkovnega trezorja.

se nadaljuje

Tabela 13: Primerjava modelov (nad.)

Vidik modeliranja	Model podatkovnega trezorja	Dimenzijski model
Učinkovitost poizvedb	Model podatkovnega trezorja vsebuje več tabel, kar pomeni kompleksnejše združevanje teh. Poleg tega na povečanje časovne zahtevnosti poizvedb vpliva tudi visoka standardizacija podatkov.	Poizvedbe, ki temeljijo na zvezdni shemah, so prilagojene orodjem BI. Učinkovitost poizvedb je ena ključnih lastnosti dimenzijskega modela.
Enostavnost dostopa	Priporočljivo je kreirati dodatno podatkovno plast, do katere dostopajo končni uporabniki. To je lahko poslovni podatkovni trezor ali področne zvezde sheme.	Dimenzijski model je uporabnikom bolj intuitiven. Pristop je znan večini razvijalcev na tem področju, kar lahko pozitivno vpliva na samostojnost in vpeljavo novega kadra.

Vir: lastno delo.

6 SKLEP

V magistrskem delu smo obravnavali primerjavo dimenzijskega modela in modela podatkovnega trezorja. Dimenzijski model je razvil Ralpha Kimball in odražata ga enostavnost ter razumljivost. Sestavljen je iz zvezdnih shem, ki vključujejo dimenzijske tabele in tabele dejstev. V večini primerov so orodja za prikaz vizualizacij zasnovana tako, da učinkovito izkoristijo prednosti dimenzijskega modela in s tem zagotovijo hitre vpoglede. To je tudi razlog, da dimenzijsko modeliranje že dlje časa velja za uveljavljen pristop k modeliranju podatkovnih skladišč. Model podatkovnega trezorja, ki ga je ustvaril Dan Linstedt, lahko opišemo kot poslovno usmerjen model, ki temelji na treh osnovnih objektih, to so vozlišča, sateliti in povezave. V zadnjem desetletju so se na področju modeliranja podatkovnega trezorja pojavile številne nove knjige, ki dopolnjujejo prvotno verzijo modela, sledili so tudi razni članki, ki opisujejo uporabo modela na realnih primerih v praksi.

Poleg podrobne predstavitve obeh modelov smo v teoretičnemu delu opredelili vidike modeliranja podatkovnih skladišč, ki so ključnega pomena za razumevanje in odločanje pri izbiri pristopa k modeliranju.

Na podlagi predstavljenega poslovnega problema smo pripravili najprej model podatkovnega trezorja in nato dimenzijski model. Sledila je analiza in primerjava modelov. K celovitemu razumevanju prednosti in slabosti obeh pristopov je prispevala predvsem primerjava na podlagi prej omenjenih teoretičnih vidikov modeliranja podatkovnih skladišč. Raziskali smo, kako na modela vplivajo nove analitične zahteve in spremembe podatkovnih struktur s simulacijo različnih dogodkov, kjer smo ugotovili, da v modelu podatkovnega trezorja takšne primere rešujemo z ustvarjanjem novih objektov, zato s tem ne vplivamo na obstoječe objekte in podatke, kar je zagotovo prednost v primerjavi z dimenzijskim modelom, hkrati pa lahko na tak način model hitro postane nepregleden in kompleksen.

Primerjava obvladovanja zgodovinskih zapisov je pokazala, da sta si koncept počasi se spreminjajočih dimenzij tipa 2 in pravilo za spremljanje zgodovine v modelu podatkovnega trezorja zelo podobna, razlikujeta pa se v načinu beleženja datumskih atributov, kar lahko precej vpliva na kompleksnost poizvedb. To je hkrati eden izmed razlogov, da se dimenzijski model izkaže kot boljši na področju hitrosti in učinkovitosti poizvedb. Kot eno izmed glavnih prednosti modela podatkovnega trezorja lahko izpostavimo enostavno integracijo in hitrost nalaganja podatkov v končni model, kar je posledica neodvisnosti med tabelami in uporabe zgoščenih ključev.

Sklenemo lahko, da imata oba predstavljena modela svoje prednosti in slabosti. Ugotavljamo, da ni enostavnega postopka za nedvoumno izbiro najboljšega pristopa. Z magistrskim delom smo prispevali k boljšem razumevanju osnovnih konceptov obeh pristopov in predlagali ključne vidike modeliranja, ki jih je smiselno upoštevati ob izbiri.

Pri pripravi magistrskega dela smo se soočili z omejitvami pri izboru literature, saj smo opazili omejeno število raziskav s področja modeliranja podatkovnega trezorja na realnih primerih, še posebej v zadnjih treh letih in v primerjavi s številom raziskav s področja dimenzijskega modeliranja.

