

# MODEL REPOZITORIJA METAPODATAKA ZA EVOLUCIJU SCHEME SKLADIŠTA PODATAKA TE INTEGRACIJU SA SUSTAVOM ZA UPRAVLJANJE MATIČNIM PODACIMA

---

**Jakšić, Danijela**

**Doctoral thesis / Disertacija**

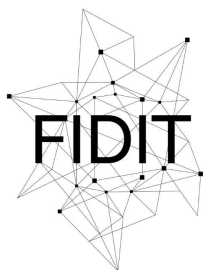
**2016**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **University of Rijeka / Sveučilište u Rijeci**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:195:329578>

*Rights / Prava:* [In copyright](#)/[Zaštićeno autorskim pravom.](#)

*Download date / Datum preuzimanja:* **2025-01-04**



Sveučilište u Rijeci  
**Fakultet informatike  
i digitalnih tehnologija**

*Repository / Repozitorij:*

[Repository of the University of Rijeka, Faculty of Informatics and Digital Technologies - INFORI Repository](#)



SVEUČILIŠTE U RIJECI  
ODJEL ZA INFORMATIKU

Danijela Jakšić

**MODEL REPOZITORIJA META-  
PODATAKA ZA EVOLUCIJU SCHEME  
SKLADIŠTA PODATAKA TE  
INTEGRACIJU SA SUSTAVOM ZA  
UPRAVLJANJE MATIČNIM PODACIMA**

DOKTORSKI RAD

Rijeka, 2016.



SVEUČILIŠTE U RIJECI  
ODJEL ZA INFORMATIKU

Danijela Jakšić

**MODEL REPOZITORIJA META-  
PODATAKA ZA EVOLUCIJU SCHEME  
SKLADIŠTA PODATAKA TE  
INTEGRACIJU SA SUSTAVOM ZA  
UPRAVLJANJE MATIČNIM PODACIMA**

DOKTORSKI RAD

Mentor: prof. dr. sc. Vladan Jovanović

Komentor: izv. prof. dr. sc. Patrizia Pošćić

Rijeka, 2016.

UNIVERSITY OF RIJEKA  
DEPARTMENT OF INFORMATICS

Danijela Jakšić

**METADATA REPOSITORY MODEL FOR  
DATA WAREHOUSE SCHEMA  
EVOLUTION AND INTEGRATION WITH  
MASTER DATA MANAGEMENT SYSTEM**

DOCTORAL THESIS

Rijeka, 2016.

Mentor rada: prof. dr. sc. Vladan Jovanović

Komentor rada: izv. prof. dr. sc. Patrizia Pošćić

Doktorski rad obranjen je dana 24. studenoga 2016. na Odjelu za informatiku Sveučilišta u Rijeci, pred povjerenstvom u sastavu:

1. Prof. dr. sc. Mile Pavlić
2. Prof. dr. sc. Mario Radovan
3. Doc. dr. sc. Alen Jakupović

## ZAHVALA

Najljepše zahvaljujem mentoru, prof. dr. sc. Vladanu Jovanoviću te komentorici, izv. prof. dr. sc. Patriziji Pošćić na vodstvu, savjetima te pruženoj stručnoj i osobnoj podršci tijekom izrade ovog doktorskog rada.

Također iskreno zahvaljujem svim kolegama s Odjela za informatiku Sveučilišta u Rijeci koji su svojim savjetima i sugestijama pridonijeli kvaliteti ovog rada.

Zahvaljujem i svim članovima moje obitelji na pruženom strpljenju i osobnoj podršci tijekom mog rada i napredovanja.

## Sažetak

Skladište podataka (SP) se u današnje vrijeme nalazi u iznimno dinamičnom poslovnom okruženju. S jedne strane imamo brojne (heterogene) izvore podataka koji su podložni čestim promjenama podataka i strukture, dok s druge strane imamo brojne promjene u informacijskim zahtjevima koje postavljaju poslovni korisnici. Višedimenzionalna shema (VDS) u svakom trenutku mora moći usvojiti promjene iz izvora podataka te im se prilagoditi, kao i zadovoljiti korisničke zahtjeve za informacijama, što je iznimno složen zadatak. Problem koji se istražuje kod evolucije skladišta podataka jest pamćenje promjena opsega te strukture podataka i meta-podataka, u dužem vremenskom periodu. Akademska zajednica je do danas napravila određene korake prema rješavanju ovoga problema, no uvijek ima prostora za poboljšanje postojećih istraživanja, kao i za osmišljanje novih rješenja. Cilj ovog doktorskog istraživanja bio je razviti model repozitorija meta-podataka (MDV) koji se zasniva na Data Vault (DV) metodi za modeliranje baza podataka. Ovako definiran repozitorij meta-podataka služi za integraciju skladišta podataka (SP) i sustava za upravljanje matičnim podacima (UMP) te za praćenje i upravljanje promjenama u SP/UMP podacima i meta-podacima, kao i u njihovim shemama. Na ovaj način evolucija sheme skladišta podataka provodi se isključivo uz proširenje postojeće sheme i bez gubitka informacija. Također, složenost provedbe evolucije SP/UMP sheme je smanjena u odnosu na tradicionalne pristupe zasnovane na relacijskom modelu, a repozitorij zasnovan na MDV modelu služi kao proširenje tradicionalnog relacijskog sistemskog kataloga. U svrhu izgradnje praktičnog prototipa i testiranja predloženog rješenja razvijen je trajni i obuhvatni model repozitorija meta-podataka za integraciju i praćenje promjena shema SP i UMP, sistematiziran je formalni konačni skup osnovnih promjena nad shemom SP i UMP, definirana je formalna algebra za održavanje SP i UMP sheme te je razvijena arhitektura integriranog SP i UMP. Na kraju je razvijen i sam praktični prototip koji služi za empirijsku verifikaciju predloženog rješenja.

**Ključne riječi:** skladište podataka, upravljanje matičnim podacima, relacijski model, data vault model, evolucija podataka, evolucija sheme, meta-podatak, repozitorij meta-podataka, sistemski katalog



## Abstract

Data Warehouse (DW) environment nowadays is an extremely dynamic one. On the one hand we have a number of (heterogeneous) data sources that are subject to frequent changes of data and structure, while on the other hand we have the frequent changes in the information requirements (set by business users). DW has an extremely complex task – it must at all times be able to adapt to changes from data sources as well as to satisfy user's requests for information. This problem that we explore here is known and recognized in literature as a DW evolution problem – tracking and storing the scope and structure changes of data and metadata for a very long time period. The academic community has taken some steps towards solving this problem but there is always some room for an improvement of the existing research, as well as for a development of new solutions. The goal of this doctoral thesis was to develop a metadata repository model (MDV) which is based on the Data Vault (DV) method for database modeling. Thus defined metadata repository model is used for integrating a data warehouse (DW) system and a master data management (MDM) system and for tracking and managing changes in the DW/MDM data and metadata, as well as in their schemas. In this way, a DW schema evolution is carried out only with the expansion of the existing schema and without loss of information. Also, the complexity of the DW schema evolution implementation is decreased compared to traditional approaches based on the relational model. Additionally, MDV repository serves as an extension of traditional relational database system catalog. In order to build a practical prototype and to test the proposed solution, a permanent and comprehensive metadata repository model for integration and tracking of DW/MDM data and schema changes was developed, a final set of fundamental changes over the DW/MDM schema was systematized, a formal algebra for DW/MDM schema maintenance was developed, an architecture of integrated DW/MDM was proposed, and a prototype of our dual DW/MDM solution was developed and empirically verified.

**Keywords:** data warehouse, master data management, relational model, data vault model, data evolution, schema evolution, metadata, metadata repository, system catalog

# Sadržaj

<b>1. UVOD</b>	<b>1</b>
1.1. MOTIVACIJA	2
1.2. CILJ, HIPOTEZE I ZNANSTVENI DOPRINOSI	3
1.3. STRUKTURA RADA	4
<b>2. EVOLUCIJA SKLADIŠTA PODATAKA</b>	<b>6</b>
2.1. POJAM SKLADIŠTA PODATAKA	6
2.1.1. Područno skladište podataka	6
2.1.2. ETL procesi	7
2.1.3. Poslovna inteligencija	7
2.2. POJAM METAPODATAKA	7
2.3. POJAM UPRAVLJANJA MATIČNIM PODACIMA	8
2.4. POJAM DATA VAULT	9
2.5. PREGLED ISTRAŽIVANJA	10
2.5.1. Evolucija sheme	11
2.5.2. Verzioniranje sheme	14
2.5.3. Održavanje pogleda	17
2.5.4. Analiza pregleda istraživanja	19
<b>3. OPIS ISTRAŽIVANJA</b>	<b>20</b>
3.1. ARHITEKTURA SP/UMP	23
3.2. KONCEPTUALNI MDV	26
3.3. LOGIČKI MDV	27
3.4. LOGIČKI MDV – DIO INTEGRACIJE SP/UMP	32
<b>4. TEORIJSKA VALIDACIJA ISTRAŽIVANJA</b>	<b>34</b>
4.1. DEFINICIJA DV MODELA	34
4.2. DOKAZ HIPOTEZE H2	35
4.2.1. Definicija skupa promjena i operacija promjene	36
4.3. PRAKTIČNI PRIMJER ZA VALIDACIJU OPERACIJA EVOLUCIJE NAD SKUPOM PROMJENA	47
4.3.1. Dodavanje nove relacije, $D_R$ i Dodavanje veze između relacija, $D_V$	50
4.3.2. Preimenovanje relacije, $I_R$	51
4.3.3. Brisanje relacije, $B_R$ i Brisanje veze između relacija, $B_V$	52
4.3.4. Dodavanje atributa u relaciju, $D_A$	52
4.3.5. Izmjena postojećeg atributa u relaciji - preimenovanje atributa, $I_{An}$	54
4.3.6. Izmjena postojećeg atributa u relaciji - promjena tipa podatka atributa, $I_{At}$	55
4.3.7. Brisanje atributa iz relacije, $B_A$	56
4.3.8. Pretvorba relacije u vezu, $I_{RV}$	56
4.3.9. Pretvorba veze u relaciju, $I_{VR}$	59
4.3.10. Pretvorba atributa u relaciju, $I_{AR}$	62
4.3.11. Pretvorba relacije u atribut, $I_{RA}$	64
4.4. DOKAZ HIPOTEZE H3	66
4.4.1. Promjene nad 3NF repozitorijem i SDV/PMDV repozitorijem	67
<b>5. EMPIRIJSKA VALIDACIJA ISTRAŽIVANJA</b>	<b>79</b>
5.1. OPIS PROTOTIPA RJEŠENJA	79
5.1.1. ETL za integraciju i punjenje SDV, PMDV i MDV repozitorija	81
5.2. PROVOĐENJE PROMJENA U IZVORIMA, SDVDB/PMDVDB I MDVDB	95
5.3. DOKAZ HIPOTEZE H1	111
5.4. UPITI ZA PRIKAZ KORISNOSTI RJEŠENJA	115

<b>6. ZAKLJUČAK.....</b>	<b>125</b>
<b>POPIS SLIKA.....</b>	<b>135</b>
<b>POPIS TABLICA .....</b>	<b>137</b>
<b>POPIS PRIVITAKA.....</b>	<b>138</b>
PRIVITAK 1. CJELOVITI MDV MODEL REPOZITORIJA METAPODATAKA .....	139
PRIVITAK 2. KONCEPTUALNI MDV MODEL REPOZITORIJA METAPODATAKA.....	140
PRIVITAK 3. POSLOVNA DEFINICIJA MDV MODELA .....	141
PRIVITAK 4. SQL DDL SKRIPTE ZA IZRADU PROTOTIPA .....	199
<b>ŽIVOTOPIS.....</b>	<b>307</b>

# 1. UVOD

Skladište podataka (SP) predstavlja kopiju transakcijskih podataka posebno strukturiranu za provođenje upita te izvođenje poslovnog izvještavanja i analize [43]. SP se može koristiti za analizu pojedinog tematskog područja (kao što su prodaja, marketing, financije, itd.), ono integrira tekuće i povijesne podatke iz više (heterogenih) izvora podataka, čuva povijest podataka te nikada ne mijenja podatke nakon što oni uđu u SP. Jednostavniji oblik SP je područno skladište podataka (PSP, *eng. data mart*), koje je često (iako ne nužno) izgrađeno, upravljano i korišteno od strane jednog odjela unutar poslovne organizacije. PSP je obično usmjereno na jednu temu ili poslovno područje te ne vuče svoje podatke od svih izvora podataka, već od manjeg broja njih (kao što je npr. maloprodaja s dnevnim prodajnim transakcijama ili inventar u skladištu). Izvori podataka u okruženju SP (i PSP) nisu samo heterogeni, već i često mijenjaju svoju strukturu i sadržaj. Također, i poslovne potrebe i analitički zahtjevi se mijenjaju s vremenom. Ove promjene moraju se propagirati i implementirati u SP, kako bi ono moglo ispravno prikazivati trenutno stanje u stvarnom svijetu te osigurati učinkovitu poslovnu analizu. Za logičku reprezentaciju SP (ili PSP), tradicionalno se koristi denormalizirani dimenzionalni model [43] u kombinaciji s relacijskim modelom u trećoj normalnoj formi (3NF) za prikaz središnjeg, integriranog SP (često nazivanog i operativna pohrana podataka, *eng. operational data store*) [34][43], ovisno o odabranoj arhitekturi SP [27].

S druge strane, sustav za upravljanje matičnim podacima (UMP, *eng. master data management*) obuhvaća procese, upravljanja, politike, standarde i alate koji dosljedno definiraju i upravljaju kritičnim podacima organizacije, s ciljem pružanja jedinstvene referentne točke [49][10]. UMP se tradicionalno koristi (koriste ga drugi poslovni sustavi, aplikacije, baze podataka i skladišta podataka) kao fizički neovisna baza matičnih i referentnih podataka. Matični podaci predstavljaju ključne poslovne subjekte i njihove opisne atribute (npr. kupac ima ime i prezime, adresu itd., proizvod ima ime, boju, težinu, kategoriju, itd.), koji se koriste u više sustava, aplikacija i poslovnih procesa organizacije kao jedinstven izvor podataka. Referentni podaci mogu biti unutarnji ili vanjski te se koriste za vrednovanje i validaciju drugih podataka. Neke od temeljnih zadaća UMP-a su uklanjanje duplikata, standardizacija podataka te ugradnja i provedba poslovnih pravila - sve s ciljem

onemogućavanja netočnih podataka da uđu u sustav i stvaranja autoritativnog izvora matičnih podataka za daljnju distribuciju. To je metoda koja omogućava poslovnoj organizaciji povezivanje svih svojih kritičnih podataka u zajedničku referentnu točku, koja se dalje dijeli po svim odjelima i relevantnim zaposlenicima. Na taj način, kvaliteta podataka se uvelike poboljšava i organizacija može bolje služiti svojim klijentima, kao i poboljšati svoje poslovanje. Za logičku reprezentaciju UMP podataka također se može koristiti dimenzionalni model (doduše u „inverznom“ obliku). UMP okruženje je također u stanju stalne promjene - izvori podataka mijenjaju svoju strukturu i sadržaj - i te promjene moraju biti točno i učinkovito propagirane i implementirane u UMP.

## **1.1. Motivacija**

Motivacija za istraživanje proizlazi iz uviđene sličnosti SP i UMP sustava, u vidu njihove namjene, načina organizacije podataka te tekućih problema u izgradnji i održavanju ovih sustava. Sličnost u njihovoj namjeni proizlazi iz činjenice da je cilj oba sustava pružanje što ispravnijeg skupa izvorišnih podataka kao temelja za točno i učinkovito poslovno izvještavanje. Dalje, oba sustava zasnivaju se na istom okruženju - integriraju iste, brojne izvore podataka te koriste sličnu vrstu reprezentacije podataka (za logičku reprezentaciju oba sustava koristi se dimenzionalni model). I na kraju, u oba sustava postoji problem evolucije podataka i sheme – prosljeđivanja i provođenja promjena nakon određenog evolucijskog događaja. Kod SP taj događaj može biti promjena u izvorima podataka ili u korisničkim zahtjevima, dok je kod UMP to češće samo promjena u izvorima podataka. SP treba sačuvati povijest promjena podataka i meta-podataka, kao i povijest promjena sheme i korisničkih zahtjeva, za vrlo dugo vremensko razdoblje. S druge strane, UMP treba sačuvati kvalitetu podataka organizacije kroz postizanje i održavanje "jedne verzije istine" skupa podataka. Zbog svih ovih razloga odlučili smo integrirati ova dva sustava, kako se problem evolucije SP i problem evolucije UMP ne bi morali rješavati odvojeno (a na taj način i smanjiti složenost i zahtjevnost samog procesa evolucije), već na zajedničkoj razini – unutar jedinstvenog sustava. U radu će biti predstavljen meta-model za integraciju ova dva sustava, dok će pristup rješavanju problema evolucije vrijediti jednako za oba ova sustava.

## **1.2. Cilj, hipoteze i znanstveni doprinosi**

Cilj ovoga rada je razviti i implementirati model repozitorija meta-podataka (MDV) za integraciju skladišta podataka (SP) i sustava za upravljanje matičnim podacima (UMP), koji će služiti za praćenje i upravljanje promjenama u podacima i meta-podacima SP i UMP, kao i u njihovim shemama. Opseg istraživanja ograničen je na relacijske izvore podataka te pristup evoluciji skladišta podataka kroz verzioniranje sheme.

U radu su postavljene tri istraživačke hipoteze:

### **H1: MDV repozitorij meta-podataka zasnovan na Data Vault modelu integrirat će skladište podataka i sustav za upravljanje matičnim podacima**

Hipoteza će se smatrati potvrđenom ukoliko se definira i razvije MDV model te se prema razvijenom MDV modelu izradi verzija prototipa sustava u kojem MDV repozitorij meta-podataka prikuplja i pohranjuje meta-podatke o povezivanju shema repozitorija „sirovih“ podataka (SDV) i repozitorija poslovnih i matičnih podataka (PMDV) te definirani upiti nad prototipnim implementacijama SP i UMP (MdmDB i DmDB) vraćaju isti skup rezultata. Potvrđenost ove hipoteze opisana je u Poglavlju 5.3.

### **H2: Evolucija sheme skladišta podataka provest će se isključivo uz nadogradnju (proširenje) postojeće sheme i bez gubitka informacija**

Hipoteza će se smatrati potvrđenom ukoliko se, prema izrađenom modelu i formaliziranom skupu osnovnih promjena, pokaže da se evolucija sheme može provesti koristeći samo dvije osnovne operacije evolucije te da upiti provedeni nad različitim dijelovima arhitekture prototipa vraćaju isti ili veći skup rezultata (očekivano centralni podatkovni repozitorij vraća isti ili veći skup rezultata u odnosu na repozitorij izvora podataka, a isto vrijedi i za usporedbu rezultata tradicionalnog sistemskog kataloga i njegovog proširenja u vidu MDV repozitorija). Potvrđenost ove hipoteze opisana je u Poglavlju 4.2.

### **H3: Složenost provedbe evolucije sheme skladišta podataka bit će smanjena, u odnosu na tradicionalne pristupe (zasnovane na relacijskom modelu)**

Hipoteza će se smatrati potvrđenom ukoliko se nad razvijenim modelom provede predloženi formalni skup osnovnih promjena te se pokaže da se za svaku osnovnu promjenu u izvorima podataka nad SP/UMP predloženim rješenjem koristi manji broj operacija evolucije, nego nad tradicionalnim relacijskim 3NF centralnim repozitorijem podataka (za koji se koriste principi

klasičnog verzioniranja sheme izradom kopija sheme, preuzetog iz pregleda dosadašnjih istraživanja i učestale prakse). Dodatno, bit će prikazana korisnost predloženog modela i programskog rješenja u odnosu na postojeći sistemski katalog izvođenjem temporalnih upita nad MDV repozitorijem i sistemskim katalogom. Potvrđenost ove hipoteze opisana je u Poglavlju 4.4.

Prilikom izrade ovoga rada postignuti su sljedeći znanstveni doprinosi:

- Definiran je i razvijen trajni model repozitorija meta-podataka za praćenje promjena shema skladišta podataka i sustava za upravljanje matičnim podacima
- Definiran je i razvijen obuhvatni model za integraciju skladišta podataka i sustava za upravljanje matičnim podacima
- Sistematiziran je formalni konačni skup osnovnih promjena nad shemom skladišta podataka i sustava za upravljanje matičnim podacima
- Definirana je formalna algebra za održavanje sheme skladišta podataka i sustava za upravljanje matičnim podacima
- Razvijena je i opisana arhitektura integriranog skladišta podataka i sustava za upravljanje matičnim podacima
- Razvijen je i empirijski verificiran prototip integriranog rješenja

### **1.3. Struktura rada**

Rad je organiziran kako slijedi:

U Poglavlju 1 predstavljen je kratki uvod u temu ovoga doktorskoga istraživanja, kao i motivacija za istraživanje te osnovni cilj, hipoteze i znanstveni doprinosi istraživanja.

U Poglavlju 2 predstavljeni su i definirani neki osnovni pojmovi važni za ovo istraživanje (poput skladišta podataka, upravljanja matičnim podacima, meta-podataka, Data Vault metode za modeliranje i sl.). Dodatno, u ovome poglavlju nalazi se pregled dosadašnjih istraživanja u području evolucije skladišta podataka, kao i kratka analiza dosadašnjih pristupa.

U Poglavlju 3 nalazi se opis osnovnih ideja istraživanja te je predstavljen i opisan MDV model repozitorija meta-podataka, kao i arhitektura predloženog rješenja.

U Poglavlju 4 nalazi se teorijska validacija predloženog istraživanja. Ovdje je predstavljen i definiran osnovni skup promjena nad izvorišnom shemom, kao i osnovne operacije promjene sheme. Formalno je definiran DV (kao i MDV) model te su dokazane hipoteze H2 i H3.

U Poglavlju 5 opisana je empirijska validacija istraživanja. Ovdje je predstavljen i opisan prototip rješenja, ETL paketi korišteni za izradu prototipa, kao i skup promjena koje su nad prototipom provedene. Prikazan je i skup upita nad prototipom (meta-podatkovnim MDV repozitorijem, kao i podatkovnim repozitorijima) za prikaz korisnosti rješenja. Također, u ovome poglavlju nalazi se dokaz hipoteze H1.

U Poglavlju 6 nalazi se kratka rasprava o trenutno ostvarenome istraživanju prikazanome u ovome doktorskomu radu, planovi za budućnost te zaključak.



## 2. EVOLUCIJA SKLADIŠTA PODATAKA

U ovome poglavlju definirani su pojmovi skladišta podataka, upravljanja matičnim podacima i metapodataka. Opisan je i pregled dosadašnjih istraživanja u području evolucije skladišta podataka.

### 2.1. *Pojam skladišta podataka*

Skladište podataka (SP) predstavlja posebno organiziranu centralnu bazu podataka koja integrira brojne heterogene izvore podataka te omogućuje brzu i efikasnu analizu za potrebe poslovanja. Dvije su najpoznatije definicije SP:

- “Skladište podataka je kolekcija podataka koja pruža podršku procesima odlučivanja. Skladište podataka ima sljedeće karakteristike: tematski je orijentirano, integrirano je i konzistentno, prikazuje vlastiti razvoj kroz vrijeme i nepromijenljivo je.” [34]
- “Skladište podataka je kopija transakcijskih podataka specifično strukturirana za postavljanje upita, analizu i izvještavanje.” [43]

Dakle, nužne karakteristike SP su prikupljanje i pružanje informacija prema pojedinim tematskim cjelinama, konzistentno spajanje i integracija podataka iz brojnih heterogenih izvora podataka te čuvanje i neuništavanje podataka kroz veoma veliki vremenski period.

#### 2.1.1. *Područno skladište podataka*

Područno skladište podataka (*eng. data mart*) sadrži podskup podataka pohranjenih u skladištu podataka, najčešće pohranjenih i organiziranih prema tematskim cjelinama, skupu korisnika kojima su namijenjeni ili dijelovima poslovne organizacije. Podaci pohranjeni u područnom SP obično su na neki način transformirani, denormalizirani i agregirani, u odnosu na podatke pohranjene u SP te su tradicionalno organizirani prema dimenzijskom modelu [43]. Područno SP izrađeno prema dimenzijskom modelu (popularno nazvanom „zvjezdana shema“) najčešće ima tablicu činjenica u centru te tablice dimenzija (često nazivane samo „dimenzije“) koje je okružuju. Tablice činjenica pohranjuju takozvane „mjere“, odnosno numeričke podatke koji se mogu mjeriti, dok dimenzije pohranjuju opisne podatke koji daju značenje i kontekst mjerama.

### 2.1.2. *ETL procesi*

ETL procesi (*eng. extract, transform and load*) predstavljaju procese izvlačenja, transformacije i punjenja podataka, iz izvora podataka pa kroz sve slojeve SP arhitekture. To je skup koraka i aktivnosti potrebnih da bi se podaci iz izvora podataka unijeli u centralno SP te se iz centralnog SP transformirali i unijeli u područna SP. Ukratko, izvlačenje (*eng. extraction*) predstavlja izvlačenje podataka iz homogenih ili heterogenih izvora podataka; transformacija (*eng. transformation*) predstavlja pretvorbu podataka kako bi se oni mogli u centralnom SP ili u područnim SP pohraniti u odgovarajuće strukture ili u odgovarajućem formatu; punjenje (*eng. loading*) predstavlja punjenje podataka u konačnu destinaciju (najčešće centralno SP ili područno SP) [34][43]. Na tržištu danas postoji veliki broj komercijalnih ETL alata i aplikacija, kao i onih otvorenog koda.

### 2.1.3. *Poslovna inteligencija*

Poslovna inteligencija (PI) je pojam koji se općenito odnosi na veliki broj funkcionalnosti pruženih krajnjem poslovnom korisniku, u svrhu što bolje prezentacije i analize poslovnih podataka na temelju kojih se donose svakodnevne i strateške poslovne odluke. Svi alati i aplikacije poslovne inteligencije pretražuju i rade upite nad podacima u skladištu podataka (najčešće direktno nad područnim SP, jer su podaci ovdje denormalizirani i agregirani te samim time i pogodniji za brže provođenje upita te brzu poslovnu analizu) [43].

## 2.2. *Pojam metapodataka*

Za meta-podatke možemo tradicionalno reći da su to „podaci o drugim podacima“. Oni pružaju kontekst i dodatne informacije o podacima pohranjenima u bazu podataka. Promatrano u kontekstu skladišta podataka, meta-podaci imaju važnu ulogu u standardizaciji, pročišćavanju i transformiranju podataka, kao i povezivanju podataka između različitih struktura u SP. Kimball [42] opisuje meta-podatke kao DNA skladišta podataka, jer meta-podaci definiraju sve elemente SP te opisuju na koji način oni zajedno funkcioniraju. Prema [42] postoje tri glavne kategorije meta-podataka: tehnički meta-podaci, poslovni meta-podaci i procesni meta-podaci. Tehnički meta-podaci definiraju i opisuju objekte i procese u SP sustavu, promatrano iz tehničke perspektive. U ovu vrstu meta-podataka pripadaju sistemski meta-podaci koji

definiraju baze podataka, tablice, stupce, tipove podataka, indekse, dimenzije, mjere i sl. Poslovni meta-podaci su nešto više prilagođeni krajnjem korisniku – oni opisuju koje podatke imamo u SP, otkuda su oni došli, što znače i kako su povezani s ostalim podacima u SP. Oni najčešće služe kao dokumentacija za SP. Procesni meta-podaci opisuju rezultate raznih operacija u SP, poput rezultata ETL procesa. To su najčešće početak i završetak ETL zadatka, broj obrađenih redaka i sl. Također, neke dodatne kategorizacije i pristupi organizaciji meta-podataka korišteni prilikom izrade ovoga doktorskoga rada mogu se naći u [51][68][37].

### **2.3. Pojam upravljanja matičnim podacima**

Već smo rekli da sustav za upravljanje matičnim podacima (UMP, *eng. master data management*) obuhvaća procese, upravljanja, politike, standarde i alate koji dosljedno definiraju i upravljaju kritičnim podacima organizacije, s ciljem pružanja jedinstvene referentne točke [49][10] te da se UMP tradicionalno koristi kao fizički neovisna baza matičnih i referentnih podataka. Podaci u UMP su također pohranjeni prema dimenzijskom modelu, gdje su matični podaci ekvivalentni dimenzijama. Obzirom da se matični podaci s velikom vjerojatnošću pojavljuju u gotovo svim aplikacijama poslovne organizacije, vrlo je važno na koristan i ispravan način upravljati njima. Idealni scenarij bio bi kad bi sve poslovne aplikacije koristile iste matične podatke (prikupljene na razini cijele poslovne organizacije). Da bi se to postiglo, za implementaciju UMP sustava može se birati između nekoliko pristupa [49][10]:

- Bez centralnog UMP – u ovome pristupu sustavi i aplikacije uopće ne komuniciraju, već se podaci spajaju ad-hoc (po potrebi) te je izvedba jako skupa s vremenom
- Centralni repozitorij meta-podataka – u ovome pristupu postoje centralne, integrirane definicije matičnih podataka, koje svi sustavi i aplikacije implementiraju kako bi ad-hoc spajanja podataka bila nešto jednostavnija
- Centralni repozitorij meta-podataka s povezivanjem identiteta – ovaj pristup, u odnosu na prethodni, pohranjuje još i ključeve koji povezuju tablice u UMP sustavu, što ubrzava proces integracije podataka, ali i otežava održavanje matičnih podataka kroz vrijeme
- Centralni repozitorij meta-podataka i centralni podaci – u ovome pristupu pohranjuju se i matični podaci i meta-podaci u UMP sustav, međutim matični podaci se i dalje

unose i ažuriraju u izvorišnim sustavima te se integriraju i pročišćavaju u UMP-u (što je i dalje prilično skup pristup).

- Centralni UMP s jednom kopijom – u ovome pristupu imamo posebnu UMP aplikaciju koja prikuplja, integrira i održava matične podatke skupa s njihovim meta-podacima u jednoj centralnoj lokaciji, a svi izvorišni sustavi i aplikacije koriste te podatke. Nedostatak ovog pristupa je potreba za (skupom i dugotrajnom) nadogradnjom svih izvorišnih sustava i aplikacija, kako bi koristili samo tu centralnu lokaciju.
- Centralni UMP s više kopija - u ovome pristupu također imamo centralnu UMP aplikaciju za integriranje i održavanje matičnih podataka i njihovih meta-podataka, međutim ovdje svaki izvorišni sustav održava i svoju kopiju matičnih podataka, a matični podaci iz UMP se repliciraju u izvorišne sustave, što također povlači određene nedostatke u vidu konflikata kod ažuriranja te (skupa) konstantna spajanja.

## **2.4. Pojam Data Vault**

Data Vault (DV) je metoda za modeliranje baza iskladišta podataka, posebno osmišljena za izgradnju SP za dugotrajnu pohranu povijesnih podataka, prikupljenih iz raznih izvora [47],[48]. Strukturalni podaci su eksplicitno razdvojeni od opisnih atributa, bez obzira iz kojeg izvora dolaze, što omogućava brzo paralelno punjenje te smanjuje troškove. Također, model postaje fleksibilan na promjene u poslovnom okruženju. DV metoda se zasniva na pretpostavci da je okruženje SP u stanju stalne promjene te naglašava potrebu za praćenjem podrijetla podataka sadržanih u bazi, kroz empirijski definiran skup metapodataka (poput *datum\_punjenja*, *datum\_kraja\_punjenja* i *izvor\_podatka*). Samim time se omogućuje praćenje vrijednosti natrag do izvora podataka i praćenje povijesti promjena. Dodatno, DV metoda ne razlikuje dobre i loše podatke, već se svi podaci pohranjuju, bez obzira iz kojeg izvora dolaze i jesu li prilagodljivi poslovnim pravilima. Na ovaj način se izbjegava gubitak informacija – svi podaci se pohranjuju, u svako vrijeme. Sve promjene se u model implementiraju isključivo kao nezavisna proširenja postojećeg modela, što znači da promjene ne utječu na postojeće aplikacije te se sve verzije aplikacija mogu zasnivati na istoj, razvijajućoj bazi podataka. Sve verzije sheme su podskup DV sheme.

## 2.5. Pregled istraživanja

Problem korištenja višestrukih heterogenih shema javio se već kod rada s relacijskim i objektno-orijentiranim bazama podataka. Unatoč promjenama u strukturi baze podataka, potreba za zadržavanjem postojećih podataka je ostala - imperativ je da podaci prežive promjene u shemi baze podataka. Kao rješenje ovoga problema, pojavila su se dva pristupa – evolucija sheme i verzioniranje sheme. Oba pristupa zasnivaju se na uvođenju inteligentnog upravljanja vremenskim neusklađenostima između podataka i strukture podataka. Kod evolucije sheme pohranjuje se samo trenutna verzija sheme, dok se kod verzioniranja sheme pohranjuje više odabranih verzija sheme, uz korištenje simboličkih oznaka ili, češće, oznaka datuma i vremena. Zapravo evoluciju sheme, radi jednostavnosti, možemo smatrati posebnim slučajem verzioniranja sheme. Roddick u [61] daje izvrstan pregled glavnih pitanja i karakteristika ova dva pristupa na razini relacijskih i objektno-orijentiranih baza podataka. U slučaju skladišta podataka, u literaturi razlikujemo tri pristupa evoluciji skladišta podataka – evolucija sheme ([4],[9],[11],[15],[22],[23],[31],[32],[52],[53],[67],[69]), verzioniranje sheme ([6],[7],[12],[20],[25],[26],[50],[55],[65],[72]) i održavanje pogleda ([1],[2],[5],[8],[13],[16],[30],[36],[46],[59],[60],[62],[63],[70]). Prva dva pristupa se zasnivaju na skladištu podataka definiranom kao višedimenzionalna shema, dok se održavanje pogleda zasniva na skladištu podataka koje je postavljeno kao skup materijaliziranih pogleda.

Kao što je već spomenuto, problem evolucije sheme je prisutan i kod baza podataka, gdje su istraživanja uglavnom usmjerena na pamćenje stanja baze podataka. Zanimljiva su i neka rana istraživanja poput [3], u kojem je predstavljen model verzija za evoluciju sheme baze podataka, [18] u kojem autori definiraju povijesni relacijski model podataka uz pripadajući skup temporalnih operatora nad oznakama datuma i vremena i intervalima, te [17] u kojem autori predstavljaju PRISM, sustav koji podupire evoluciju sheme baze podataka, tako što pruža jezik operatora za izmjenu sheme (SMO) za opis kompleksnih promjena sheme te povećava predvidivost evolucije kroz automatsku provjeru očuvanja informacija, redundantnosti i podrške upitima. Također, PRISM omogućuje automatsku migraciju podataka te potpunu dokumentaciju procesa evolucije sheme. U [19] i [38] autori dalje proučavaju problem temporalnih baza podataka, odnosno korištenja temporalnih podataka u relacijskim bazama podataka. U [19] autori su opisali osnovne probleme temporalnih

podataka te su predstavili temeljne konstrukte i operatore za rješavanje tih problema. U [38] autori su predstavili Asertivno verzioniranje, konceptualni pristup pomoću kojeg se smanjuje trošak i povećava vrijednost temporalnih podataka.

U odnosu na baze podataka, zahtjevi skladišta podataka su povećani u pogledu opsega i pamćenja promjena te su i istraživanja problema evolucije skladišta podataka nešto šireg opsega. U [56] dan je kratki pregled istraživanja unutar navedena tri pristupa evoluciji skladišta podataka.

Evolucija sheme pretpostavlja da shema može imati samo jednu verziju u određenom vremenu – onu trenutnu. Svi podaci su pohranjeni u trenutnoj verziji te se sve promjene vrše nad trenutnom verzijom, koja se zatim transformira u novu verziju. Problem ovoga pristupa je gubitak povijesti – ne čuvaju se prethodne verzije pa prethodno dostupne strukture i podaci u novoj verziji mogu postati nedostupni. Evoluciju sheme možemo promatrati kao slabiji slučaj verzioniranja sheme. Verzioniranje sheme pretpostavlja da više verzija sheme mogu biti važeće u različitim vremenskim intervalima. Verzioniranjem sheme čuva se povijest verzija – promjene u shemi skladišta podataka uzrokuju kreiranje nove verzije sheme, kojoj se dodjeljuje vremenska oznaka. Čuvaju se sve verzije sheme skladišta podataka. Održavanje pogleda pretpostavlja da je skladište podataka skup materijaliziranih pogleda, direktno napunjenih podacima iz izvora podataka. Materijalizirane poglede zatim treba osvježavati i prilagođavati, ovisno o raznim promjenama u izvorima podataka.

U nastavku slijedi pregled radova i istraživanja vezanih uz navedene pristupe.

### 2.5.1. *Evolucija sheme*

Ovaj pristup odnosi se na skladište podataka definirano kao višedimenzionalna shema s tablicama činjenica i dimenzija te podatkovnim kockama. Evolucija sheme sadrži više različitih razina ažuriranja, poput ažuriranja dimenzija, instanci, razina, činjenica, atributa, podatkovnih kocki, hijerarhija, mjera, ograničenja, kvalitete ili strukture. Naravno, neke vrste ažuriranja predstavljaju više statički aspekt razvoja skladišta podataka (poput ažuriranja dimenzija), dok neke vrste ažuriranja predstavljaju više dinamički aspekt razvoja skladišta podataka (npr. ažuriranje strukture). Kada se u literaturi govori o evoluciji sheme, naglasak je na ažuriranju dimenzija, instanci, razina, hijerarhija, činjenica i atributa.

U [31] autori su, uz 5 operatora za ažuriranje dimenzija, predložili i 2 operatora za ažuriranje instanci: Generalizirati, Specijalizirati, Povezati razine, Nepovezati razine, Brisati razinu, Dodati instancu i Brisati instancu. Kako bi kontrolirali utjecaj strukturalnih promjena u dimenziji na podatkovnu kocku, autori su predložili prilagodbu podatkovne kocke nakon korištenja operatora za dodavanje i brisanje dimenzije te dodavanje i brisanje instance. Za svaki pogled podatkovne kocke, predložili su izračunavanje izraza za njegovo održavanje. U [32] autori su dodatno predložili 4 složena operatora za ažuriranje instanci. U [11] autori su definirali formalni okvir za opis evolucije višedimenzionalnih shema, koji sadrži opis samih shema i instanci. Model je definiran na sljedeći način: MD model  $M$  je 6-torka  $(F, L, A, Gran, Class, Attr)$  gdje je  $F$  konačan skup naziva činjenica,  $L$  je konačan skup naziva razina dimenzija,  $A$  je konačan skup naziva atributa,  $Gran$  je funkcija koja povezuje činjenicu sa skupom naziva razina dimenzija,  $Class$  je relacija definirana na nazivu razine,  $Attr$  je funkcija koja veže atribut uz određenu činjenicu ili razinu dimenzije. Nakon što su definirali model, autori su predložili i skup formalnih operacija evolucije skladišta podataka. Ideja je bila da se dizajneru omogući da odredi željenu evoluciju sheme na konceptualnoj razini, nakon čega slijedi automatska prilagodba implementacije. Međutim, autori su se fokusirali samo na promjene sheme skladišta podataka koje nastaju zbog promjene poslovnih zahtjeva. U [69] autori su predložili proširenje rada iz [31] i [32], tako što su predložili alat za vizualizaciju za dimenzije i podatkovne kocke. U [15] autori su definirali sljedećih 6 operatora za evoluciju sheme: Dodati atribut, Brisati atribut, Preimenovati atribut, Dodati tablicu, Brisati tablicu, Preimenovati tablicu. U [52] autori su predložili mehanizam za dobivanje logičke sheme skladišta podataka putem unaprijed definiranih transformacija koje se primjenjuju na logičke sheme izvora. Ovime se omogućuje praćenje i dokumentiranje dizajna i povezivanja (eng. *mapping*) između logičkih struktura skladišta podataka i izvora podataka. Mehanizam sadrži dvije vrste pravila – pravila konzistentnosti za osiguranje dosljednosti dobivene sheme, te strategije dizajna koje sadrže različita rješenja za dizajniranje skladišta podataka. Također, autori su predložili skup transformacija s njihovim opisima i primjerima upotrebe. U [22] autori su predstavili AutoMed, heterogeni sustav za transformaciju i integraciju podataka, koji ima sposobnost integracije podataka preko višestrukih modela podataka. Shema se inkrementalno mijenja, kako se na nju primjenjuje niz primitivnih transformacija, a svaka primitivna transformacija dodaje, briše ili preimenuje jedan konstrukt sheme. Pomoću AutoMed-ovih transformacija autori su prikazali evoluciju sheme skladišta podataka uzrokovanu promjenama u shemama izvora,

promjenama u jeziku za modeliranje, ili objema promjenama. U [9] autori su definirali WHES (eng. *Warehouse Evolution System*) prototip koji opisuje evoluciju skladišta podataka te omogućava ažuriranje dimenzija i podatkovnih kocki. Također, proširili su SQL jezik u MDL (eng. *Multidimensional Data definition Language*) jezik. Autori su predložili 8 operatora za ažuriranje dimenzija, te odgovarajućih 8 operatora za ažuriranje podatkovnih kocki. U [41] autori su predložili osam operatora evolucije sheme, koji su uglavnom usmjereni na ažuriranje dimenzija i razina te ažuriranje atributa dimenzija i mjera. Predloženi operatori su sljedeći: Dodati dimenziju u činjenicu, Brisati dimenziju, Dodati razinu, Brisati razinu, Spojiti atribut s razinom dimenzije, Razdvojiti atribut od razine dimenzije, Dodati mjeru, Brisati mjeru. U [4] autori su predložili formalizam za predstavljanje sheme skladišta podataka i određivanje ispravnosti operatora evolucije primijenjenih na shemi. Također, autori su predstavili alat koji brzo može kreirati shemu skladišta podataka te provjeriti valjanost operacija za evoluciju sheme, kroz upotrebu okidača i pohranjenih procedura. Predloženi su i operatori za ažuriranje proširenih hijerarhija: Višestruke hijerarhije, Ne-pokrivajuće hijerarhije, Ne-prema hijerarhije, Ne-stroge hijerarhije. U [53] autori su dali pregled dosadašnjih istraživanja na području evolucije sheme skladišta podataka te su predložili pet operatora za ažuriranje agregirane tablice činjenica. Agregirana tablica činjenica sadrži unaprijed sastavljene izračune na najvišoj razini zrnitosti te sadrži nove metrike (agregirane činjenice ili sažete statistike) dobivene primjenom zbirnih funkcija. Autori su predložili sljedeće operatore: Kreirati agregaciju, Brisati agregaciju, Izmijeniti agregaciju, Preimenovati agregaciju, Ukloniti agregaciju. U [67] autori su proučavali složene proširene hijerarhije te su predložili operatore evolucije i ograničenja koja treba zadovoljiti, kako bi se osigurao integritet podataka i ispravnost sheme. Za razliku od [4], ovdje autori razlikuju varijacije višestrukih hijerarhija te su predložili dodatna tri operatora za ažuriranje hijerarhija: Višestruka alternativna hijerarhija, Paralelna zavisna hijerarhija i Paralelna nezavisna hijerarhija. Definirana su i ograničenja nad navedenim vrstama višestrukih hijerarhija, koja moraju biti zadovoljena radi postizanja ispravnosti sheme. U [23] autori proučavaju utjecaj evolucije sheme skladišta podataka na skup od njega zavisnih tržišta podataka (eng. *data mart*). Predložena je arhitektura za evoluciju skladišta podataka, koja uključuje horizontalnu i vertikalnu evoluciju. Vertikalna evolucija, između ostalog, sadrži generični model propagacije (eng. *Generic Propagation Model, GPM*), koji opisuje kako se promjene u shemi skladišta podataka transformiraju u promjene sheme tržišta podataka.



Također, napravljen je pregled operatora za evoluciju tržišta podataka te je definiran skup pravila za propagaciju.

### 2.5.2. *Verzioranje sheme*

Verzioranje sheme sastoji se od prebacivanja podataka iz postojeće sheme u novu shemu te se sve promjene vrše na novoj shemi. Pritom se čuvaju sve stare verzije shema kroz korištenje vremenske ekstenzije na verziju sheme ili kroz fizičku pohranu verzija. U [20] autori su predložili proširenje višedimenzionalnog modela – temporalni višedimenzionalni model, koji omogućuje pohranjivanje vremenskih verzija dimenzijskih podataka. Predložena su povezivanja za prijenos podataka između različitih vremenskih verzija, kako bi sustav ispravno mogao odgovoriti na upite nad različitim verzijama dimenzijskih podataka. U [12] autori su predložili operatore za ažuriranje dimenzija (Kreirati/Brisati dimenziju, Kreirati/Brisati hijerarhiju, Kreirati/Brisati razinu, Pomaknuti razinu u hijerarhiji) i instanci (Kreirati/Brisati člana, Izmijeniti naziv ili svojstvo člana, Spojiti n članova u jednog člana, Podijeliti jednog člana u n članova, Reklasifikacija člana u dimenziji). Za svaku promjenu definira se nova, vremenski ograničena verzija. U [55] autori su predložili petnaest operatora za izmjenu sheme skladišta podataka: Kreirati novu tablicu razine, Povezati tablicu razine s njenim pod/nad tablicama razine, Razdvojiti tablicu razine iz njene hijerarhije dimenzije, Brisati razinu iz sheme, Dodati atribut u razinu, Brisati atribut iz razine, Izmijeniti domen atributa razine, Kreirati novu tablicu činjenica, Dodati atribut u tablicu činjenica, Povezati tablicu činjenica s dimenzijom, Brisati atribut koji nije primarni ili vanjski ključ iz tablice činjenica, Brisati vezu (vanjski ključ) između tablice činjenica i dimenzije, Brisati tablicu činjenica, Preimenovati atribut, Brisati tablicu. Autori predlažu primjenu ovih operatora na novoj verziji skladišta podataka. U [25] autori su proučavali evoluciju hijerarhija unutar dimenzija. Autori su predložili sljedeće operatore za evoluciju sheme: DodatiA (dodavanje novog atributa u dimenziju), BrisatiA (brisanje atributa iz dimenzije), DodatiF (unos nove hijerarhije između dva atributa u dimenziji), BrisatiF (brisanje hijerarhije iz dimenzije). Također, autori su predložili spremanje uvećane sheme zajedno s novom verzijom sheme. U [6] autori su predstavili prototip više-verzijskog skladišta podataka koje podupire promjene u strukturi sheme te „what-if“ analizu. Također, autori su prikazali korištenje tehnike dijeljenja podataka, kojom se u novoj verziji skladišta podataka pohranjuju samo oni podaci koji su novi ili izmijenjeni u odnosu na

prethodnu verziju, te podaci koji su vezani uz roditeljsku verziju (a dijele ih verzije-djeca). Definirana su dva tipa verzija: prava verzija i alternativna verzija. Prava verzija prati promjene u stvarnom svijetu, tj. promjene u vanjskim izvorima podataka. Alternativna verzija prati promjene koje se događaju u „what-if“ analizi. Važno je za napomenuti da su autori postavili i vremensko ograničenje na verziju, pri čemu svaka verzija ima svoje vrijeme trajanja unutar kojega je ona važeća verzija. U [7] autori su proširili rad iz [6] tako što su predložili dvije grupe operatora: operatore za promjenu sheme (petnaest operatora, uglavnom usmjereni na ažuriranje dimenzija, instanci, atributa, razina i činjenica) i operatore za strukturnu promjenu instanci dimenzija (pet operatora, uglavnom usmjereni na ažuriranje razina instanci). U [57] autori su proučavali problem izvođenja „what-if“ analize za promjene koje se događaju nad shemom izvora podataka te su predstavili model grafa koji uniformno modelira relacije, upite, poglede, ETL aktivnosti i njihova stanja. Ovaj model grafa omogućuje predviđanje učinka promjene na sveukupni sustav. Autori definiraju okvir i opći mehanizam za izvođenje „what-if“ analize nad potencijalnim promjenama izvora podataka. U [26] autori su predstavili X-Time, istraživački prototip za upravljanje verzioniranjem shema u relacijskim skladištima podataka. X-Time je specifično orijentiran na formulaciju među-verzijskih upita te podupire definiranje i punjenje uvećanih shema i omogućava korisniku da vizualno formulira među-verzijske upite nad grafom sheme. U [65] predstavljen je okvir za evoluciju skladišta podataka, koji podupire sljedeće promjene: unos, brisanje i preimenovanje izvorne relacije te unos, brisanje i promjena tipa izvorne relacije. Predloženi okvir automatizira evoluciju sheme skladišta podataka i kreiranje novih verzija te omogućuje prilagodbu ETL procesa i postojećih izvještaja nad shemom skladišta podataka. Za razliku od prethodnih radova, koji razmatraju samo jedan vid evolucijskih problema (promjene sheme uzrokovane promjenama poslovnih zahtjeva ili promjene sheme uzrokovane promjenama u izvorima podataka), predloženi okvir promatra oba aspekta evolucije sheme skladišta podataka. U [72] autori su predložili model MV-DW, za upravljanje više-verzijskim skladištima podataka, zasnovan na verzioniranju sheme i instanci skladišta podataka. U svrhu očuvanja konzistentnosti, MV-DW model podupire nekoliko postojećih strukturalnih i temporalnih ograničenja te definira ograničenje temporalne zavisnosti između dimenzije i činjenice, kako bi se osigurala zbrojivost mjera kroz više dimenzijskih verzija.

**Tablica 1. Prikaz predloženih operacija evolucije po istraživanjima**

	Promjene atributa	Promjene ograničenja	Promjene kocke	Promjene dimenzije	Promjene činjenice	Promjene hijerarhije	Promjene instance	Promjene razine	Promjene mjere	Promjene agregirane činjenice	Nova verzija sheme	Prava i alternativna verzija	Promjene ETL procesa
Hurtado				•			•	•					
Hurtado							•						
Biaschka	•			•	•			•					
Vaisman			•	•			•	•					
Chen	•			•	•								
Marotta	•			•	•	•			•				
Fan	•			•	•	•	•						
Benitez	•		•	•	•			•	•				
Kaas	•			•				•	•				
Banerjee			•	•	•	•		•					
Meenakshi										•			
Talwar		•				•							
Feki	•	•		•	•	•		•	•				
Eder	•			•		•	•	•			•		
Body	•			•		•	•	•			•		
Morzy	•	•		•	•	•		•			•		
Golfarelli	•			•		•		•			•		
Bebel											•	•	
Bebel	•			•	•		•	•			•		
Papastefanatos	•	•											•
Golfarelli											•		
Solodovnikova											•		•
Zouari Turki		•		•	•		•				•		

### 2.5.3. *Održavanje pogleda*

Pogled predstavlja relaciju izvedenu nad osnovnom relacijom (pohranjenom u izvoru podataka), odnosno funkciju definiranu između skupa osnovnih tablica i izvedene tablice. Svaki put kad je pogled pozvan, on se izračunava ispočetka. Pogled je materijaliziran kada se njegove n-torke pohrane u bazu podataka. Kod materijaliziranog pogleda nije potrebno ponovno izračunavati funkciju, već je dovoljno samo pristupiti podacima pohranjenima u bazi podataka. Velika prednost materijaliziranog pogleda je to što omogućava brz pristup podacima. Za još brži pristup podacima, nad materijaliziranim pogledom mogu se kreirati indeksi.

Skladište podataka može biti definirano kao skup materijaliziranih pogleda nad izvorima podataka [13],[8],[62],[63]. Problemi koji se ovdje javljaju jesu kako efikasno održavati poglede (kako prilagoditi i sinkronizirati poglede nakon promjena u definiciji ili opsegu pogleda)[30],[63][63] te kako efikasno odabrati poglede za veći učinak (koje agregacije na kojoj razini povećavaju učinak s obzirom na prostorna ograničenja) [13].

Problem održavanja materijaliziranog pogleda razmatra se u odnosu na održavanje kvalitete podataka, promjene u podacima te promjene u izvorima podataka. Održavanje pogleda možemo podijeliti na dva pristupa: prilagodba pogleda i sinkronizacija pogleda. Kod prilagodbe pogleda, da bi se materijalizirani pogled prilagodio promjenama, dodaju mu se meta-podaci koji sadrže najnovije strukturalne informacije. Za razliku od prilagodbe pogleda, sinkronizacija pogleda predstavlja ponovno pisanje pogleda. U [30] autori su opisali materijalizirane poglede, vrste primjene te probleme i tehnike za njihovo održavanje. Autori su predstavili klasifikaciju problema održavanja pogleda, prema kojoj se problem održavanja pogleda može proučavati kroz četiri dimenzije – dimenzija informacije, dimenzija promjene, dimenzija jezika i dimenzija instance. Također, opisali su nekoliko algoritama za inkrementalno održavanje pogleda. U [8] autori su predstavili formalni model za evoluciju sheme skladišta podataka, koji ima samo dva operatora – Dodati pogled i Brisati pogled. U [60] autor je predstavio dinamičku prilagodbu pogleda vezanu uz promjene u izvorima podataka. Ovim pristupom dobiva se na jednostavnosti – novi pogled se kreira iz starog pogleda, a ne ponovno iz različitih izvora podataka. Autor je ovaj problem proučavao iz dvije perspektive – korisnika i izvora podataka. Do korisničkih promjena nad shemom materijaliziranog pogleda dolazi kad

se postave novi zahtjevi. S druge strane, izvori podataka mogu mijenjati svoju shemu, što utječe na strukturalnu konzistentnost skladišta podataka. U [60] autori su definirali C-SQL (eng. *Cooperative SQL*), proširenje SQL jezika za pisanje pogleda. U [46] autori su definirali S-SQL (eng. *Schema SQL*) za integraciju relacijskih baza podataka i meta-podataka. U [16] autori su proučavali problem praćenja podrijetla podataka (eng. *lineage tracing*). Formalizirali su problem i razvili algoritam za praćenje podrijetla podataka za relacijske ASPJ (eng. *aggregate-select-project-join*) poglede. Također, predložili su nekoliko shema za pohranjivanje pomoćnih pogleda te su prezentirali studiju izvodivosti predloženih shema u određenim okruženjima. U [62] autori su predstavili prototip EVE, sustav za automatiziranje ponovnog pisanja definicija pogleda. Cilj EVE sustava je rješavanje problema nefleksibilnosti pogleda, odnosno očuvanje najvećeg broja definicija pogleda po promjenama sheme u izvorima podataka. EVE sustav pomoću omotača integrira izvore podataka te pretvara modele izvora podataka u zajednički relacijski model. Autori i tvorci EVE sustava predstavili su i E-SQL jezik, prošireni SQL jezik za predstavljanje evolucije definicije pogleda te MISD, model za opis izvora podataka. U [63] autori su predstavili formalni okvir SDCC (eng. *Schema change and Data update Concurrency Control*), kao pristup za rješavanje problema istovremenog ažuriranja sheme i podataka. U [59] fokus je na kvaliteti podataka za vrijeme evolucije sheme. Autor je predstavio više operatora na osnovne relacije i poglede te njihov utjecaj na kvalitetu podataka. Autor je za svaki operator definirao njegov utjecaj na kvalitetu podataka, pa tako operatori mogu utjecati na točnost, cjelovitost i konzistentnost logičke sheme (Dodati osnovnu relaciju/pogled, Obrisati osnovnu relaciju/pogled, Dodati atribut u osnovnu relaciju/pogled, Obrisati atribut iz osnovne relacije/pogleda, Izmjeniti definiciju pogleda), razumljivost pogleda i njegovih atributa (Preimenovati relaciju, pogled ili atribut), razumljivost i mogućnost interpretacije podataka (Izmjeniti domenu atributa) te vjerodostojnost i konzistentnost podataka (Dodati ograničenje integriteta, Obrisati ograničenje integriteta). U [1] autori su na temelju sustava EVE definirali MAVIE, sustav za sinkronizaciju pogleda u dinamičnijem i distribuiranom okruženju, baziran na tehnologijama mobilnih agenata. MAVIE sustav smanjuje vrijeme sinkronizacije zbog paralelnosti kod mobilnih agenata te izbjegava zasićenost mreže. U [2] autori su predložili algoritam za inkrementalno održavanje pogleda, pri čemu su koristili koncept pohranjivanja verzija za starije verzije tablica ažuriranih na izvoru podataka te im pridruživali broj transakcije. Time su pokušali riješiti problem anomalija i nekonzistentnih promjena nad pogledima. U [36] nalazi se dobar pregled istraživanja problema održavanja pogleda. Autori su radove koji se

bave istraživanjem problema održavanja pogleda grupirali po tri pristupa – osnovno održavanje pogleda, inkrementalno održavanje pogleda i samoodrživo održavanje pogleda.

**Tablica 2. Prikaz predloženih pristupa održavanja pogleda po istraživanjima**

	Osnovno održavanje	Inkrementalno održavanje	Samoodrživo održavanje	Odabir pogleda	Select-Project_Join pogled	Meta-podaci	Implementacija i jezik	Kvaliteta
Gupta	●	●			●		●	
Baralis				●				
Bouzeghoub	●							
Bellahsene		●						
Rajaraman							●	
Lakshmanan						●	●	
Cui		●			●			
Rundensteiner							●	
Zhang		●						
Rundensteiner		●						
Quix	●							●
Akaichi							●	
Almazayad		●				●		

#### 2.5.4. Analiza pregleda istraživanja

Problem evolucije sheme skladišta podataka može se promatrati i kao dvostruki problem, na razini skladišta podataka i na razini upravljanja osnovnim podacima (eng. *master data management, UMP*) ([49],[45]). Iz perspektive skladišta podataka, prati se svaki događaj koji je povezan s dimenzijama. S druge strane, kod UMP se prate osnovni podaci (dimenzije), a

kontekst im čine događaji. I u slučaju UMP-a, kao i kod skladišta podataka, javlja se problem evolucije sheme nakon promjena u izvorima podataka ili korisničkim zahtjevima. Međutim, proces evolucije ili verzioniranja sheme još uvijek zahtjeva puno resursa – vremena te mrežnih i ljudskih resursa. Potrebno je uravnotežiti zahtjeve za resursima i kvalitetu provođenja evolucije skladišta podataka. Možda najveći problem kod verzioniranja sheme jest očuvanje konzistentnosti sheme i integriteta podataka, kao i izvođenje povijesnih upita nad više verzija sheme istovremeno. U pregledu dosadašnjih istraživanja prezentirani su različiti pristupi rješavanju problema evolucije skladišta podataka, uključujući razne tehnike, algoritme, algebre, modele, prototipe, metodologije i okvire, međutim i dalje ne postoji šire prihvaćeno rješenje. Neki autori definiraju metodologije za evoluciju skladišta podataka, koje su vezane uz konkretnu implementaciju, međutim one su ograničene za upotrebu. Dalje, javlja se problem temporalnih upita. Kako bi se postigla savršena arhivska kvaliteta sustava i podataka, ključno je pohraniti podatke u onu verziju sheme u kojoj su se oni prvi put pojavili. Mnogi autori smatraju da kod verzioniranja sheme nije potrebno razlikovati vrijeme transakcije i vrijeme važenja. Međutim, treba uzeti u obzir i mogućnost istovremeno važećih verzija, gdje je potrebno razlikovati ova dva vremena, kako bi se upiti nad više verzija mogli izvršiti nezavisno od vremena kad su oni postavljeni. Dosadašnji temporalni upitni jezici, poput TSQL2 [64], omogućili su rad s temporalnim elementima i ograničenjima, međutim ne podupiru verzioniranje u potpunosti, odnosno ne podupiru među-verzijske upite uopće, ili ih podupiru djelomično. Iako je akademska zajednica napravila korake u pravcu postavljanja i upravljanja među-verzijskim upitima, tu još ima prostora za poboljšanje (prvenstveno vezano uz softversku podršku, temporalne upitne jezike i performanse upita), kao i za pronalazak opće prihvaćenog rješenja, te u potpunosti efikasnog komercijalnog rješenja. Najvažnije, dosadašnja istraživanja ne naglašavaju činjenicu da su zahtjevi skladišta podataka, u današnje doba, povećani u pogledu opsega i strukture podataka i meta-podataka (sve veći broj izvora podataka te više novih i različitih tipova podataka, poput npr. polustrukturiranog teksta), što dodatno zahtijeva osmišljanje novih pristupa i rješenja problema evolucije skladišta podataka.

### **3. OPIS ISTRAŽIVANJA**

U ovome poglavlju nalazi se opis metoda i postupaka primijenjenih u doktorskom istraživanju. Doktorsko istraživanje zasniva se na paradigmi dizajna znanstvenog istraživanja

(eng. Design Science Research, DSR) ([29],[44],[14],[28],[58]). DSR predstavlja važnu i danas široko prihvaćenu istraživačku paradigmu u području projektiranja informacijskih sustava (IS), iako još nije ostvarila optimalni učinak zbog nesporazuma u shvaćanju i primjeni DSR koncepata i metoda, kao i neodgovorenim pitanjima kako i u kojoj mjeri DSR doprinosi znanju i općenito teoriji [29][44][28]. U kontekstu informacijskih sustava DSR uključuje razvoj i izgradnju brojnih društveno-tehnoloških (IT) objekata (eng. *artifact*), poput sustava za podršku odlučivanju, alata za modeliranje, strategija za upravljanje podacima, metoda za evaluaciju IS, postupaka za izmjenu IS i sl., te ističe kako bi ova istraživanja trebala imati jasan doprinos u stvarnom poslovnom okruženju na način da rezultati istraživanja imaju utjecaja na primjenu u poslovanju (odnosno području projektiranja IS). U centru DSR paradigme nalazi se jasno identificiranje, definiranje i komuniciranje teorijskog ili praktičnog doprinosa postojećem ljudskome znanju, kao i učinkovita komunikacija istog. Pri tome imamo dvije struje – jednu koja veću važnost pridaje teoriji dizajna kao doprinosu istraživanja [28] i drugoj koja veći naglasak stavlja na sam IT objekt kao važan doprinos istraživanja [58]. Međutim, u posljednje vrijeme ojačao je stav da doprinos znanju može biti nematerijalizirana teorija (potpuna, djelomična ili nepotpuna) ili neka posebno zanimljiva empirijska generalizacija u formi novog, materijaliziranog dizajna IT objekta (npr. model, metoda, softver, utjelovljenje) [29]. Teorija i znanje mogu se razvijati i sazrijevati počevši od stvaranja novog IT objekta, a zatim se granice teorije i znanja šire i testiraju te sama teorija na ovaj način sazrijeva. Ovim stavom smo se vodili i prilikom izrade ovoga doktorskoga istraživanja - specifično metodologijom iz [58] koja sintezom raznih pristupa nudi koristan generalizirani model.

Ona uključuje sljedeće korake:

1) identifikacija problema

- a. u uvodnome dijelu (Poglavlje 1) identificiran je i predstavljen problem istraživanja
- b. u Poglavlju 2 predstavljen je pregled dosadašnjih istraživanja u ovome području, napravljena je analiza dosadašnjih rješenja te je opisana važnost rješavanja problema evolucije sheme SP u okviru stvarne prakse

2) definiranje ciljeva istraživanja

- a. u uvodnome dijelu (Poglavlje 1) definiran je osnovni cilj i opseg istraživanja te su predstavljene hipoteze istraživanja i izvorni znanstveni doprinosi istraživanja



### 3) dizajn i razvoj

- a. u Poglavlju 3 opisana je ideja istraživanja, predstavljen je dizajn arhitekture dvostrukog SP/UMP rješenja te je predstavljen i opisan dizajn konceptualnog i logičkog MDV modela za repozitorij meta-podataka (koji predstavlja proširenje i poboljšani dizajn postojećeg sistemskog kataloga)
- b. u Poglavlju 4 razvijen je i definiran skup promjena i operacija promjene te su razvijeni praktični primjeri za teorijsku validaciju rješenja
- c. Poglavlja 3 i 4 služe za prikaz apstraktnog i generaliziranog znanja, odnosno teorijskog doprinosa znanju, prikazanih kroz teorijsko rezoniranje - ovo predstavlja teoriju dizajna prema DSR [29]

### 4) demonstracija

- a. u Poglavlju 5 prikazan je opis praktičnog prototipa rješenja, kao i koraci primijenjeni u njegovom razvoju
- b. Prototip rješenja predstavlja materijalizirani dizajn IT objekta, odnosno praktični doprinos znanju prema DSR [29]

### 5) evaluacija

- a. u Poglavlju 5 prototip rješenja koji predstavlja „dokaz koncepta“ (eng. *proof of concept*) testiran je kroz provedbu skupa upita nad MDV repozitorijem i sistemskim katalogom, nakon provedbe prethodno definiranih promjena nad izvorima podataka te SDV/PMDV shemama, kao i kroz provedbu skupa upita za prikaz korisnosti i primjenjivosti rješenja

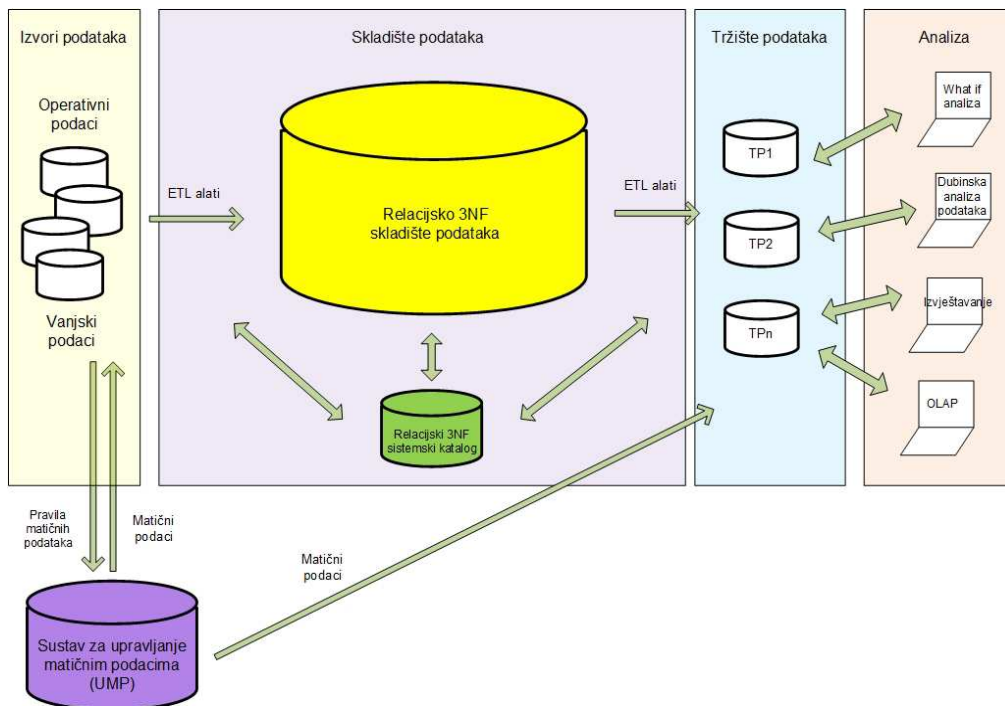
### 6) komunikacija

- a. u nekoliko radova prije objave ove doktorske disertacije predstavile su se neke ideje i doprinosi ovog istraživanja, a u planu je i objaviti još nekoliko članaka prema DSR metodi s cjelovitim doprinosima doktorske disertacije
- b. rasprava o postignutome dosada, važnosti teorijskih i praktičnih rezultata ovog istraživanja, kao i ograničenjima trenutnog rješenja nalazi se u Poglavlju 6, gdje se nalazi i zaključak te planovi za budući rad

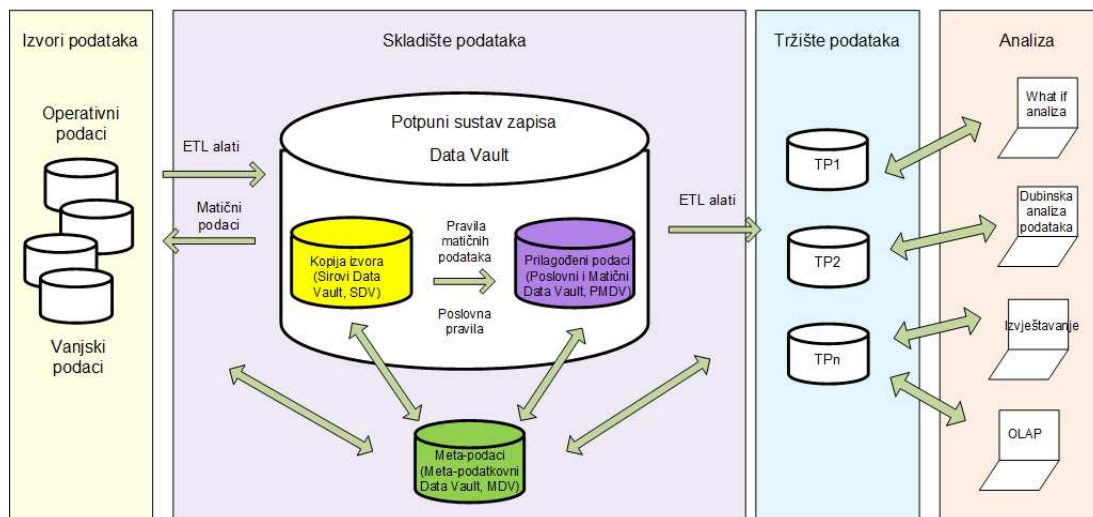
U nastavku slijedi detaljni opis predložene arhitekture, kao i MDV modela repozitorija meta-podataka.

### 3.1. Arhitektura SP/UMP

Unutar tipične troslojne SP arhitekture [27] možemo vidjeti da su SP i UMP nezavisni i fizički razdvojeni, iako oba ova sustava izvlače podatke iz istog skupa izvora te izvještavaju isti skup poslovnih korisnika (Slika 1).



Slika 1. Tipična troslojna arhitektura SP i UMP sustava



Slika 2. Arhitektura integriranog SP/UMP sustava

U svrhu smanjenja složenosti i zahtjevnosti procesa evolucije te opsega posla za SP/UMP arhitekta i dizajnera, naša integrirana arhitektura nalazi se na Slika 2. Centralni SP/UMP koji se sastoji od dva dijela – izvorišne kopije podataka (sirovih podataka) i usklađenih podataka (poslovnih matičnih podataka) – zasniva se na Data Vault (DV) metodi za modeliranje baza podataka [47],[39],[48], za razliku od tradicionalnih pristupa zasnovanih na relacijskom modelu. Samim time, ova dva dijela možemo promatrati kao sirovi DV (SDV) orijentiran na izvore podataka i kao poslovni matični DV (PMDV) orijentiran na korisničke zahtjeve. Integracija ova dva dijela odvija se preko proširenog sistemskog kataloga sustava za upravljanje bazom podataka, odnosno repozitorija meta-podataka, koji je također zasnovan na DV modelu te ćemo ga skraćeno nazvati MDV repozitorij. MDV repozitorij nam služi za integraciju ova dva sustava te za praćenje povijesti promjena meta-podataka i njihovih shema, kao i poslovnih pravila i transformacija te mapiranja između ova dva sustava. MDV repozitorij sadrži povijest meta-podataka o izvorima podataka (njihovim domenama i shemama), shemi i integraciji centralnog SP/UMP repozitorija te u budućnosti i shemama područnih SP (uključujući i korisničke zahtjeve) i shemi sigurnosti (kontrola i povijest korističkih pristupa podacima). Možemo reći da MDV repozitorij predstavlja SP o SP te će on biti ključni dio arhitekture u našem istraživanju (više u Poglavljima 3.2 i 3.3.).

Vratimo se zasad na predloženu arhitekturu - centralni SP/UMP repozitorij puni se preko ETL procesa podacima iz relacijskih izvora podataka te poslovnim pravilima. Podaci se prvo pune u SDV, u obliku u kojem su došli iz izvora podataka (čuva se neizmjenjena izvorišna kopija za učinkovitije očuvanje povijesti i provođenje postupka revizije), dok se poslovna pravila pohranjuju u MDV. Jednom kad podaci uđu u SDV, više se ne brišu i ne mijenjaju – što znači da se kopije originala trajno čuvaju u SDV. Na ovaj način izbjegava se gubitak informacija i stvara se osnova za proces revizije (*eng. audit*).

Na podatke u SDV se zatim primjenjuju poslovna pravila i transformacije iz MDV, s ciljem poslovnog „usklađivanja“ i integracije te tako nastaje PMDV. PMDV predstavlja nadograđeni SDV, pročišćen i prilagođen za daljnje prosljeđivanje podataka u područna SP. SDV i PMDV su na Slika 2 prikazani kao odvojeni sustavi te ćemo ih kao takve fizički i implementirati. SDV i PMDV strukture bit će međusobno povezane kroz „isti-kao“ vezu (*eng. same-as link*) [47],[48] unutar MDV repozitorija, kao i shema i pravila mapiranja. Međutim oni se fizički mogu

implementirati i kao jedinstvena baza podataka s međusobno nezavisnim, ali povezanim strukturama.

Također, zbog ove razdvojenosti SDV i PMDV dijela možemo razlikovati dvije vrste transformacija – povratne (lake) i nepovratne (teške)[40]. Povratne transformacije koristimo za punjenje iz izvora podataka u SDV (odnosno za kreiranje izvorišne kopije podataka) te se njihov učinak može preokrenuti natrag na početni sustav zapisa (osnova za reviziju). Nepovratne transformacije zasnivaju se na poslovnim pravilima i pravilima matičnih podataka te služe za punjenje materijaliziranih SPS-a, a njihov učinak je obično nepovratan.

U materijaliziranom SPS pohranjuju se matični podaci, kao i podaci za poslovnu analizu i izvještavanje. Podaci se pohranjuju prema korisničkim zahtjevima, a dimenzijski model se koristi za reprezentaciju (podaci su sumirani, agregirani i izračunati). Ovdje je tradicionalni SPS integriran s UMP, odnosno PMDV „hrani“ podacima UMP, a UMP zatim „hrani“ podacima SPS i izvore podataka. Krajnji korisnik preko odabranih alata za analizu i izvještavanje zatim pristupa SPS-ima te vrši potrebnu analizu nad podacima. Dodatno, „usklađeni“ matični podaci se vraćaju u izvore podataka (na Slici 2 povratna veza iz SP na izvore podataka), radi usklađivanja podataka unutar poslovne organizacije te krajnji korisnik sad preko svojih poslovnih aplikacija može pristupiti tim podacima. Ovdje se koristi u najvećoj mjeri prethodno opisani pristup centralnog UMP-a s jednom kopijom (Poglavlje 2.3). Matični podaci se iz PMDV prikupljaju i održavaju u jednoj centralnoj UMP lokaciji, a svi izvorišni sustavi i aplikacije koriste te podatke. Ovime se direktno utječe na kvalitetu podataka koji kasnije ponovno ulaze u sirovi SDV te opet prolaze kroz opisane slojeve arhitekture. Također, ovime se smanjuje količina podataka nad kojima je kasnije potrebno vršiti „teške“ transformacije te se na taj način ubrzava proces integracije podataka [40].

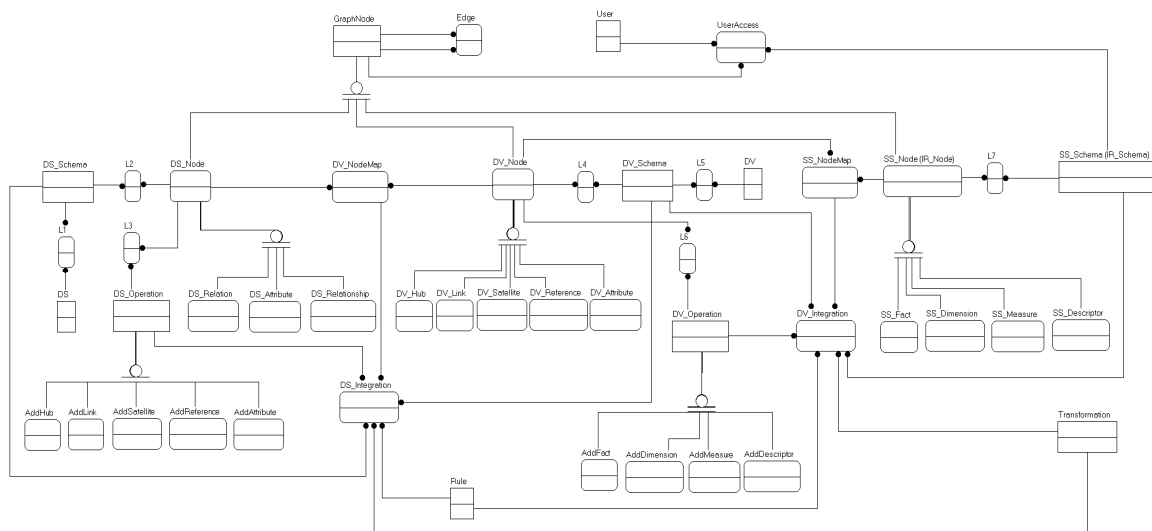
Čuvanjem i izvorišne kopije podataka i podataka prilagođenih poslovnim i matičnim pravilima (kroz DV reprezentaciju) omogućeno je praćenje podataka natrag do izvora podataka te po potrebi njihova rekonstrukcija, što je ključna ideja u ostvarivanju integriranog SP/UMP sustava i postizanju obuhvatnog upravljanja podacima (*eng. data governance*). Na taj način repozitorij podataka (SDV+PMDV) čuva izvorišne i matične podatke te povijest njihovih promjena, dok repozitorij metapodataka (MDV) čuva meta-podatke te povijest njihovih promjena.

Data Vault (DV) model ovdje koristimo zbog njegovih brojnih prednosti u dizajniranju SP, a najviše iz razloga što vjerujemo da relacijski model kao logički model jednostavno nije prikladan za učinkovitu i jednostavnu evoluciju SP, pogotovo iz razloga što ne razdvaja identitet od svojstava (kao što to čini DV model).

### 3.2. Konceptualni MDV

Na Slici 3 (a i u Pravitku 2) nalazi se konceptualni MDV model. Model (kao i svi ostali modeli u radu) je izrađen prema IDEF1X metodi [33], korištenjem CA ERwin Data Modeler 9.5 alata za modeliranje [21] (a neki pomoćni modeli su kreirani u SQL Data Modeler alatu [66]). Također, svi modeli i praktični primjeri izrađeni su na engleskom jeziku, radi jednostavnijeg usklađivanja s postojećim sustavima za upravljanje bazama podataka i njihovim sistemskim katalozima.

Konceptualni MDV model sastoji se od 4 područja – izvori podataka (DS\_Schema), centralni repozitorij podataka (DV\_Schema), područno skladište podataka (podatkovna kocka – SS\_Schema) i sigurnost i kontrola pristupa (User i UserAccess). Možemo vidjeti da su sva 4 dijela međusobno povezana kroz koncepte integracije shema (DS\_Integration i DV\_Integration), gdje se prate povezivanja između struktura izvorišnih shema i struktura SDV/PMDV shema, kao i povezivanja između PMDV/UMP/SP shema. Više će biti prikazano u sljedećem poglavlju, kod opisa logičkog MDV modela.



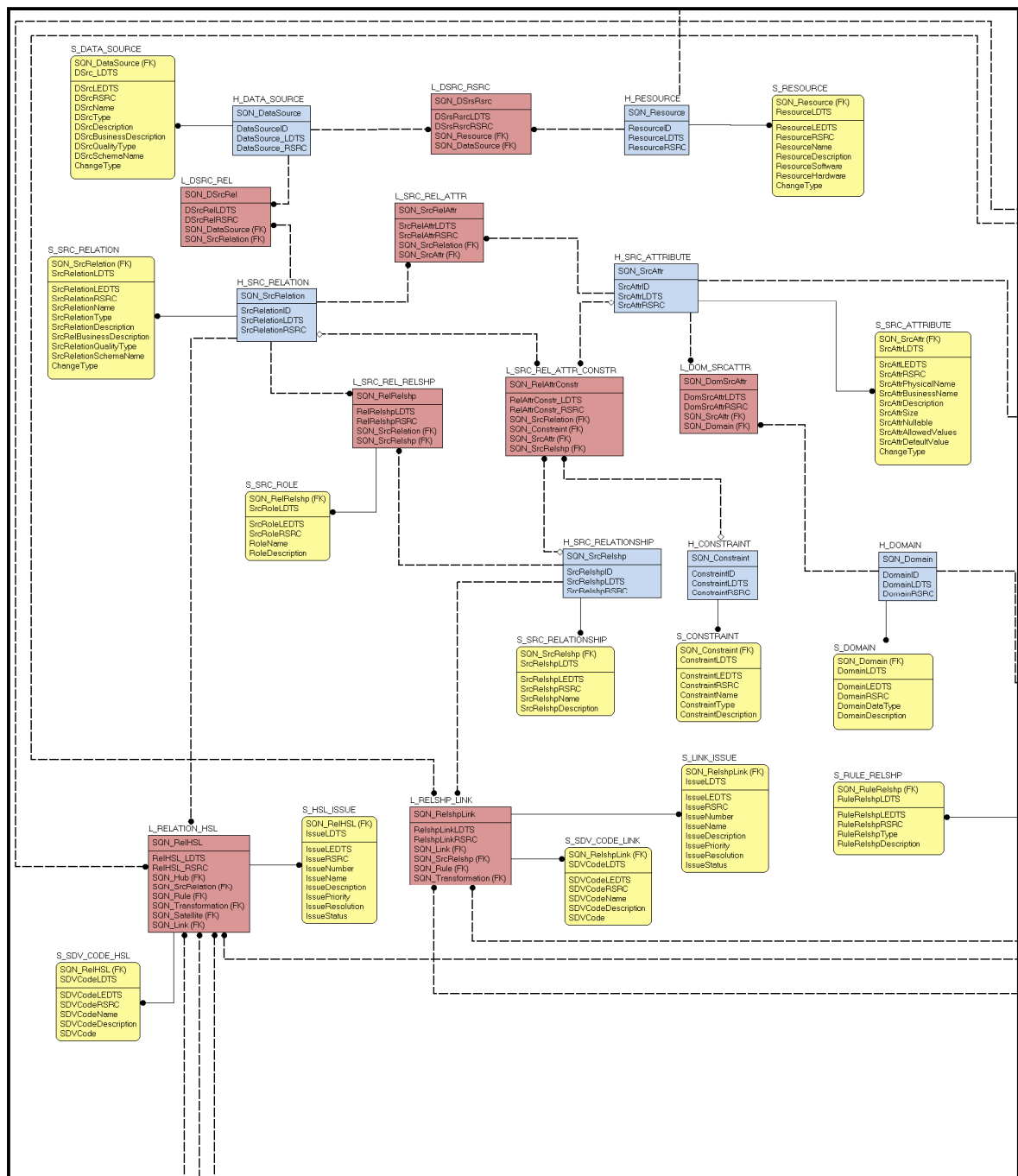
Slika 3. Konceptualni MDV model

### **3.3. Logički MDV**

Kako bi lakše razumjeli MDV model, kratko ćemo spomenuti osnovne koncepte DV metode, koji su korišteni za izradu MDV modela. Osnovna 4 koncepta DV metode su hub, link, satelit i referentna tablica. Hubovi predstavljaju identitete, odnosno poslovne ključeve te su radi čitljivosti modela prikazani u plavoj boji. Linkovi predstavljaju veze između dva ili više hubova, odnosno reprezentacije M:M veze iz relacijskog modela, te su prikazani u crvenoj boji. Dodatno možemo razlikovati takozvane isti-kao linkove koji nam pomažu definirati semantički jednake podatkovne strukture. Sateliti predstavljaju historizirane opisne atribute, odnosno sve neključne atribute huba ili linka, te su prikazani u žutoj boji. Svaki hub ili link može imati više satelita, a oni se mogu dijeliti prema izvoru iz kojeg dolaze, tipu podataka koji sadrže ili brzini promjene. I na kraju, referentna tablica najčešće predstavlja referentne podatke (šifarnike, unutarne ili vanjske standardizirane podatke te klasifikacije ili kategorizacije).

Cjeloviti logički MDV model repozitorija metapodataka nalazi se u Pravitku 1. On trenutno obuhvaća 4 područja metapodataka – izvore podataka (Slika 4), centralni repozitorij (Slika 5), područna skladišta podataka (Slika 6) te kontrolu pristupa podacima (Slika 7). Treba istaknuti da je, radi korištenja DV metode, MDV model iznimno fleksibilan i proširiv te se može stalno obogaćivati novim i korisnim vrstama metapodataka.

Na Slici 4. možemo vidjeti dio MDV modela koji predstavlja metapodatke o izvorima podataka. Ovaj dio služi za pohranu općih podataka o resursima i fizičkim izvorima podataka (hubovi RESOURCE i DATA\_SOURCE te njihovi pripadajući sateliti) te o izvorišnim strukturama poput tablica, stupaca, veza i ograničenja (hubovi SRC\_RELATION, SRC\_ATTRIBUTE, SRC\_RELATIONSHIP, CONSTRAINT sa svojim satelitima), vezama između njih u izvoru podataka (linkovi DSRC\_RSRC, DSRC\_REL, SRC\_REL\_ATTR, SRC\_REL\_RELSHP, DOM\_SRCATTR, SRC\_REL\_ATTR\_CONSTR), kao i o njihovim mapiranjima u SDV dio MDV modela (linkovi RELATION\_HSL za vezu između izvorišne tablice i huba/linka/satelita u SDV ili RELSHP\_LINK za mapiranje izvorišne veze u SDV vezu).

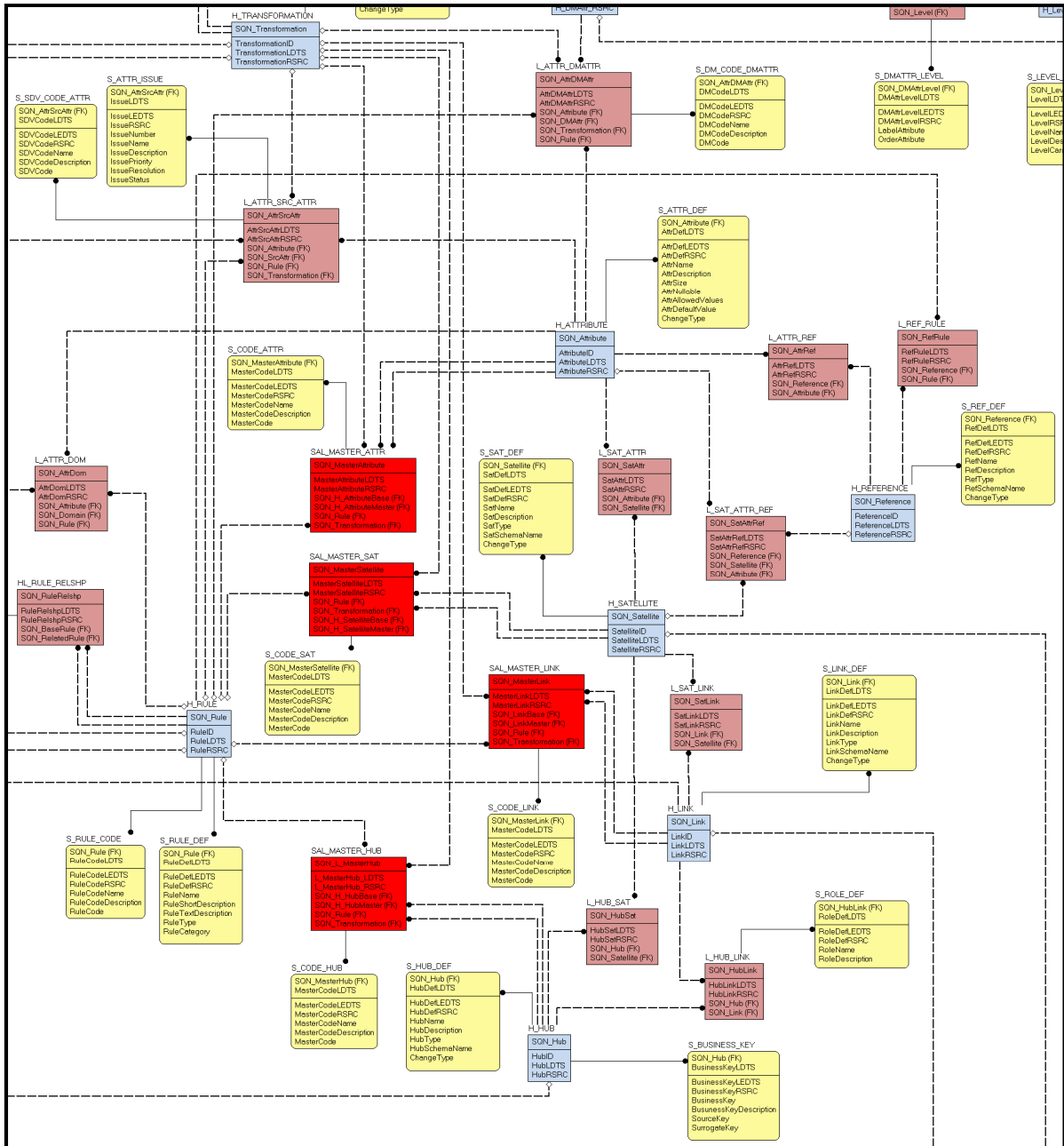


Slika 4. Dio MDV modela koji se odnosi na izvore podataka

Na Slici 5. možemo vidjeti dio MDV modela koji predstavlja meta-podatke o centralnom repozitoriju podataka (SDV+PMDV). Ovaj dio pohranjuje podatke o strukturi centralnog repozitorija podataka SP/UMP, odnosno podatke o hubovima i linkovima te njihovim pripadajućim satelitima i atributima. Ovdje se pohranjuju informacije o mapiranjima struktura iz izvora podataka u SDV dio centralnog repozitorija (npr. SRC\_RELATION hub koji predstavlja popis izvorišnih tablica se preko RELATION\_HSL linka povezuje s HUB hubom koji predstavlja

popis hubova iz centralnog repozitorija), kao i mapiranja iz SDV u PMDV (npr. hub HUB koji sadrži popis hubova iz podatkovnog repozitorija ima dvostruku vezu na MASTER\_HUB isti-kao link čime se SDV hub povezuje s PMDV hubom) što je zapravo ključ SP/UMP integracije (nešto više o tome u Poglavlju 3.4) te mapiranja s podatkovnom kockom (npr. hub ATTRIBUTE iz PMDV dijela je vezan preko linka ATTR\_DMATTR s hubom DM\_ATTRIBUTE koji predstavlja popis atributa podatkovne kocke, bilo mjera ili dimenzijskih razina). Također, svim ovim mapiranjima su pridruženi podaci o poslovnim pravilima i transformacijama koji se koriste (hubovi RULE i TRANSFORMATION s pripadajućim satelitima koji ih dodatno opisuju). Dodatno, u ovom dijelu meta-modela pohranjuju se podaci o domenama podataka i referentnim tablicama. Možemo primjetiti da u našem MDV modelu nema referentnih tablica – to je iz razloga što su referentni podaci unaprijed konsolidirani i nad njima su već primjenjena unutarnja poslovna pravila ili vanjske standardizacije pa se oni pohranjuju na podatkovnoj razini, odnosno u centralnom repozitoriju podataka, a u MDV modelu će koncept H\_REFERENCE jednostavno pohranjivati njihove meta-podatke.