Za nadgradnjo in bolj poglobljeno analizo bi v prihodnosti lahko oba modela uvedli v isti organizaciji in opravili obsežnejšo primerjavo, ki bi vključevala testiranje časov nalaganja podatkov, trajanje poizvedb in časov ter stroškov pri implementaciji, dopolnjevanju in vzdrževanju modelov. Smiselno bi bilo vključiti mnenja končnih uporabnikov, ki bi dejansko uporabljali oba modela za izdelavo poročil in analiz. Na ta način bi pridobili informacije o uporabniški izkušnji in uporabnosti modelov v realnem okolju. Testirali bi lahko tudi hitrost vpeljave novega kadra, ki bi skrbel za model podatkovnega skladišča. Upoštevati bi bilo treba tudi druge dejavnike, kot so količina podatkov, uporabljene tehnologije in orodja, velikost podjetja, panoga delovanja ter posebne zahteve in potrebe podjetja, kar bi nam omogočilo bolj celovit vpogled v primernost vsakega modela glede na zahteve posameznega podjetja.

LITERATURA IN VIRI

1. Albright, S. C. in Winston, W. L. (2014). *Business Analytics: Data Analysis & Decision Making*. Cengage Learning.
2. Anandarajan, M., Anandarajan, A. in Srinivasan, C. A. (2004). *Business Intelligence Techniques: A Perspective from Accounting and Finance* (1. izd.). Springer.
3. Bagui, S. in Earp, R. (2003). *Database Design Using Entity-Relationship Diagrams.* CRC Press.
4. Balali, F., Nouri, J., Nasiri, A. in Zhao, T. (2020). *Data Intensive Industrial Asset Management: IoT-based Algorithms and Implementation* (1. izd.). Springer.

5. Boddu, P. (2023). Data Vault vs. Dimensional Modeling: Choosing the Right Data Warehousing Approach. Pridobljeno 2. oktobra 2023 s https://medium.com/@prasad.imdb/data-vault-vs-dimensional-modeling-choosing-the-right-data-warehousing-approach-b89b0c89e9f2
6. Caserta, J. in Kimball, R. (2004). *The Data Warehouse ETL Toolkit: Practical Techniques for Extracting, Cleaning, Conforming, and Delivering Data* (1. izd.). Wiley.
7. Coronel, C., Morris, S. in Rob, P. (2011). *Database systems - Design, Implementation and Management* (9. izd.). Cengage Learning.
8. Craig, S. (2021). What is data modeling? *TechTarget*. Pridobljeno 10. marca 2023 s https://www.techtarget.com/searchdatamanagement/definition/data-modeling
9. Cuba, P. (2020). *The Data Vault Guru: A Pragmatic Guide on Building a Data Vault*. Independently Published.
10. DataVaultAlliance. (brez datuma). *Data Vault 2.0 Certified Software Tools Program*. Pridobljeno 26. novembra 2023 s https://datavaultalliance.com/certified-software-tools/
11. Date, C. J. (2019). *Database Design and Relational Theory: Normal Forms and All That Jazz* (2. izd.). Apress.
12. Dratwa, T. (2023). *Data Vault Part 2 - Data modeling*. Pridobljeno 22. oktobra 2023 iz: https://bitpeak.pl/dv-2/
13. Giebler, C., Gröger, C., Hoos, E., Schwarz, H. in Mitschang, B. (2019). *Modeling Data Lakes with Data Vault: Practical Experiences, Assessment, and Lessons Learned*. Cham: Conceptual Modeling.
14. Gluchowski, P. (2021). Data Vault as a Modeling Concept for the Data Warehouse. V *Engineering the Transformation of the Enterprise: A Design Science Research Perspective* (str. 277-286). Cham: Springer.
15. Golfarelli, M. in Rizzi, S. (2009). *Data Warehouse Design: Modern Principles and Methodologies* (1. izd.). McGraw Hill.
16. Gribova, S. (2022). *Literature Review on Data Vaults – What is the State of the Art of Literature on Data Vaults?* Pridobljeno 4. oktobra 2023 s https://www.fh-wedel.de/fileadmin/Mitarbeiter/Records/Gribova_2022_-_Literature_Review_on_Data_Vaults_-_What_is_the_State_of_the_Art_of_Literature_on_Data_Vaults.pdf.
17. Grigoriev, Y., Ermakov, E. in Ermakov, O. (2021). Hadoop/Hive Data Query Performance Comparison Between Data Warehouses Designed by Data Vault and Snowflake Methodologies. V *Modern Information Technology and IT Education* (str. 147-156). Springer, Cham.
18. Hoberman, S. (2009). *Data modeling made simple: a practical guide for business and IT professionals* (92. izd.). Technics Publications.
19. Hospodka, A. (2022). *Overview and Analysis of Data Vault 2.0 - Flexible Data Warehousing Methodology* (magistrsko delo). Masaryk University.