Slika 5. Dio MDV modela koji se odnosi na centralni repozitorij podataka (SDV+PMDV)

Na Slici 6. možemo vidjeti dio MDV modela koji predstavlja meta-podatke o područnim skladištima podataka (PSP), odnosno podatkovnim kockama. Možemo vidjeti da se čuvaju podaci o činjenicama i dimenzijama te njihovim pripadajućim atributima (mjerama i elementima dimenzija), kao i dimenzionalnim razinama i hijerarhijama te podatkovnim kockama. Podatkovna kocka prikazana je kao link svih relevantnih hubova te jedan zapis u podatkovnoj kocki predstavlja jedan korisnički zahtjev. Dodatno, ovdje se pohranjuju i podaci



### **3.4. Logički MDV – dio integracije SP/UMP**

Na Slici 8 nalazi se manji dio MDV modela koji prikazuje na koji način će se integrirati SP i UMP sustavi, odnosno SDV i PMDV područja podataka. Možemo vidjeti da se integracija vrši preko isti-kao linkova (skraćeno SAL; same-as link) nad četiri glavna huba (H\_HUB, H\_LINK, H\_SATELLITE, and H\_ATTRIBUTE) koji čuvaju podatke o tri osnovna DV koncepta u podatkovnom repozitoriju (s dodatkom H\_ATTRIBUTE huba koji pohranjuje podatke o atributima satelita, odnosno meta-podatke o stupcima podatkovnog repozitorija). Uzmimo za primjer H\_HUB hub i njegove satelite S\_BUSINESS\_KEY i S\_HUB\_DEF. Ovaj hub čuva meta-podatke o svim hubovima iz centralnog repozitorija (i iz SDV i iz PMDV dijela), poput naziva hubova, naziva njihovih poslovnih ključeva, tipova hubova i njihovih općenitih opisa te datuma punjenja i izvora zapisa. Međutim, u ovome trenu SP i UMP hubovi nisu još integrirani – jer meta-hub H\_HUB sa svojim pripadajućim meta-satelitima sadrži samo popis i prošireni opis svih hubova iz podatkovnog repozitorija. Kako bismo povezali hubove iz SDV dijela s hubovima iz PMDV dijela podatkovnog repozitorija, koristimo isti-kao link SAL\_MASTER\_HUB koji povezuje određeni hub iz popisa svih hubova u H\_HUB s drugim hubom iz istog tog popisa, pri čemu se koristi veza tipa podređeni-nadređeni (odnosno SDV hub je baza, a PMDV hub je matični podatak). Dodatno, na SAL\_MASTER\_HUB vežu se i hubovi H\_RULE i H\_TRANSFORMATION, meta-koncepti koji pohranjuju podatke o poslovnim pravilima koja se primjenjuju na podatke prilikom pretvorbe iz SDV u PMDV te o (ETL) transformacijama koje su potrebne za provedbu poslovnih pravila. Na ovaj način, osim što možemo pratiti povijest integracije SP/UMP, odnosno povijest SDV i PMDV struktura (SAL\_MASTER\_HUB sadrži load\_date i record\_source atribut za svaku pojedinu integraciju), možemo pratiti i povijest svih poslovnih pravila i transformacija primjenjenih na pojedinu integraciju. SAL\_MASTER\_HUB ima i pripadajući satelit, S\_CODE\_HUB, koji sadrži opisne podatke o ETL kodu korištenom za pojedinu SDV/PMDV integraciju. Na isti način stvaramo matične podatke za preostala 3 huba u MDV modelu (SAL\_MASTER\_LINK za H\_LINK, SAL\_MASTER\_SAT za H\_SATELLITE and SAL\_MASTER\_ATTR za H\_ATTRIBUTE) te na temelju SDV izvorišne kopije podataka kreiramo kompletnu i historiziranu PMDV „zlatnu kopiju“ poslovnih i matičnih podataka.



## 4. TEORIJSKA VALIDACIJA ISTRAŽIVANJA

U ovome poglavlju teorijski i formalno ćemo validirati MDV model i predloženo rješenje za evoluciju SP i UMP. Također, bit će teorijski validirane i dokazane hipoteze H2 i H3.

### 4.1. Definicija DV modela

U nastavku slijedi formalna definicija osnovnih struktura i koncepata poopćenog DV modela prikazanog kao graf. Definicija je primjenjiva i na podatkovnoj i na meta-podatkovnoj razini modeliranja DV metodom.

**Definicija 1 (Hub).** Hub H je pravi podskup konačnog skupa čvorova C. Za svaki Hub čvor postoji samo jedan put do njegovih Satelit čvorova.

**Definicija 2 (Link).** Link L je pravi podskup konačnog skupa čvorova C. Svaki link čvor je jako povezan s hub čvorom – svaki link čvor ima najmanje dva brida prema hub čvorovima. Za svaki link čvor postoji samo jedan put do njegovih satelit čvorova.

**Definicija 3 (Satelit).** Satelit S je pravi podskup konačnog skupa čvorova C. Do svakog satelit čvora postoji samo jedan put od njegovog hub ili link čvora.

**Definicija 4 (Atribut).** Atribut A je pravi podskup konačnog skupa čvorova C. Do svakog atribut čvora postoji samo jedan put od njegovog satelit čvora.

**Definicija 5 (Referenca).** Referenca R je pravi podskup konačnog skupa čvorova C.

**Definicija 6 (Domena).** Domena D je pravi podskup konačnog skupa čvorova C.

**Definicija 7 (DV shema).** DV shema  $SH_{ESP}$  je 6-torka [H, R, L, S, A, D], gdje je H konačan skup hubova, R konačan skup referenci, L konačan skup linkova, S konačan skup satelita, A konačan skup atributa i D konačan skup domena.

Skup koncepata DV modela iz [47],[48] proširen je s Atributom i Domenom, radi detaljnijeg praćenja promjena u podacima i shemama.

Imajući u vidu formalnu definiciju općih DV koncepata, naš logički MDV meta-model iz Privitka 1 možemo definirati kao:

- aciklični usmjereni multigraf (C,B) s plitko povezanim čvorovima, gdje je C konačan skup čvorova (vrhova), a B konačan skup bridova. Čvor predstavlja koncepte u DV modelu (hub, link, satelit, referenca, atribut). Brid predstavlja vezu između dva koncepta u DV modelu.

MDV graf ima sljedeća svojstva:

- usmjeren je – predstavlja konačan skup uređenih parova čvorova, a bridovi imaju pridružen smjer (usmjereni bridovi - lukovi)
- bez petlji – jednostavan je, nema petlje/višestruke bridove (odnosno bridove koji počinju i završavaju u istom čvoru)
- multigraf je - graf može imati višestruke (paralelne) bridove, odnosno dva čvora mogu biti povezana s više od jednog brida (bridovi formiraju multiskup)
- acikličan je – nema usmjerene cikluse
- ravnomjerno je plitak – sadrži visinu 1 ili 2 (Hub->Satelit ili Hub->Link->Satelit)
- čvorovi su imenovani – postoji konačan skup uloga U za svaki čvor,  $U \in \{\text{Hub, Link, Satelit, Atribut, Referenca}\}$
- bridovi su imenovani – postoji konačan skup uloga U za svaki brid,  $U \in \{\text{HL brid, HS brid, LS brid}\}$
- Za svaki Hub i Link čvor postoji samo jedan put do njihovih Satelit čvorova (listova)
- Link čvorovi su snažno povezani s Hub čvorovima – postoje najmanje dva brida, a broj puteva od Hub čvora do linkovog Satelit čvora je jednak broju linkovih bridova prema hubovima

## **4.2. Dokaz hipoteze H2**

Hipoteza H2 kaže da se MDV model smatra trajnim iz dva razloga (Poglavlje 1.2):

- a) Sve promjene nad podacima implementirane su u model samo kao dodavanja – nema gubitka podataka, i
- b) Sve promjene u shemi modela implementirane su kao jednostavna proširenja modela – nema gubitka sheme.

Da bismo dokazali tvrdnje a) i b) izrađena je formalna algebra za održavanje podataka i sheme SP i UMP te je definiran formalni konačni skup osnovnih promjena nad shemom SP i UMP.

#### 4.2.1. *Definicija skupa promjena i operacija promjene*

Krenut ćemo od šest operacija za promjenu DV sheme:

- Dodaj<sub>C</sub> - dodavanje novog čvora
- Briši<sub>C</sub> – brisanje čvora
- Izmjeni<sub>C</sub> – izmjena čvora
- Dodaj<sub>B</sub> - dodavanje novog brida
- Briši<sub>B</sub> – brisanje brida
- Izmjeni<sub>B</sub> – izmjena brida

Pokazat ćemo da su dovoljne osnovne dvije operacije za promjenu sheme (Dodaj<sub>C</sub> i Dodaj<sub>B</sub>)- sve ostale operacije nad DV shemom su kombinacija ove dvije osnovne operacije.

Neka je  $Sh(C, B)$  DV shema,  $A$  skup atributa,  $S$  skup satelita,  $H$  skup hubova,  $L$  skup linkova,  $R$  skup referenci i  $I(A_i)$  skup instanci pridruženih atributu  $A_i$ , gdje vrijedi  $A_i \rightarrow I(A_i)$ .

## Dodaj<sub>c</sub> - dodavanje novog čvora

### Dodavanje novog čvora – atributa:

Novi (Sh, Dodaj<sub>c</sub>(A)) := (C ∪ {A}, B ∪ {Ŝ<sub>A</sub>})

- razina podataka u MDV (A → I(A))

### Dodavanje novog čvora – satelita:

Novi (Sh, Dodaj<sub>c</sub>(S)) := (C ∪ {S}, B ∪ {Ŝ<sub>A<sub>1</sub></sub>, Ŝ<sub>A<sub>2</sub></sub>, ..., Ŝ<sub>A<sub>k</sub></sub>} ∪ {HL̂<sub>S</sub>}), gdje HLE {H, L}

- razina podataka u MDV (A → I(A))

### Dodavanje novog čvora – linka:

Novi (Sh, Dodaj<sub>c</sub>(S)) := (C ∪ {L} ∪ {S} ∪ {A}, B ∪ {Ŝ<sub>A<sub>1</sub></sub>, Ŝ<sub>A<sub>2</sub></sub>, ..., Ŝ<sub>A<sub>k</sub></sub>} ∪ {L̂<sub>S<sub>1</sub></sub>, L̂<sub>S<sub>2</sub></sub>, ..., L̂<sub>S<sub>k</sub></sub>} ∪ {L̂<sub>H<sub>1</sub></sub>, L̂<sub>H<sub>2</sub></sub>, ..., L̂<sub>H<sub>k</sub></sub>})

- razina podataka u MDV (A → I(A))

### Dodavanje novog čvora – huba:

Novi (Sh, Dodaj<sub>c</sub>(H)) := (C ∪ {H} ∪ {L} ∪ {S} ∪ {A}, B ∪ {Ĥ<sub>S<sub>1</sub></sub>, Ĥ<sub>S<sub>2</sub></sub>, ..., Ĥ<sub>S<sub>k</sub></sub>} ∪ {Ŝ<sub>A<sub>1</sub></sub>, Ŝ<sub>A<sub>2</sub></sub>, ..., Ŝ<sub>A<sub>k</sub></sub>} ∪ {Ĥ<sub>L<sub>1</sub></sub>, Ĥ<sub>L<sub>2</sub></sub>, ..., Ĥ<sub>L<sub>k</sub></sub>} ∪ {L̂<sub>S<sub>1</sub></sub>, L̂<sub>S<sub>2</sub></sub>, ..., L̂<sub>S<sub>k</sub></sub>}), gdje je {Ŝ<sub>A<sub>1</sub></sub>, Ŝ<sub>A<sub>2</sub></sub>, ..., Ŝ<sub>A<sub>k</sub></sub>} ∈ HL

- razina podataka u MDV (A → I(A))

### Dodavanje novog čvora – reference:

Novi (Sh, Dodaj<sub>c</sub>(R)) := (C ∪ {R} ∪ {A}, B ∪ {R̂<sub>A<sub>1</sub></sub>, R̂<sub>A<sub>2</sub></sub>, ..., R̂<sub>A<sub>k</sub></sub>} ∪ {Ŝ<sub>A<sub>1</sub></sub>, Ŝ<sub>A<sub>2</sub></sub>, ..., Ŝ<sub>A<sub>k</sub></sub>})

- razina podataka u MDV (A → I(A))



## Brišic – brisanje čvora

### Brisanje čvora – atributa:

Novi (Sh, Brišic(A)) := (C, B) := A -> I(A) - nema promjene u SDV/PMDV, razina podataka u MDV

### Brisanje čvora – satelita:

Novi (Sh, Brišic(S)) := (C, B) := A -> I(A) - nema promjene u SDV/PMDV, razina podataka u MDV

### Brisanje čvora – linka:

Novi (Sh, Brišic(L)) := (C, B) := A -> I(A) - nema promjene u SDV/PMDV, razina podataka u MDV

### Brisanje čvora – huba:

Novi (Sh, Brišic(H)) := (C, B) := A -> I(A) - nema promjene u SDV/PMDV, razina podataka u MDV

### Brisanje čvora – reference:

Novi (Sh, Brišic(R)) := (C, B) := A -> I(A) - nema promjene u SDV/PMDV, razina podataka u MDV

## Izmjenic – izmjena čvora

### Preimenovanje čvora:

Novi (Sh, Izmjenic(C)) := (C, B) := A -> I(A) - nema promjene u SDV/PMDV, razina podataka u MDV

Promjena tipa podatka atributa:

Novi (Sh, Izmijeni<sub>C</sub>(A)) := Novi (Sh, Dodaj<sub>C</sub>(A)) := (C U {A}, B U { $\widehat{S \ A}$ }) - nema promjene u SDV/PMDV, razina podataka u MDV (A -> I(A))

**Dodaj<sub>B</sub> - dodavanje novog brida**

Novi (Sh, Dodaj<sub>B</sub>(b)) := (B U {b}) , gdje je  $b \in \{\widehat{SA}, \widehat{SH}, \widehat{SL}, \widehat{HL}\}$  - razina podataka u MDV (A -> I(A))

**Briši<sub>B</sub> - brisanje brida**

Novi (Sh, Briši<sub>B</sub>(b)) := (C, B) := A -> I(A) - nema promjene u SDV/PMDV, razina podataka u MDV

**Izmjeni<sub>B</sub> – izmjena brida**

Novi (Sh, Izmjeni<sub>B</sub>(b)) := Novi (Sh, Briši<sub>B</sub>(b)) := (C, B) := A -> I(A) := Novi (Sh, Dodaj<sub>B</sub>(b)) := (B U {b}) , gdje je  $b \in \{\widehat{SA}, \widehat{SH}, \widehat{SL}, \widehat{HL}\}$  - nema promjene u SDV/PMDV, razina podataka u MDV

Pokazali smo da su uistinu samo dvije operacije (Dodaj<sub>C</sub> i Dodaj<sub>B</sub>) dovoljne za promjenu DV sheme, a sve ostale operacije nad DV shemom su kombinacija ove dvije osnovne operacije. Kako bismo to dalje verificirali, potrebno je definirati konačan skup promjena nad DV shemom te utvrditi njihov učinak na predstavljenu SDV/PMDV i MDV shemu.

U kontekstu promjena promatramo promjene u izvorima podataka. Ove promjene direktno utječu na centralni podatkovni repozitorij, odnosno na SDV i PMDV shemu, dok promjene u korisničkim zahtjevima češće utječu na SPS shemu, odnosno podatkovnu kocku (te su stoga te

promjene van opsega ovoga istraživanja). Iz prethodno definiranih operacija jasno je da možemo razlikovati i promjene koje utječu na sadržaj grafa (na instancu čvora ili brida) i promjene koje utječu na strukturu grafa (na čvor ili brid), ovisno o razini s koje promatamo graf (razina podataka koja predstavlja instance i vrijednosti ili razina sheme koja predstavlja osnovne koncepte DV modela). U opsegu ovoga istraživanja nalaze se promjene na razini sheme, odnosno strukturalne promjene nad grafom, dok se promjene na razini podataka jednostavno implementiraju kroz dodavanje novih instanci u čvor ili brid grafa (odnosno dodavanje novih zapisa u bazu) te nisu u opsegu ovoga istraživanja.

Definirajmo prvo skup promjena u izvorima podataka te njihov učinak na SDV/PMDV i MDV shemu. Definiran je konačni skup promjena u postojećoj shemi izvora podataka  $I = \{D_R, I_R, B_R, D_A, I_{An}, I_{At}, B_A, D_V, B_V, I_{RV}, I_{VR}, I_{AR}, I_{RA}\}$ :

- Dodavanje nove relacije,  $D_R$
- Preimenovanje relacije,  $I_R$
- Brisanje relacije,  $B_R$
- Dodavanje atributa u relaciju,  $D_A$
- Izmjena postojećeg atributa u relaciji:
  - preimenovanje atributa,  $I_{An}$
  - promjena tipa podatka atributa,  $I_{At}$
- Brisanje atributa iz relacije,  $B_A$
- Dodavanje veze između relacija,  $D_V$
- Brisanje veze između relacija,  $B_V$
- Pretvorba relacije u vezu,  $I_{RV}$
- Pretvorba veze u relaciju,  $I_{VR}$
- Pretvorba atributa u relaciju,  $I_{AR}$
- Pretvorba relacije u atribut,  $I_{RA}$

Operaciju dodavanja novih izvora podataka promatramo kao operaciju spajanja shema izvora podataka (kao što se to većinom smatra u literaturi, [20]-[72]) te postavljamo da je konačni skup promjena kod dodavanja novih izvora podataka  $D_i$  pravi podskup od konačnog skupa  $I$ ,  $D_i \subseteq I$  te uključuje iste promjene kao nad postojećom shemom izvora podataka.

**Tablica 3. Učinak promjena u shemi izvora podataka na SDV/PMDV i MDV graf**

Promjena u shemi izvora	Efekt promjene u SDV	Efekt promjene u PMDV	Efekt promjene u MDV
Dodavanje nove relacije, $D_R$	<ul style="list-style-type: none"> <li>- proširenje SDV grafa – razina sheme/promjene nad strukturom grafa</li> <li>- dodavanje novih hub, link i satelit čvorova i odgovarajućih bridova u SDV graf - Dodaj<sub>C</sub> i Dodaj<sub>B</sub></li> </ul>	<ul style="list-style-type: none"> <li>- proširenje PMDV grafa - razina sheme/promjene nad strukturom grafa</li> <li>- dodavanje novih (master) hub, link i satelit čvorova i bridova u PMDV graf - Dodaj<sub>C</sub> i Dodaj<sub>B</sub></li> <li>- dodavanje novog linka između SDV hub čvora i PMDV master hub čvora - Dodaj<sub>C</sub> i Dodaj<sub>B</sub></li> </ul>	<ul style="list-style-type: none"> <li>- razina podataka/promjene nad sadržajem grafa – nema promjena na MDV shemi</li> <li>- dodavanje novog zapisa u odgovarajuće hubove (hub, link, satelit, referenca, atribut, pravilo, ...), linkove i satelite MDV modela</li> </ul>
Preimenovanje relacije, $I_R$	<ul style="list-style-type: none"> <li>- razina podataka/promjene nad sadržajem grafa – nema promjena na SDV shemi</li> </ul>	<ul style="list-style-type: none"> <li>- razina podataka/promjene nad sadržajem grafa – nema promjena na PMDV shemi</li> </ul>	<ul style="list-style-type: none"> <li>- razina podataka/promjene nad sadržajem grafa – nema promjena na MDV shemi</li> <li>- dodavanje novog zapisa u odgovarajuće MDV satellite (kod, hub definicija)</li> </ul>
Brisanje relacije, $B_R$	<ul style="list-style-type: none"> <li>- proširenje SDV grafa – razina sheme/promjene nad strukturom grafa</li> <li>- dodavanje novog satelit čvora (valjanost) i HS brida u SDV graf - Dodaj<sub>C</sub> i Dodaj<sub>B</sub></li> </ul>	<ul style="list-style-type: none"> <li>- proširenje PMDV grafa – razina sheme/promjene nad strukturom grafa</li> <li>- dodavanje novog (master) satelit čvora (valjanost) i HS brida u PMDV graf - Dodaj<sub>C</sub> i Dodaj<sub>B</sub></li> </ul>	<ul style="list-style-type: none"> <li>- razina podataka/promjene nad sadržajem grafa – nema promjena na MDV shemi</li> <li>- dodavanje novog zapisa u odgovarajuće MDV satellite (hub/link/satelit definicija, kod)</li> </ul>
Dodavanje atributa u relaciju, $D_A$	<ul style="list-style-type: none"> <li>- proširenje SDV grafa – razina sheme/promjene nad strukturom grafa</li> <li>- dodavanje novog satelit čvora i HS ili LS brida u SDV graf -- Dodaj<sub>C</sub> i Dodaj<sub>B</sub></li> </ul>	<ul style="list-style-type: none"> <li>- proširenje PMDV grafa – razina sheme/promjene nad strukturom grafa</li> <li>- dodavanje novog (master) satelit čvora i HS ili LS brida u PMDV graf -- Dodaj<sub>C</sub> i Dodaj<sub>B</sub></li> </ul>	<ul style="list-style-type: none"> <li>- razina podataka/promjene nad sadržajem grafa – nema promjena na MDV shemi</li> <li>- dodavanje novog zapisa u odgovarajuće MDV hubove, linkove i satelite</li> </ul>

Preimenovanje atributa u relaciji, I <sub>An</sub>	- razina podataka/promjene nad sadržajem grafa – nema promjena na SDV shemi	- razina podataka/promjene nad sadržajem grafa – nema promjena na PMDV shemi	- razina podataka/promjene nad sadržajem grafa – nema promjena na MDV shemi - dodavanje novog zapisa u odgovarajuće MDV satellite (kod, izv_atribut/satelit/atribut definicija)
Promjena tipa podatka atributa u relaciji, I <sub>At</sub>	- proširenje SDV grafa – razina sheme/promjene nad strukturom grafa - dodavanje novog satelit čvora i satelit čvora (valjanost) te odgovarajućih bridova SDV graf - Dodaj <sub>C</sub> i Dodaj <sub>B</sub>	- proširenje PMDV grafa – razina sheme/promjene nad strukturom grafa - dodavanje novog satelit čvora i satelit čvora (valjanost) te odgovarajućih bridova PMDV graf - Dodaj <sub>C</sub> i Dodaj <sub>B</sub>	- razina podataka/promjene nad sadržajem grafa – nema promjena na MDV shemi - dodavanje novog zapisa u odgovarajući MDV satellit (atribut definicija) i link (između atribut-domena)
Brisanje atributa iz relacije, B <sub>A</sub>	- proširenje SDV grafa – razina sheme/promjene nad strukturom grafa - dodavanje novog satelit čvora (valjanost) i HS brida u SDV graf - Dodaj <sub>C</sub> i Dodaj <sub>B</sub>	- proširenje PMDV grafa – razina sheme/promjene nad strukturom grafa - dodavanje novog satelit čvora (valjanost) i HS brida u PMDV graf - Dodaj <sub>C</sub> i Dodaj <sub>B</sub>	- razina podataka/promjene nad sadržajem grafa – nema promjena na MDV shemi - dodavanje novog zapisa u odgovarajuće MDV satellite (satelit/atribut definicija, kod)
Dodavanje veze između relacija, D <sub>V</sub>	- proširenje SDV grafa – razina sheme/promjene nad strukturom grafa - dodavanje novog čvora i bridova u SDV graf - Dodaj <sub>C</sub> i Dodaj <sub>B</sub>	- proširenje PMDV grafa – razina sheme/promjene nad strukturom grafa - dodavanje novog čvora i bridova u PMDV graf - Dodaj <sub>C</sub> i Dodaj <sub>B</sub>	- razina podataka/promjene nad sadržajem grafa – nema promjena na MDV shemi - dodavanje novog zapisa u odgovarajuće MDV hubove, linkove i satelite (link i uloga hub, link uloga-link-hub, definicija sateliti)
Brisanje veze između relacija, B <sub>V</sub>	- proširenje SDV grafa – razina sheme/promjene nad strukturom grafa - dodavanje novog satelit čvora (valjanost) i LS brida u SDV graf - Dodaj <sub>C</sub> i Dodaj <sub>B</sub>	- proširenje PMDV grafa – razina sheme/promjene nad strukturom grafa - dodavanje novog satelit čvora (valjanost) i LS brida u PMDV graf - Dodaj <sub>C</sub> i Dodaj <sub>B</sub>	- razina podataka/promjene nad sadržajem grafa – nema promjena na MDV shemi - dodavanje novog zapisa u odgovarajuće MDV satellite (link definicija, kod)

Promjena relacije u vezu, I <sub>RV</sub>	<ul style="list-style-type: none"> <li>- proširenje SDV grafa – razina sheme/promjene nad strukturom grafa</li> <li>- dodavanje novog satelit čvora (valjanost) i HS brida u SDV graf - Dodaj<sub>C</sub> i Dodaj<sub>B</sub></li> <li>- dodavanje novog link i satelit čvora i LS bridova u SDV graf - Dodaj<sub>C</sub> i Dodaj<sub>B</sub></li> </ul>	<ul style="list-style-type: none"> <li>- proširenje PMDV grafa – razina sheme/promjene nad strukturom grafa</li> <li>- dodavanje novog satelit čvora (valjanost) i HS brida u PMDV graf - Dodaj<sub>C</sub> i Dodaj<sub>B</sub></li> <li>- dodavanje novog link i satelit čvora i LS bridova u PMDV graf - Dodaj<sub>C</sub> i Dodaj<sub>B</sub></li> </ul>	<ul style="list-style-type: none"> <li>- razina podataka/promjene nad sadržajem grafa – nema promjena na MDV shemi</li> <li>- dodavanje novog zapisa u odgovarajuće MDV hubove, linkove i satelite (hub definicija, link hub, link definicija, satelit hub, satelit definicija, kod, link hub-link-uloga, link link-satelit)</li> </ul>
Promjena veze u relaciju, I <sub>VR</sub>	<ul style="list-style-type: none"> <li>- proširenje SDV grafa – razina sheme/promjene nad strukturom grafa</li> <li>- dodavanje novog satelit čvora (valjanost) i LS brida u SDV graf - Dodaj<sub>C</sub> i Dodaj<sub>B</sub></li> <li>- dodavanje novog hub i satelit čvora i HS bridova u SDV graf - Dodaj<sub>C</sub> i Dodaj<sub>B</sub></li> </ul>	<ul style="list-style-type: none"> <li>- proširenje PMDV grafa – razina sheme/promjene nad strukturom grafa</li> <li>- dodavanje novog satelit čvora (valjanost) i LS brida u PMDV graf - Dodaj<sub>C</sub> i Dodaj<sub>B</sub></li> <li>- dodavanje novog hub i satelit čvora i HS bridova u PMDV graf - Dodaj<sub>C</sub> i Dodaj<sub>B</sub></li> </ul>	<ul style="list-style-type: none"> <li>- razina podataka/promjene nad sadržajem grafa – nema promjena na MDV shemi</li> <li>- dodavanje novog zapisa u odgovarajuće MDV hubove, linkove i satelite (link definicija, hub hub, hub definicija, satelit hub, satelit definicija, kod, link hub-link-uloga, link hub-satelit)</li> </ul>
Promjena atributa u relaciju, I <sub>AR</sub>	<ul style="list-style-type: none"> <li>- proširenje SDV grafa – razina sheme/promjene nad strukturom grafa</li> <li>- dodavanje novog hub, link i satelit čvora te HS i LS bridova u SDV graf - Dodaj<sub>C</sub> i Dodaj<sub>B</sub></li> </ul>	<ul style="list-style-type: none"> <li>- proširenje PMDV grafa – razina sheme/promjene nad strukturom grafa</li> <li>- dodavanje novog hub, link i satelit čvora te HS i LS bridova u PMDV graf - Dodaj<sub>C</sub> i Dodaj<sub>B</sub></li> </ul>	<ul style="list-style-type: none"> <li>- razina podataka/promjene nad sadržajem grafa – nema promjena na MDV shemi</li> <li>- dodavanje novog zapisa u odgovarajuće MDV hubove, linkove i satelite (satelit definicija, atribut definicija, hub hub, hub definicija, link hub-satelit, link hub-link-uloga, kod)</li> </ul>
Promjena relacije u atribut, I <sub>RA</sub>	<ul style="list-style-type: none"> <li>- proširenje SDV grafa – razina sheme/promjene nad strukturom grafa</li> <li>- dodavanje novog satelit čvora (valjanost) i HS brida u SDV graf - Dodaj<sub>C</sub> i Dodaj<sub>B</sub></li> <li>- dodavanje novog link i satelit čvora te LS bridova u SDV graf - Dodaj<sub>C</sub> i Dodaj<sub>B</sub></li> </ul>	<ul style="list-style-type: none"> <li>- proširenje PMDV grafa – razina sheme/promjene nad strukturom grafa</li> <li>- dodavanje novog satelit čvora (valjanost) i HS brida u PMDV graf - Dodaj<sub>C</sub> i Dodaj<sub>B</sub></li> <li>- dodavanje novog link i satelit čvora te LS bridova u PMDV graf - Dodaj<sub>C</sub> i Dodaj<sub>B</sub></li> </ul>	<ul style="list-style-type: none"> <li>- razina podataka/promjene nad sadržajem grafa – nema promjena na MDV shemi</li> <li>- dodavanje novog zapisa u odgovarajuće MDV hubove, linkove i satelite (satelit definicija, atribut definicija, hub definicija, kod)</li> </ul>

Zaključujemo da ovaj konačni skup promjena u izvorima podataka utječe na izmjenu sheme u SDV/PMDV, ali nema efekta na MDV shemu/graf. U SDV/PMDV promjene se implementiraju isključivo kao proširenje postojeće sheme/grafa (kroz samo dvije operacije - Dodaj<sub>C</sub> i Dodaj<sub>B</sub>, odnosno dodavanje novih čvorova i bridova s pridruženim ulogama). Promjene se u MDV implementiraju isključivo na razini podataka (promjene nad sadržajem grafa), kao unos novih zapisa u odgovarajuće relacije. Strukturalne promjene na MDV shemi (ukoliko do njih dođe) tretiraju se jednako kao strukturalne promjene nad SDV/PMDV grafom.

Sukladno tome, možemo definirati konačni skup promjena u SDV/PMDV shemi  $M \{D_H, I_H, B_H, D_S, I_{S_n}, I_{S_t}, B_S, D_L, B_L, I_{HL}, I_{LH}, I_{SH}, I_{HS}\}$ :

- Dodavanje novog huba,  $D_H$
- Preimenovanje huba,  $I_H$
- Brisanje huba,  $B_H$
- Dodavanje satelita u hub ili link,  $D_S$
- Izmjena postojećeg satelita:
  - preimenovanje satelita,  $I_{S_n}$
  - promjena tipa podatka satelita,  $I_{S_t}$
- Brisanje satelita,  $B_S$
- Dodavanje linka između hubova,  $D_L$
- Brisanje linka između hubova,  $B_L$
- Pretvorba hub u link,  $I_{HL}$
- Pretvorba link u hub,  $I_{LH}$
- Pretvorba satelit u hub,  $I_{SH}$
- Pretvorba hub u satelit,  $I_{HS}$

Sljedeća tablica prikazuje da su efekti promjena iz skupa  $M$  nad MDV shemom/grafom ekvivalentne efektima promjena iz skupa  $I$  nad SDV/PMDV shemom/grafom, uz iznimku za  $I_H$ ,  $I_{S_n}$  i  $I_{S_t}$  – međutim možemo poopćiti jer razlikujemo samo razine primjene (podaci/schema, tj. sadržaj/struktura grafa).

**Tablica 4. Učinak promjene u SDV/PMDV grafu na MDV graf**

Promjena u shemi izvora	Efekt promjene u SDV/PMDV	Promjena u MDV shemi	Efekt promjene u MDV
Dodavanje nove relacije, $D_R$	<ul style="list-style-type: none"> <li>- proširenje SDV/PMDV grafa - razina sheme/promjene nad strukturom grafa</li> <li>- dodavanje novih (master) hub, link i satelit čvorova i bridova u SDV/PMDV graf - Dodaj<sub>C</sub> i Dodaj<sub>B</sub></li> <li>- dodavanje novog linka između SDV hub čvora i PMDV master hub čvora - Dodaj<sub>C</sub> i Dodaj<sub>B</sub></li> </ul>	Dodavanje novog huba, $D_H$	- proširenje MDV grafa – razina sheme/promjene nad strukturom grafa
Preimenovanje relacije, $I_R$	- razina podataka/promjene nad sadržajem grafa – nema promjena na SDV/PMDV shemi	Preimenovanje huba, $I_H$	- proširenje MDV grafa – razina sheme/promjene nad strukturom grafa
Brisanje relacije, $B_R$	<ul style="list-style-type: none"> <li>- proširenje SDV/PMDV grafa – razina sheme/promjene nad strukturom grafa</li> <li>- dodavanje novog satelit čvora (valjanost) i HS brida u SDV/PMDV graf - Dodaj<sub>C</sub> i Dodaj<sub>B</sub></li> </ul>	Brisanje huba, $B_H$	- proširenje MDV grafa – razina sheme/promjene nad strukturom grafa
Dodavanje atributa u relaciju, $D_A$	<ul style="list-style-type: none"> <li>- proširenje SDV/PMDV grafa – razina sheme/promjene nad strukturom grafa</li> <li>- dodavanje novog satelit čvora i HS ili LS brida u SDV/PMDV graf - - Dodaj<sub>C</sub> i Dodaj<sub>B</sub></li> </ul>	Dodavanje satelita u hub ili link, $D_S$	- proširenje MDV grafa – razina sheme/promjene nad strukturom grafa
Preimenovanje atributa u relaciji, $I_{An}$	- razina podataka/promjene nad sadržajem grafa – nema promjena na SDV/PMDV shemi	Preimenovanje satelita, $I_{Sn}$	- proširenje MDV grafa – razina sheme/promjene nad strukturom grafa
Promjena tipa podatka atributa u relaciji, $I_{At}$	<ul style="list-style-type: none"> <li>- proširenje SDV/PMDV grafa – razina sheme/promjene nad strukturom grafa</li> <li>- dodavanje novog satelit čvora i satelit čvora (valjanost) te</li> </ul>	Promjena tipa podatka satelita, $I_{St}$	- proširenje MDV grafa – razina sheme/promjene nad strukturom grafa



	odgovarajućih bridova SDV/PMDV graf - Dodaj <sub>C</sub> i Dodaj <sub>B</sub>		
Brisanje atributa iz relacije, B <sub>A</sub>	- proširenje SDV/PMDV grafa – razina sheme/promjene nad strukturom grafa - dodavanje novog satelit čvora (valjanost) i HS brida u SDV/PMDV graf - Dodaj <sub>C</sub> i Dodaj <sub>B</sub>	Brisanje satelita, B <sub>S</sub>	- proširenje MDV grafa – razina sheme/promjene nad strukturom grafa
Dodavanje veze između relacija, D <sub>V</sub>	- proširenje SDV/PMDV grafa – razina sheme/promjene nad strukturom grafa - dodavanje novog čvora i bridova u SDV/PMDV graf - Dodaj <sub>C</sub> i Dodaj <sub>B</sub>	Dodavanje linka između hubova, D <sub>L</sub>	- proširenje MDV grafa – razina sheme/promjene nad strukturom grafa
Brisanje veze između relacija, B <sub>V</sub>	- proširenje SDV/PMDV grafa – razina sheme/promjene nad strukturom grafa - dodavanje novog satelit čvora (valjanost) i LS brida u SDV/PMDV graf - Dodaj <sub>C</sub> i Dodaj <sub>B</sub>	Brisanje linka između hubova, B <sub>L</sub>	- proširenje MDV grafa – razina sheme/promjene nad strukturom grafa
Promjena relacije u vezu, I <sub>RV</sub>	- proširenje SDV/PMDV grafa – razina sheme/promjene nad strukturom grafa - dodavanje novog satelit čvora (valjanost) i HS brida u SDV/PMDV graf - Dodaj <sub>C</sub> i Dodaj <sub>B</sub> - dodavanje novog link i satelit čvora i LS bridova u SDV/PMDV graf - Dodaj <sub>C</sub> i Dodaj <sub>B</sub>	Pretvorba hub u link, I <sub>HL</sub>	- proširenje MDV grafa – razina sheme/promjene nad strukturom grafa
Promjena veze u relaciju, I <sub>VR</sub>	- proširenje SDV/PMDV grafa – razina sheme/promjene nad strukturom grafa - dodavanje novog satelit čvora (valjanost) i LS brida u SDV/PMDV graf - Dodaj <sub>C</sub> i Dodaj <sub>B</sub> - dodavanje novog hub i satelit čvora i HS bridova u SDV/PMDV graf - Dodaj <sub>C</sub> i Dodaj <sub>B</sub>	Pretvorba link u hub, I <sub>LH</sub>	- proširenje MDV grafa – razina sheme/promjene nad strukturom grafa  - dodavanje novog satelit čvora (valjanost) i LS brida u MDV graf - Dodaj <sub>C</sub> i Dodaj <sub>B</sub>

Promjena atributa u relaciju, I <sub>AR</sub>	- proširenje SDV/PMDV grafa – razina sheme/promjene nad strukturom grafa - dodavanje novog hub, link i satelit čvora te HS i LS bridova u SDV/PMDV graf - Dodaj <sub>C</sub> i Dodaj <sub>B</sub>	Pretvorba satelit u hub, I <sub>SH</sub>	- proširenje MDV grafa – razina sheme/promjene nad strukturom grafa
Promjena relacije u atribut, I <sub>RA</sub>	- proširenje SDV/PMDV grafa – razina sheme/promjene nad strukturom grafa - dodavanje novog satelit čvora (valjanost) i HS brida u SDV/PMDV graf - Dodaj <sub>C</sub> i Dodaj <sub>B</sub> - dodavanje novog link i satelit čvora te LS bridova u SDV/PMDV graf - Dodaj <sub>C</sub> i Dodaj <sub>B</sub>	Pretvorba hub u satelit, I <sub>HS</sub>	- proširenje MDV grafa – razina sheme/promjene nad strukturom grafa

Sve ostale promjene se mogu tretirati na isti način – kroz dvije operacije, Dodaj<sub>C</sub> i Dodaj<sub>B</sub>.

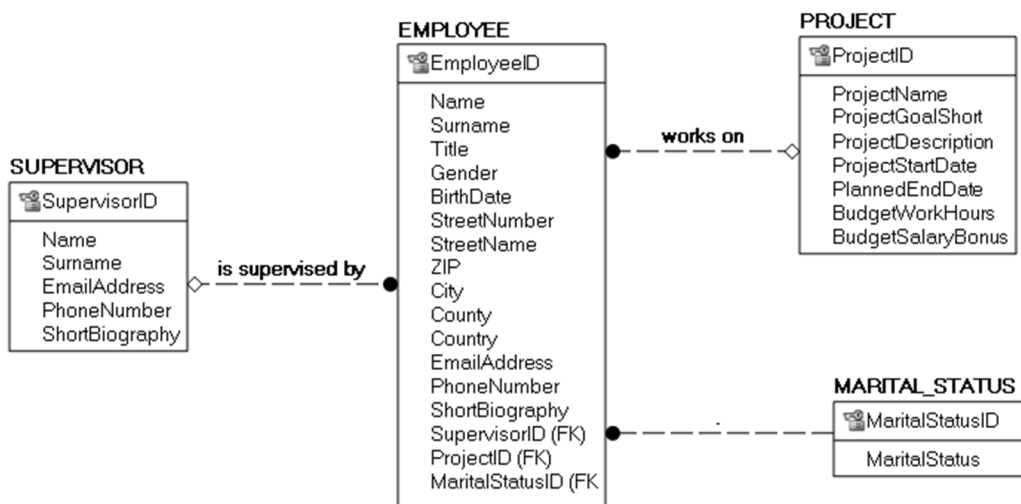
Ovime smo teoretski validirali model i teoretski dokazali hipotezu H2– pokazali smo da se sve promjene nad shemom izvora podataka u SDV/PMDV i MDV implementiraju isključivo kao jednostavna proširenja (isključivo kroz operacije dodavanja, Dodaj<sub>C</sub> i Dodaj<sub>B</sub>) – stari podaci i sheme se čuvaju te na taj način nema gubitka informacija (o podacima ili shemi). Upravo to čini model obuhvatnim i trajnim. Dodatno, ovaj dokaz hipoteze H2. potvrdit ćemo u Poglavlju 5, s empirijskom validacijom prototipa.

### ***4.3. Praktični primjer za validaciju operacija evolucije nad skupom promjena***

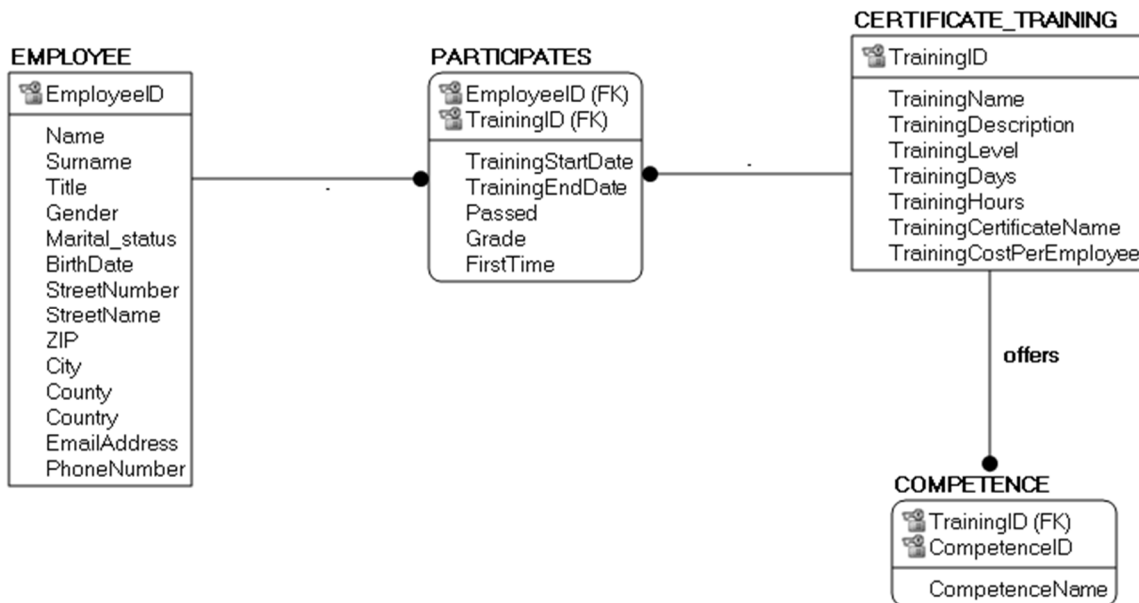
U svrhu teorijske validacije predloženog skupa promjena i operacija, razvijen je praktični primjer koji prati rad zaposlenika nekog poduzeća po projektima te sudjelovanje na poslovnim usavršavanjima (treninzima). Na Slici 9 vidimo shemu JobDB izvora, koja opisuje zaposlenike (EMPLOYEE i MARITAL\_STATUS), njihove voditelje (SUPERVISOR) te projekte na kojima

trenutno rade (PROJECT). Na Slici 10 vidimo shemu TrainingDB izvora, koja također opisuje zaposlenike (EMPLOYEE), ali ovaj put u kontekstu poslovnih treninga (CERTIFICATE\_TRAINING) na kojima sudjeluju (PARTICIPATES) te kompetencija koje kroz njih ostvaruju (COMPETENCE). Na Slici 11 nalazi se SDV shema SdvDB baze podataka koja integrira ova dva izvora u jedinstvenu bazu podataka, kroz upotrebu Data Vault modela. Možemo vidjeti da svaka relacija iz izvorišnih shema u SdvDB shemi ima svoj hub ili link (ovisno o tome posjeduje li vlastiti poslovni ključ) te ima svoje satelite (koji sadržavaju opisne attribute iz izvorišnih shema). SdvDB shema sadrži i već spomenute standardne DV meta-podatke (LDTS i LEDTS kao datume punjenja i prestanja važenja). Dodatno, radi demonstracije, u shemu sa Slike 11 dodan je i jedan matični hub sa svojim satelitima te je povezan s osnovnim hubom, iako će PMDV shema PmdvDB baze podataka biti fizički realizirana u zasebnoj bazi podataka (više u Poglavlju 5).

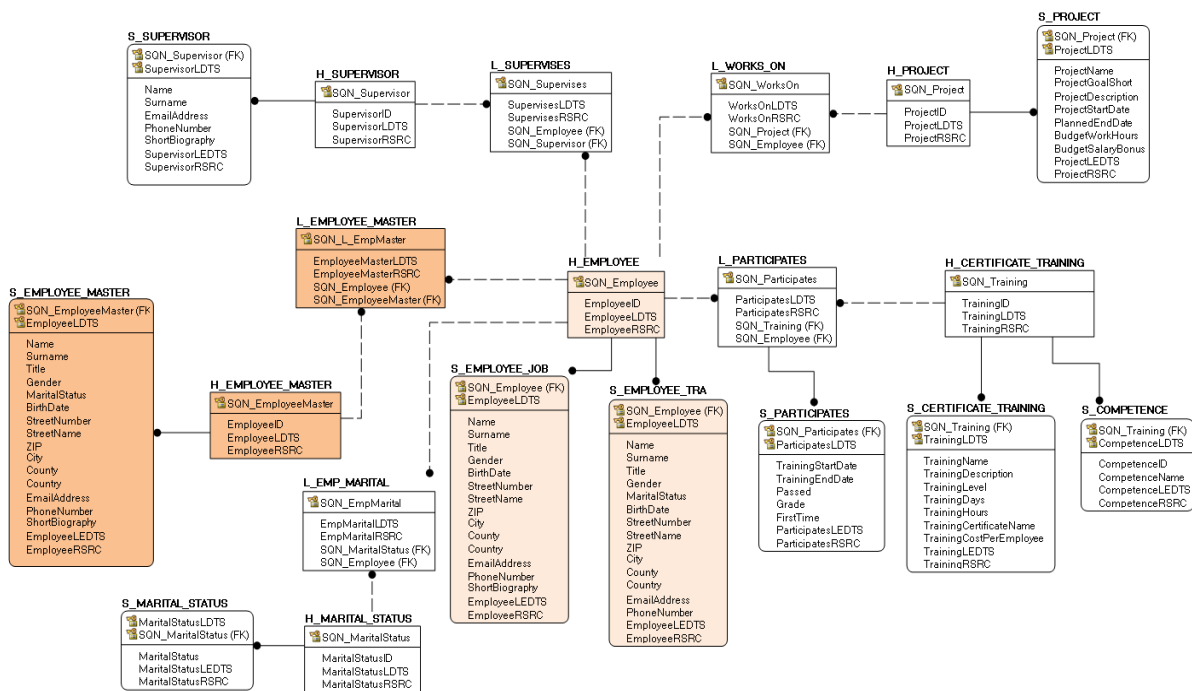
Vidjet ćemo kako definirane operacije u ovim izvorišnim shemama utječu na evidentiranje promjena u predloženim SDV/PMDV i MDV shemama te ćemo kroz praktični primjer pokazati kako su predložene operacije Dodaj<sub>C</sub> i Dodaj<sub>B</sub> dovoljne za provođenje cijelog skupa promjena.



Slika 9. JobDB shema



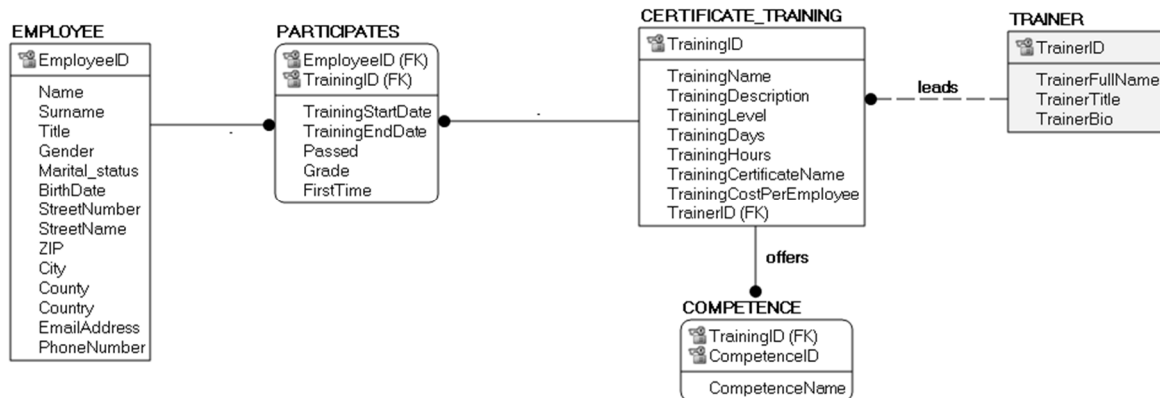
Slika 10. TrainingDB shema



Slika 11. Početna integrirana SdvDB i PmdvDB

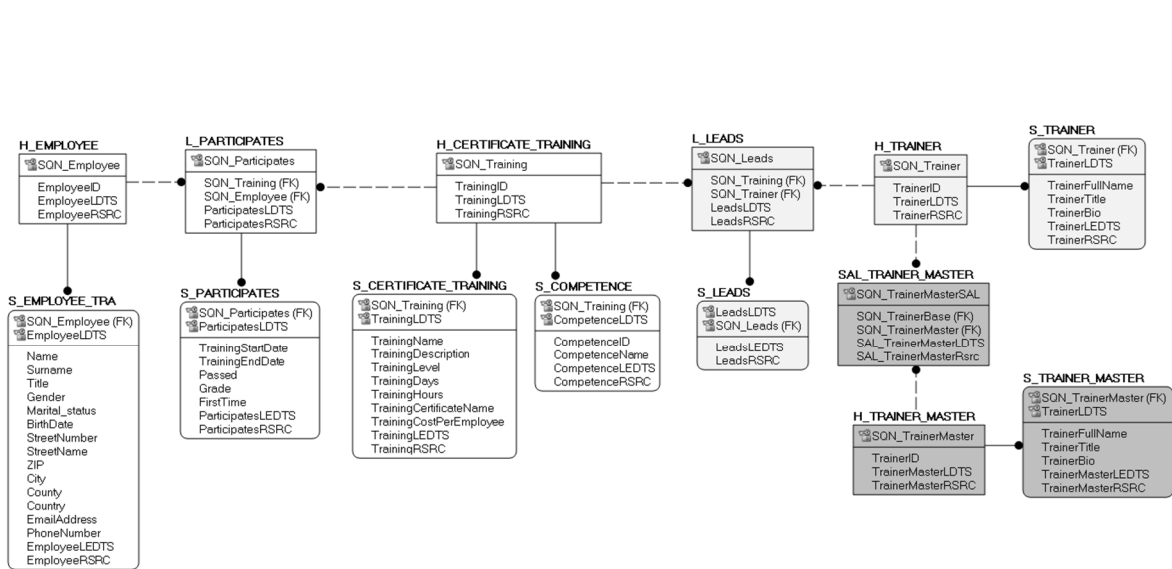
### 4.3.1. Dodavanje nove relacije, $D_R$ i Dodavanje veze između relacija, $D_V$

Na Slici 12 vidimo dodavanje relacije TRAINER u shemu izvora podataka TrainingDB, skupa s dodavanjem veze između nove relacije TRAINER i postojeće relacije CERTIFICATE\_TRAINING.



**Slika 12. Dodavanje nove relacije,  $D_R$  i Dodavanje veze između relacija,  $D_V$  u TrainingDB**

Na Slici 13 možemo vidjeti kako se ova promjena implementira dalje u shemu centralnog SP/UMP. SDV graf će se jednostavno proširiti dodavanjem novih čvorova i bridova – dodaje se novi hub H\_TRAINER, satelit S\_TRAINER i link L\_LEADS u SDV shemu (svijetlosivo obojano), kao i veze među njima. Link L\_LEADS predstavlja vezu između relacije TRAINER i postojeće relacije CERTIFICATE\_TRAINING, odnosno njihovih hubova u SDV. PMDV shema se također mijenja – dodaje se novi matični hub H\_TRAINING\_MASTER sa svojim satelitom S\_TRAINING\_MASTER te link SAL\_TRAINER\_MASTER koji povezuje odgovarajući originalni i matični hub (tamnosivo obojano). MDV shema strukturalno se ne mijenja, već se ova promjena pohranjuje na podatkovnoj razini – dodaje se novi redak zapisa u odgovarajuće hubove (H\_HUB, H\_SATELLITE, H\_LINK, H\_ATTRIBUTE, ...).

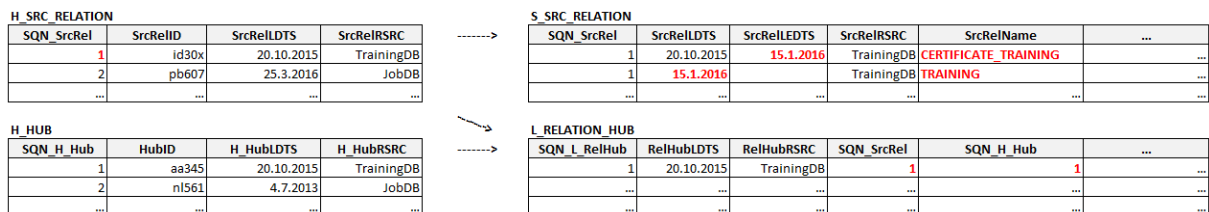


Slika 13. Dodavanje nove relacije,  $D_R$  i Dodavanje veze između relacija,  $D_V$  u SdvDB

#### 4.3.2. Preimenovanje relacije, $I_R$

U shemi izvora podataka TrainingDB relaciju CERTIFICATE\_TRAINING preimenovat ćemo u TRAINING.

U SDV i PMDV shemi nema promjene, dok se u MDV shemi sve promjene odvijaju na podatkovnoj razini - ispunjava se LEDTS za trenutni zapis te se stvara novi zapis u odgovarajućim hubovima, linkovima i satelitima (H\_SRC\_RELATION i S\_SRC\_RELATION te H\_HUB L\_RELATION\_HUB), odnosno kreira se novi zapis za izvorišnu preimenovanu relaciju te se kreira mapiranje između nje i postojećeg huba H\_CERTIFICATE\_TRAINING u SDV/PMDV shemi. Taj zapis se kasnije koristi u ETL procesima za dohvaćanje ispravnih i ažuriranih podataka te kreiranje upita.



Slika 14. Preimenovanje relacije,  $I_R$  u TrainingDB shemi i efekt u MDV repozitoriju

### 4.3.3. *Brisanje relacije, B<sub>R</sub> i Brisanje veze između relacija, B<sub>V</sub>*

U izvoru podataka TrainingDB briše se relacija TRAINER zajedno sa svim svojim vezama (u ovom slučaju veza *leads* na CERTIFICATE\_TRAINING relaciju). U SDV i PMDV shemi nema strukturalne promjene, a na podatkovnoj razini se u atributu TrainerLEDTS satelita S\_TRAINER i LeadsLEDTS satelita S\_LEADS upisuje datum brisanja relacije iz izvora. Također, u MDV shemi na podatkovnoj razini evidentira se brisanje izvorišne relacije i njenih veza kroz dodavanje novih zapisa o valjanosti - unos datuma brisanja u LEDTS atributu odgovarajućih satelita (S\_SRC\_RELATION, S\_HUB\_DEF, S\_LINK\_DEF, S\_SRC\_TAB\_REL\_DEF). Na Slici 15 vidimo kako je izvorišna tablica TRAINER kreirana 21.11.2015., a „izbrisana“ 23.6.2016. te za nju u LEDTS stupcima više nema NULL vrijednosti. Brisanje veze između hubova H\_TRAINER i H\_CERTIFICATE\_TRAINING prikazano je u H\_LINK i S\_LINK\_DEF tablicama. Ovi zapisi u MDV se kasnije koriste u ETL procesima za dohvaćanje ispravnih i ažuriranih podataka te kreiranje upita.

H_SRC_RELATION					S_SRC_RELATION						
SQN_SrcRel	SrcRelID	SrcRelLDTS	SrcRelRSRC		SQN_SrcRel	SrcRelLDTS	SrcRelLEDTS	SrcRelRSRC	SrcRelName	StatusDescription	...
3	sd209	21.11.2015	TrainingDB		3	21.11.2015	23.6.2016	TrainingDB	TRAINER	Deleted from source	...

H_HUB				S_HUB_DEF						
SQN_SrcTable	SrcTableID	SrcTableLDTS	SrcTableRSRC	SQN_SrcTable	SrcTableLDTS	SrcTableLEDTS	HubName	HubDescription	StatusDescription	...
3		21.11.2015	TrainingDB	3	21.11.2015	23.6.2016	H_Trainer	All trainers business keys...	Deleted from source	...

L_SRC_TAB_REL					S_SRC_TAB_REL_DEF						
SQN_L_TabRel	SQN_SrcRel	SQN_H_SrcRelsh	L_TabRelLDTS	L_TabRelRSRC	SQN_L_TabRel	S_TabRelLDTS	S_TabRelLEDTS	S_TabRelRSRC	TabRelshName	StatusDescription	...
1	3	2	21.11.2015	TrainingDB	1	21.11.2015	23.6.2016	TrainingDB	Leads	Deleted from source	...

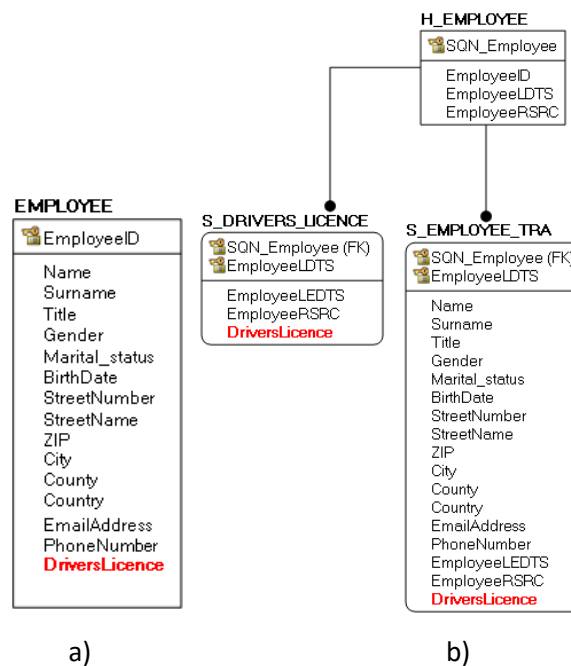
H_LINK				S_LINK_DEF						
SQN_H_Link	LinkID	LinkLDTS	LinkRSRC	SQN_H_Link	LinkLDTS	LinkLEDTS	LinkName	LinkDescription	StatusDescription	...
1	op309	21.11.2015	TrainingDB	1	21.11.2015	23.6.2016	L_LEADS	H_TRAINER and H_CERTIFICATE_TRAINING relationship	Deleted from source	...

Slika 15. Brisanje relacije, B<sub>R</sub> i Brisanje veze između relacija, B<sub>V</sub> u MDV repozitoriju

### 4.3.4. *Dodavanje atributa u relaciju, D<sub>A</sub>*

U TrainingDB je EMPLOYEE izmijenjen – dodan mu je atribut DriversLicence. Na podatkovnoj razini u SDV i PMDV mijenja se S\_EMPLOYEE na način da se na zadnje aktivne zapise postavi LEDTS vrijednost kraja važenja, a zatim se doda novi stupac DriversLicence (SDV graf će se jednostavno proširiti dodavanjem novih čvorova i bridova) te se ponovno učitaju podaci u

tablicu. U MDV shemi se promjene događaju samo na podatkovnoj razini - dodaju se novi zapisi u odgovarajuće hubove (H\_SRC\_ATTRIBUTE, H\_ATTRIBUTE), satelite (S\_SRC\_ATTRIBUTE, S\_ATTRIBUTE\_DEF) i linkove (L\_SRC\_RELATION\_ATTRIBUTE, L\_ATTRIBUTE\_SRC\_ATTRIBUTE, L\_SATELLITE\_ATTRIBUTE). Alternativa je jednostavno dodavanje novog satelita S\_DRIVERS\_LICENCE s odgovarajućim atributima u SDV/PMDV shemu (dakle dodavanje novih čvorova i brigova u graf) te kasnije preko JOIN-a spajanje sa S\_EMPLOYEE. U tome slučaju se u MDV shemi dodaje novi zapis i u hub H\_SATELLITE te njegov pripadajući satelit S\_SATELLITE.



Slika 16. Dodavanje atributa u relaciju  $D_A$  u TrainingDB shemu (a) i u SdvDB shemu (b)

a)

S_EMPLOYEE							
SQN_Employee	EmployeeLDTs	EmployeeLEDTS	EmployeeRSRC	Name	Surname	DriversLicence	...
1	13.5.2015	21.10.2016	TrainingDB	Daniel	Hamilton		...
2	18.10.2015	21.10.2016	TrainingDB	Gabriela	Novak		...
1	21.10.2016		TrainingDB	Daniel	Hamilton	YES	...
2	21.10.2016		TrainingDB	Gabriela	Novak	NO	...

b)

S_EMPLOYEE						
SQN_Employee	EmployeeLDTs	EmployeeLEDTS	EmployeeRSRC	Name	Surname	...
1	13.5.2015		TrainingDB	Daniel	Hamilton	...
2	18.10.2015		TrainingDB	Gabriela	Novak	...

S_DRIVERS_LICENCE				
SQN_Employee	EmployeeLDTs	EmployeeLEDTS	EmployeeRSRC	DriversLicence
1	21.10.2016		TrainingDB	YES
2	21.10.2016		TrainingDB	NO

Slika 17. Dodavanje atributa u relaciju  $D_A$  u TrainingDB na podatkovnoj razini



Na Slici 18 možemo vidjeti primjer unosa novih zapisa u izvorišni SRC\_ATTRIBUTE hub, SDV/PMDV hub H\_ATTRIBUTE te link koji sadrži njihovo mapiranje. Na isti način evidentiramo promjene i u ostalim (gore navedenim) hubovima, satelitima i linkovima.

H_SRC_ATTRIBUTE			
SQN_SrcAttr	SrcAttrID	SrcAttrLDTS	SrcAttrRSRC
8	ax404	24.8.2016	TrainingDB

H_ATTRIBUTE			
SQN_Attribute	AttributeID	AttributeLDTS	AttributeRSRC
6	bb776	24.8.2016	TrainingDB

L_ATTRIBUTE_SRC_ATTRIBUTE					
SQN_L_Attr_SrcAttr	SrcAttr_SrcAttrLDTS	SrcAttr_SrcAttrRSRC	SQN_SrcAttr	SQN_Attribute	...
25	24.8.2016	TrainingDB	8	6	...

Slika 18. Dodavanje atributa u relaciju D<sub>A</sub> u SdvDB na podatkovnoj razini

#### 4.3.5. Izmjena postojećeg atributa u relaciji - preimenovanje atributa, I<sub>An</sub>

U izvorišnoj shemi atribut Name preimenovan je u First\_Name. Slično kao i kod preimenovanja relacije, u SDV/PMDV shemi nema promjene, dok se u MDV shemi sve promjene odvijaju na podatkovnoj razini – ispunjava se LEDTS za trenutni zapis te se stvara novi zapis u odgovarajućim hubovima, linkovima i satelitima (H\_SRC\_ATTRIBUTE i S\_SRC\_ATTRIBUTE te H\_ATTRIBUTE, S\_ATTRIBUTE, L\_ATTR\_SRC\_ATTR i L\_SATELLITE\_ATTRIBUTE), odnosno kreira se novi zapis za izvorišni preimenovani atribut te se kreiraju mapiranja između njega i postojećeg atributa i satelita u SDV/PMDV shemi (A\_Name --> A\_First\_Name, A\_First\_Name --> S\_EMPLOYEE). Taj zapis se kasnije koristi u ETL procesima za dohvaćanje ispravnih i ažuriranih podataka te kreiranje upita.

H_SRC_ATTRIBUTE				S_SRC_ATTR					
SQN_H_SrcAttr	SrcAttrID	H_SrcAttrLDTS	H_SrcAttrRSRC	SQN_H_SrcAttr	S_SrcAttrLDTS	S_SrcAttrLEDTS	S_SrcAttrRSRC	SrcAttrName	...
1	gg560	20.10.2015	TrainingDB	1	20.10.2015	15.1.2016	TrainingDB	Name	...
2	za232	25.3.2016	JobDB	1	15.1.2016		TrainingDB	First_Name	...
...	...	...	...	...	...	...	...	...	...

H_ATTRIBUTE				L_ATTR_SRC_ATTR					
SQN_H_Attribute	AttributeID	H_AttrLDTS	H_AttrRSRC	SQN_L_AttrSrcAttr	AttrSrcAttrLDTS	AttrSrcAttrRSRC	SQN_H_Attribute	SQN_H_SrcAttr	...
1	gg560	20.10.2015	TrainingDB	1	20.10.2015	TrainingDB	1	1	...
2	za232	4.7.2013	JobDB	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...	...

Slika 19. Preimenovanje atributa I<sub>An</sub> u MDV na podatkovnoj razini

#### 4.3.6. *Izmjena postojećeg atributa u relaciji - promjena tipa podatka atributa, I<sub>At</sub>*

U izvorišnoj relaciji CERTIFICATE\_TRAINING atribut TrainingLevel mijenja tip podatka iz NUMERIC u STRING. Dosada su razine treninga bile u rasponu od 1 do 5, međutim postoji potreba uvođenja među-razina treninga (npr. 1a i 1b). U SDV/PMDV shemi promjenu implementiramo na način da se u S\_CERTIFICATE\_TRAINING stupcu TrainingLevel direktno promijeni tip podatka (ukoliko je moguće), ili se radi kopija tablice S\_CERTIFICATE\_TRAINING s novim tipom podatka TrainingLevel stupca (operacije dodavanja nove relacije D<sub>R</sub> i dodavanja veze između relacija D<sub>V</sub>).

U MDV shemi se ta promjena može evidentirati na dva načina:

1. Na zadnji aktivni zapis u S\_ATTRIBUTE za TrainingLevel postavljamo LEDTS vrijednost kraja važenja, a zatim unosimo novi zapis s promjenom vrijednosti iz NUMERIC u STRING u AttributeType stupcu. Dodatno, u S\_SAT\_DEF satelitu za staru S\_CERTIFICATE\_TRAINING relaciju postavljamo LEDTS vrijednost kraja važenja (te tako nova kopija postaje važeća relacija).
2. Domene mogu biti pohranjene u zasebnom konceptu (npr. H\_DOMAIN i S\_DOMAIN) te preko linka L\_ATTRIBUTE\_DOMAIN biti dodijeljene pojedinim atributima. U tome slučaju u MDV shemi jednostavno dodajemo novi zapis u L\_ATTRIBUTE\_DOMAIN link u kojem pridružujemo ključ STRING vrijednosti iz H\_DOMAIN ključu TrainingLevel vrijednosti iz H\_ATTRIBUTE. Na prethodno mapiranje (TrainingLevel – NUMERIC) postavljamo vrijednost LEDTS.

L\_ATTRIBUTE\_DOMAIN

SQN_L_AttrDom	AttrDomLDTS	AttrDomRSRC	SQN_H_Attribute	SQN_H_Domain	...
1	15.7.2015	TrainingDB	7	5	...
2	22.12.2015	TrainingDB	7	6	...
...	...	...	...	...	...

H\_ATTRIBUTU

SQN_H_Attribute	AttributeID	H_AttrLDTS	H_AttrRSRC
7	rr456	21.11.2014	TrainingDB
8	ds264	14.3.2015	JobDB
...	...	...	...

H\_DOMAIN

SQN_H_Domain	DomainID	H_DomainLDTS	H_DomainRSRC
5	df956	5.2.2013	Oracle11g
6	jk789	5.2.2013	Oracle11g
...	...	...	...

S\_ATTR\_DEF

SQN_H_Attribute	S_AttrDefLDTS	S_AttrDefLEDTS	S_AttrDefRSRC	AttrName	AttrType	...
7	21.11.2014	22.12.2015	TrainingDB	TrainingLevel	NUMERIC	...
7	22.12.2015		TrainingDB	TrainingLevel	STRING	...
...	...	...	...	...	...	...

S\_DOM\_DEF

SQN_H_Domain	S_DomDefLDTS	S_DomDefLEDTS	S_DomDefRSRC	DataType	...
5	5.2.2013		Oracle11g	NUMERIC	...
6	5.2.2013		Oracle11g	STRING	...
...	...	...	...	...	...

Slika 20. Promjena tipa podataka atributa I<sub>At</sub> u MDV na podatkovnoj razini

#### 4.3.7. Brisanje atributa iz relacije, B<sub>A</sub>

Iz relacije EMPLOYEE izbrisat ćemo atribut DriversLicence. Slično kao i kod brisanja izvorišne relacije, kod brisanja izvorišnog atributa u SDV i PMDV shemi nema strukturalne promjene, a u MDV shemi na podatkovnoj razini evidentira se brisanje izvorišnog atributa kroz dodavanje novog zapisa o valjanosti - unos datuma brisanja u LEDTS attribute odgovarajućih satelita (S\_SRC\_ATTR i S\_ATTR\_DEF). Taj zapis se kasnije koristi u ETL procesima za dohvaćanje ispravnih i ažuriranih podataka te kreiranje upita.

S\_SRC\_ATTR

SQN_H_SrcAttr	S_SrcAttrLDTS	S_SrcAttrLEDTS	S_SrcAttrRSRC	SrcAttrName	StatusDescription	...
9	20.10.2015	18.6.2016	TrainingDB	DriversLicence	Deleted from source	...
...	...	...	...	...	...	...

S\_ATTR\_DEF

SQN_H_Attribute	S_AttrDefLDTS	S_AttrDefLEDTS	S_AttrDefRSRC	AttrName	StatusDescription	...
13	20.10.2015	18.6.2016	TrainingDB	DriversLicence	Deleted from source	...
...	...	...	...	...	...	...

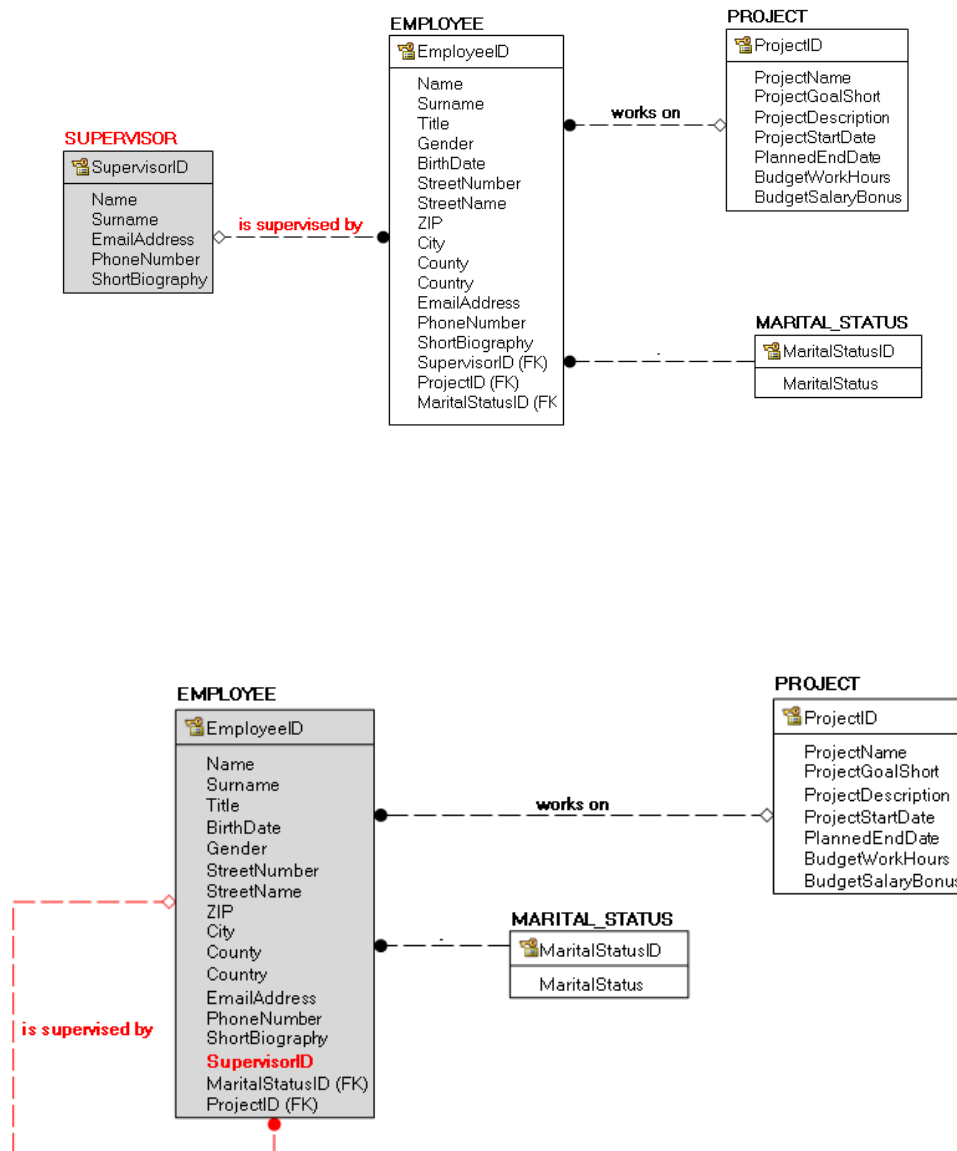
Slika 21. Brisanje atributa iz relacije B<sub>A</sub> u MDV na podatkovnoj razini

#### 4.3.8. Pretvorba relacije u vezu, I<sub>RV</sub>

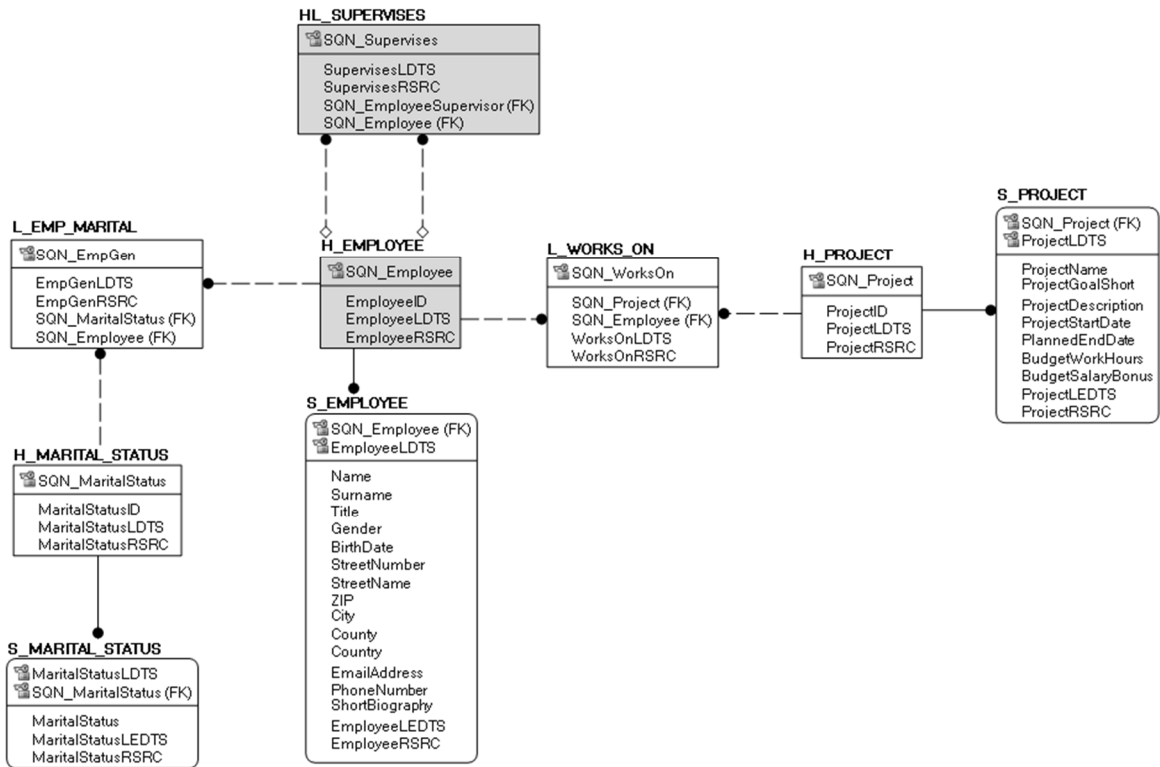
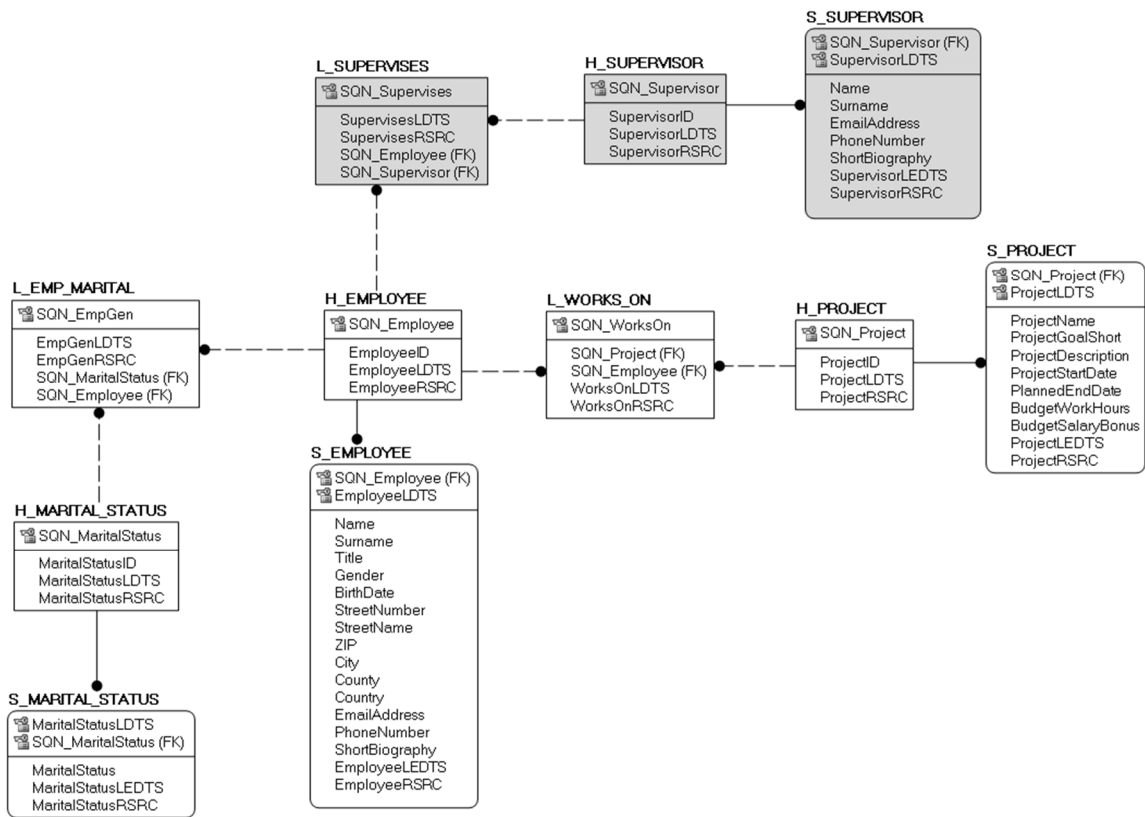
Obzirom da relacija SUPERVISOR u JobDB sadrži podskup atributa relacije EMPLOYEE, a semantički ukazuje na zaposleničku hijerarhiju, ovu relaciju ćemo u izvoru podataka pretvoriti u rekurzivnu vezu *is supervised by* relacije EMPLOYEE.

U SDV/PMDV na relaciju SUPERVISOR primjenjujemo prethodno opisanu promjenu brisanja relacije B<sub>R</sub> i brisanja veze između relacija B<sub>V</sub>. Zatim primjenjujemo promjenu dodavanja veze

između relacija  $D_V$  nad hubom  $H\_EMPLOYEE$  koja će kreirati novi hijerarhijski link  $HL\_SUPERVISES$  te ga vezati s dvostrukom vezom na  $H\_EMPLOYEE$  (dodajemo novi čvor i bridove u SDV/PMDV graf). U MDV shemi nema strukturalnih promjena, već se ova promjena pohranjuje na podatkovnoj razini – dodaju se novi zapisi u odgovarajuće tablice ( $S\_SRC\_RELATION$ ,  $S\_SRC\_RELATIONSHIP$ ,  $S\_HUB\_DEF$ ,  $S\_LINK\_DEF$ ,  $H\_LINK$ ), prema prethodno opisanim primjerima.



Slika 22. Pretvorba relacije u vezu  $I_{RV}$  u JobDB shemi

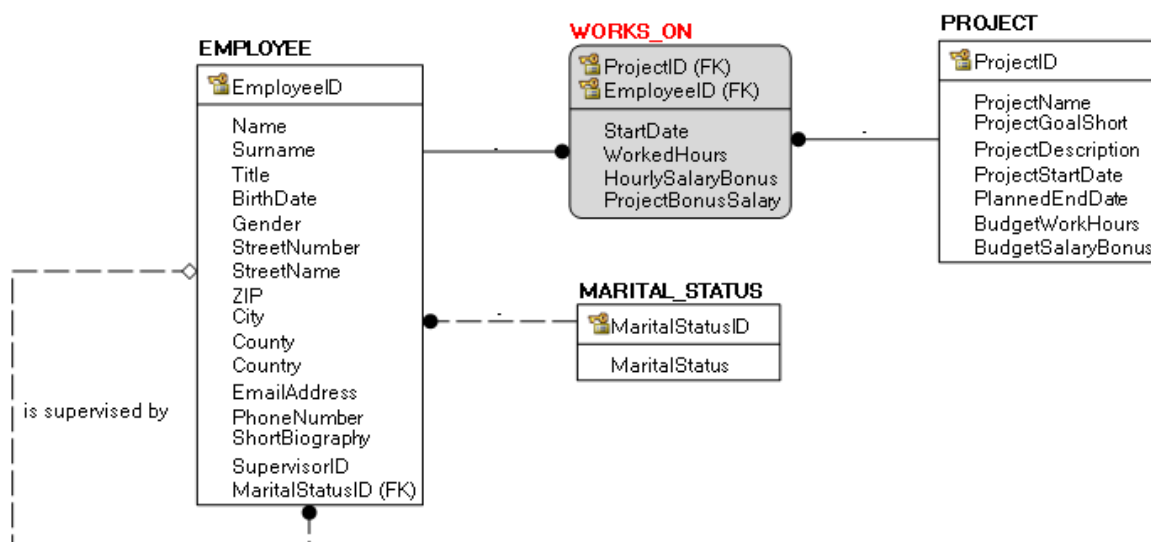
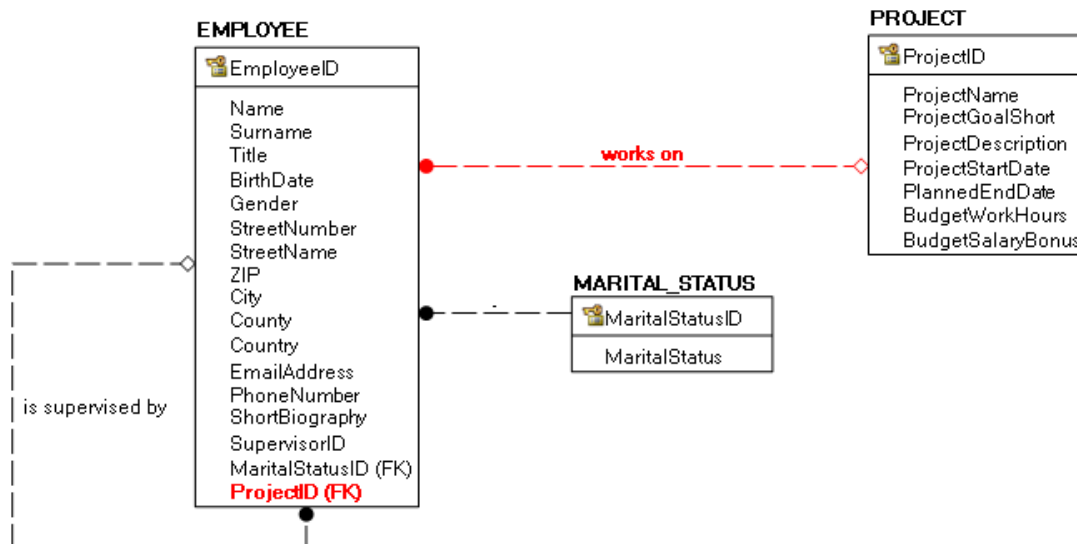


Slika 23. Pretvorba relacije u vezu I<sub>RV</sub> u SDV shemi

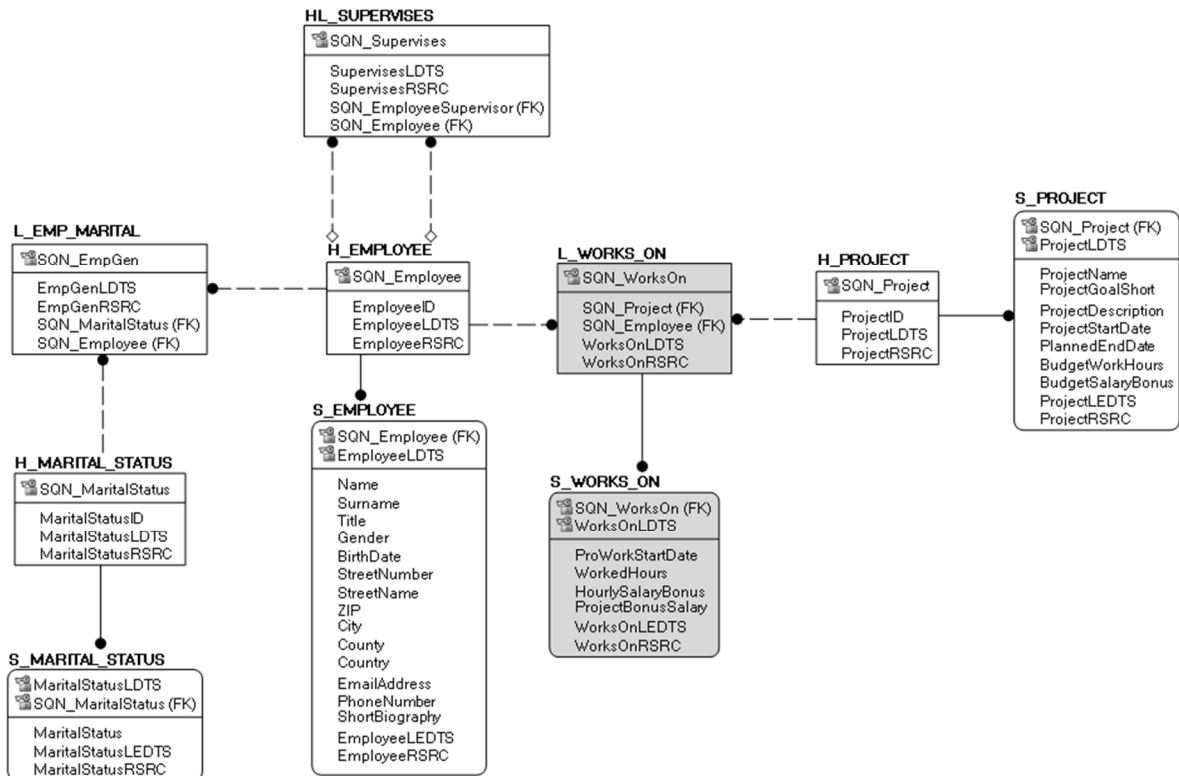
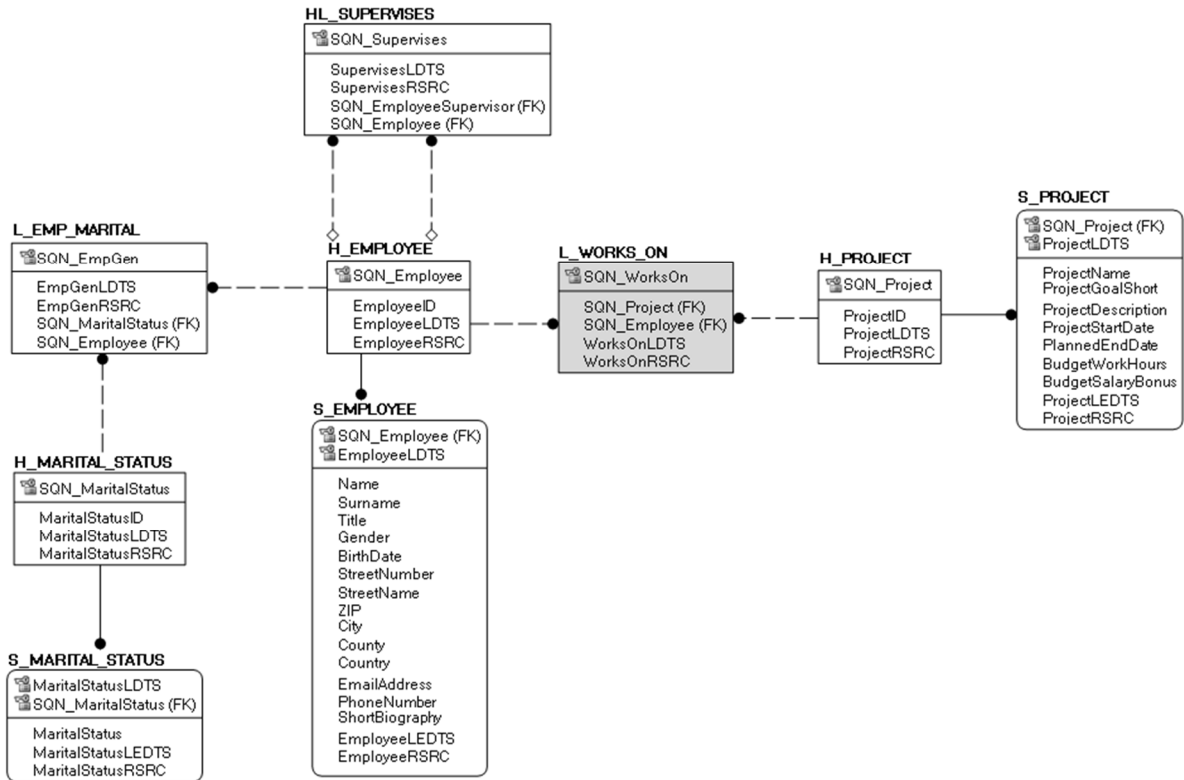
#### 4.3.9. Pretvorba veze u relaciju, $I_{VR}$

U JobDB shemi možemo vidjeti da svaki radnik trenutno radi na jednom projektu. Međutim, želimo omogućiti da svaki radnik može raditi na više projekata te pratiti koliko je ukupno sati radnik radio na svakom projektu i koliki mu je iznos bonusa. To znači da izvorišnu vezu *works\_on* moramo pretvoriti u relaciju WORKS\_ON koja će biti povezana s relacijama EMPLOYEE i PROJECT.

U SDV/PMDV shemi na vezu *works\_on* primjenjujemo prethodno opisanu promjenu brisanja veze između relacija  $B_V$ . Zatim primjenjujemo promjene dodavanja nove relacije  $D_R$  i dodavanja nove veze između relacija  $D_V$ , kako bismo dodali izvorišnu relaciju WORKS\_ON te je vezali na relacije EMPLOYEE i PROJECT – inače bismo dodali novi hub i njegov satelit H\_WORKS\_ON i S\_WORKS\_ON, međutim ova relacija zadržava svoj ključ (odnosno nema prirodni poslovni ključ) te je možemo implementirati u SDV/PMDV kao link – dakle dodajemo link L\_WORKS\_ON i satelit S\_WORKS\_ON te u njih pohranjujemo zapise o vezama između zaposlenika i projekata. U MDV shemi nema strukturalnih promjena, već se ova promjena pohranjuje na podatkovnoj razini (dodaju se novi zapisi u H\_SRC\_RELATIONSHIP, H\_SRC\_RELATION, S\_SRC\_RELATION, S\_SRC\_ATTRIBUTE, H\_LINK, H\_SATELLITE i H\_ATTRIBUTE, kao i njihove pripadajuće S\_DEF satelite), prema prethodno opisanim primjerima.