20. Hultgren, H. (2012). *Modeling the Agile Data Warehouse with Data Vault*. Brighton Hamilton.
21. IBM. (brez datuma a). *What is data lifecycle management?* Pridobljeno 9. aprila 2023 s https://www.ibm.com/topics/data-lifecycle-management.
22. IBM. (brez datuma b). *What is data modeling?* Pridobljeno 9. aprila 2023 s https://www.ibm.com/topics/data-modeling.
23. Imhoff, C., Galemmo, N. in Geiger, J. G. (2003). *Mastering Data Warehouse Design: Relational and Dimensional Techniques*. Wiley.
24. Informatica. (brez datuma). *What is ETL (extract transform load)?* Pridobljeno 28. aprila 2023 s https://www.informatica.com/nz/resources/articles/what-is-etl.html.
25. Inmon, W. H. (2005). *Building the Data Warehouse* (3. izd.). Wiley.
26. Inmon, W. H. in Hackathorn, R. D. (1994). *Using the data warehouse* (1. izd.). Wiley.
27. Inmon, W. H. H., Linstedt, D. in Levins, M. (2019). *Data Architecture: A Primer for the Data Scientist: A Primer for the Data Scientist* (2. izd.). Academic Press.
28. Kimball, R. in Ross, M. (2013). *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling* (3. izd.). Wiley.
29. Kimball, R., Ross, M., Thornthwaite, W., Mundy, J. in Becker, B. (2008). *The data warehouse lifecycle toolkit: practical techniques for building data warehouse and business intelligence systems* (2. izd.). Wiley.
30. Krneta D., K. S. (2020). Data Vault as a Decision Support Platform for an Electricity Supplier in the Open Electricity Market. *19th International Symposium INFOTEH-JAHORINA* (str. 185-188).
31. Laudon, K. C. in Laudon, J. P. (2003). *Essentials of Management Information Systems: Managing the Digital Firm* (5. izd.). Prentice Hall.
32. Linstedt, D. in Olschimke, M. (2015). *Building a Scalable Data Warehouse with Data Vault 2.0* (1. izd.). Morgan Kaufmann.
33. Loshin, D. (2013). *Business intelligence: the savvy manager's guide, second edition* (2. izd.). Morgan Kaufmann.
34. McIntyre, S. in Zdechovan, R. (2023). *Data Vault 2.0 with dbt Cloud.* Pridobljeno 26. november 2023 s https://docs.getdbt.com/blog/data-vault-with-dbt-cloud#automatedv-formerly-known-as-dbtvault
35. Naamane, Z. in Jovanovic, V. (2016). Effectiveness of Data Vault compared to Dimensional Data Marts on Overall Performance of a Data Warehouse System. *IJCSI International Journal of Computer Science Issues, 13*(4), 16-31.
36. Oppel, A. (2009). *Databases A Beginner's Guide* (1. izd.). McGraw Hill.
37. Ponniah, P. (2001). *Data warehousing fundamentals: a comprehensive guide for IT professionals* (1. izd.). Wiley-Interscience.
38. Rainardi, V. (2008). *Building a Data Warehouse: With Examples in SQL Server* (1. izd.). Apress.
39. Reis, J. in Housley, M. (2022). *Fundamentals of Data Engineering: Plan and Build Robust Data Systems* (1. izd.). O'Reilly Media.

40. Runkler, T. A. (2020). *Data Analytics : Models and Algorithms for Intelligent Data Analysis* (3. izd.). Springer Vieweg.
41. Scalefree. (2010). *Visual Data Vault*. Pridobljeno 5. septembra 2023 iz: https://www.visualdatavault.com/.
42. Schalkwyk, M. V. (2014). *A comparison of the impact of data vault and dimensional modelling on data warehouse performance and maintenance* (magistrsko delo). North-West University.
43. Schnider, D., Martino, A. in Eschermann, M. (2014). *Comparison of Data Modeling Methods for a Core Data Warehouse*. Pridobljeno 20. julija 2022 s https://danischnider.files.wordpress.com/2019/09/comparison_dwh_core_modeling.pdf.
44. Sherman, R. (2014). *Business Intelligence Guidebook: From Data Integration to Analytics* (1. izd.). Morgan Kaufmann.
45. Subotic, D., Jovanovic, V. M. in Poščić, P. (2014). Data Warehouse and Master Data Management Evolution – A Meta-Data-Vault Approach. *Issues in Information Systems, 15*(2), 14-23.
46. Yessad, L. in Labiod, A. (2016). Comparative study of data warehouses modeling approaches: Inmon, Kimball and Data Vault. *2016 International Conference on System Reliability and Science (ICSRS)*, 95-99.