Slika 24. Pretvorba veze u relaciju 1vR u JobDB shemi

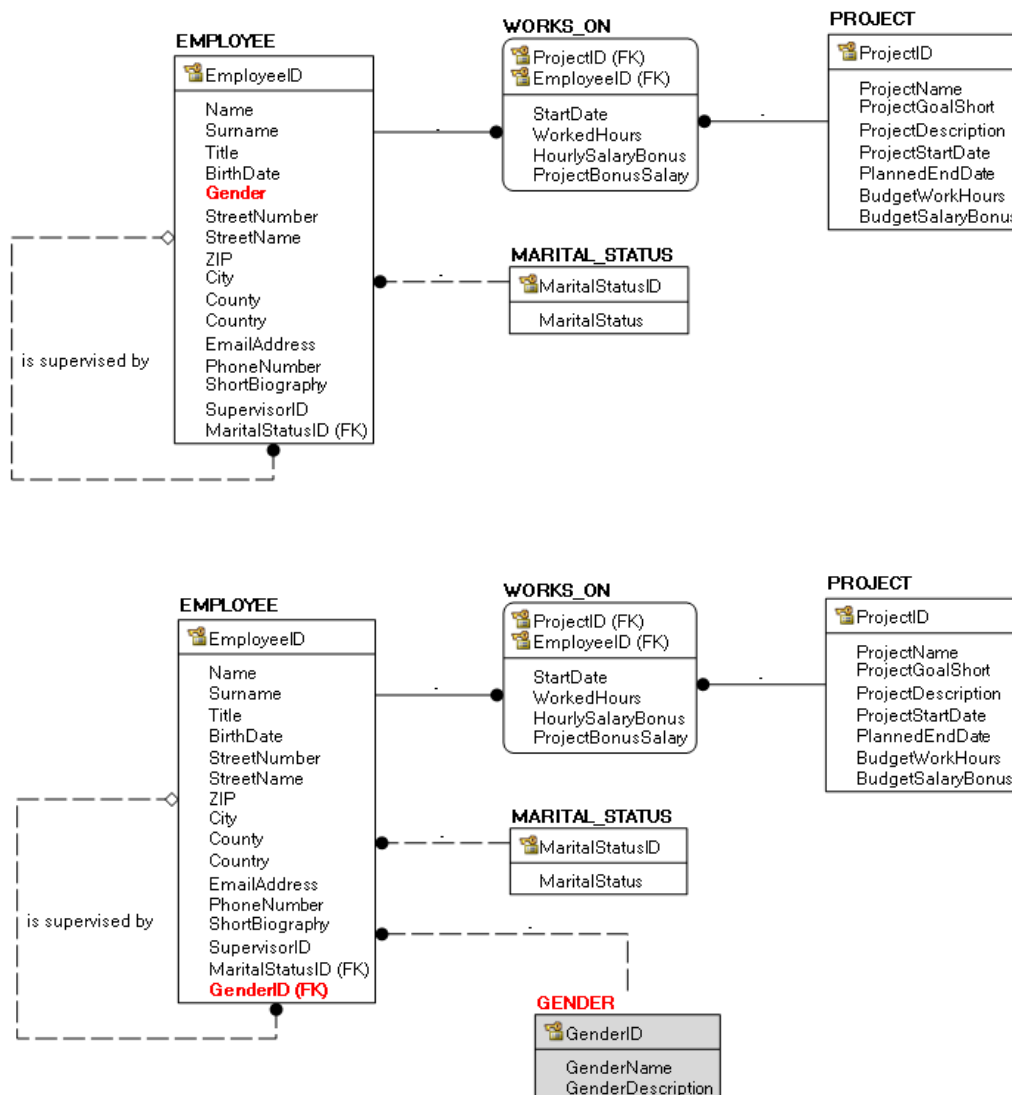


Slika 25. Pretvorba veze u relaciju I<sub>VR</sub> u SDV shemi

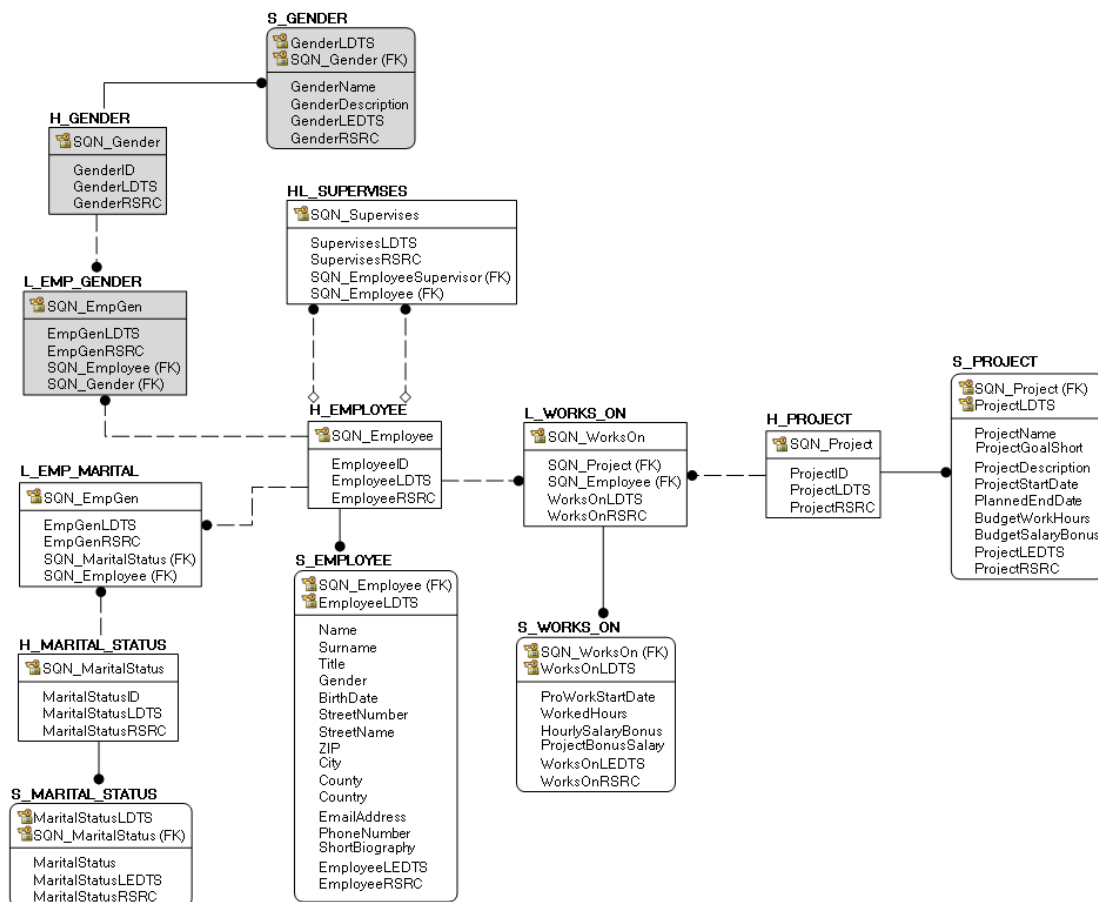
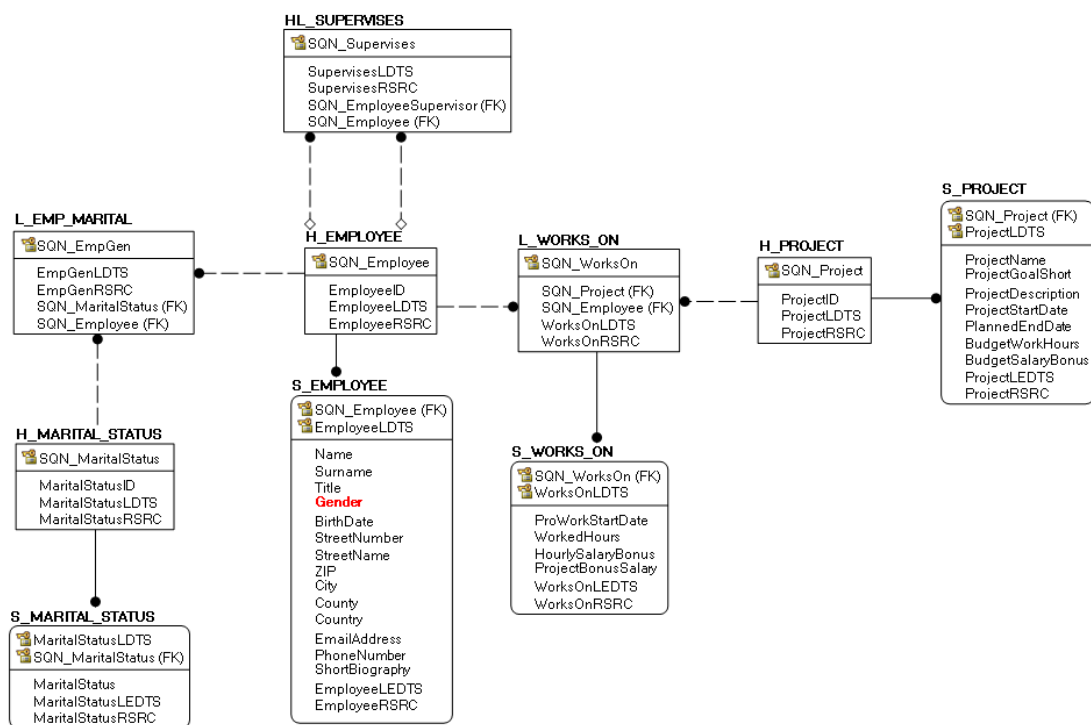


#### 4.3.10. Pretvorba atributa u relaciju, $I_{AR}$

Atribut Gender iz relacije EMPLOYEE želimo kao lookup tablicu izvući u zasebnu relaciju te pojedinoj vrijednosti spola dodijeliti šifru. Ovdje u SDV/PMDV shemi prvo primjenjujemo prethodno opisanu promjenu brisanja atributa iz relacije  $B_A$  nad atributom Gender, a zatim promjenu dodavanja nove relacije  $D_R$  i dodavanja nove veze između relacija  $D_V$  kako bismo dodali novi hub  $H\_GENDER$ , njegov satelit  $S\_GENDER$  te vezu  $L\_EMP\_GENDER$  s  $H\_EMPLOYEE$ . U MDV shemi nema strukturalnih promjena, već se ova promjena pohranjuje na podatkovnoj razini – dodaju se novi zapisi u  $S\_SRC\_ATTRIBUTE$  te  $H\_HUB$ ,  $H\_LINK$ ,  $H\_SATELLITE$  i  $H\_ATTRIBUTE$ , kao i njihove pripadajuće  $S\_DEF$  satelite.



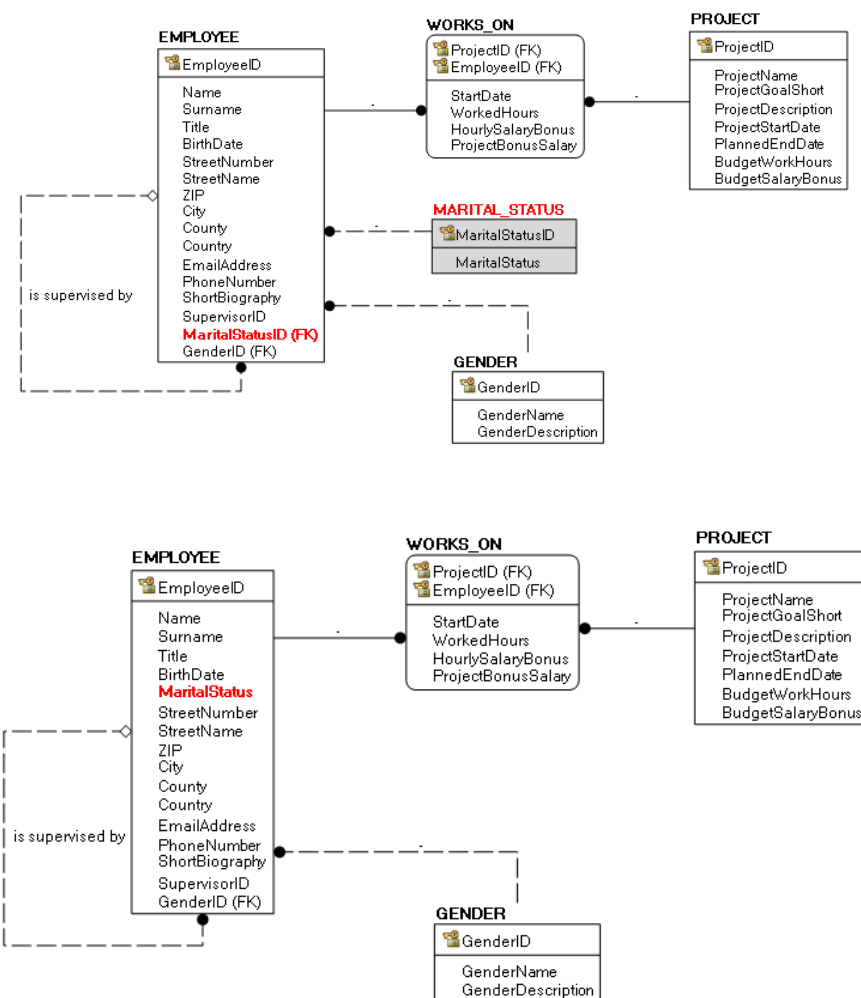
Slika 26. Pretvorba atributa u relaciju  $I_{AR}$  u JobDB shemi



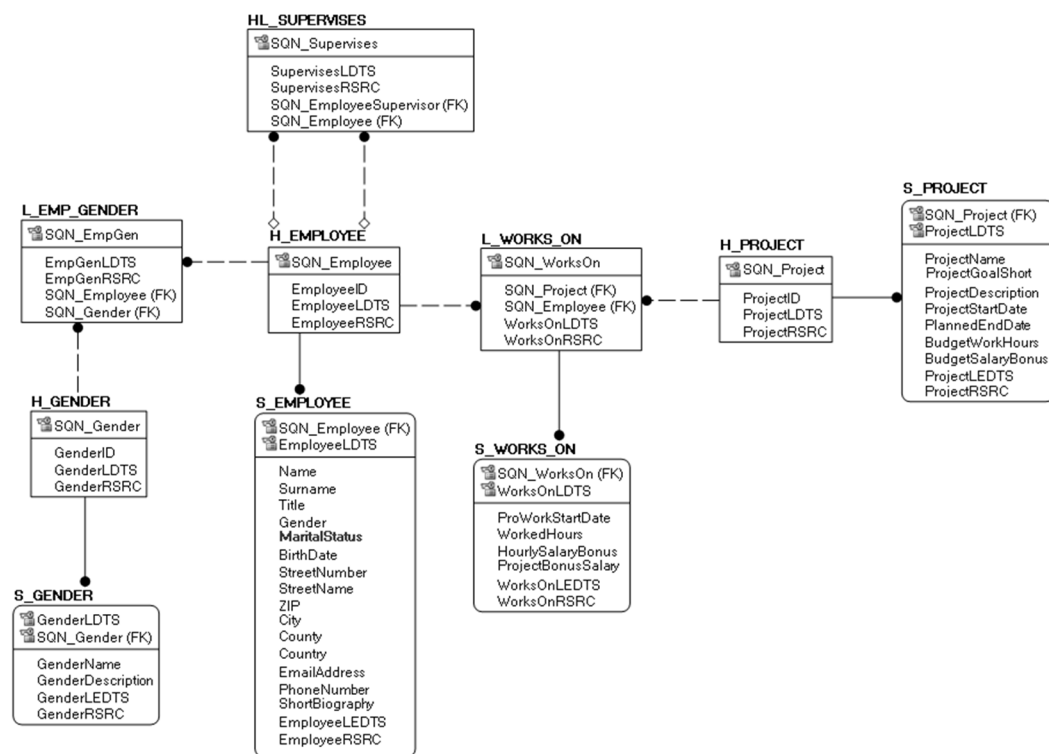
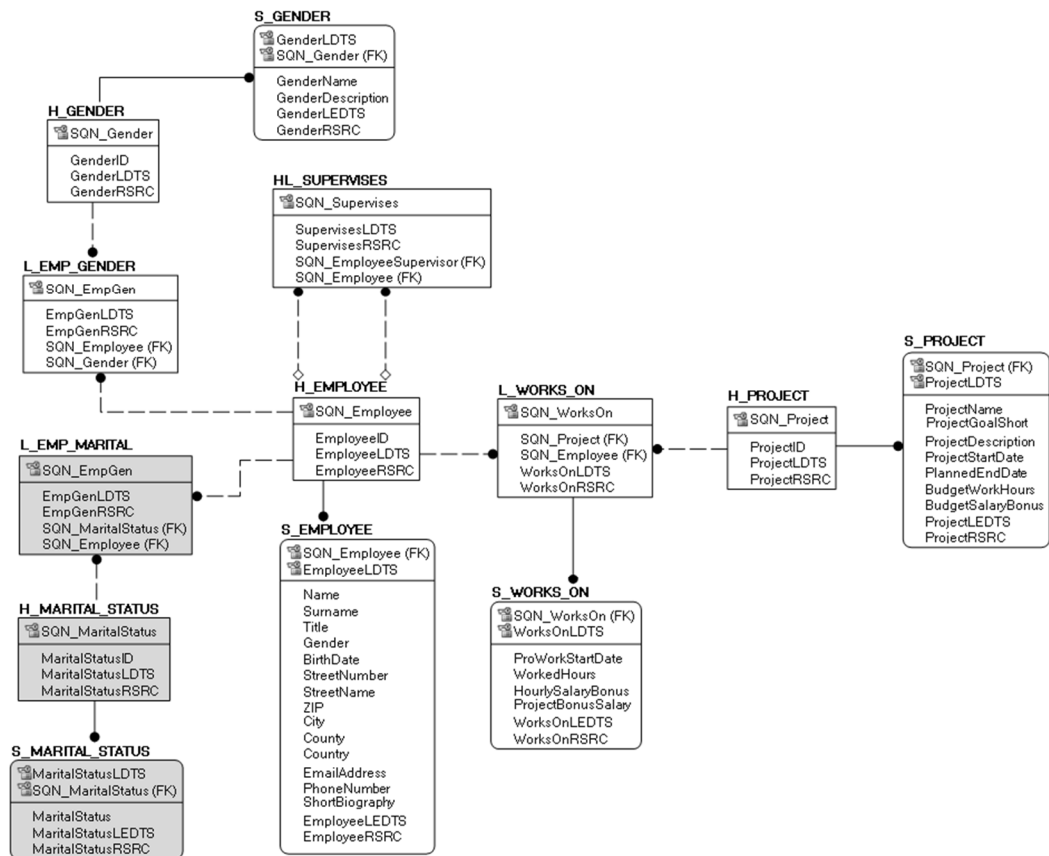
Slika 27. Pretvorba atributa u relaciju I<sub>AR</sub> u SDV shemi

#### 4.3.11. Pretvorba relacije u atribut, I<sub>RA</sub>

Relacija MARITAL\_STATUS ne zahtjeva jedinstvenu šifru za pojedini bračni status, već su dovoljne samo dvije vrijednosti unosa (udana/oženjen ili neudana/neoženjen). Te vrijednosti možemo pohraniti u jednom atributu MaritalStatus unutar relacije EMPLOYEE. Ovdje u SDV/PMDV shemi prvo primjenjujemo promjene brisanja relacije B<sub>R</sub> i brisanja veze između relacija B<sub>V</sub> nad relacijom H\_MARITAL\_STATUS i njenom vezom L\_EMP\_MARITAL s relacijom H\_EMPLOYEE. Zatim primjenjujemo promjenu dodavanja atributa u relaciju D<sub>A</sub> nad relacijom H\_EMPLOYEE – odnosno hubu H\_EMPLOYEE dodajemo novi satelit S\_MARITAL\_STATUS. U MDV shemi nema strukturalnih promjena, već se ova promjena pohranjuje na podatkovnoj razini – dodaju se novi zapisi u H\_SRC\_RELATIONSHIP, H\_SRC\_RELATION, S\_SRC\_RELATION, S\_SRC\_ATTRIBUTE te H\_HUB, H\_LINK, H\_SATELLITE i H\_ATTRIBUTE, kao i njihove pripadajuće S\_DEF satelite.



Slika 28. Pretvorba relacije u atribut I<sub>RA</sub> u JobDB shemi

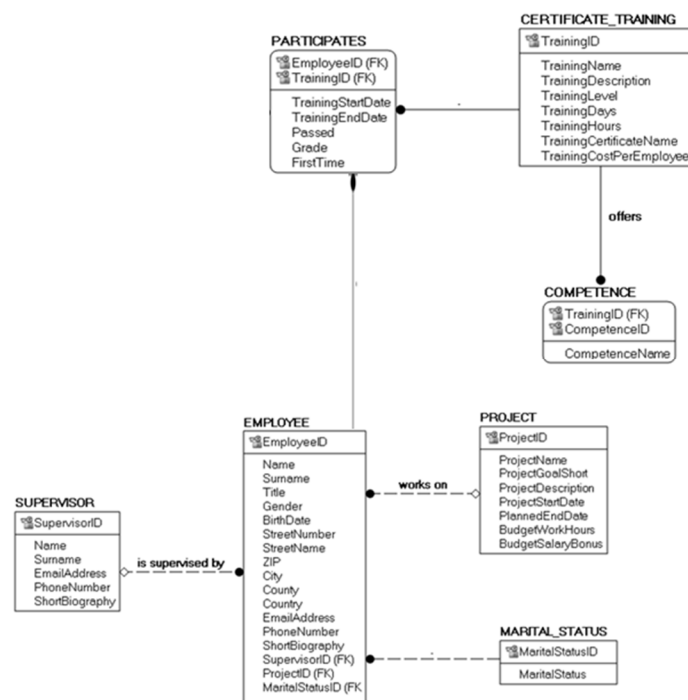


Slika 29. Pretvorba relacije u atribut I<sub>RA</sub> u JobDB shemi

## 4.4. Dokaz hipoteze H3

Hipoteza H3 pretpostavlja da je složenost provedbe evolucije u predloženom rješenju manja nego u tradicionalnom relacijskom pristupu (Poglavlje 1.2).

Složenost provedbe evolucije sheme ocijenit će se kroz prebrojavanje operacija evolucije (dvije definirane operacije, Dodaj<sub>c</sub> i Dodaj<sub>B</sub>) koje su potrebne za svaku prethodno definiranu promjenu sheme (Poglavlje 4.2.1) u našem rješenju te u općenito promatranima tradicionalnim pristupima verzioniranja sheme iz ([6], [7], [12], [20], [25], [26], [50], [55], [65], [72]). Pokazat ćemo da se za svaku osnovnu promjenu u izvorima podataka nad SP/UMP koristi manji ukupni broj operacija evolucije, nego nad tradicionalnim relacijskim 3NF centralnim repozitorijem podataka (za koji se koriste principi klasičnog verzioniranja sheme izradom kopija sheme). Podrazumijevamo da se evolucija u tradicionalnom pristupu vrši kroz najčešće korišteni pristup verzioniranja relacijske sheme, odnosno izrade novih verzija relacijske baze podataka za svaku definiranu promjenu. Za mjerenje ukupnog broja potrebnih operacija koristit ćemo prethodno definirane primjere izvorišnih shema JobDB (Slika 9) i TrainingDB (Slika 10), kao i početnu centralnu SdvDB shemu sa Slike 11. Na Slici 30 nalazi se primjer sheme centralnog 3NF repozitorija, koju ćemo za potrebe usporedbe također promatrati kao prethodno definirani graf (Poglavlje 4.1).



**Slika 30. Početna centralna 3NF shema za integrirani JobDB i TrainingDB**

**4.4.1. Promjene nad 3NF repozitorijem i SDV/PMDV repozitorijem**

Dodavanje nove relacije,  $D_R$  i Dodavanje nove veze između relacija,  $D_V$

U izvorišnu shemu TrainingDB dodajemo novu relaciju Trainer. Da bismo izradili novu verziju sheme 3NF centralnog repozitorija moramo prvo kopirati cijelu prethodnu shemu sa Slike 30 te joj dodati novu relaciju i njenu pripadajuću vezu. Međutim, da bismo izradili novu verziju SDV/PMDV repozitorija, jednostavno nadograđujemo postojeću shemu (što je dokazano u Poglavlju 4.2). Tablice prikazuju ukupan broj potrebnih operacija za provođenje ove promjene nad 3NF centralnim repozitorijem i nad SDV/PMDV podatkovnim repozitorijem. Vidimo kako je broj operacija za provedbu ove promjene nad SDV/PMDV shemom manji nego nad tradicionalnom 3NF shemom.

**Tablica 5. Broj operacija za  $D_R$  i  $D_V$  nad 3NF repozitorijem**

Operacija	Objašnjenje	Ukupno
Dodaj <sub>c</sub>	<ol style="list-style-type: none"> <li>1) Dodavanje čvora tablice Employee</li> <li>2) Dodavanje čvora tablice CertificateTraining</li> <li>3) Dodavanje čvora tablice Participates</li> <li>4) Dodavanje čvora tablice Competence</li> <li>5) Dodavanje čvora tablice Trainer</li> <li>6) Dodavanje čvora tablice Supervisor</li> <li>7) Dodavanje čvora tablice Project</li> <li>8) Dodavanje čvora tablice Marital</li> </ol>	8
Dodaj <sub>B</sub>	<ol style="list-style-type: none"> <li>1) Dodavanje brida veze Employee-Participates</li> <li>2) Dodavanje brida veze CertificateTrainingParticipates</li> <li>3) Dodavanje brida veze CertificateTrainingCompetence</li> <li>4) Dodavanje brida veze CertificateTrainingTrainer</li> <li>5) Dodavanje brida veze EmployeeSupervisor</li> </ol>	7

	6) Dodavanje brida veze EmployeeProject 7) Dodavanje brida veze EmployeeMarital	
<b>UKUPNO</b>		<b>15</b>

**Tablica 6. Broj operacija za D<sub>R</sub> i D<sub>V</sub> nad SDV/PMDV repozitorijem**

<b>Operacija</b>	<b>Objašnjenje</b>	<b>Ukupno</b>
Dodaj <sub>c</sub>	1) Dodavanje hub čvora Trainer 2) Dodavanje satelit čvora Trainer 3) Dodavanje link čvora Leads 4) Dodavanje satelit čvora Leads	4
Dodaj <sub>B</sub>	1) Dodavanje brida TrainerLeads 2) Dodavanje brida CertificateTrainingTrainer 3) Dodavanje brida TrainerSatTrainer 4) Dodavanje brida LeadsSatLeads	4
<b>UKUPNO</b>		<b>8</b>

#### Preimenovanje relacije, I<sub>R</sub>

U shemi izvora podataka TrainingDB relaciju CERTIFICATE\_TRAINING preimenovat ćemo u TRAINING. I u 3NF i u SDV/PMDV shemi nema promjene (jednostavno se mijenja naziv relacije). U 3NF sistemskom katalogu mijenja se postojeći meta-podatkovni zapis (operacijom UPDATE na naziv tablice), a MDV repozitoriju se dodaje novi meta-podatkovni zapis s novim nazivom tablice u odgovarajuće tablice (npr. Prema MDV modelu S\_HUB\_DEF). Obzirom da ova promjena u izvorišnoj shemi ne utječe na promjenu sheme u 3NF i SDV/PMDV repozitorijima, nije relevantna za ovu analizu.

#### Brisanje relacije, B<sub>R</sub> i Brisanje veze između relacija, B<sub>V</sub>

U izvoru podataka TrainingDB briše se relacija TRAINER skupa sa svim svojim vezama (u ovom slučaju veza *leads* na CERTIFICATE\_TRAINING relaciju). U 3NF centralnom repozitoriju izrađuje se nova verzija sheme bez relacije TRAINER, dok u SDV i PMDV shemi nema strukturalne

promjene, već se promjena evidentira kao novi zapis u SDV/PMDV i MDV repozitorijima. Vidimo kako je broj operacija za provedbu ove promjene nad SDV/PMDV shemom manji nego nad tradicionalnom 3NF shemom.

**Tablica 7. Broj operacija za B<sub>R</sub> i B<sub>V</sub> nad 3NF repozitorijem**

Operacija	Objašnjenje	Ukupno
Dodaj <sub>C</sub>	1) Dodavanje čvora tablice Employee 2) Dodavanje čvora tablice CertificateTraining 3) Dodavanje čvora tablice Participates 4) Dodavanje čvora tablice Competence 5) Dodavanje čvora tablice Supervisor 6) Dodavanje čvora tablice Project 7) Dodavanje čvora tablice Marital	7
Dodaj <sub>B</sub>	1) Dodavanje brida veze Employee-Participates 2) Dodavanje brida veze CertificateTrainingParticipates 3) Dodavanje brida veze CertificateTrainingCompetence 4) Dodavanje brida veze EmployeeSupervisor 5) Dodavanje brida veze EmployeeProject 6) Dodavanje brida veze EmployeeMarital	6
<b>UKUPNO</b>		<b>13</b>

**Tablica 8. Broj operacija za B<sub>R</sub> i B<sub>V</sub> nad SDV/PMDV repozitorijem**

Operacija	Objašnjenje	Ukupno
Dodaj <sub>C</sub>		0
Dodaj <sub>B</sub>		0
<b>UKUPNO</b>		<b>0</b>

Dodavanje atributa u relaciju, D<sub>A</sub>

U TrainingDB dodajemo atribut DriversLicence u EMPLOYEE. U SDV/PMDV se dodaje novi satelit S\_DRIVERS\_LICENCE s odgovarajućim atributima (atributom DriversLicence) te veza na



hub H\_EMPLOYEEETRA. U 3NF repozitorij se kreira nova verzija sheme (kopije stare) te joj se u tablicu EMPLOYEE sa svim starim atributima dodaje novi atribut DriversLicence. Vidimo kako je broj operacija za provedbu ove promjene nad SDV/PMDV shemom manji nego nad tradicionalnom 3NF shemom.

**Tablica 9. Broj operacija za D<sub>A</sub> nad 3NF repozitorijem**

Operacija	Objašnjenje	Ukupno
Dodaj <sub>c</sub>	1) Dodavanje čvora tablice Employee 2) Dodavanje čvora tablice CertificateTraining 3) Dodavanje čvora tablice Participates 4) Dodavanje čvora tablice Competence 5) Dodavanje čvora tablice Supervisor 6) Dodavanje čvora tablice Project 7) Dodavanje čvora tablice Marital 8) Dodavanje čvora atributa DriversLicence	8
Dodaj <sub>B</sub>	1) Dodavanje brida veze Employee-Participates 2) Dodavanje brida veze CertificateTrainingParticipates 3) Dodavanje brida veze CertificateTrainingCompetence 4) Dodavanje brida veze EmployeeSupervisor 5) Dodavanje brida veze EmployeeProject 6) Dodavanje brida veze EmployeeMarital	6
<b>UKUPNO</b>		<b>14</b>

**Tablica 10. Broj operacija za D<sub>A</sub> nad SDV/PMDV repozitorijem**

Operacija	Objašnjenje	Ukupno
Dodaj <sub>c</sub>	1) Dodavanje čvora satelita S_DriversLicence	1
Dodaj <sub>B</sub>	1) Dodavanje brida veze huba EmployeeTra i satelita DriversLicence	1
<b>UKUPNO</b>		<b>2</b>

#### Izmjena postojećeg atributa u relaciji – preimenovanje atributa, I<sub>An</sub>

U izvorišnoj shemi atribut Name preimenovan je u First\_Name. Slično kao i kod preimenovanja relacije, i u SDV/PMDV shemi i u 3NF shemi nema promjene, a u MDV i u 3NF sistemskom katalogu sve promjene odvijaju se na podatkovnoj razini. Obzirom da ova promjena u izvorišnoj shemi ne utječe na promjenu sheme u 3NF i SDV/PMDV repozitorijima, nije relevantna za ovu analizu.

#### Izmjena postojećeg atributa u relaciji – promjena tipa podatka atributa, I<sub>At</sub>

U izvorišnoj relaciji CERTIFICATE\_TRAINING atribut TrainingLevel mijenja tip podatka iz NUMERIC u STRING. U SDV/PMDV shemi promjenu implementiramo na način da se u S\_CERTIFICATE\_TRAINING stupcu TrainingLevel direktno promijeni tip podatka (ukoliko je moguće), ili se radi kopija tablice S\_CERTIFICATE\_TRAINING s novim tipom podatka TrainingLevel stupca (koristimo operacije dodavanja nove relacije D<sub>R</sub> i dodavanja veze između relacija D<sub>V</sub>). U 3NF shemi promjenu implementiramo na isti način – tako da se u CERTIFICATE\_TRAINING stupcu TrainingLevel direktno promijeni tip podatka (ukoliko je moguće), ili se radi kopija tablice CERTIFICATE\_TRAINING s novim tipom podatka TrainingLevel stupca (također koristimo operacije dodavanja nove relacije D<sub>R</sub> i dodavanja veze između relacija D<sub>V</sub>). Vidimo kako je broj operacija za provedbu ove promjene nad SDV/PMDV shemom manji nego nad tradicionalnom 3NF shemom.

**Tablica 11. Broj operacija za I<sub>AT</sub> nad 3NF repozitorijem**

Operacija	Objašnjenje	Ukupno
Dodajc	1) Dodavanje čvora tablice Employee 2) Dodavanje čvora tablice Participates 3) Dodavanje čvora tablice Competence 4) Dodavanje čvora tablice Supervisor 5) Dodavanje čvora tablice Project 6) Dodavanje čvora tablice Marital 7) Dodavanje čvora nove tablice CertificateTraining	7
	1) Dodavanje brida veze Employee-Participates	6

Dodaj <sub>B</sub>	2) Dodavanje brida veze CertificateTrainingCompetence 3) Dodavanje brida veze EmployeeSupervisor 4) Dodavanje brida veze EmployeeProject 5) Dodavanje brida veze EmployeeMarital 6) Dodavanje brida nove veze CertificateTrainingParticipates	
<b>UKUPNO</b>		<b>13</b>

**Tablica 12. Broj operacija za I<sub>AT</sub> nad SDV/PMDV repozitorijem**

Operacija	Objašnjenje	Ukupno
Dodaj <sub>C</sub>	1) Dodavanje čvora novog satelita S_CertificateTraining	1
Dodaj <sub>B</sub>	1) Dodavanje brida veze huba H_CertificateTraining i novog satelita S_CertificateTraining	1
<b>UKUPNO</b>		<b>2</b>

Brisanje atributa iz relacije, B<sub>A</sub>

Iz relacije EMPLOYEE izbrisat ćemo atribut DriversLicence. U 3NF centralnom repozitoriju izrađuje se nova verzija sheme bez atributa DriversLicence u relaciji Employee, dok u SDV i PMDV shemi nema strukturalne promjene, već se promjena evidentira kao novi zapis u SDV/PMDV i MDV repozitorijima. Vidimo kako je broj operacija za provedbu ove promjene nad SDV/PMDV shemom manji nego nad tradicionalnom 3NF shemom.

**Tablica 13. Broj operacija za B<sub>A</sub> nad 3NF repozitorijem**

Operacija	Objašnjenje	Ukupno
Dodaj <sub>C</sub>	1) Dodavanje čvora nove tablice Employee 2) Dodavanje čvora tablice CertificateTraining 3) Dodavanje čvora tablice Participates	7

	4) Dodavanje čvora tablice Competence 5) Dodavanje čvora tablice Supervisor 6) Dodavanje čvora tablice Project 7) Dodavanje čvora tablice Marital	
Dodaj <sub>B</sub>	1) Dodavanje brida veze Employee-Participates 2) Dodavanje brida veze CertificateTrainingParticipates 3) Dodavanje brida veze CertificateTrainingCompetence 4) Dodavanje brida veze EmployeeSupervisor 5) Dodavanje brida veze EmployeeProject 6) Dodavanje brida veze EmployeeMarital	6
<b>UKUPNO</b>		<b>13</b>

**Tablica 14. Broj operacija za B<sub>A</sub> nad SDV/PMDV repozitorijem**

Operacija	Objašnjenje	Ukupno
Dodaj <sub>C</sub>		0
Dodaj <sub>B</sub>		0
<b>UKUPNO</b>		<b>0</b>

#### Pretvorba relacije u vezu, I<sub>RV</sub>

Relaciju SUPERVISOR ćemo u izvoru podataka pretvoriti u rekurzivnu vezu *is supervised by* relacije EMPLOYEE. U SDV/PMDV na relaciju SUPERVISOR primjenjujemo prethodno opisanu promjenu brisanja relacije B<sub>R</sub> i brisanja veze između relacija B<sub>V</sub>. Zatim primjenjujemo promjenu dodavanja veze između relacija D<sub>V</sub> nad hubom H\_EMPLOYEE koja će kreirati novi hijerarhijski link HL\_SUPERVISES te ga vezati s dvostrukom vezom na H\_EMPLOYEE. U 3NF u novoj verziji sheme brišemo relaciju SUPERVISOR te na relaciju EMPLOYEE dodajemo rekurzivnu vezu *isSupervisedBy*. Vidimo kako je broj operacija za provedbu ove promjene nad SDV/PMDV shemom manji nego nad tradicionalnom 3NF shemom.

**Tablica 15. Broj operacija za I<sub>RV</sub> nad 3NF repozitorijem**

Operacija	Objašnjenje	Ukupno
Dodaj <sub>c</sub>	1) Dodavanje čvora tablice Employee 2) Dodavanje čvora tablice CertificateTraining 3) Dodavanje čvora tablice Participates 4) Dodavanje čvora tablice Competence 5) Dodavanje čvora tablice Project 6) Dodavanje čvora tablice Marital	6
Dodaj <sub>B</sub>	1) Dodavanje brida veze Employee-Participates 2) Dodavanje brida veze CertificateTrainingParticipates 3) Dodavanje brida veze CertificateTrainingCompetence 4) Dodavanje brida veze EmployeeProject 5) Dodavanje brida veze EmployeeMarital 6) Dodavanje brida rekurzivne veze EmployeeSupervises	6
<b>UKUPNO</b>		<b>12</b>

**Tablica 16. Broj operacija za I<sub>RV</sub> nad SDV/PMDV repozitorijem**

Operacija	Objašnjenje	Ukupno
Dodaj <sub>c</sub>	1) Dodavanje čvora hijerarhijskog linka HL_Supervises	1
Dodaj <sub>B</sub>	1) Dodavanje 1.brida veze huba H_Employee i linka HL_Supervises 2) Dodavanje 2.brida veze huba H_Employee i linka HL_Supervises	2
<b>UKUPNO</b>		<b>3</b>

Pretvorba veze u relaciju, I<sub>VR</sub>

U JobDB shemi izvorišnu vezu *works\_on* pretvaramo u relaciju WORKS\_ON koja će biti povezana s relacijama EMPLOYEE i PROJECT. U SDV/PMDV shemi na vezu *works\_on* primjenjujemo prethodno opisanu promjenu brisanja veze između relacija B<sub>v</sub>. Zatim

primjenjujemo promjene dodavanja nove relacije  $D_R$  i dodavanja nove veze između relacija  $D_V$ , kako bismo dodali izvorišnu relaciju WORKS\_ON (kao link i njegov pripadajući satelit) te je vezali na relacije EMPLOYEE i PROJECT. U 3NF shemi na novoj verziji sheme dodajemo relaciju WORKS\_ON s pripadajućim atributima, te veze te relacije s relacijom EMPLOYEE i s relacijom PROJECT. Vidimo kako je broj operacija za provedbu ove promjene nad SDV/PMDV shemom manji nego nad tradicionalnom 3NF shemom.

**Tablica 17. Broj operacija za  $I_{VR}$  nad 3NF repozitorijem**

Operacija	Objašnjenje	Ukupno
Dodaj <sub>c</sub>	1) Dodavanje čvora tablice Employee 2) Dodavanje čvora tablice Supervisor 3) Dodavanje čvora tablice CertificateTraining 4) Dodavanje čvora tablice Participates 5) Dodavanje čvora tablice Competence 6) Dodavanje čvora tablice Project 7) Dodavanje čvora tablice Marital 8) Dodavanje čvora tablice Works_On	8
Dodaj <sub>B</sub>	1) Dodavanje brida veze EmployeeParticipates 2) Dodavanje brida veze EmployeeSupervisor 3) Dodavanje brida veze CertificateTrainingParticipates 4) Dodavanje brida veze CertificateTrainingCompetence 5) Dodavanje brida veze EmployeeMarital 6) Dodavanje brida veze EmployeeWorksOn 7) Dodavanje brida veze ProjectWorksOn	7
<b>UKUPNO</b>		<b>15</b>

**Tablica 18. Broj operacija za  $I_{VR}$  nad SDV/PMDV repozitorijem**

Operacija	Objašnjenje	Ukupno
Dodaj <sub>c</sub>	1) Dodavanje čvora linka L_WorksOn 2) Dodavanje čvora satelita S_WorksOn	2

Dodaj <sub>B</sub>	1) Dodavanje brida veze huba H_Project i linka L_WorksOn 2) Dodavanje brida veze huba H_Employee i linka L_WorksOn 3) Dodavanje brida veze linka L_WorksOn i satelita S_WorksOn	3
<b>UKUPNO</b>		<b>5</b>

#### Pretvorba atributa u relaciju, I<sub>AR</sub>

Atribut Gender iz relacije EMPLOYEE ćemo izvući u zasebnu relaciju te pojedinoj vrijednosti spola dodijeliti šifru. U SDV/PMDV shemi prvo primjenjujemo prethodno opisanu promjenu brisanja atributa iz relacije B<sub>A</sub> nad atributom Gender, a zatim promjenu dodavanja nove relacije D<sub>R</sub> i dodavanja nove veze između relacija D<sub>V</sub> kako bismo dodali novi hub H\_GENDER, njegov satelit S\_GENDER te vezu L\_EMP\_GENDER s H\_EMPLOYEE. U 3NF shemi kreiramo novu verziju sheme u kojoj se u relaciji EMPLOYEE briše atribut Gender, dodaje se nova relacija Gender s pripadajućim atributima te se dodaje veza između relacije GENDER i EMPLOYEE. Vidimo kako je broj operacija za provedbu ove promjene nad SDV/PMDV shemom manji nego nad tradicionalnom 3NF shemom.

**Tablica 19. Broj operacija za I<sub>AR</sub> nad 3NF repozitorijem**

Operacija	Objašnjenje	Ukupno
Dodaj <sub>C</sub>	1) Dodavanje čvora tablice Employee 2) Dodavanje čvora tablice Supervisor 3) Dodavanje čvora tablice CertificateTraining 4) Dodavanje čvora tablice Participates 5) Dodavanje čvora tablice Competence 6) Dodavanje čvora tablice Project 7) Dodavanje čvora tablice Marital 8) Dodavanje čvora tablice Gender	8
Dodaj <sub>B</sub>	1) Dodavanje brida veze EmployeeParticipates 2) Dodavanje brida veze EmployeeSupervisor	6

	3) Dodavanje brida veze CertificateTrainingParticipates 4) Dodavanje brida veze CertificateTrainingCompetence 5) Dodavanje brida veze EmployeeMarital 6) Dodavanje brida veze EmployeeGender	
<b>UKUPNO</b>		<b>14</b>

**Tablica 20. Broj operacija za I<sub>AR</sub> nad SDV/PMDV repozitorijem**

<b>Operacija</b>	<b>Objašnjenje</b>	<b>Ukupno</b>
Dodaj <sub>c</sub>	1) Dodavanje čvora linka L_EmpGender 2) Dodavanje čvora huba H_Gender 3) Dodavanje čvora satelita S_Gender	3
Dodaj <sub>B</sub>	1) Dodavanje brida veze huba H_Employee i linka L_EmpGender 2) Dodavanje brida veze huba H_Gender i linka L_EmpGender 3) Dodavanje brida veze huba H_Gender i satelita S_Gender	3
<b>UKUPNO</b>		<b>6</b>

#### Pretvorba relacije u atribut, I<sub>RA</sub>

Relacija MARITAL\_STATUS pretvara se u izvorišnoj shemi JobDB u atribut MaritalStatus unutar relacije EMPLOYEE. U SDV/PMDV shemi prvo primjenjujemo promjene brisanja relacije B<sub>R</sub> i brisanja veze između relacija B<sub>V</sub> nad relacijom S\_MARITAL\_STATUS, H\_MARITAL\_STATUS i njenom vezom L\_EMP\_MARITAL s relacijom H\_EMPLOYEE. Zatim primjenjujemo promjenu dodavanja atributa u relaciju D<sub>A</sub> nad relacijom H\_EMPLOYEE – odnosno hubu H\_EMPLOYEE dodajemo novi satelit S\_MARITAL\_STATUS. U 3NF shemi kreiramo novu verziju sheme u kojoj prvo brišemo relacije MARITAL\_STATUS i EMPLOYEE, a zatim dodajemo relaciju EMPLOYEE sa svim starim atributima i novim atributom MaritalStatus. Vidimo kako je broj operacija za provedbu ove promjene nad SDV/PMDV shemom manji nego nad tradicionalnom 3NF shemom.



**Tablica 21. Broj operacija za I<sub>RA</sub> nad 3NF repozitorijem**

Operacija	Objašnjenje	Ukupno
Dodaj <sub>c</sub>	1) Dodavanje čvora tablice Employee 2) Dodavanje čvora tablice Supervisor 3) Dodavanje čvora tablice CertificateTraining 4) Dodavanje čvora tablice Participates 5) Dodavanje čvora tablice Competence 6) Dodavanje čvora tablice Project 7) Dodavanje čvora tablice Gender 8) Dodavanje čvora atributa MaritalStatus	8
Dodaj <sub>B</sub>	1) Dodavanje brida veze EmployeeParticipates 2) Dodavanje brida veze EmployeeSupervisor 3) Dodavanje brida veze CertificateTrainingParticipates 4) Dodavanje brida veze CertificateTrainingCompetence 5) Dodavanje brida veze EmployeeGender	5
<b>UKUPNO</b>		<b>13</b>

**Tablica 22. Broj operacija za I<sub>RA</sub> nad SDV/PMDV repozitorijem**

Operacija	Objašnjenje	Ukupno
Dodaj <sub>c</sub>	1) Dodavanje čvora satelita S_MaritalStatus	1
Dodaj <sub>B</sub>	1) Dodavanje brida veze huba H_Employee i satelita S_MaritalStatus	1
<b>UKUPNO</b>		<b>2</b>

Kao što je prethodno pokazano, za provedbu svake promjene iz formalnog skupa promjena u SDV/PMDV shemi je potreban manji ukupni broj operacija evolucije, u odnosu na tradicionalnu relacijsku 3NF centralnu shemu. Prethodno dokazano možemo poopćiti na evoluciju sheme MDV repozitorija u odnosu na tradicionalni relacijski sistemski katalog. Mijenjamo samo razinu podataka na razinu meta-podataka na kojoj promatramo modele te na temelju dokazanog u Poglavlju 4.2 zaključujemo da vrijede isti dokazi. Ovime je dokazana

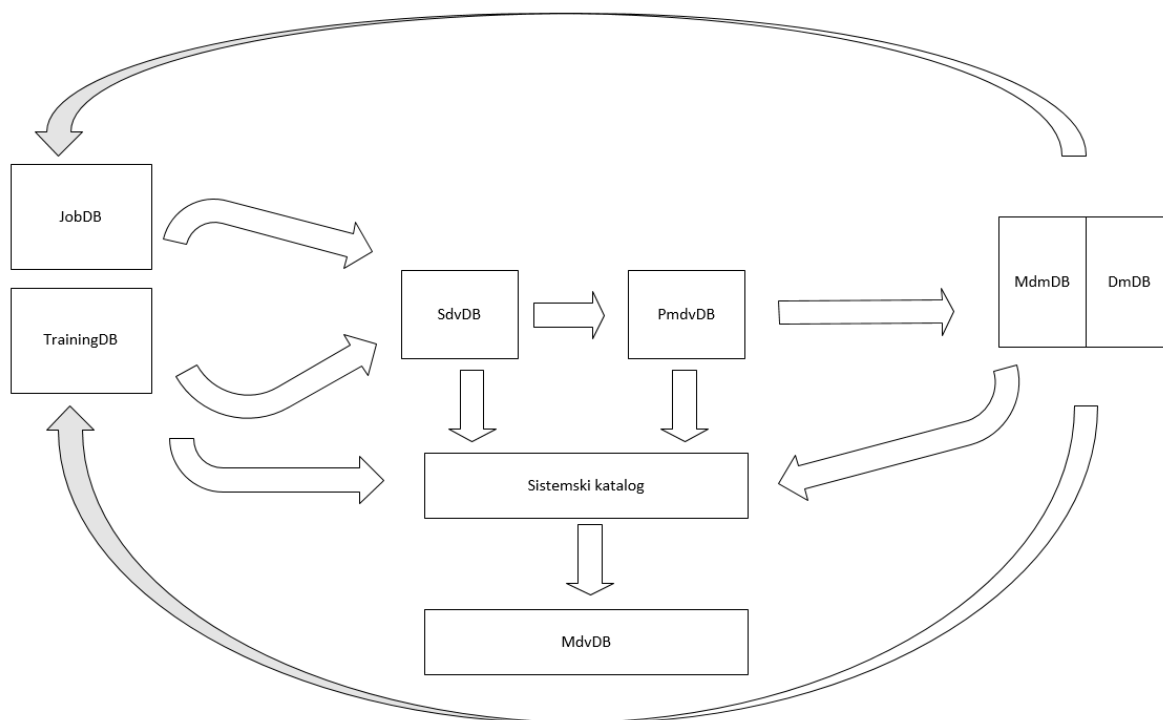
hipoteza H3. Dodatno, korisnost predloženog modela i programskog rješenja u vidu MDV repozitorija meta-podataka u odnosu na postojeći sistemski katalog i prikazana je u Poglavlju 5 (kroz izvođenje upita nad MDV repozitorijem i sistemskim katalogom).

## **5. EMPIRIJSKA VALIDACIJA ISTRAŽIVANJA**

U ovome poglavlju predstavljena je i opisana trenutna verzija prototipa rješenja koja služi za empirijsku validaciju istraživanja te dodatni prikaz korisnosti predloženog pristupa.

### ***5.1. Opis prototipa rješenja***

Prototip rješenja izrađen je prema prikazanim primjerima iz Poglavlja 4.3 te je testiran kroz provođenje skupa upita nad SDV, PMDV i MDV bazama podataka, kao i sistemskim katalogom sustava u kojemu se sve ove baze podataka nalaze. Na Slici 31 prikazana je arhitektura prototipa. Možemo vidjeti da postoje dvije izvorišne baze podataka JobDB i TrainingDB koje se zatim integriraju u sirovu kopiju izvorišnih podataka SdvDB, kreiranu prema prethodno opisanom modelu i metodama. Na temelju SdvDB izgrađuje se i puni PmdvDB koja čuva „pročišćene“ podatke iz SdvDB. PmdvDB zatim puni MdmDB koja praktično predstavlja UMP te čuva matične podatke. U MdmDB se u principu nalaze isti podaci kao i u PmdvDB, ali organizirani prema dimenzijskom modelu [43] – u dimenzijsku bazu podataka. MdmDB dalje puni DmDB, koja se također zasniva na dimenzijskom modelu te praktično predstavlja područno SP. Na ovaj način integrirani su UMP i SP - MdmDB služi kao osnova matičnih podataka o zaposlenicima za punjenje područnog SP te ima mogućnost (prema definiranoj arhitekturi iz Poglavlja 3.1, a za implementaciju u budućnosti) vraćanja „zlatne kopije“ matičnih podataka o zaposlenicima natrag u izvore podataka, ovisno o načinu implementacije UMP sustava (opisanima u Poglavlju 2.3).



**Slika 31. Dijagram arhitekture prototipa**

Prototip, odnosno sve baze podataka koje sadrži, nalaze se na istom Windows 10 Education x64 operacijskom sustavu te su izrađene u Microsoft SQL Server 2012 [54] sustavu za upravljanje bazama podataka, uz korištenje Microsoft SQL Server Integration Services 2012 (SSIS) i SQL Server Data Tool for Visual Studio 2012 za punjenje i transformaciju podataka između navedenih baza podataka. Za generiranje podataka kojima su inicijalno napunjene izvorišne baze podataka JobDB i TrainingDB korišten je web alat FreeDataGenerator [24]. SQL skripte za kreiranje i punjenje navedenih baza podataka nalaze se u Privitku 4.

Treba istaknuti da je u radu prikazana trenutna verzija prototipa, koja ne uključuje implementaciju cjelovitog MDV modela za repozitorij meta-podataka. U trenutnoj verziji prototipa u MdvDB repozitorij implementirani su dijelovi MDV modela koji se odnose na izvore podataka i centralni SdvDB/PmdvDB repozitorij, a za budućnost ostaje implementacija dijelova koji se odnose na područna SP i sigurnost podataka. Međutim, kao što će biti prikazano, trenutna implementirana verzija prototipa dovoljna je za empirijsku validaciju predloženog pristupa. Također, sve buduće implementacije zasnivat će se isključivo na nadogradnji trenutne verzije.

### 5.1.1. ETL za integraciju i punjenje SDV, PMDV i MDV repozitorija

U ovome poglavlju bit će prikazani i opisani postupci punjenja i transformacije navedenih baza podataka iz prototipa, kroz opise Microsoft SSIS ekrana za izradu toka kontrole (*eng. control flow*) i toka podataka (*eng. data flow*).

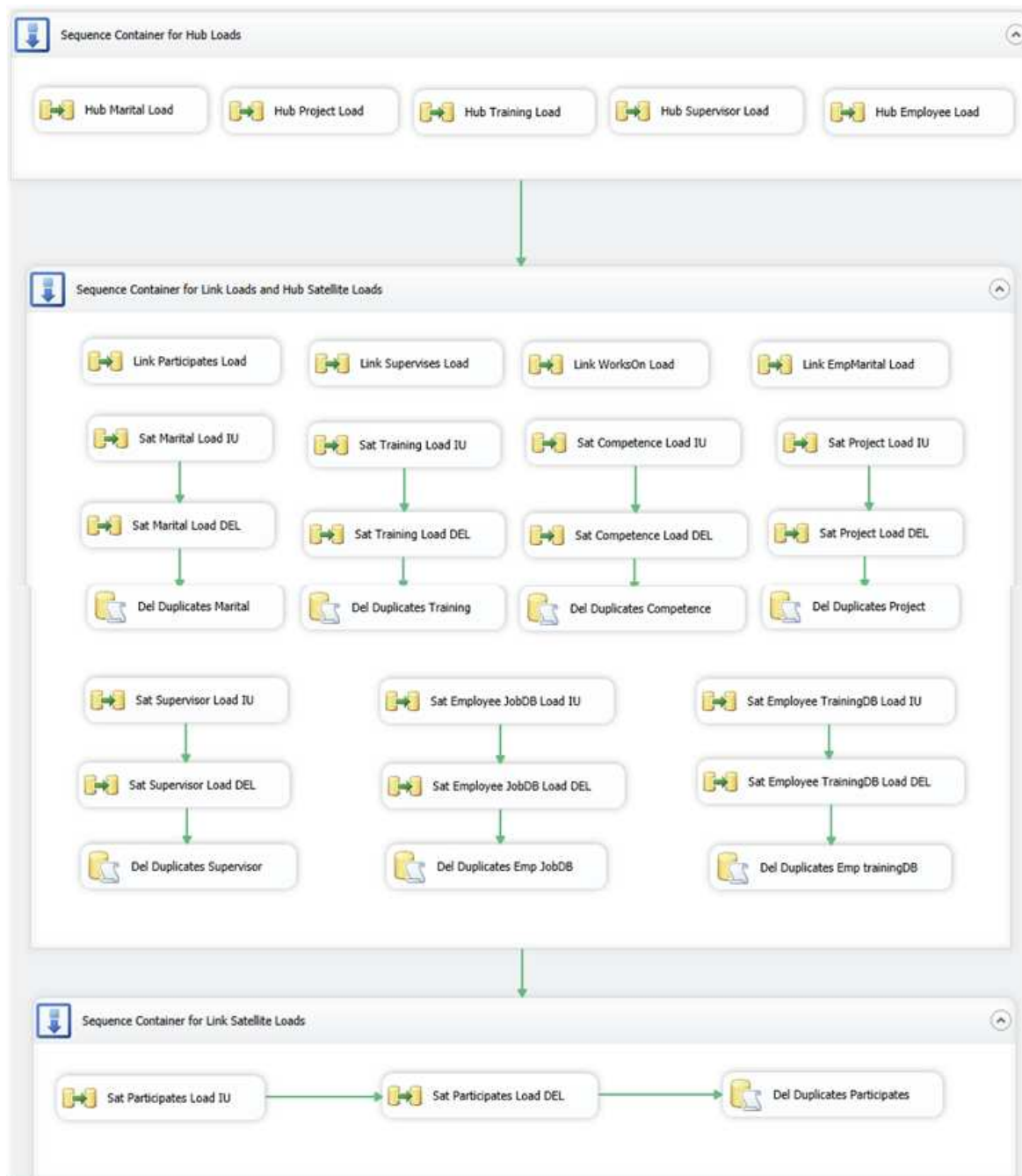
Na Slici 32 vidimo da su za punjenje SdvDB iz JobDB i TrainingDB izvora podataka potrebna tri ciklusa punjenja:

1. Punjenje Hubova
2. Punjenje Linkova i Hub-Satelita
3. Punjenje Link-Satelita

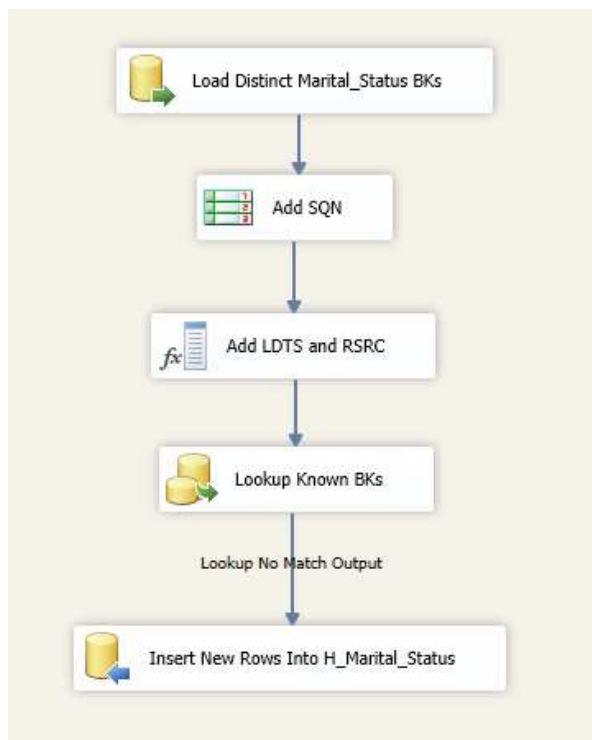
Ovo je iz razloga što je prvo potrebno za svaki jedinstveni poslovni ključ iz izvora podataka generirati surogat ključ te ga pohraniti u Hub. Nakon toga, pune se Linkovi za koje se također generira surogat ključ za svaku jedinstvenu kombinaciju Hubovih surogat ključeva (koji se pohranjuju u Link kao vanjski ključevi). Paralelno se pune i Hub-Sateliti kojima treba surogat ključ Huba kako bi s meta-podatkom LDTs (datum i vrijeme punjenja podatka u Satelit, *eng. load\_datetimestamp*) kreirao primarni ključ Satelita. I na kraju se pune Link-Sateliti, jer oni traže surogat ključ linka kako bi sa svojim LDTs kreirali primarni ključ Link-Satelita.

Na Slici 33 vidimo SSIS tok podataka za punjenje svih Hubova u SdvDB, na primjeru toka podataka za H\_Marital\_Status. Koraci za punjenje Huba u SdvDB su sljedeći:

1. Pronaći popis svih različitih poslovnih ključeva (Load Distinct Marital\_Status BKs)
2. Generirati surogat ključ za poslovni ključ (Add SQN)
3. Dodati DV meta-podatke, LDTs (datum i vrijeme punjenja) te RSRC (izvor podatka) (Add LDTs and RSRC)
4. Ako poslovni ključ već postoji u hubu, izbaciti ga iz popisa (Lookup Known BKs, No match Output)
5. Unijeti retke u ciljani Hub (Insert New Rows Into H\_Marital\_Status)

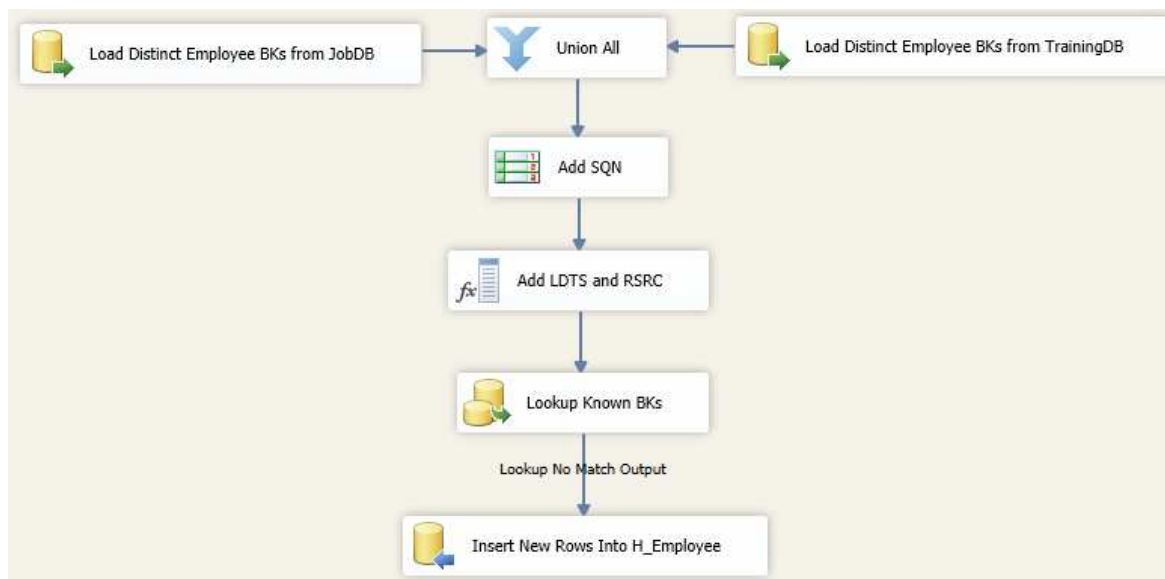


Slika 32. SSIS tok kontrole za punjenje SdvDB iz izvora



**Slika 33. SSIS tok podataka za punjenje SdvDB Huba (primjer H\_Marital\_Status)**

Slika 34 dodatno prikazuje nešto drugačiji SSIS tok podataka za punjenje huba H\_Employee. Ovdje u prvom koraku radimo uniju poslovnih ključeva iz dva izvora (jer Employee zapisi postoje i u JobDB i u TrainingDB) te ih dalje na isti način punimo u jedinstveni hub H\_Employee u SdvDB.

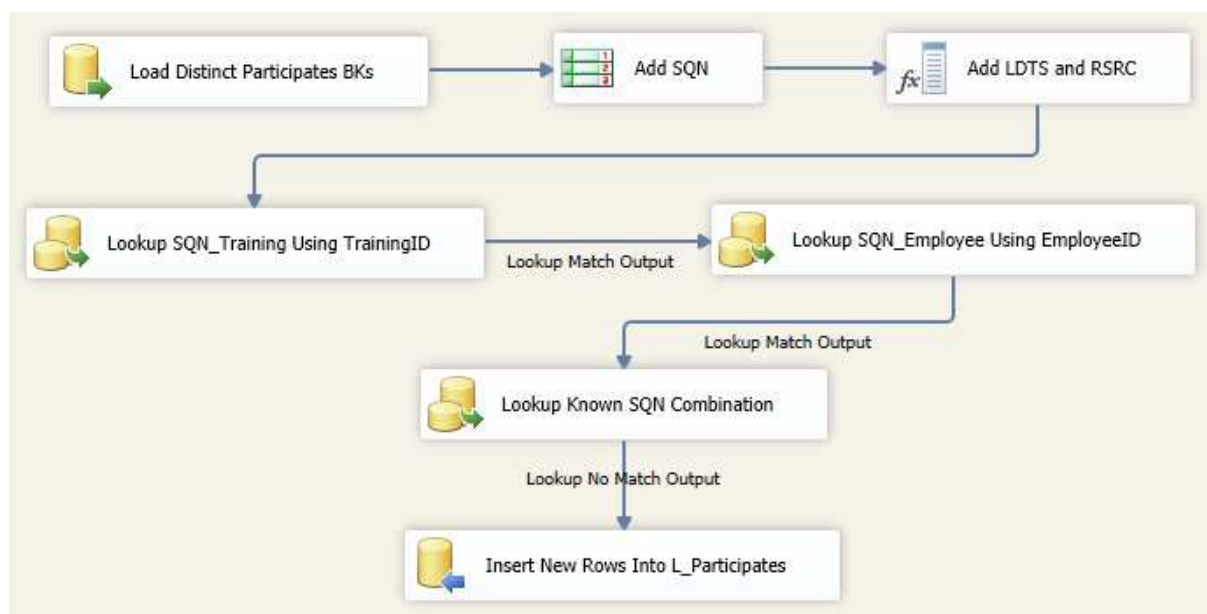


**Slika 34. SSIS tok podataka za punjenje SdvDB Huba (primjer H\_Employee)**

Slika 35 prikazuje SSIS tok podataka za punjenje Linkova, na primjeru linka L\_Participates.

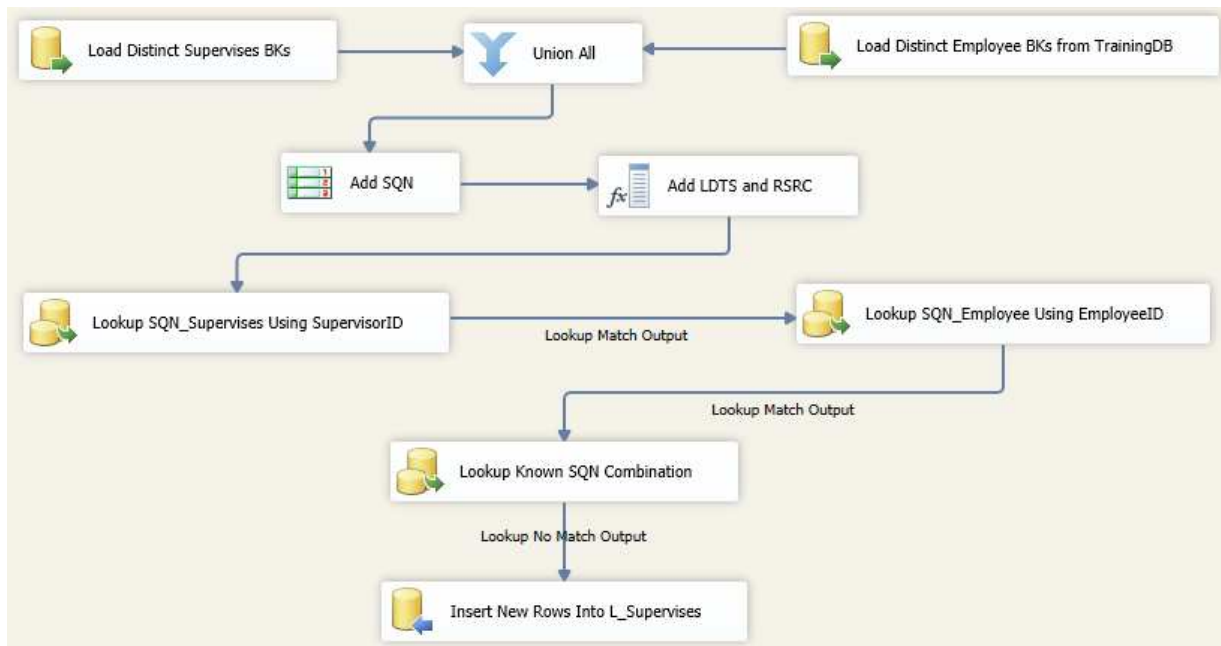
Koraci za punjenje Linka u SdvDB su sljedeći:

1. Pronaći popis svih različitih poslovnih ključeva Linka (Load Distinct Participates BKs)
2. Generirati surogat ključ za poslovni ključ (Add SQN)
3. Dodati DV meta-podatke, LDTs (datum i vrijeme punjenja) te RSRC (izvor podatka) (Add LDTs and RSRC)
4. Provjeriti surogat ključeve svih Hubova koji sudjeluju u Linku (Lookup SQN\_Training Using TrainingID, Lookup SQN\_Employee Using EmployeeID)
5. Provjeriti sve poznate kombinacije Hubovih surogat ključeva (Lookup Known SQN Combination)
6. Ako poslovni ključ već postoji u hubu, izbaciti ga iz popisa (Lookup No Match Output)
7. Unijeti retke u ciljni Link (Insert New Rows Into L\_Participates)



**Slika 35. SSIS tok podataka za punjenje SdvDB Linka (primjer L\_Participates)**

Na Slici 36 također je prikazan tok podataka za punjenje Linka (L\_Supervises), ali ovaj put s unijom iz JobDB i TrainingDB skupa podataka u prvom koraku. Ostali koraci su identični kao na Slici 35.



**Slika 36. SSIS tok podataka za punjenje SdvDB Linka (primjer L\_Supervises)**

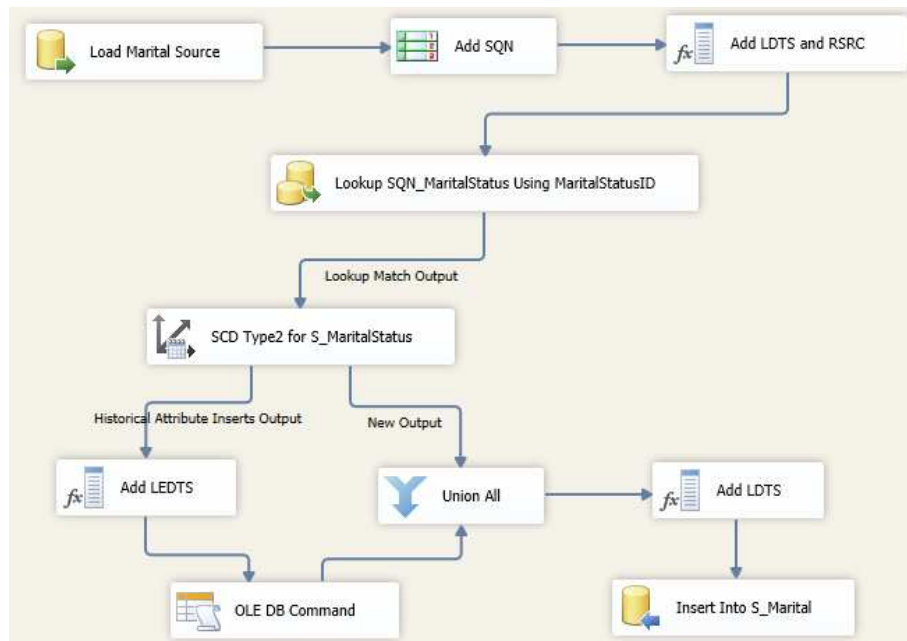
Slike 37 i 38 prikazuju SSIS tok podataka za punjenje Satelita u SdvDB, na primjeru satelita S\_MaritalSource. Koraci za punjenje Satelita su sljedeći (Slika 37):

1. Pronaći sve različite Satelit zapise (Load Marital Source)
2. Generirati surogat ključ za poslovni ključ (Add SQN)
3. Dodati DV meta-podatke, LDTS (datum i vrijeme punjenja) te RSRC (izvor podatka) (Add LDTS and RSRC)
4. Provjeriti surogat ključ vezanog Huba (Lookup SQN\_MaritalStatus Using MaritalStatusID)
5. Provjeriti postoji li Satelit zapis već u Satelitu; ako NE unijeti novi zapis u Satelit; ako DA provjeriti je li duplikat;; ako DA izbaciti ga s popisa za unos;; ako NE unijeti novi zapis, a na prethodni zapis s tim SQN postaviti vrijednost kraja važenja, LEDTS (SCD Type2 for S\_MaritalStatus)
6. Unijeti retke u ciljani Link (Insert Into S\_Marital)

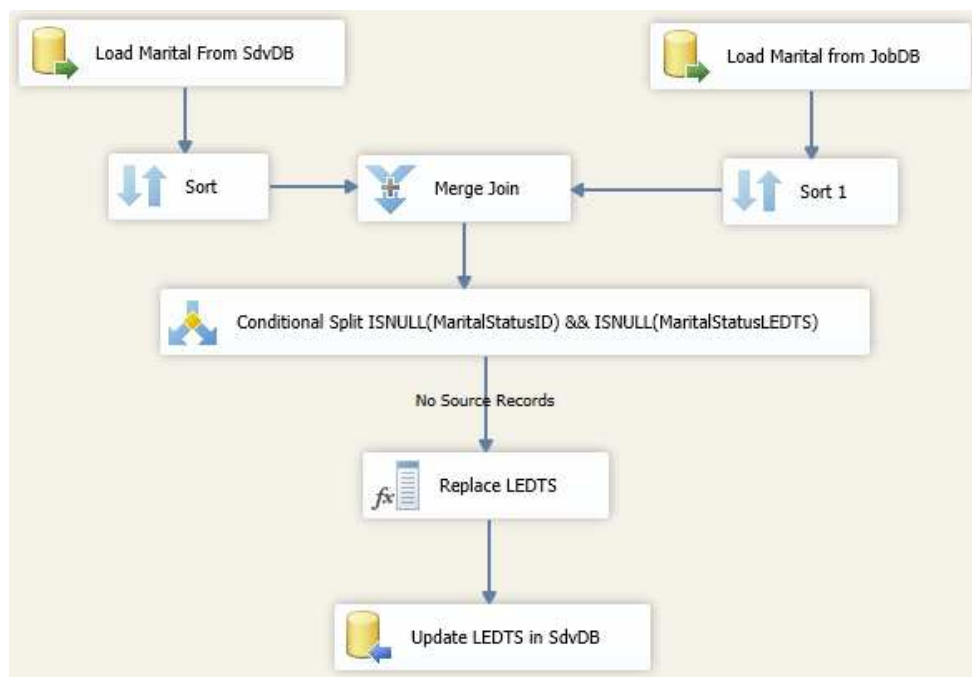
Dodatno, nakon provedenog toka podataka sa Slike 37 (koji ubacuje u Satelit retke koji su u izvoru podataka novi ili izmijenjeni), slijedi izvršavanje toka podataka sa Slike 38, koji provjerava postoje li u Satelitu retci koji su u izvoru u međuvremenu izbrisani – ako postoje



na taj zapis se u Satelitu dodaje vrijednost kraja važenja, LEDTS (Conditional Split, Replace LEDTS).



**Slika 37. SSIS tok podataka za punjenje i ažuriranje SdvDB Satelita**



**Slika 38. SSIS tok podataka za ažuriranje obrisanih redaka SdvDB Satelita**

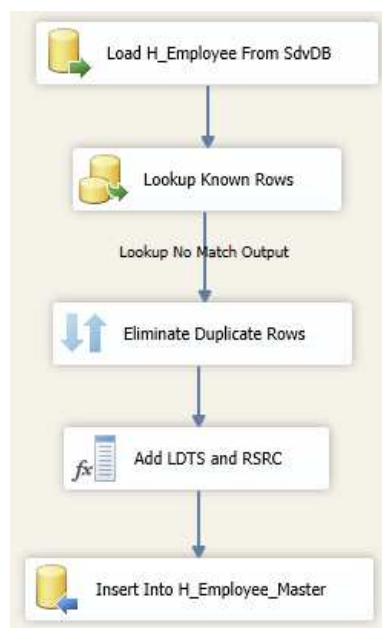
U nastavku slijedi nekoliko prikaza toka podataka za punjenje PmdvDB iz SdvDB. Početni tok kontrole nije prikazan, jer je identičan toku kontrole sa Slike 32 (PmdvDB se puni u tri ciklusa,

kao i SdvDB). I tokovi podataka za punjenje Hubova, Linkova i Satelita su kreirani prema istim koracima koji su prethodno opisani za SdvDB te će iz toga razloga biti prikazani samo neki tokovi podataka koji imaju dodatne transformacije specifične za PmdvDB.

Obzirom da PmdvDB predstavlja repozitorij u kojem su podaci integrirani, standardizirani, te prilagođeni određenim poslovnim pravilima (ovdje se vrše „teške“ transformacije), za potrebe izgradnje prototipa izveli smo 3 takve transformacije u PmdvDB:

- a. Eliminacija duplih redaka za sve Hubove, Linkove i Satelite
- b. Integracija dva zasebna satelita iz SdvDB (S\_Employee i S\_EmployeeTra) u jedan satelit u PmdvDB
- c. Primjena poslovnog pravila nad satelitom S\_Project (ProjectStartDate < ProjectEndDate)

Na Slici 39. vidimo primjer SSIS toka podataka za punjenje H\_EmployeeMaster huba u PmdvDB, s primjenjenom transformacijom eliminacije duplih redaka (Eliminate Duplicate Rows).

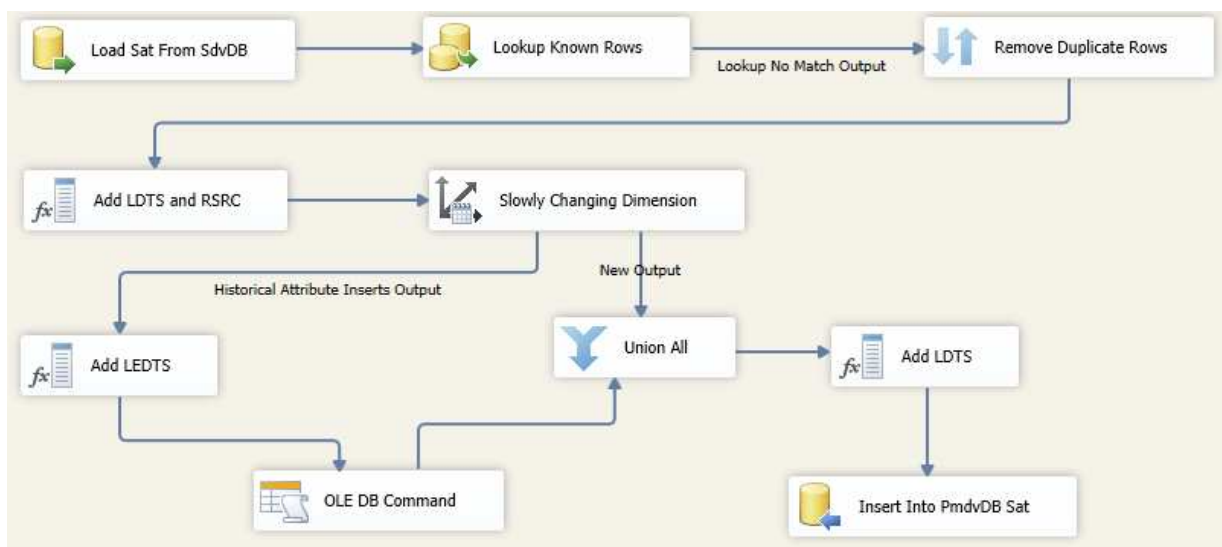


**Slika 39. SSIS tok podataka za eliminaciju duplih redaka u PmdvDB**

Slika 40 prikazuje primjer SSIS toka podataka za punjenje master satelita S\_EmployeeMaster. Ovdje je važan SQL upit koji već u prvom koraku (Load Sat From SdvDB) vrši transformaciju tablica iz SdvDB, na način da se selektiraju svi zapisi iz S\_Employee i S\_EmployeeTra te se vrši

unija oba skupa rezultata. Pritom se vodi računa da je u JobDB MaritalStatus u zasebnoj tablici, dok je u TrainingDB MaritalStatus atribut u tablici Employee (te su na taj način i pohranjeni u SdvDB). Također, u S\_Employee postoji atribut ShortBiography, dok u S\_EmployeeTra isti atribut ne postoji.

Dodatno, i ovdje se na tok podataka sa Slike 40 veže tok podataka sa Slike 41 – provjera i ažuriranje redaka iz PmdvDB, koji su u međuvremenu u izvorima izbrisani, a u SdvDB im se promijenila vrijednost LEDTS.

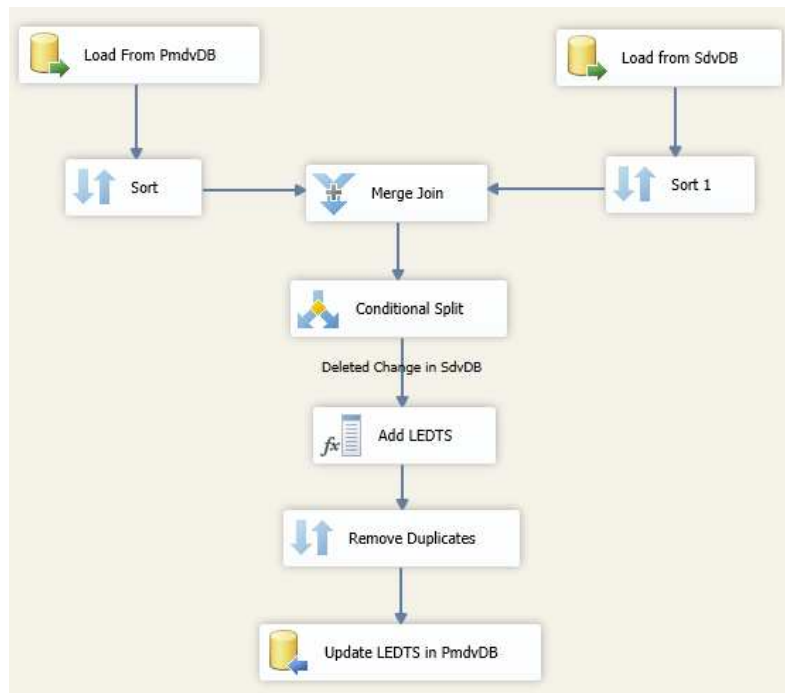


```

SELECT s.H_EMPLOYEE_SQN_Employee,s.EmployeeLDTS,Name,Surname,Title,Gender,BirthDate,
StreetNumber,StreetName,ZIP,City,County,Country,EmailAddress,PhoneNumber,ShortBiography,
s.EmployeeLEDTS,s.EmployeeRSRC, maritalStatus, maritalStatusLEDTS
FROM S_EMPLOYEE s INNER JOIN H_EMPLOYEE h
ON s.H_EMPLOYEE_SQN_Employee=h.SQN_Employee INNER JOIN L_EMP_MARITAL l
ON h.SQN_Employee=l.H_EMPLOYEE_SQN_Employee INNER JOIN H_MARITAL_STATUS m
ON m.SQN_MaritalStatus=l.H_MARITAL_STATUS_SQN_MaritalStatus INNER JOIN S_MARITAL_STATUS k
ON k.H_MARITAL_STATUS_SQN_MaritalStatus=m.SQN_MaritalStatus
AND EmployeeLEDTS IS NULL AND maritalStatusLEDTS is NULL
UNION
SELECT g.H_EMPLOYEE_SQN_Employee, g.EmployeeLDTS, Name,Surname,Title,Gender,BirthDate,
,StreetName,ZIP,City,County,Country,EmailAddress,PhoneNumber,
NULL as ShortBiography,g.EmployeeLEDTS,g.EmployeeRSRC,MaritalStatus, NULL as maritalStatusLEDTS
FROM S_EMPLOYEETRA g INNER JOIN H_EMPLOYEE h
ON g.H_EMPLOYEE_SQN_Employee=h.SQN_Employee AND EmployeeLEDTS IS NULL

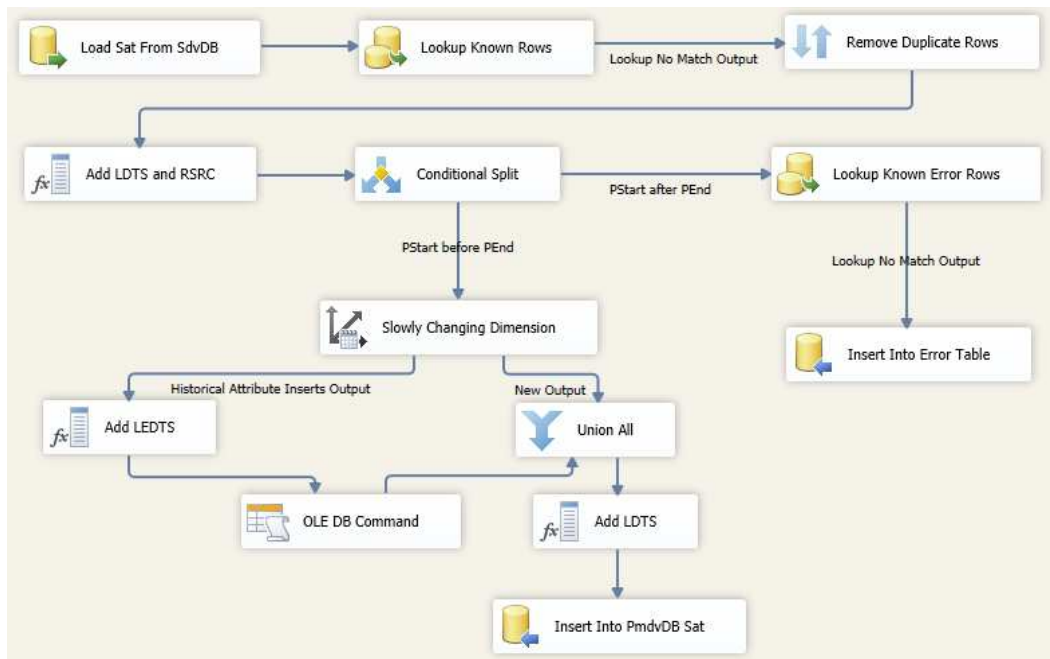
```

**Slika 40. SSIS tok podataka za integraciju više tablica u jednu u PmdvDB**



**Slika 41. SSIS tok podataka za ažuriranje Satelita u PmdvDB**

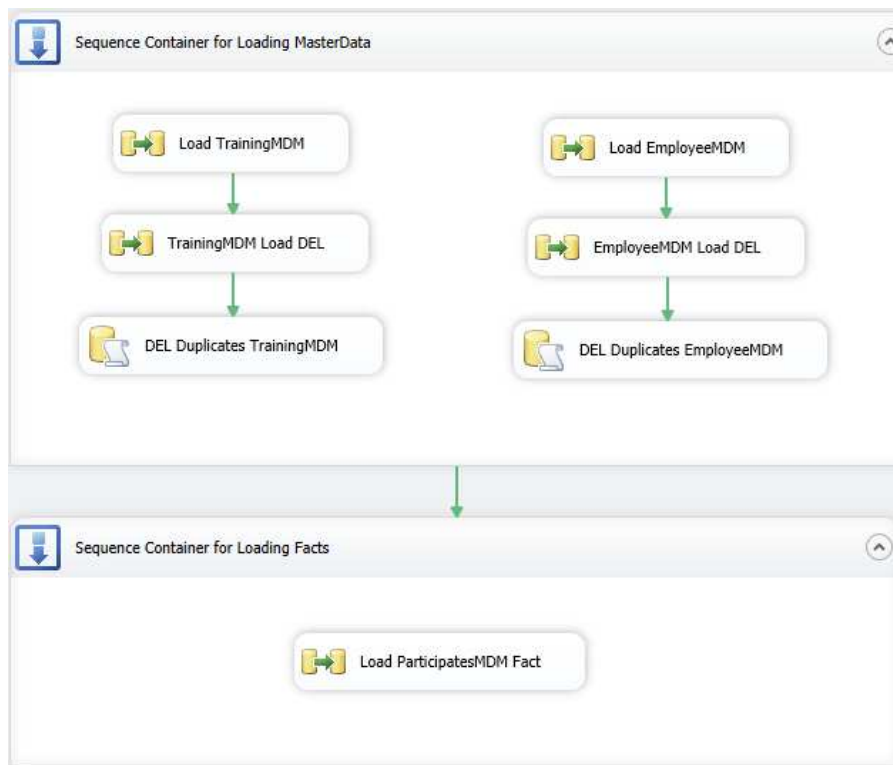
I zadnje vezano uz PmdvDB, Slika 42 prikazuje punjenje satelita S\_Project iz SdvDB u PmdvDB, uz provjeru poslovnog pravila. Poslovno pravilo glasi: „Datum početka projekta mora biti manji od datuma kraja projekta (ProjectStartDate < ProjectEndDate)“. Na Slici 42 vidimo da postoje dvije destinacije za pronađene retke, nakon Conditional Split transformacije. Jedan tok puni retke koji zadovoljavaju uvjet u S\_Project (Pstart before Pend, Insert Into PmdvDB Sat), dok drugi puni tablicu u PmdvDB posebno kreiranu za obradu grešaka (Pstart After Pend, Insert Into Error Table), uz eliminaciju identičkih redaka grešaka (Lookup Known Error Rows).



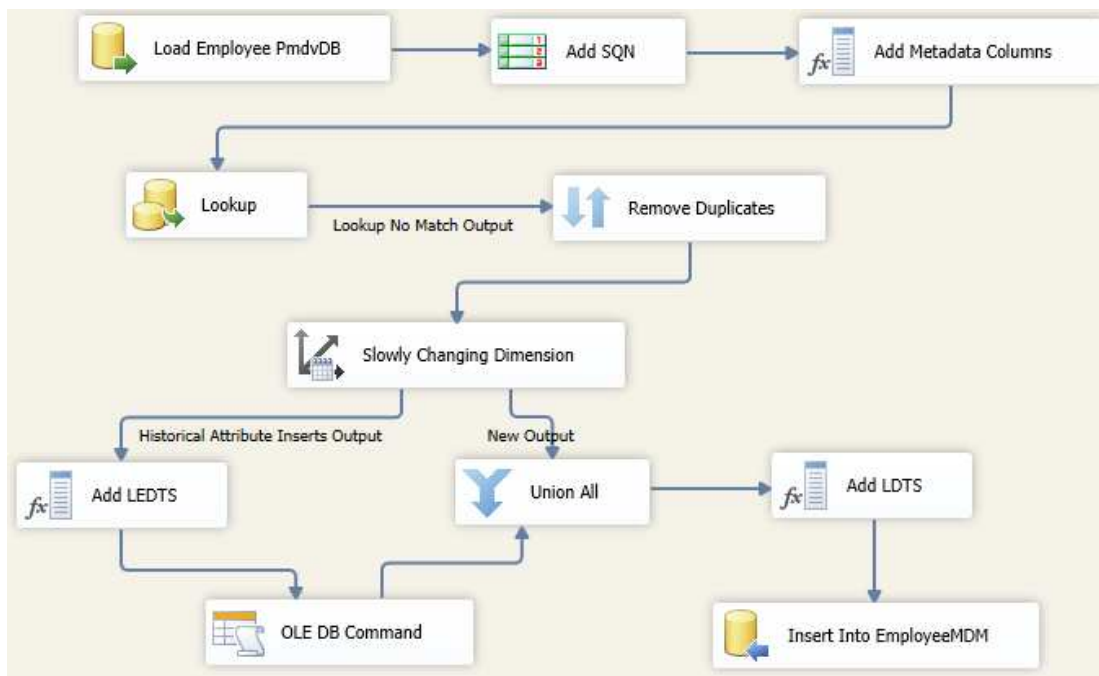
**Slika 42. SSIS tok podataka za punjenje Satelita s poslovnim pravilom u PmdvDB**

Nakon što je PmdvDB napunjena te su nad podacima pohranjenima u njoj provedene navedene “teške” transformacije, ona sada predstavlja osnovu za punjenje UMP i područnog SP. U tu svrhu kreirane su još dvije dodatne baze podataka: MdmDB koja praktično predstavlja UMP sustav, odnosno pohranjuje matične podatke iz PmdvDB unutar dimenzijskog modela te DmDB koja predstavlja područno SP. Ovo je u prototipu izvedeno na način da PmdvDB puni MdmDB (podaci se transformiraju kako bi mogli biti pohranjeni prema dimenzijskom modelu), a MdmDM dalje puni DmDB (također organiziranu prema dimenzijskom modelu). Na ovaj način integrirali smo ove dvije baze podataka. Na Slici 43 vidimo SSIS tok kontrole za punjenje matičnih podataka o zaposlenicima te o treninzima. Slika 44 prikazuje SSIS tok podataka za punjenje matičnih podataka o zaposlenicima (EmployeeMDM) iz PmdvDB huba H\_EMPLOYEE\_MASTER i njegovog satelita S\_EMPLOYEE\_MASTERALL.

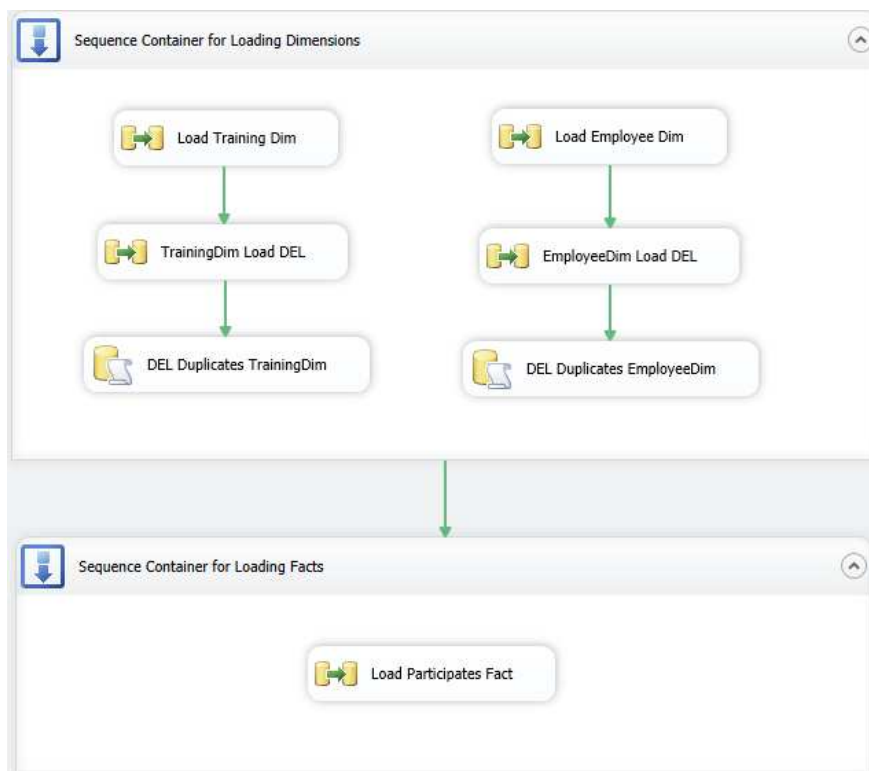
Dodatno, na Slikama 45 i 46 vidimo tok kontrole za punjenje DmDB i unutar toga tok podataka za punjenje dimenzije Employee, koja čuva podatke o zaposlenicima (na temelju podataka iz MdmDB).



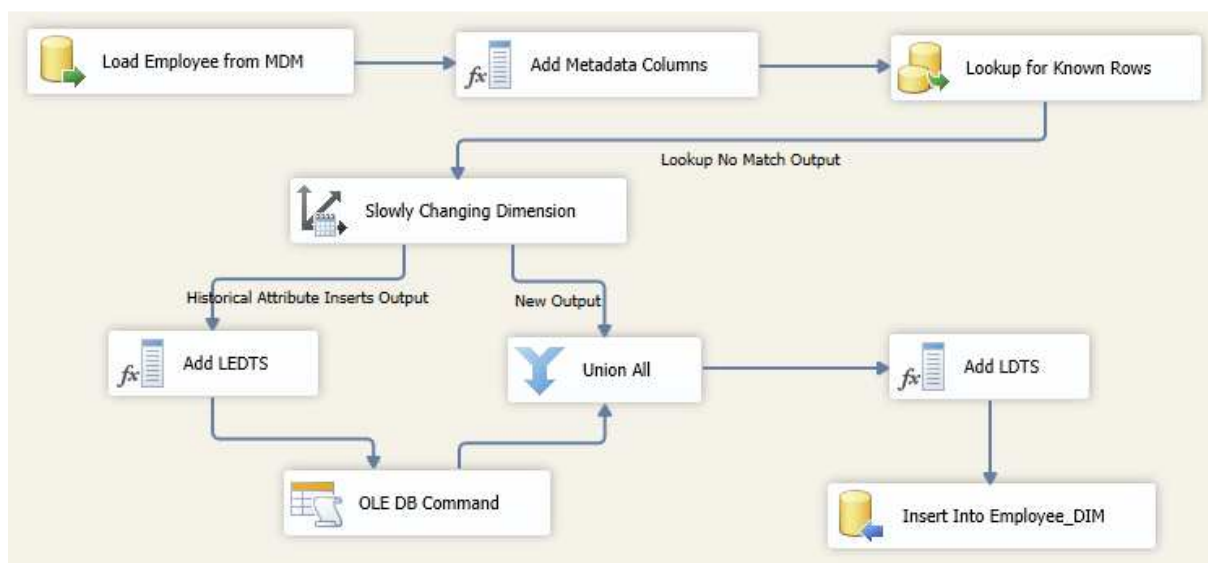
Slika 43. SSIS tok kontrole za punjenje MdmDB matičnih podataka o Employee i Training



Slika 44. SSIS tok podataka za punjenje EmployeeMDM



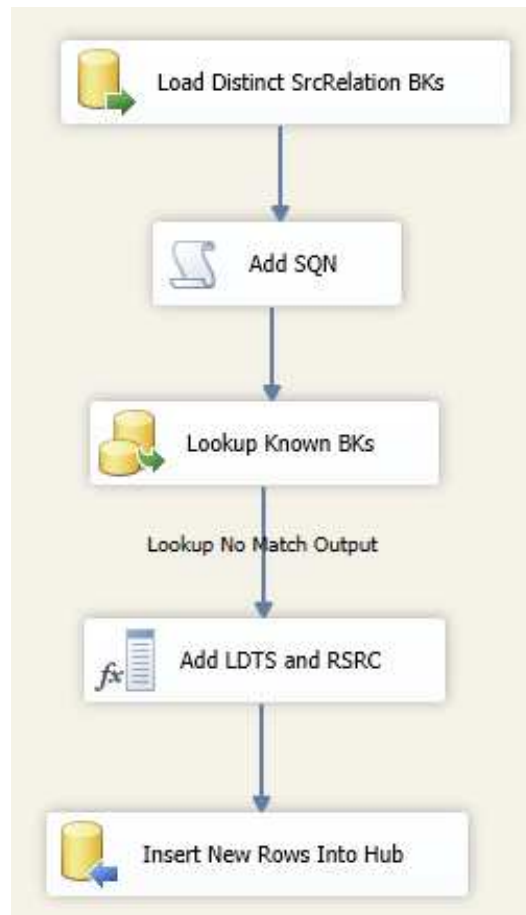
**Slika 45. SSIS tok kontrole za punjenje DmDB dimenzija Employee i Training i tablice činjenica Participates**



**Slika 46. SSIS tok podataka za punjenje dimenzije Employee**

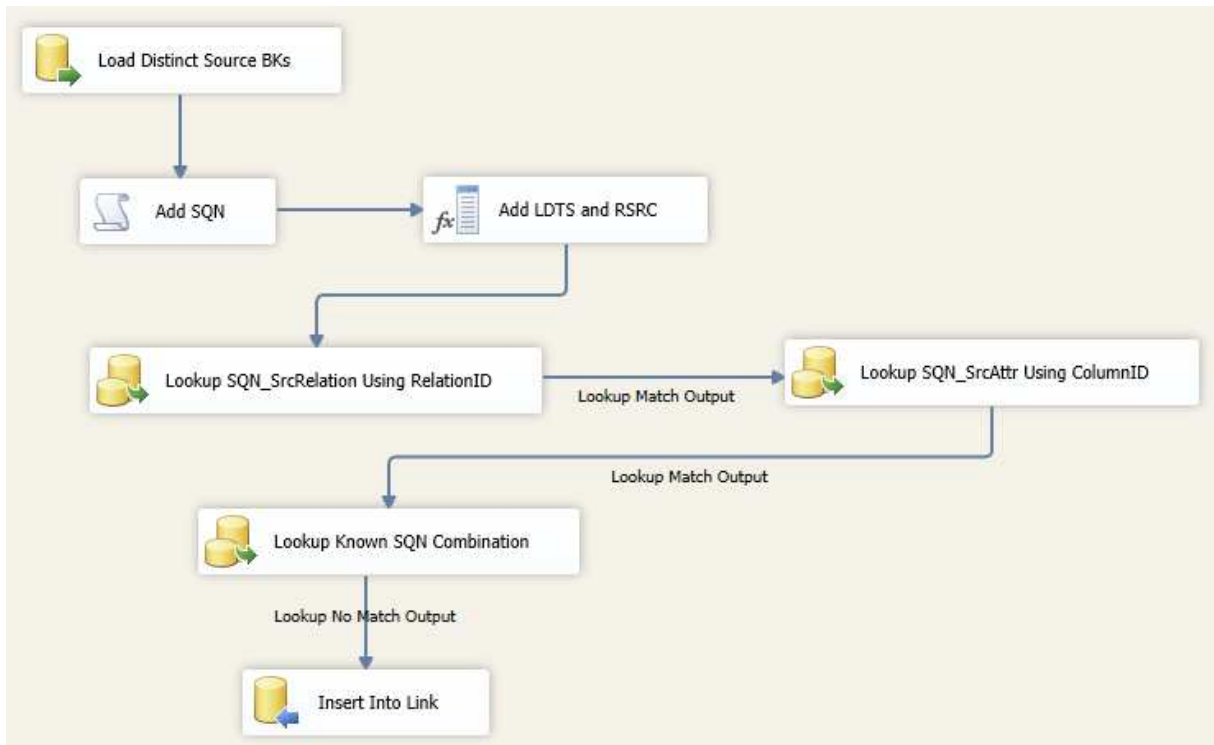
I na kraju imamo punjenje MdvDB repozitorija na temelju sistemskog kataloga. MdvDB baza podataka u prototipu predstavlja MDV repozitorij te je kreirana prema definiranom MDV modelu (Poglavlja 3 i 4). U ovoj trenutnoj verziji prototipa implementiran je samo dio MDV modela – dio koji prati promjene u shemi izvora podataka te dio koji prati promjene u SDV i

PMDV shemama. Obzirom da MdvDB predstavlja proširenje sistemskog kataloga, dio podataka vuče iz njega, a dio je kroz ETL procese polu-automatski definiran. Općeniti princip punjenja MDV možemo poopćiti i na princip punjenja SdvDB/PmdvDB, jer se sve ove baze podataka zasnivaju na DV modelu. Na Slikama 47, 48 i 49 možemo vidjeti primjere SSIS toka podataka za punjenje MdvDB hubova, linkova i satelita.

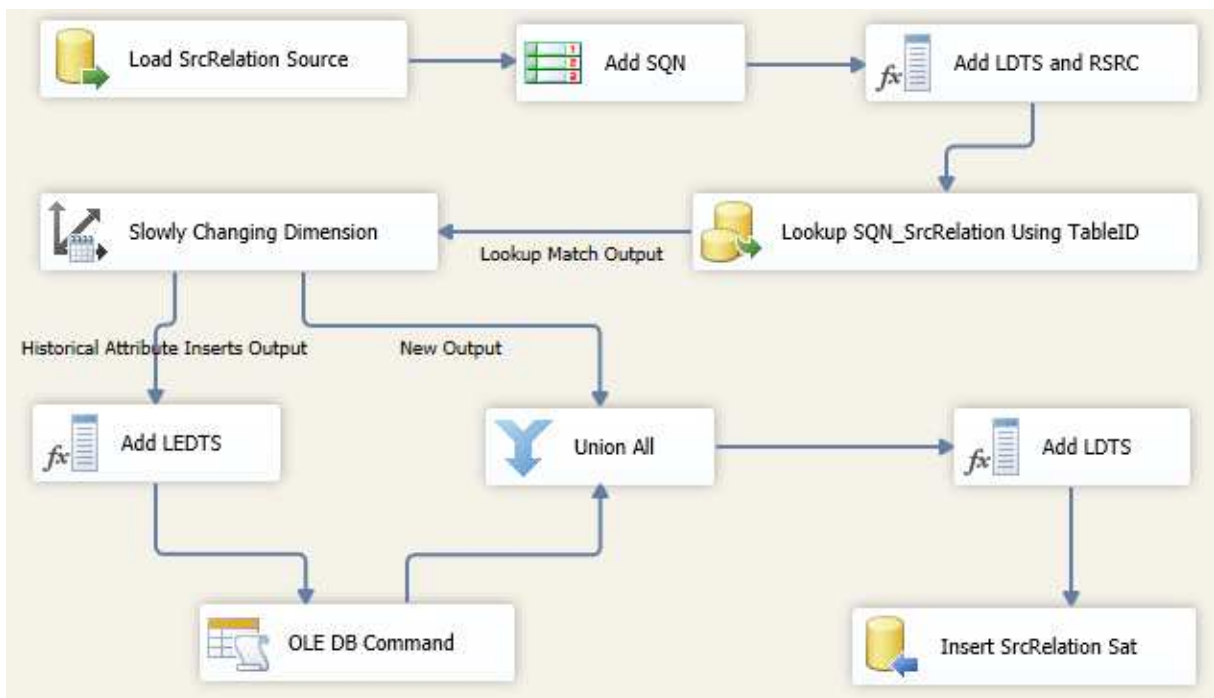


**Slika 47. SSIS tok podataka za punjenje MdvDB hubova**





Slika 48. SSIS tok podataka za punjenje MdvDB linkova



Slika 49. SSIS tok podataka za punjenje MdvDB satelita

## 5.2. Provođenje promjena u izvorima, SdvDB/PmdvDB i

### MdvDB

Osnovne promjene definirane u Poglavlju 4.3 provukli smo kroz prototip te smo promatrali kako se one implementiraju i kako utječu na SdvDB, PmdvDB i MdvDB.

#### Dodavanje nove relacije $D_R$ i dodavanje veze $D_V$

U Tablici 23 vidimo SQL naredbe za dodavanje tablice TRAINER u TrainingDB izvorišnu bazu podataka te njene veze *leads* na postojeću tablicu CERTIFICATE\_TRAINING, kao i naredbe za dodavanje odgovarajućih hubova, linkova i satelita u SdvDB i PmdvDB.

**Tablica 23. Dodavanje nove relacije,  $D_R$**

```
-- Kreiranje tablice u TrainingDB
USE [TrainingDB]
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[TRAINER](
    [TrainerID] [numeric](5, 0) NOT NULL,
    [TrainerFullName] [nvarchar](50) NOT NULL,
    [TrainerTitle] [nvarchar](50) NULL,
    [TrainerBio] [nvarchar](1500) NULL,
    CONSTRAINT [TRAINER_PK] PRIMARY KEY CLUSTERED
(
    [TrainerID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO

-- Kreiranje tablica u SdvDB
USE [SdvDB]
GO

SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[H_TRAINER](
    [SQN_Trainer] [numeric](5, 0) NOT NULL,
    [TrainerID] [numeric](5, 0) NULL,
    [TrainerLDTS] [datetime] NULL,
    [TrainerRSRC] [nvarchar](300) NULL,
    CONSTRAINT [H_TRAINER_PK] PRIMARY KEY CLUSTERED
(
```

```

        [SQN_Trainer] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
    ) ON [PRIMARY]

GO

CREATE TABLE [dbo].[S_TRAINER](
    [TrainerLDTS] [datetime] NOT NULL,
    [TrainerFullName] [nvarchar](200) NOT NULL,
    [TrainerTitle] [nvarchar](50) NULL,
    [TrainerBio] [nvarchar](1500) NULL,
    [TrainerLEDTS] [datetime] NULL,
    [TrainerRSRC] [nvarchar](100) NULL,
    [H_TRAINER_SQN_Trainer] [numeric](5, 0) NOT NULL,
    CONSTRAINT [S_TRAINER_PK] PRIMARY KEY CLUSTERED
    (
        [TrainerLDTS] ASC,
        [H_TRAINER_SQN_Trainer] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
    ) ON [PRIMARY]

GO

ALTER TABLE [dbo].[S_TRAINER] WITH CHECK ADD CONSTRAINT [S_TRAINER_TRAINER_FK]
FOREIGN KEY([H_TRAINER_SQN_Trainer])
REFERENCES [dbo].[H_TRAINER] ([SQN_Trainer])
GO

ALTER TABLE [dbo].[S_TRAINER] CHECK CONSTRAINT [S_TRAINER_TRAINER_FK]
GO

CREATE TABLE [dbo].[L_LEADS](
    [SQN_Leads] [numeric](5, 0) NOT NULL,
    [LeadsLDTS] [datetime] NULL,
    [LeadsRSRC] [nvarchar](100) NULL,
    [H_TRAINER_SQN_Trainer] [numeric](5, 0) NOT NULL,
    [H_CERTIFICATE_TRAINING_SQN_Training] [numeric](5, 0) NOT NULL,
    CONSTRAINT [L_LEADS_PK] PRIMARY KEY CLUSTERED
    (
        [SQN_Leads] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
    ) ON [PRIMARY]

GO

ALTER TABLE [dbo].[L_LEADS] WITH NOCHECK ADD CONSTRAINT
[L_LEADS_CERTIFICATE_TRAINING_FK] FOREIGN KEY([H_CERTIFICATE_TRAINING_SQN_Training])
REFERENCES [dbo].[H_CERTIFICATE_TRAINING] ([SQN_Training])
GO

ALTER TABLE [dbo].[L_LEADS] CHECK CONSTRAINT [L_LEADS_CERTIFICATE_TRAINING_FK]
GO

ALTER TABLE [dbo].[L_LEADS] WITH NOCHECK ADD CONSTRAINT [L_LEADS_TRAINER_FK]
FOREIGN KEY([H_TRAINER_SQN_Trainer])
REFERENCES [dbo].[H_TRAINER] ([SQN_Trainer])
GO

```

```

ALTER TABLE [dbo].[L_LEADS] CHECK CONSTRAINT [L_LEADS_TRAINER_FK]
GO

CREATE TABLE [dbo].[S_LEADS](
    [LeadsLDTS] [datetime] NOT NULL,
    [LeadsLEDTS] [datetime] NULL,
    [LeadsRSRC] [nvarchar](100) NULL,
    [L_LEADS_SQN_Leads] [numeric](5, 0) NOT NULL,
    CONSTRAINT [S_LEADS_PK] PRIMARY KEY CLUSTERED
(
    [LeadsLDTS] ASC,
    [L_LEADS_SQN_Leads] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO

ALTER TABLE [dbo].[S_LEADS] WITH CHECK ADD CONSTRAINT [S_LEADS_L_LEADS_FK] FOREIGN
KEY([L_LEADS_SQN_Leads])
REFERENCES [dbo].[L_LEADS] ([SQN_Leads])
GO

ALTER TABLE [dbo].[S_LEADS] CHECK CONSTRAINT [S_LEADS_L_LEADS_FK]
GO

-- Kreiranje tablica u PmdvDB
USE [PmdvDB]
GO

SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[H_TRAINER_MASTER](
    [SQN_Trainer] [numeric](5, 0) NOT NULL,
    [TrainerID] [numeric](5, 0) NULL,
    [TrainerLDTS] [datetime] NULL,
    [TrainerRSRC] [nvarchar](300) NULL,
    CONSTRAINT [H_TRAINER_MASTERPK] PRIMARY KEY CLUSTERED
(
    [SQN_Trainer] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO

CREATE TABLE [dbo].[S_TRAINER_MASTER](
    [TrainerLDTS] [datetime] NOT NULL,
    [TrainerFullName] [nvarchar](200) NOT NULL,
    [TrainerTitle] [nvarchar](50) NULL,
    [TrainerBio] [nvarchar](1500) NULL,
    [TrainerLEDTS] [datetime] NULL,
    [TrainerRSRC] [nvarchar](100) NULL,
    [H_TRAINER_MASTER_SQN_Trainer] [numeric](5, 0) NOT NULL,
    CONSTRAINT [S_TRAINER_MASTER_PK] PRIMARY KEY CLUSTERED

```

```

(
    [TrainerLDTS] ASC,
    [H_TRAINER_MASTER_SQN_Trainer] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO

ALTER TABLE [dbo].[S_TRAINER_MASTER] WITH CHECK ADD CONSTRAINT
[S_TRAINER_MAS_TRAINER_MAS_FK] FOREIGN KEY([H_TRAINER_MASTER_SQN_Trainer])
REFERENCES [dbo].[H_TRAINER_MASTER] ([SQN_Trainer])
GO

ALTER TABLE [dbo].[S_TRAINER_MASTER] CHECK CONSTRAINT [S_TRAINER_MAS_TRAINER_MAS_FK]
GO

CREATE TABLE [dbo].[L_LEADS_MASTER](
    [SQN_Leads] [numeric](5, 0) NOT NULL,
    [LeadsLDTS] [datetime] NULL,
    [LeadsRSRC] [nvarchar](100) NULL,
    [H_TRAINER_SQN_Trainer] [numeric](5, 0) NOT NULL,
    [H_CERT_TRA_MAST_SQN_Training] [numeric](5, 0) NOT NULL,
    CONSTRAINT [L_LEADS_MASTER_PK] PRIMARY KEY CLUSTERED
(
    [SQN_Leads] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO

ALTER TABLE [dbo].[L_LEADS_MASTER] WITH NOCHECK ADD CONSTRAINT
[L_LEADS_MAS_CERT_TRAIN_FK] FOREIGN KEY([H_CERT_TRA_MAST_SQN_Training])
REFERENCES [dbo].[H_CERTIFICATE_TRAINING_MASTER] ([SQN_Training])
GO

ALTER TABLE [dbo].[L_LEADS_MASTER] CHECK CONSTRAINT [L_LEADS_MAS_CERT_TRAIN_FK]
GO

ALTER TABLE [dbo].[L_LEADS_MASTER] WITH NOCHECK ADD CONSTRAINT
[L_LEADS_MAS_TRAINER_FK] FOREIGN KEY([H_TRAINER_SQN_Trainer])
REFERENCES [dbo].[H_TRAINER_MASTER] ([SQN_Trainer])
GO

ALTER TABLE [dbo].[L_LEADS_MASTER] CHECK CONSTRAINT [L_LEADS_MAS_TRAINER_FK]
GO

CREATE TABLE [dbo].[S_LEADS_MASTER](
    [LeadsLDTS] [datetime] NOT NULL,
    [LeadsLEDTS] [datetime] NULL,
    [LeadsRSRC] [nvarchar](100) NULL,
    [L_LEADS_SQN_Leads] [numeric](5, 0) NOT NULL,
    CONSTRAINT [S_LEADS_MASTER_PK] PRIMARY KEY CLUSTERED
(
    [LeadsLDTS] ASC,
    [L_LEADS_SQN_Leads] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO

```

```

ALTER TABLE [dbo].[S_LEADS_MASTER] WITH CHECK ADD CONSTRAINT
[S_LEADS_MAS_L_LEADS_FK] FOREIGN KEY([L_LEADS_SQN_Leads])
REFERENCES [dbo].[L_LEADS_MASTER] ([SQN_Leads])
GO

ALTER TABLE [dbo].[S_LEADS_MASTER] CHECK CONSTRAINT [S_LEADS_MAS_L_LEADS_FK]
GO

```

Nakon dodavanja nove tablice, u nju smo ubacili podatke o 20 trenera te smo u CERTIFICATE\_TRAINING tablicu na postojećih 5 treninga pridružili trenere.

Ova promjena u izvorima i u SdvDB/PmdvDB povlači punjenje MdvDB repozitorija s novim zapisima, kako bi se zabilježilo novo stanje u shemi izvora podataka TrainingDB, kao i u shemama SdvDB i PmdvDB. Slika 50 prikazuje stanje MdvDB repozitorija prije ove promjene, a Slika 51 nakon promjene, promatrano u dijelu repozitorija za izvore podataka. Vidimo da je nakon promjene u MdvDB hub H\_SRC\_RELATION i njegov satelit S\_SRC\_RELATION dodan novi zapis za izvorišnu tablicu TRAINER.

H_SRC_REL_SQN_SrcRel	SrcRelationID	SrcRelationName	SrcRelationLDTS	SrcRelationLEDTS	SrcRelationRSRC
1	565577053	CERTIFICATE_TRAINING	2016-10-11 14:29:53.000	2016-10-12 13:35:57.000	System Catalog
2	565577053	EMPLOYEE	2016-10-12 13:35:57.000	2016-10-12 13:38:28.000	System Catalog
3	565577053	EMPLOYEE	2016-10-12 15:44:10.000	NULL	System Catalog
4	629577281	COMPETENCE	2016-10-11 14:29:53.000	NULL	System Catalog
5	677577452	EMPLOYEE	2016-10-11 14:32:08.000	NULL	System Catalog
6	341576255	PARTICIPATES	2016-10-11 14:32:08.000	NULL	System Catalog

Slika 50. MdvDB stanje u izvorima prije dodavanja relacije

H_SRC_REL_SQN_SrcRel	SrcRelationID	SrcRelationName	SrcRelationLDTS	SrcRelationLEDTS	SrcRelationRSRC
1	565577053	CERTIFICATE_TRAINING	2016-10-11 14:29:53.000	2016-10-12 13:35:57.000	System Catalog
2	565577053	EMPLOYEE	2016-10-12 13:35:57.000	2016-10-12 13:38:28.000	System Catalog
3	565577053	CERTIFICATE_TRAINING	2016-10-12 17:38:50.000	NULL	System Catalog
4	629577281	COMPETENCE	2016-10-11 14:29:53.000	NULL	System Catalog
5	677577452	EMPLOYEE	2016-10-11 14:32:08.000	NULL	System Catalog
6	341576255	PARTICIPATES	2016-10-11 14:32:08.000	NULL	System Catalog
7	1381579960	TRAINER	2016-10-12 17:38:50.000	NULL	System Catalog

Slika 51. MdvDB stanje u izvorima nakon dodavanja relacije

Dodavanje novih tablica u SdvDB i PmdvDB prikazano je na Slici 52 (možemo vidjeti da su na postojeće stanje dodana dva zadnja retka zapisa – novi „sirovi“ hub H\_TRAINER i njegov matični hub H\_TRAINER\_MASTER te njihovi pripadajući sateliti) i na Slici 53 (zbog velikog broja atributa u MdvDB, ovdje su prikazani samo novi sateliti i njihovi atributi). U stupcu ChangeType vidimo koja promjena je uzrokovala upis novih redaka u MdvDB te vidimo da te nove tablice postoje u 2. verziji sheme (u odnosu na NULL vrijednost za početnu verziju sheme izvora, SdvDB i PmdvDB).

```

SELECT distinct a.H_HUB_SQN_Hub, a.HubName, e.H_SAT_SQN_Sat, e.SatName, a.ChangeType, a.HubSchemaName
FROM S_HUB_DEF a INNER JOIN H_HUB b ON a.H_HUB_SQN_Hub=b.SQN_Hub
INNER JOIN L_HUB_SAT c ON b.SQN_Hub=c.H_HUB_SQN_Hub INNER JOIN H_SATELLITE d ON d.SQN_Satellite=c.H_SATELLITE_SQN_Satellite
INNER JOIN S_SAT_DEF e ON d.SQN_Satellite=e.H_SAT_SQN_Sat

```

	H_HUB_SQN_Hub	HubName	H_SAT_SQN_Sat	SatName	ChangeType	HubSchemaName
1	1	H_CERTIFICATE_TRAINING	1	S_CERTIFICATE_TRAINING	NULL	NULL
2	1	H_CERTIFICATE_TRAINING	3	S_COMPETENCE	NULL	NULL
3	2	H_CERTIFICATE_TRAINING_MASTER	2	S_CERTIFICATE_TRAINING_MASTER	NULL	NULL
4	2	H_CERTIFICATE_TRAINING_MASTER	4	S_COMPETENCE_MASTER	NULL	NULL
5	3	H_EMPLOYEE	5	S_EMPLOYEE	NULL	NULL
6	3	H_EMPLOYEE	9	S_EMPLOYEEETRA	NULL	NULL
7	4	H_EMPLOYEE_MASTER	6	S_EMPLOYEE_MASTER	NULL	NULL
8	4	H_EMPLOYEE_MASTER	7	S_EMPLOYEE_MASTERALL	NULL	NULL
9	4	H_EMPLOYEE_MASTER	8	S_EMPLOYEEETRA	NULL	NULL
10	5	H_EMPLOYEE_MASTER	6	S_EMPLOYEE_MASTER	NULL	NULL
11	5	H_EMPLOYEE_MASTER	7	S_EMPLOYEE_MASTERALL	NULL	NULL
12	5	H_EMPLOYEE_MASTER	8	S_EMPLOYEEETRA	NULL	NULL
13	6	H_MARITAL_STATUS	10	S_MARITAL_STATUS	NULL	NULL
14	7	H_MARITAL_STATUS_MASTER	11	S_MARITAL_STATUS_MASTER	NULL	NULL
15	8	H_PROJECT	15	S_PROJECT	NULL	NULL
16	9	H_PROJECT_MASTER	14	S_PRO_MAS_STARTERR	NULL	NULL
17	9	H_PROJECT_MASTER	16	S_PROJECT_MASTER	NULL	NULL
18	10	H_SUPERVISOR	17	S_SUPERVISOR	NULL	NULL
19	11	H_SUPERVISOR_MASTER	18	S_SUPERVISOR_MASTER	NULL	NULL
20	23	H_TRAINER	39	S_TRAINER	Add relation	2
21	24	H_TRAINER_MASTER	40	S_TRAINER_MASTER	Add relation	2

**Slika 52. MdvDB stanje u SDV/PMDV hub-satelit shemama nakon dodavanja relacije**

SQLQuery95.sql - O...NIRI\dsuotic (53) SQLQuery62.sql - O...NIRI\dsuotic (58) X

```

SELECT distinct a.H_SAT_SQN_Sat, a.SatName, e.H_ATTRIBUTE_SQN_Attribute, e.AttrName, a.ChangeType
FROM S_SAT_DEF a INNER JOIN H_SATELLITE b ON a.H_SAT_SQN_Sat=b.SQN_Satellite
INNER JOIN L_SAT_ATTR c ON b.SQN_Satellite=c.H_SATELLITE_SQN_Satellite INNER JOIN H_ATTRIBUTE d ON d.SQN_Attribute=c.H_ATTRIBUTE_SQN_Attribute
INNER JOIN S_ATTR_DEF e ON e.H_ATTRIBUTE_SQN_Attribute=d.SQN_Attribute AND a.SatSchemaName=2

```

100 %

Results Messages

	H_SAT_SQN_Sat	SatName	H_ATTRIBUTE_SQN_Attribute	AttrName	ChangeType
1	28	S_LEADS	536	L_LEADS_SQN_Leads	Add relation
2	28	S_LEADS	540	LeadsLDTS	Add relation
3	28	S_LEADS	544	LeadsLEDTS	Add relation
4	28	S_LEADS	547	LeadsRSRC	Add relation
5	29	S_LEADS_MASTER	537	L_LEADS_SQN_Leads	Add relation
6	29	S_LEADS_MASTER	541	LeadsLDTS	Add relation
7	29	S_LEADS_MASTER	545	LeadsLEDTS	Add relation
8	29	S_LEADS_MASTER	549	LeadsRSRC	Add relation
9	39	S_TRAINER	534	H_TRAINER_SQN_Trainer	Add relation
10	39	S_TRAINER	753	TrainerBio	Add relation
11	39	S_TRAINER	755	TrainerFullName	Add relation
12	39	S_TRAINER	759	TrainerLDTS	Add relation
13	39	S_TRAINER	763	TrainerLEDTS	Add relation
14	39	S_TRAINER	767	TrainerRSRC	Add relation
15	39	S_TRAINER	769	TrainerTitle	Add relation
16	40	S_TRAINER_MASTER	531	H_TRAINER_MASTER_SQN_Trainer	Add relation
17	40	S_TRAINER_MASTER	754	TrainerBio	Add relation
18	40	S_TRAINER_MASTER	756	TrainerFullName	Add relation
19	40	S_TRAINER_MASTER	760	TrainerLDTS	Add relation
20	40	S_TRAINER_MASTER	764	TrainerLEDTS	Add relation
21	40	S_TRAINER_MASTER	768	TrainerRSRC	Add relation
22	40	S_TRAINER_MASTER	770	TrainerTitle	Add relation

**Slika 53. MdvDB stanje u SDV/PMDV satelit-atribut shemama nakon dodavanja relacije**

Na Slici 54 vidimo samo matične hubove koji su dodani u PmdvDB, odnosno prvi put su se pojavili u drugoj verziji PmdvDB sheme.

SQLQuery107.sql - ...NIRI\dsuotic (54) X

```

SELECT distinct a.H_HUB_SQN_Hub, a.HubName, e.H_SAT_SQN_Sat, e.SatName, a.ChangeType, a.HubSchemaName as SchemaVersion
FROM S_HUB_DEF a INNER JOIN H_HUB b ON a.H_HUB_SQN_Hub=b.SQN_Hub
INNER JOIN L_HUB_SAT c ON b.SQN_Hub=c.H_HUB_SQN_Hub INNER JOIN H_SATELLITE d ON d.SQN_Satellite=c.H_SATELLITE_SQN_Satellite
INNER JOIN S_SAT_DEF e ON d.SQN_Satellite=e.H_SAT_SQN_Sat AND a.HubName like '%MASTER%' AND a.HubSchemaName=2

```

100 %

Results Messages

	H_HUB_SQN_Hub	HubName	H_SAT_SQN_Sat	SatName	ChangeType	SchemaVersion
1	24	H_TRAINER_MASTER	40	S_TRAINER_MASTER	Add relation	2

**Slika 54. MdvDB stanje u PMDV shemi nakon dodavanja relacije**

### Preimenovanje relacije, I<sub>R</sub>

U TrainingDB preimenovali smo tablicu CERTIFICATE\_TRAINING u TRAINING izvođenjem sljedeće naredbe:

```
EXEC sp_rename 'CERTIFICATE_TRAINING', 'TRAINING';
```

U ETL paketu za punjenje SdvDB iz izvora podataka (SdvDBLoad paket) moramo ažurirati izvorišnu tablicu te pokrećemo ažuriranje MDV repozitorija kroz ETL paket za punjenje MdvDB (MdvDBLoad paket). Stanje u MDV repozitoriju nakon promjene prikazano je na Slici 55, a u sistemskom katalogu na Slici 56. Vidimo da MdvDB pamti i stari i novi naziv tablice, dok



sistemski katalog samo prati datum i vrijeme promjene, međutim ne vidimo o kojoj se promjeni radi.

	H_SRC_REL_SQN_SrcRel	SrcRelationID	SrcRelationName	SrcRelationLDTs	SrcRelationLEDTs	SrcRelationRSRC	Change Type
1	2	565577053	CERTIFICATE_TRAINING	2016-10-12 20:42:22.000	2016-10-13 11:02:43.000	System Catalog	NULL
2	2	565577053	TRAINING	2016-10-13 11:02:43.000	NULL	System Catalog	Rename relation
3	3	629577281	COMPETENCE	2016-10-12 20:42:22.000	NULL	System Catalog	NULL
4	5	677577452	EMPLOYEE	2016-10-12 20:42:22.000	NULL	System Catalog	NULL
5	7	341576255	PARTICIPATES	2016-10-12 20:42:22.000	NULL	System Catalog	NULL
6	10	1381579960	TRAINER	2016-10-12 20:42:22.000	NULL	System Catalog	NULL

**Slika 55. MdvDB stanje u izvorišnoj shemi nakon preimenovanja relacije**

```

/***** Script for SelectTopNRows command from SSMS *****/
SELECT TOP 1000 [name] ,[object_id] ,[type] ,[type_desc]
,[create_date] ,[modify_date]
FROM [TrainingDB].[sys].[tables]

```

	name	object_id	type	type_desc	create_date	modify_date
1	PARTICIPATES	341576255	U	USER_TABLE	2016-10-03 13:48:15.157	2016-10-06 12:02:33.373
2	TRAINING	565577053	U	USER_TABLE	2016-10-06 12:01:58.630	2016-10-13 10:56:21.380
3	COMPETENCE	629577281	U	USER_TABLE	2016-10-06 12:02:07.567	2016-10-06 12:02:07.587
4	EMPLOYEE	677577452	U	USER_TABLE	2016-10-06 12:02:33.257	2016-10-06 12:02:33.370
5	TRAINER	1381579960	U	USER_TABLE	2016-10-12 15:47:44.430	2016-10-12 15:53:02.180

**Slika 56. Stanje u sistemskom katalogu nakon preimenovanja relacije**

### Brisanje relacije, B<sub>R</sub> i brisanje veze B<sub>V</sub>

U TrainingDB izvoru brišemo tablicu TRAINER, kao i vezu na tablicu TRAINING (u njoj brišemo vanjski ključ - stupac TrainerID\_FK). Kroz ETL pakete za punjenje SdvDB i PmdvDB provlačimo promjene. U SdvDB i PmdvDB shemama nema promjena, odnosno zadržavaju se postojeće tablice kako bi se pamtilo stanje te verzije sheme, a promjena u izvorima se ažurira na podatkovnoj razini. Ovo možemo vidjeti na Slici 57 – imamo dva upita, gornji nad SdvDB i donji nad PmdvDB koji vraćaju isti skup podataka iz tablice TRAINER te vidimo da je svakome pojedinom treneru u stupcu TrainerLEDTs upisana vrijednost kraja važenja (odnosno datum i vrijeme brisanja). U MDV repozitoriju imamo stanje sa Slike 58 – vidimo da je tablica TRAINER

na datum 13.listopad 2016 izbrisana iz izvora (vrijednost SrcRelationLEDTS stupca je postavljena na datum i vrijeme brisanja, a vrijednost ChangeType stupca postavljena je na vrstu promjene koja je dovela do upisa ovog retka u MdvDB) – te sa Slike 59, gdje vidimo koji stupci su izbrisani u izvoru i kada.

TrainerLDTs	TrainerFullName	TrainerTitle	TrainerBio	TrainerLEDTS	TrainerRSRC	H_TRAINER_SQN_Trainer
2016-10-13 12:14:22.000	Marc O'Neill	Mr.	Lorem ipsum dolor sit amet, mauris dapibus nunc ...	2016-10-13 14:31:11.000	Load Trainer Sat	1
2016-10-13 12:14:22.000	Laura McNelly	Mrs.	Libero diam magna habitasse vestibulum mi. Mole...	2016-10-13 14:31:11.000	Load Trainer Sat	2
2016-10-13 12:14:22.000	Emm Avery	Mrs.	Incidunt elit lacinia maecenas ac, purus est feugi...	2016-10-13 14:31:11.000	Load Trainer Sat	3

TrainerLDTs	TrainerFullName	TrainerTitle	TrainerBio	TrainerLEDTS	TrainerRSRC	H_TRAINER_MASTER_SQN_Trainer
2016-10-13 12:14:22.000	Marc O'Neill	Mr.	Lorem ipsum dolor sit amet, mauris dapibus nunc ...	2016-10-13 14:32:09.000	Load Trainer Sat	1
2016-10-13 12:14:22.000	Laura McNelly	Mrs.	Libero diam magna habitasse vestibulum mi. Mole...	2016-10-13 14:32:09.000	Load Trainer Sat	2
2016-10-13 12:14:22.000	Emm Avery	Mrs.	Incidunt elit lacinia maecenas ac, purus est feugi...	2016-10-13 14:32:09.000	Load Trainer Sat	3

Slika 57. Podatkovno stanje u SdvDB i PmdvDB nakon brisanja relacije

SrcRelationLDTs	SrcRelationLEDTS	SrcRelationName	H_SRC_REL_SQN_SrcRel	ChangeType
2016-10-12 20:42:22.000	2016-10-13 11:02:43.000	CERTIFICATE_TRAINING	2	NULL
2016-10-12 20:42:22.000	NULL	COMPETENCE	3	NULL
2016-10-12 20:42:22.000	NULL	EMPLOYEE	4	NULL
2016-10-12 20:42:22.000	NULL	EMPLOYEE	5	NULL
2016-10-12 20:42:22.000	NULL	MARITAL_STATUS	6	NULL
2016-10-12 20:42:22.000	NULL	PARTICIPATES	7	NULL
2016-10-12 20:42:22.000	NULL	PROJECT	8	NULL
2016-10-12 20:42:22.000	NULL	SUPERVISOR	9	NULL
2016-10-12 20:42:22.000	NULL	TRAINER	10	NULL
2016-10-13 11:02:43.000	NULL	TRAINING	2	Rename relation
2016-10-13 14:32:14.000	2016-10-13 14:32:14.000	TRAINER	10	Delete relation

Slika 58. MdvDB stanje u izvorišnoj shemi tablica nakon brisanja relacije

SQLQuery117.sql -...NIRI\dsubotic (53))\* x ODJ-O422-102.SdvDB - dbo.S\_TRAINER SQLQuery107.sql -...NIRI\dsubotic (54))\*

```

SELECT [SrcAttrLDTs] , [SrcAttrLEDTs] , [SrcAttrRSRC] , [SrcAttrPhysicalName] , [SrcAttrChangeType] , [H_SRC_ATTR_SQN_SrcAttr]
FROM [MdvDB].[dbo].[S_SRC_ATTRIBUTE] WHERE [SrcAttrChangeType]='Delete relation'

```

100 %

Results Messages

	SrcAttrLDTs	SrcAttrLEDTs	SrcAttrRSRC	SrcAttrPhysicalName	SrcAttrChangeType	H_SRC_ATTR_SQN_SrcAttr
1	2016-10-13 14:52:40.000	2016-10-13 14:52:40.000	System Catalog	TRAINER_TrainerID	Delete relation	58
2	2016-10-13 14:52:40.000	2016-10-13 14:52:40.000	System Catalog	TrainerBio	Delete relation	59
3	2016-10-13 14:52:40.000	2016-10-13 14:52:40.000	System Catalog	TrainerFullName	Delete relation	60
4	2016-10-13 14:52:40.000	2016-10-13 14:52:40.000	System Catalog	TrainerID	Delete relation	61
5	2016-10-13 14:52:40.000	2016-10-13 14:52:40.000	System Catalog	TrainerTitle	Delete relation	62

**Slika 59. MdvDB stanje u izvorišnoj shemi stupaca nakon brisanja relacije**

Dotadno, na Slikama 60 i 61 vidimo stanje nakon brisanja izvorišne tablice u SDV/PMDV dijelu MdvDB repozitorija. Možemo vidjeti da je za hubove H\_TRAINER i H\_TRAINER\_MASTER (a isto vrijedi i za njihove satellite) postavljen datum prestanka važenja u HubDefLEDTs stupcu (Slika 60), isto kao i za attribute u njihovim satelitima (Slika 61).

SQLQuery125.sql -...NIRI\dsubotic (51))\* x

```

SELECT [HubDefLDTs]
, [HubDefLEDTs]
, [HubName]
, [HubSchemaName]
, [H_HUB_SQN_Hub]
, [ChangeType]
FROM [MdvDB].[dbo].[S_HUB_DEF]

```

100 %

Results Messages

	HubDefLDTs	HubDefLEDTs	HubName	HubSchemaName	H_HUB_SQN_Hub	ChangeType
1	2016-10-12 12:59:27.000	NULL	H_CERTIFICATE_TRAINING	NULL	1	NULL
2	2016-10-12 12:59:27.000	NULL	H_CERTIFICATE_TRAINING_MASTER	NULL	2	NULL
3	2016-10-12 12:59:27.000	NULL	H_EMPLOYEE	NULL	3	NULL
4	2016-10-12 12:59:27.000	NULL	H_EMPLOYEE_MASTER	NULL	4	NULL
5	2016-10-12 12:59:27.000	NULL	H_EMPLOYEE_MASTER	NULL	5	NULL
6	2016-10-12 12:59:27.000	NULL	H_MARITAL_STATUS	NULL	6	NULL
7	2016-10-12 12:59:27.000	NULL	H_MARITAL_STATUS_MASTER	NULL	7	NULL
8	2016-10-12 12:59:27.000	NULL	H_PROJECT	NULL	8	NULL
9	2016-10-12 12:59:27.000	NULL	H_PROJECT_MASTER	NULL	9	NULL
10	2016-10-12 12:59:27.000	NULL	H_SUPERVISOR	NULL	10	NULL
11	2016-10-12 12:59:27.000	NULL	H_SUPERVISOR_MASTER	NULL	11	NULL
12	2016-10-13 12:17:22.000	NULL	H_TRAINER	2	23	Add relation
13	2016-10-13 12:17:22.000	NULL	H_TRAINER_MASTER	2	24	Add relation
14	2016-10-13 14:32:14.000	2016-10-13 14:32:14.000	H_TRAINER	3	23	Delete relation
15	2016-10-13 14:32:14.000	2016-10-13 14:32:14.000	H_TRAINER_MASTER	3	24	Delete relation

**Slika 60. MdvDB stanje u SDV/PMDV shemama nakon brisanja relacije – hubovi**

SQLQuery129.sql -...NIRI\dsubotic (51))\* X

```
SELECT [AttrDefLDTS] , [AttrDefLEDTS], [AttrName], [ChangeType]
FROM [MdvDB].[dbo].[S_ATTR_DEF] WHERE [ChangeType]='Delete relation'
```

100 %

Results Messages

	AttrDefLDTS	AttrDefLEDTS	AttrName	ChangeType
1	2016-10-13 14:52:40.000	2016-10-13 14:52:40.000	H_CERT_TRA_MAST_SQN_Training	Delete relation
2	2016-10-13 14:52:40.000	2016-10-13 14:52:40.000	H_CERTIFICATE_TRAINING_SQN_Training	Delete relation
3	2016-10-13 14:52:40.000	2016-10-13 14:52:40.000	H_TRAINER_MASTER_SQN_Trainer	Delete relation
4	2016-10-13 14:52:40.000	2016-10-13 14:52:40.000	H_TRAINER_SQN_Trainer	Delete relation
5	2016-10-13 14:52:40.000	2016-10-13 14:52:40.000	H_TRAINER_SQN_Trainer	Delete relation
6	2016-10-13 14:52:40.000	2016-10-13 14:52:40.000	H_TRAINER_SQN_Trainer	Delete relation
7	2016-10-13 14:52:40.000	2016-10-13 14:52:40.000	SQN_Trainer	Delete relation
8	2016-10-13 14:52:40.000	2016-10-13 14:52:40.000	SQN_Trainer	Delete relation
9	2016-10-13 14:52:40.000	2016-10-13 14:52:40.000	TrainerBio	Delete relation
10	2016-10-13 14:52:40.000	2016-10-13 14:52:40.000	TrainerBio	Delete relation
11	2016-10-13 14:52:40.000	2016-10-13 14:52:40.000	TrainerFullName	Delete relation
12	2016-10-13 14:52:40.000	2016-10-13 14:52:40.000	TrainerFullName	Delete relation
13	2016-10-13 14:52:40.000	2016-10-13 14:52:40.000	TrainerID	Delete relation
14	2016-10-13 14:52:40.000	2016-10-13 14:52:40.000	TrainerID	Delete relation
15	2016-10-13 14:52:40.000	2016-10-13 14:52:40.000	TrainerTitle	Delete relation
16	2016-10-13 14:52:40.000	2016-10-13 14:52:40.000	TrainerTitle	Delete relation

**Slika 61. MdvDB stanje u SDV/PMDV shemama nakon brisanja relacije - atributi**

Za usporedbu, upitom nad sistemskim katalogom možemo vidjeti njegovo stanje nakon promjene u izvorima. Vidimo da sistemski katalog ne pamti povijest stanja izvorišne sheme, odnosno tablica TRAINER kao da nikad nije niti postojala u TrainingDB izvoru (Slika 62).

SQLQuery124.sql -...NIRI\dsubotic (57))\* X ODJ-O422-102.Mdv...o.S\_SRC\_RELATION SQLQuery123.sql -...NIRI\dsubotic (51))\* X

```
SELECT [name], [object_id] , [type] , [type_desc] , [create_date], [modify_date]
FROM [TrainingDB].[sys].[tables]
```

100 %

Results Messages

	name	object_id	type	type_desc	create_date	modify_date
1	PARTICIPATES	341576255	U	USER_TABLE	2016-10-03 13:48:15.157	2016-10-13 13:59:41.223
2	TRAINING	565577053	U	USER_TABLE	2016-10-06 12:01:58.630	2016-10-13 14:01:17.603
3	COMPETENCE	629577281	U	USER_TABLE	2016-10-06 12:02:07.567	2016-10-13 13:59:41.227
4	EMPLOYEE	677577452	U	USER_TABLE	2016-10-06 12:02:33.257	2016-10-13 13:59:41.227

**Slika 62. Stanje u sistemskom katalogu nakon brisanja relacije**

### Dodavanje atributa u relaciju, DA

U TrainingDB dodajemo stupac DriversLicence u tablicu EMPLOYEE. U SdvDB i PmdvDB ovu promjenu realiziramo na način da dodamo novi satelit S\_DRIVERS\_LICENCE na hub H\_EMPLOYEE. U Tablici 24 prikazan je skup SQL naredbi za dodavanje novog satelita u SdvDB i PmdvDB.

**Tablica 24. SQL naredbe za dodavanje novog satelita u SdvDB i PmdvDB**

```
CREATE TABLE [dbo].[S_DRIVERS_LICENCE](
    [DriversLicenceLDTS] [datetime] NOT NULL,
    [DriversLicence] [nvarchar](50) NULL,
    [DriversLicenceLEDTS] [datetime] NULL,
    [DriversLicenceRSRC] [nvarchar](100) NULL,
    [H_EMPLOYEE_SQN_Employee] [numeric](5, 0) NOT NULL,
    CONSTRAINT [S_LicenceEmp_PK] PRIMARY KEY CLUSTERED
(
    [DriversLicenceLDTS] ASC,
    [H_EMPLOYEE_SQN_Employee] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[S_DRIVERS_LICENCE] ADD CONSTRAINT [S_LICENCE_EMPLOYEE_FK]
FOREIGN KEY([H_EMPLOYEE_SQN_Employee])
REFERENCES [dbo].[H_EMPLOYEE] ([SQN_Employee])
GO
CREATE TABLE [dbo].[S_DRIVERS_LICENCE_MASTER](
    [DriversLicenceLDTS] [datetime] NOT NULL,
    [DriversLicence] [nvarchar](50) NULL,
    [DriversLicenceLEDTS] [datetime] NULL,
    [DriversLicenceRSRC] [nvarchar](100) NULL,
    [H_EMP_MASTER_SQN_Employee] [numeric](5, 0) NOT NULL,
    CONSTRAINT [S_LicenceEmp_PK] PRIMARY KEY CLUSTERED
(
    [DriversLicenceLDTS] ASC,
    [H_EMP_MASTER_SQN_Employee] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[S_DRIVERS_LICENCE_MASTER] ADD CONSTRAINT
[S_LICENCE_EMPLOYEEEM_FK] FOREIGN KEY([H_EMP_MASTER_SQN_Employee])
REFERENCES [dbo].[H_EMPLOYEE_MASTER] ([SQN_Employee])
GO
```

Na Slici 63 prikazano je podatkovno stanje u SdvDB (gornji upit/skup rezultata) i u PmdvDB (donji upit/skup rezultata). Nakon dodavanja stupca u izvorišnu tablicu EMPLOYEE i ažuriranja podataka (za jednog zaposlenika dodana je vrijednost YES u DriversLicence stupac), putem ETL procesa podaci su ažurirani u SdvDB i PmdvDB.



SQLQuery145.sql -...NIR\dsuotic (58))\* x SQLQuery144.sql -...NIR\dsuotic (57)

```

/***** Script for SelectTopNRows command from SSMS *****/
SELECT distinct c.Name, c.Surname, c.BirthDate, c.Gender,
[DriversLicenceLDTS], [DriversLicence], [DriversLicenceLEDTS], [DriversLicenceRSRC], a.[H_EMPLOYEE_SQN_Employee]
FROM [SdvDB].[dbo].[S_DRIVERS_LICENCE] a INNER JOIN [SdvDB].[dbo].[H_EMPLOYEE] b ON b.SQN_Employee=a.[H_EMPLOYEE_SQN_Employee]
INNER JOIN [SdvDB].[dbo].[S_EMPLOYEEETRA] c ON b.SQN_Employee=c.H_EMPLOYEE_SQN_Employee

SELECT distinct c.Name, c.Surname, c.BirthDate, c.Gender,
[DriversLicenceLDTS], [DriversLicence], [DriversLicenceLEDTS], [DriversLicenceRSRC], a.[H_EMP_MASTER_SQN_Employee]
FROM [PmdvDB].[dbo].[S_DRIVERS_LICENCE_MASTER] a INNER JOIN [PmdvDB].[dbo].[H_EMPLOYEE_MASTER] b ON b.SQN_Employee=a.[H_EMP_MASTER_SQN_Employee]
INNER JOIN [PmdvDB].[dbo].[S_EMPLOYEE_MASTERALL] c ON b.SQN_Employee=c.H_EMPLOYEE_SQN_Employee

```

100 %

Results Messages

Name	Surname	BirthDate	Gender	DriversLicenceLDTS	DriversLicence	DriversLicenceLEDTS	DriversLicenceRSRC	H_EMPLOYEE_SQN_Employee
1	Quinton	Mcbride	1995-10-26	m	2016-10-13 16:15:12.000	YES	NULL	Load Licence Sat 71

Name	Surname	BirthDate	Gender	DriversLicenceLDTS	DriversLicence	DriversLicenceLEDTS	DriversLicenceRSRC	H_EMP_MASTER_SQN_Employee
1	Quinton	Mcbride	1995-10-26	m	2016-10-13 16:15:12.000	YES	NULL	Load Licence Sat 71

Slika 63. Podatkovno stanje u SdvDB i PmdvDB nakon dodavanja atributa

Na Slici 64 prikazano je stanje u MdvDB repozitoriju, nakon provedene promjene. Vidimo da su za hubove H\_EMPLOYEE i H\_EMPLOYEE\_MASTER dodani sateliti S\_DRIVERS\_LICENCE i S\_DRIVERS\_LICENCE\_MASTER te isto možemo vidjeti koji stupci su za njih dodani.

SQLQuery147.sql -...NIR\dsuotic (54))\* x

```

SELECT distinct a.H_HUB_SQN_Hub, a.HubName, e.H_SAT_SQN_Sat, e.SatName, h.H_ATTRIBUTE_SQN_Attribute, h.AttrName
FROM S_HUB_DEF a INNER JOIN H_HUB b ON a.H_HUB_SQN_Hub=b.SQN_Hub
INNER JOIN L_HUB_SAT c ON b.SQN_Hub=c.H_HUB_SQN_Hub INNER JOIN H_SATELLITE d ON d.SQN_Satellite=c.H_SATELLITE_SQN_Satellite
INNER JOIN S_SAT_DEF e ON d.SQN_Satellite=e.H_SAT_SQN_Sat INNER JOIN L_SAT_ATTR f ON d.SQN_Satellite=f.H_SATELLITE_SQN_Satellite
INNER JOIN H_ATTRIBUTE g ON f.H_ATTRIBUTE_SQN_Attribute=g.SQN_Attribute INNER JOIN S_ATTR_DEF h ON h.H_ATTRIBUTE_SQN_Attribute=g.SQN_Attribute
WHERE a.ChangeType='Add attribute' OR e.ChangeType='Add attribute'

```

100 %

Results Messages

H_HUB_SQN_Hub	HubName	H_SAT_SQN_Sat	SatName	H_ATTRIBUTE_SQN_Attribute	AttrName	
1	3	H_EMPLOYEE	45	S_DRIVERS_LICENCE	820	DriversLicence
2	3	H_EMPLOYEE	45	S_DRIVERS_LICENCE	822	DriversLicenceLDTS
3	3	H_EMPLOYEE	45	S_DRIVERS_LICENCE	824	DriversLicenceLEDTS
4	3	H_EMPLOYEE	45	S_DRIVERS_LICENCE	826	DriversLicenceRSRC
5	3	H_EMPLOYEE	45	S_DRIVERS_LICENCE	901	H_EMPLOYEE_SQN_Employee
6	4	H_EMPLOYEE_MASTER	46	S_DRIVERS_LICENCE_MASTER	821	DriversLicence
7	4	H_EMPLOYEE_MASTER	46	S_DRIVERS_LICENCE_MASTER	823	DriversLicenceLDTS
8	4	H_EMPLOYEE_MASTER	46	S_DRIVERS_LICENCE_MASTER	825	DriversLicenceLEDTS
9	4	H_EMPLOYEE_MASTER	46	S_DRIVERS_LICENCE_MASTER	827	DriversLicenceRSRC
10	4	H_EMPLOYEE_MASTER	46	S_DRIVERS_LICENCE_MASTER	894	H_EMP_MASTER_SQN_Employee
11	5	H_EMPLOYEE_MASTER	46	S_DRIVERS_LICENCE_MASTER	821	DriversLicence
12	5	H_EMPLOYEE_MASTER	46	S_DRIVERS_LICENCE_MASTER	823	DriversLicenceLDTS
13	5	H_EMPLOYEE_MASTER	46	S_DRIVERS_LICENCE_MASTER	825	DriversLicenceLEDTS
14	5	H_EMPLOYEE_MASTER	46	S_DRIVERS_LICENCE_MASTER	827	DriversLicenceRSRC
15	5	H_EMPLOYEE_MASTER	46	S_DRIVERS_LICENCE_MASTER	894	H_EMP_MASTER_SQN_Employee

Slika 64. MdvDB stanje u SDV/PMDV shemi nakon dodavanja atributa

### Izmjena postojećeg atributa u relaciji - preimenovanje atributa, I<sub>An</sub>

U TrainingDB preimenovat ćemo stupac Name iz tablice EMPLOYEE u FirstName naredbom:

```
EXEC sp_rename 'EMPLOYEE.Name', 'EMPLOYEE.FirstName';
```

U ETL paketu za punjenje SdvDB (SdvDBLoad paket) moramo ažurirati izvorišni atribut te pokrećemo ažuriranje MdvDB repozitorija kroz MdvDBLoad ETL paket. Stanje u MdvDB

repozitoriju nakon promjene, kao i u sistemskom katalogu, prikazano je na Slici 65. Gornjim upitom nad MdvDB prikazali smo povijest stupca Name. Donjim upitom nad sistemskim katalogom prikazani su stupci iz relacije EMPLOYEE. Možemo vidjeti da MdvDB skup rezultata prikazuje i stari i novi naziv stupca Name/FirstName, dok se u sistemskom katalogu vidi samo novi naziv stupca FirstName (ne pamti se povijest promjena).

The screenshot shows two SQL queries and their results in SQL Server Enterprise Manager.

**Query 1 (Top):**

```
SELECT [H_SRC_ATTR_SQN_SrcAttr], [SrcAttrLDTs], [SrcAttrLEDTs], [SrcAttrRSRC], [SrcAttrPhysicalName], [SrcAttrChangeType]
FROM [MdvDB].[dbo].[S_SRC_ATTRIBUTE] WHERE H_SRC_ATTR_SQN_SrcAttr=31
```

**Query 2 (Bottom):**

```
SELECT TOP 1000 a.[object_id], b.name as TableName, a.[name] as ColumnName, [column_id]
FROM [TrainingDB].[sys].[columns] a INNER JOIN [TrainingDB].[sys].[tables] b ON a.object_id=b.object_id AND b.name='EMPLOYEE'
```

**Results of Query 1:**

	H_SRC_ATTR_SQN_SrcAttr	SrcAttrLDTs	SrcAttrLEDTs	SrcAttrRSRC	SrcAttrPhysicalName	SrcAttrChangeType
1	31	2016-10-13 08:49:03.000	2016-10-13 17:53:31.000	SystemCatalog	Name	NULL
2	31	2016-10-13 17:53:31.000	NULL	System Catalog	EMPLOYEE.FirstName	Rename attribute

**Results of Query 2:**

	object_id	TableName	ColumnName	column_id
1	677577452	EMPLOYEE	EmployeeID	1
2	677577452	EMPLOYEE	EMPLOYEE.FirstName	2
3	677577452	EMPLOYEE	Surname	3
4	677577452	EMPLOYEE	Title	4
5	677577452	EMPLOYEE	Gender	5
6	677577452	EMPLOYEE	MaritalStatus	6
7	677577452	EMPLOYEE	BirthDate	7
8	677577452	EMPLOYEE	StreetNumber	8
9	677577452	EMPLOYEE	StreetName	9
10	677577452	EMPLOYEE	ZIP	10
11	677577452	EMPLOYEE	City	11
12	677577452	EMPLOYEE	County	12
13	677577452	EMPLOYEE	Country	13
14	677577452	EMPLOYEE	EmailAddress	14
15	677577452	EMPLOYEE	PhoneNumber	15
16	677577452	EMPLOYEE	DriversLicence	16

**Slika 65. Stanje u MdvDB i sistemskom katalogu nakon preimenovanja atributa**

Izmjena postojećeg atributa u relaciji - promjena tipa podatka atributa, I<sub>At</sub>

U JobDB mijenja se tip podatka stupca ZIP (EMPLOYEE) iz NUMERIC u NVARCHAR. Stanje nakon promjene u Mdv repozitoriju, kao i u sistemskom katalogu, prikazano je na Slici 66. Možemo vidjeti da sistemski katalog vraća samo postojeći tip podatka stupca ZIP (gornji upit/skup rezultata), dok MdvDB vraća trenutno i povijestno stanje (donji upit/skup rezultata).

SQLQuery162.sql - ...NIRI\dsuotic (57)\*

```

SELECT [TABLE_CATALOG], [TABLE_NAME], [COLUMN_NAME], [ORDINAL_POSITION], [IS_NULLABLE], [DATA_TYPE]
FROM [JobDB].[INFORMATION_SCHEMA].[COLUMNS] where column_name='ZIP'

SELECT s.H_SRC_ATTR_SQN_SrcAttr, s.SrcAttrLDTs, s.SrcAttrLEDTs, s.SrcAttrPhysicalName, s.SrcAttrChangeType, c.DomainDataType
FROM [MdvDB].[dbo].[S_SRC_ATTRIBUTE] s INNER JOIN [MdvDB].[dbo].[H_SRC_ATTRIBUTE] h ON s.H_SRC_ATTR_SQN_SrcAttr=h.SQN_SrcAttr
INNER JOIN [MdvDB].[dbo].[L_DOM_SRCATTR] a ON a.H_SRC_ATTR_SQN_SrcAttr=h.SQN_SrcAttr INNER JOIN H_DOMAIN b
ON b.SQN_Domain=a.H_DOMAIN_SQN_Domain INNER JOIN S_DOMAIN c ON c.H_DOMAIN_SQN_Domain=b.SQN_Domain AND s.SrcAttrPhysicalName='ZIP'

```

100 %

Results Messages

	TABLE_CATALOG	TABLE_NAME	COLUMN_NAME	ORDINAL_POSITION	IS_NULLABLE	DATA_TYPE
1	JobDB	EMPLOYEE	ZIP	9	NO	nvarchar

---

	H_SRC_ATTR_SQN_SrcAttr	SrcAttrLDTs	SrcAttrLEDTs	SrcAttrPhysicalName	SrcAttrChangeType	DomainDataType
1	74	2016-10-13 08:49:03.000	2016-10-13 18:13:43.000	ZIP	NULL	numeric
2	157	2016-10-13 18:13:43.000	NULL	ZIP	Change attribute type	nvarchar
3	157	2016-10-13 18:13:43.000	NULL	ZIP	Change attribute type	sysname
4	157	2016-10-13 18:13:43.000	NULL	ZIP	Change attribute type	nvarchar

### Slika 66. Stanje u MdvDB i sistemskom katalogu nakon promjene tipa podatka atributa

#### Brisanje atributa iz relacije, B<sub>A</sub>

U TrainingDB izbrisat ćemo stupac DriversLicence (u tablici EMPLOYEE). Stanje nakon promjene u MdvDB repozitoriju, kao i u sistemskom katalogu, prikazano je na Slici 67. Možemo vidjeti da MdvDB čuva sve zapise o izbrisanoj stupcu - vrijednost stupca SrcAttrLEDTs postaje datum brisanja, a stupac SrcAttrChangeType bilježi promjenu koja je utjecala na zapis (gornji upit/skup rezultata), dok sistemski katalog ne čuva zapis o stupcu DriversLicence – kao da nikad nije ni postojao (donji upit/skup rezultata).

SQLQuery7.sql - OD...NIRI\dsuotic (58)\* SQLQuery5.sql - OD...NIRI\dsuotic (54)\* SQLQuery4.sql - OD...NIRI\dsuotic (51)\*

```

SELECT [H_SRC_ATTR_SQN_SrcAttr], [SrcAttrLDTs], [SrcAttrLEDTs], [SrcAttrRSRC], [SrcAttrPhysicalName], [SrcAttrChangeType]
FROM [MdvDB].[dbo].[S_SRC_ATTRIBUTE] where [SrcAttrPhysicalName]='DriversLicence'

SELECT [TABLE_CATALOG], [TABLE_SCHEMA], [TABLE_NAME], [COLUMN_NAME]
FROM [TrainingDB].[INFORMATION_SCHEMA].[COLUMNS] where TABLE_NAME='EMPLOYEE'

```

100 %

Results Messages

	H_SRC_ATTR_SQN_SrcAttr	SrcAttrLDTs	SrcAttrLEDTs	SrcAttrRSRC	SrcAttrPhysicalName	SrcAttrChangeType
1	89	2016-10-13 16:13:12.000	2016-10-13 19:33:12.000	System Catalog	DriversLicence	Add attribute
2	89	2016-10-13 19:33:12.000	2016-10-13 19:33:12.000	System Catalog	DriversLicence	Delete attribute

---

	TABLE_CATALOG	TABLE_SCHEMA	TABLE_NAME	COLUMN_NAME
1	TrainingDB	dbo	EMPLOYEE	EmployeeID
2	TrainingDB	dbo	EMPLOYEE	EMPLOYEE.First Name
3	TrainingDB	dbo	EMPLOYEE	Surname
4	TrainingDB	dbo	EMPLOYEE	Title
5	TrainingDB	dbo	EMPLOYEE	Gender
6	TrainingDB	dbo	EMPLOYEE	Marital Status
7	TrainingDB	dbo	EMPLOYEE	BirthDate
8	TrainingDB	dbo	EMPLOYEE	Street Number
9	TrainingDB	dbo	EMPLOYEE	Street Name
10	TrainingDB	dbo	EMPLOYEE	ZIP
11	TrainingDB	dbo	EMPLOYEE	City
12	TrainingDB	dbo	EMPLOYEE	County
13	TrainingDB	dbo	EMPLOYEE	Country
14	TrainingDB	dbo	EMPLOYEE	Email Address
15	TrainingDB	dbo	EMPLOYEE	Phone Number

### Slika 67. Stanje u MdvDB i sistemskom katalogu nakon brisanja atributa



I ostaju još 4 promjene više razine, za koje smo u Poglavljima 4.2.1 i 4.3 dokazali da se sastoje od određenih kombinacija ovdje implementiranih osnovnih promjena te je njihovo provođenje ekvivalentno već predstavljanim primjerima (stoga neće ponovno biti prikazane kroz primjere). U Tablici 25 nalazi se redosljed promjena u izvorima za svaku od ove 4 složene promjene (iz Poglavlja 4.3), dok se promjene u SdvDB i PmdvDB dalje implementiraju prema prethodno opisanim primjerima.

**Tablica 25. Redosljed promjena u izvorima za složene promjene**

SLOŽENA PROMJENA U IZVORU	PROVEDBA U IZVORIMA
Pretvorba relacije u vezu, I <sub>RV</sub>	<ol style="list-style-type: none"> <li>1. Brisanje veze između relacija, B<sub>V</sub> (is supervised by)</li> <li>2. Brisanje relacije, B<sub>R</sub> (SUPERVISOR)</li> <li>3. Dodavanje atributa u relaciju, D<sub>A</sub> (EMPLOYEE -&gt; SupervisorID)</li> <li>4. Dodavanje veze između relacija, D<sub>V</sub> (is supervised by)</li> </ol>
Pretvorba veze u relaciju, I <sub>VR</sub>	<ol style="list-style-type: none"> <li>1. Brisanje veze između relacija, B<sub>V</sub> (works on)</li> <li>2. Dodavanje nove relacije, D<sub>R</sub> (WORKS_ON)</li> <li>3. Dodavanje atributa u relaciju, D<sub>A</sub> (svi atributi iz WORKS_ON)</li> <li>4. Dodavanje veze između relacija, D<sub>V</sub> (veza na EMPLOYEE)</li> <li>5. Dodavanje veze između relacija, D<sub>V</sub> (veza na PROJECT)</li> </ol>
Pretvorba atributa u relaciju, I <sub>AR</sub>	<ol style="list-style-type: none"> <li>1. Brisanje atributa iz relacije, B<sub>A</sub> (Gender)</li> <li>2. Dodavanje nove relacije, D<sub>R</sub> (GENDER)</li> <li>3. Dodavanje atributa u relaciju, D<sub>A</sub> (EMPLOYEE-&gt; GenderID)</li> <li>4. Dodavanje veze između relacija, D<sub>V</sub> (EMPLOYEE has GENDER)</li> </ol>
Pretvorba relacije u atribut, I <sub>RA</sub>	<ol style="list-style-type: none"> <li>1. Brisanje atributa iz relacije, B<sub>A</sub> (MaritalStatusID)</li> </ol>

	<p>2. Brisanje veze između relacija, B<sub>V</sub> (EMPLOYEE-&gt; MARITAL_STATUS)</p> <p>3. Brisanje relacije, B<sub>R</sub> (EMPLOYEE has MARITAL_STATUS)</p> <p>4. Dodavanje atributa u relaciju, D<sub>A</sub> (EMPLOYEE-&gt; MaritalStatus)</p>
--	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

### 5.3. Dokaz hipoteze H1

Hipoteza H1 kaže da će MDV repozitorij meta-podataka zasnovan na Data Vault (DV) modelu integrirati skladište podataka (SP) i upravljanje matičnim podacima (UMP). Hipoteza se smatra potvrđenom ukoliko se definira i razvije MDV model (što je definirano i dokazano u Poglavljima 3 i 4) te se prema razvijenom MDV modelu izradi verzija prototipa sustava u kojem: a) MDV repozitorij meta-podataka prikuplja i pohranjuje meta-podatke o povezivanju SDV i PMDV shema i b) definirani upiti nad MdmDB i DmDB vraćaju isti skup rezultata.

Kao što je već opisano u Poglavlju 5.1, PmdvDB i MdmDB prikupljaju i čuvaju poslovno prilagođene matične podatke – PmdvDB prema DV modelu, a MdmDB prema dimenzijskom modelu. PmdvDB svoje podatke prosljeđuje u MdmDB koji predstavlja UMP, a MdmDB zatim svoje podatke prosljeđuje u DMDB koji predstavlja područno SP. Dodatno, i SdvDB i PmdvDB također predstavljaju centralno SP za pohranu, integraciju i čuvanje trenutnih i povijesnih podataka (i svih promjena nad njima). Na ovaj način, kako je i praktično prikazano u Poglavljima 5.1 i 5.2, integrirali smo SP i UMP sustav u našem prototipu. Da bismo dokazali da je ova integracija valjana (te da vrijedi hipoteza H1) proveli smo dva testa na podatkovnoj razini cijelog prototipa:

- a) upit nad MdvDB repozitorijem meta-podataka koji nam vraća podatke o povezivanju SDV i PMDV shema
- b) dva ekvivalentna upita nad MdmDB i DmDB koji nam vraćaju isti skup rezultata

Da bismo proveli test a) definirali smo upit nad MdvDB repozitorijem koji bi nam trebao vratiti podatke o povezivanju struktura iz SdvDB s elementima iz PmdvDB, odnosno podatke o tome koji hub (ili bilo koji drugi element) iz SdvDB postaje hub (ili bilo koji drugi element) u PmdvDB.

Takva dva upita (za hubove i satellite) i njihove skupove rezultata možemo vidjeti na Slikama 68 i 69. Ovim upitima nad MdvDB možemo pratiti integraciju struktura koje čuvaju “sirove” podatke u SdvDB sa strukturama koje čuvaju matične podatke u PmdvDB (što su zapravo i matični podaci u MdmDB u ovoj verziji prototipa). Također, vidimo da se osim integracije ovih sustava prate i promjene nad strukturama (stupci ChangeType i LEDTS).

SQLQuery98.sql - O...NIRI\dsubotic (57)\* x SQLQuery62.sql - O...NIRI\dsubotic (58)\*

```

SELECT m.[SQN_MasterHub], m.[MasterHubLDTs], m.[H_HUB_SQN_HubBase], m.[H_HUB_SQN_HubMaster]
, sa.HubName as BaseHubName, sb.HubName as MasterHubName, sb.HubSchemaName as SchemaVersion, sb.ChangeType
FROM [MdvDB].[dbo].[SAL_MASTER_HUB] m INNER JOIN H_HUB a ON a.SQN_Hub=m.H_HUB_SQN_HubBase
INNER JOIN H_HUB b ON b.SQN_Hub=m.H_HUB_SQN_HubMaster INNER JOIN S_HUB_DEF sa ON sa.H_HUB_SQN_Hub=a.SQN_Hub
INNER JOIN S_HUB_DEF sb ON sb.H_HUB_SQN_Hub=b.SQN_Hub

```

	SQN_MasterHub	MasterHubLDTs	H_HUB_SQN_HubBase	H_HUB_SQN_HubMaster	BaseHubName	MasterHubName	SchemaVersion	ChangeType
1	1	2016-10-11 19:01:19.000	1	2	H_CERTIFICATE_TRAINING	H_CERTIFICATE_TRAINING_MASTER	NULL	NULL
2	2	2016-10-11 19:01:19.000	3	5	H_EMPLOYEE	H_EMPLOYEE_MASTER	NULL	NULL
3	3	2016-10-11 19:01:19.000	6	7	H_MARITAL_STATUS	H_MARITAL_STATUS_MASTER	NULL	NULL
4	4	2016-10-11 19:01:19.000	8	9	H_PROJECT	H_PROJECT_MASTER	NULL	NULL
5	5	2016-10-11 19:01:19.000	10	11	H_SUPERVISOR	H_SUPERVISOR_MASTER	NULL	NULL
6	11	2016-10-13 12:17:22.000	23	24	H_TRAINER	H_TRAINER_MASTER	2	Add relation

**Slika 68. Integracija SdvDB i PmdvDB hubova u MdvDB**

SQLQuery98.sql - O...NIRI\dsubotic (57)\* x SQLQuery62.sql - O...NIRI\dsubotic (58)\*

```

SELECT distinct [MasterSatLDTs], [H_SAT_SQN_SatBase], [H_SAT_SQN_SatMaster]
, sa.SatName as BaseSatName, sb.SatName as MasterSatName, sb.SatSchemaName as SchemaVersion, sb.ChangeType
FROM [MdvDB].[dbo].[SAL_MASTER_SAT] m INNER JOIN H_SATELLITE a ON a.SQN_Satellite=m.H_SAT_SQN_SatBase
INNER JOIN H_SATELLITE b ON b.SQN_Satellite=m.H_SAT_SQN_SatMaster INNER JOIN S_SAT_DEF sa ON sa.H_SAT_SQN_Sat=a.SQN_Satellite
INNER JOIN S_SAT_DEF sb ON sb.H_SAT_SQN_Sat=b.SQN_Satellite

```

	MasterSatLDTs	H_SAT_SQN_SatBase	H_SAT_SQN_SatMaster	BaseSatName	MasterSatName	SchemaVersion	ChangeType
1	2016-10-11 19:13:54.000	1	2	S_CERTIFICATE_TRAINING	S_CERTIFICATE_TRAINING_MASTER	NULL	NULL
2	2016-10-11 19:13:54.000	3	4	S_COMPETENCE	S_COMPETENCE_MASTER	NULL	NULL
3	2016-10-11 19:13:54.000	5	6	S_EMPLOYEE	S_EMPLOYEE_MASTER	NULL	NULL
4	2016-10-11 19:13:54.000	5	7	S_EMPLOYEE	S_EMPLOYEE_MASTERALL	NULL	NULL
5	2016-10-11 19:13:54.000	9	6	S_EMPLOYEEETRA	S_EMPLOYEE_MASTER	NULL	NULL
6	2016-10-11 19:13:54.000	9	7	S_EMPLOYEEETRA	S_EMPLOYEE_MASTERALL	NULL	NULL
7	2016-10-11 19:13:54.000	10	11	S_MARITAL_STATUS	S_MARITAL_STATUS_MASTER	NULL	NULL
8	2016-10-11 19:13:54.000	12	13	S_PARTICIPATES	S_PARTICIPATES_MASTER	NULL	NULL
9	2016-10-11 19:13:54.000	15	16	S_PROJECT	S_PROJECT_MASTER	NULL	NULL
10	2016-10-11 19:13:54.000	17	18	S_SUPERVISOR	S_SUPERVISOR_MASTER	NULL	NULL
11	2016-10-13 12:17:22.000	28	29	S_LEADS	S_LEADS_MASTER	2	Add relation
12	2016-10-13 12:17:22.000	39	40	S_TRAINER	S_TRAINER_MASTER	2	Add relation

**Slika 69. Integracija SdvDB i PmdvDB satelita u MdvDB**

Dodatno, MdvDB na isti način prati povezivanja između izvora podataka i SdvDB, što možemo vidjeti na Slici 70 (prikaz samo za hubove u SdvDB, ali prati se i za ostale strukture) te će na isti način pratiti i integracije između ostalih baza podataka u sljedećoj verziji prototipa (u budućnosti).

SQLQuery95.sql - O...NIR\dsbotoc (53)) SQLQuery62.sql - O...NIR\dsbotoc (58))\* X

```

SELECT distinct a.H_SRC_REL_SQN_SrcRel, a.SrcRelationName, e.H_HUB_SQN_Hub, e.HubName, e.HubDefLDTS
FROM S_SRC_RELATION a INNER JOIN H_SRC_RELATION b ON a.H_SRC_REL_SQN_SrcRel=b.SQN_SrcRelation
INNER JOIN L_RELATION_HSL c ON c.H_REL_SQNRelation=b.SQN_SrcRelation INNER JOIN H_HUB d ON d.SQN_Hub=c.H_HUB_SQN_Hub
INNER JOIN S_HUB_DEF e ON e.H_HUB_SQN_Hub=d.SQN_Hub

```

100 %

Results Messages

	H_SRC_REL_SQN_SrcRel	SrcRelationName	H_HUB_SQN_Hub	HubName	HubDefLDTS
1	2	CERTIFICATE_TRAINING	1	H_CERTIFICATE_TRAINING	2016-10-12 12:59:27.000
2	2	TRAINING	1	H_CERTIFICATE_TRAINING	2016-10-12 12:59:27.000
3	4	EMPLOYEE	3	H_EMPLOYEE	2016-10-12 12:59:27.000
4	5	EMPLOYEE	3	H_EMPLOYEE	2016-10-12 12:59:27.000
5	6	MARITAL_STATUS	6	H_MARITAL_STATUS	2016-10-12 12:59:27.000
6	8	PROJECT	8	H_PROJECT	2016-10-12 12:59:27.000
7	9	SUPERVISOR	10	H_SUPERVISOR	2016-10-12 12:59:27.000
8	10	TRAINER	23	H_TRAINER	2016-10-13 12:17:22.000

**Slika 70. Integracija izvora i SdvDB hubova u MdvDB**

Obzirom da iz prikazanih i provedenih upita možemo vidjeti kako je nastao PmdvDB iz SdvDB te možemo pratiti trenutno stanje i povijest ove integracije, zaključujemo da je provedbom testa a) hipoteza H1 dokazana.

Da bismo proveli test b), u izvoru JobDB izvršili smo promjenu prezimena jednog zaposlenika – Zaposlenica Desiree Farmer postaje Desiree Farmer-Matthews. Zatim smo tu promjenu propagirali u SdvDB (pokrenuli smo izvršavanje ETL paketa SdvDBLoad) te u PmdvDB (ETL paketom PmdvDBLoad). Podatkovno stanje u SdvDB i PmdvDB nakon ove promjene prikazano je na Slici 71. Možemo vidjeti da i SdvDB i PmdvDB čuvaju i stari i novi zapis, s time da su u PmdvDB podaci dodatno integrirani i pripremljeni za punjenje u MdmDB (npr. stupac MaritalStatus je ovdje unutar tablice EMPLOYEE, dok je u SdvDB on u zasebnoj tablici vezan na EMPLOYEE).

ODI-0422-102.Pm...LOVEE\_MASTERALL SQLQuery2.sql - OD...NIRI(dsubotic (52))

```

SELECT [H_EMPLOYEE_SQN_Employee] as SQN, [EmployeeLDTs] ,[Name] ,[Surname],[Title],[Gender]
,[BirthDate],[StreetNumber],[StreetName],[ZIP],[City],[County],[Country]
,[EmailAddress],[PhoneNumber],[EmployeeLEDTs],[EmployeeRSRC]
FROM [SdvDB].[dbo].[S_EMPLOYEE] WHERE [H_EMPLOYEE_SQN_Employee]=11

SELECT [H_EMPLOYEE_SQN_Employee] as SQN, [EmpMastAllLDTs],[Name],[Surname],[Title],[Gender],[MaritalStatus],[BirthDate],[StreetNumber] as StrNum
,[StreetName] as StrName,[ZIP],[City],[County],[Country],[EmailAddress],[PhoneNumber],[EmpMastAllLEDTs],[EmpMastAllRSRC]
FROM [PmdvDB].[dbo].[S_EMPLOYEE_MASTERALL] WHERE [H_EMPLOYEE_SQN_Employee]=11

```

100 %

Results Messages

SQN	EmployeeLDTs	Name	Surname	Title	Gender	BirthDate	StreetNumber	StreetName	ZIP	City	County	Country	EmailAddress	PhoneNumber	EmployeeLEDTs	EmployeeRSRC	
1	11	2016-10-12 15:19:41.000	Desiree	Famer	Ms.	f	1955-07-02	6d1	Midway Park	33252	Fort Lauderdale	Florida	United States	Desire.FARME5495@yahoo.com	(424) 493-5312	2016-10-14 11:02:21.000	Sat Employee JobDB Load IU
2	11	2016-10-14 11:02:21.000	Desiree	Famer-Matthews	Ms.	f	1955-07-02	6d1	Midway Park	33252	Fort Lauderdale	Florida	United States	Desire.FARME5495@yahoo.com	(424) 493-5312	NULL	Sat Employee JobDB Load IU

SQN	EmpMastAllLDTs	Name	Surname	Title	Gender	MaritalStatus	BirthDate	StrNum	StrName	ZIP	City	County	Country	EmailAddress	PhoneNumber	EmpMastAllLEDTs	EmpMastAllRSRC	
1	11	2016-10-14 10:58:45.000	Desiree	Famer	Ms.	f	Married	1955-07-02	6d1	Midway Park	33252	Fort Lauderdale	Florida	United States	Desire.FARME5495@yahoo.com	(424) 493-5312	2016-10-14 11:02:36.000	Sat Employee Master Load
2	11	2016-10-14 11:02:36.000	Desiree	Famer-Matthews	Ms.	f	Married	1955-07-02	6d1	Midway Park	33252	Fort Lauderdale	Florida	United States	Desire.FARME5495@yahoo.com	(424) 493-5312	NULL	Sat Employee Master Load

**Slika 71. Stanje u SdvDB i PmdvDB nakon podatkovne promjene u JobDB izvoru**

Na Slici 72 vidimo stanje u MdmDB, nakon proslijeđivanja ove podatkovne promjene. Možemo vidjeti da MdmDB matične podatke o zaposlenicima čuva u tipu 2 sporo mijenjajućih dimenzija [43], što znači da pamti i stari i novi zapis. Ovo čak ovdje nije nužno (moguće je realizirati MdmDB da čuva povijest u tipu 1 – da jednostavno promjenu izvrši na postojećem zapisu), iz razloga što SdvDB i PmdvDB zajedno čuvaju potpunu povijest podataka te je moguće u bilo kojem trenutku izgraditi bilo koju verziju matičnih podataka u MdmDB, na temelju podataka iz PmdvDB i SdvDB.

SQLQuery15.sql - O...NIRI(dsubotic (57)) SQLQuery13.sql - O...NIRI(dsubotic (72)) SQLQuery2.sql - OD...NIRI(dsubotic (52))

```

SELECT [SQM_mdm_Employee] as SQN, [EmployeeID] ,[Name] ,[Surname],[Title],[Gender],[MaritalStatus],[BirthDate],[StreetNumber],[StreetName],[ZIP]
,[City],[County],[Country],[EmailAddress],[RecordStartDate],[RecordEndDate],[ETL_LoadJob],[RecordSource]
FROM [MdmDB].[dbo].[MDM_EMPLOYEE] where EmployeeID=110

```

100 %

Results Messages

SQN	EmployeeID	Name	Surname	Title	Gender	MaritalStatus	BirthDate	StreetNumber	StreetName	ZIP	City	County	Country	EmailAddress	PhoneNumber	RecordStartDate	RecordEndDate	
1	11	110	Desiree	Famer	Ms.	f	1955-07-02	6d1	Midway Park	33252	Fort Lauderdale	Florida	United States	Desire.FARME5495@yahoo.com	(424) 493-5312	2016-10-10 00:00:00.000	2016-10-14 11:42:55.000	
2	11	110	Desiree	Famer-Matthews	Ms.	f	Married	1955-07-02	6d1	Midway Park	33252	Fort Lauderdale	Florida	United States	Desire.FARME5495@yahoo.com	(424) 493-5312	2016-10-14 11:42:55.000	NULL

**Slika 72. Stanje u MdmDB poslije podatkovne promjene**

Na Slici 73 prikazana su 2 ekvivalentna upita nad MdmDB i DmDB, koji traže sve podatke o djelatnici sa šifrom 110 (što je u ovome primjeru Desiree). Vidimo da upiti vraćaju isti skup rezultata (2 retka – staro i novo prezime) u MdmDB (gornji upit/skup rezultata) i DmDB (donji upit/skup rezultata). Na Slici 74 prikazana su također dva ekvivalentna upita nad MdmDB i DmDB koji traže ukupan broj zapisa u obje baze podataka te vraćaju isti rezultat. Ovime zaključujemo da je testom b) dokazana hipoteza H1. Dodatno, na temelju izrađenog prototipa i provedena gore navedena 2 testa (a i b) smatramo da je hipoteza H1 dokazana te su SP i UMP uspješno integrirani.

SQLQuery19.sql - O...NIR\dsbotoc (61)\*

```

SELECT [SQM_mdm_Employee] as SQM, [EmployeeID], [Name], [Surname], [Title], [Gender], [MaritalStatus], [BirthDate], [StreetNumber], [StreetName], [ZIP], [City], [County], [Country], [EmailAddress], [PhoneNumber], [RecordStartDate], [RecordEndDate], [ETL_LoadJob], [RecordSource]
FROM [MdmDB].[dbo].[MDM_EMPLOYEE] where EmployeeID=118

SELECT [SQM_Dim_Employee] as SQM, [EmployeeID], [Name], [Surname], [Title], [Gender], [MaritalStatus], [BirthDate], [StreetNumber], [StreetName], [ZIP], [City], [County], [Country], [EmailAddress], [PhoneNumber], [RecordStartDate], [RecordEndDate], [ETL_LoadJob], [RecordSource]
FROM [DmDB].[dbo].[DIM_EMPLOYEE] where EmployeeID=118

```

SQM	EmployeeID	Name	Surname	Title	Gender	MaritalStatus	BirthDate	StreetNumber	StreetName	ZIP	City	County	Country	EmailAddress	PhoneNumber	RecordStartDateE	RecordEndDate	ETL_LoadJob
11	110	Deesre	Farmer	Ms.	f	Married	1955-07-02	6d1	Mdway Park	33252	Fort Lauderdale	Florida	United States	Deesre.FARME5495@yahoo.com	(424) 493-5312	2016-10-10 00:00:00.000	2016-10-14 11:42:55.000	MdmDBLoa
11	110	Deesre	Farmer-Matthews	Ms.	f	Married	1955-07-02	6d1	Mdway Park	33252	Fort Lauderdale	Florida	United States	Deesre.FARME5495@yahoo.com	(424) 493-5312	2016-10-14 12:02:55.000	NULL	MdmDBLoa

SQM	EmployeeID	Name	Surname	Title	Gender	MaritalStatus	BirthDate	StreetNumber	StreetName	ZIP	City	County	Country	EmailAddress	PhoneNumber	RecordStartDate	RecordEndDate	ETL_LoadJob
11	110	Deesre	Farmer	Ms.	f	Married	1955-07-02	6d1	Mdway Park	33252	Fort Lauderdale	Florida	United States	Deesre.FARME5495@yahoo.com	(424) 493-5312	2016-10-10 00:00:00.000	2016-10-14 12:02:55.000	DmDBLoad
11	110	Deesre	Farmer-Matthews	Ms.	f	Married	1955-07-02	6d1	Mdway Park	33252	Fort Lauderdale	Florida	United States	Deesre.FARME5495@yahoo.com	(424) 493-5312	2016-10-14 12:02:55.000	NULL	DmDBLoad

Slika 73. Stanje u DmDB (i MdmDB) poslije podatkovne promjene

SQLQuery19.sql - O...NIR\dsbotoc (61)\*

```

SELECT COUNT(*) as MDM_rows FROM [MdmDB].[dbo].[MDM_EMPLOYEE];

SELECT COUNT(*) as DIM_rows FROM [DmDB].[dbo].[DIM_EMPLOYEE];

```

MDM_rows
121

DIM_rows
121

Slika 74. Broj redaka u MdmDB i DmDB poslije podatkovne promjene

## 5.4. Upiti za prikaz korisnosti rješenja

Da bismo dodatno potvrdili korisnost predloženog MDV repozitorija za proširenje standardnog relacijskog kataloga, kreirali smo i 7 dodatnih upita nad MdvDB. Obzirom da MdvDB čuva potpunu povijest meta-podataka, odnosno sve podatkovne i strukturalne promjene nad SDV/PMDV shemom, kao i shemama izvora podataka – ovim upitima možemo provjeravati trenutno stanje izvorišnih i SDV/PMDV shema te ispisati povijest svih (ili određenih) promjena. Na taj način možemo dobiti informacije o evoluciji poslovne sheme i podataka, kao i vršiti postupak revizije nad podacima i shemom. Upiti su definirani u nastavku u odgovarajućim Tablicama (26-32), prema [71]. Rezultati svakog pojedinog upita prikazani su odgovarajućim Slikama (75-82).

**Tablica 26. Upit 1 – Koje tablice su i kada u svim izvorima promijenile naziv i kako su se prije zvale?**

```

WITH upit AS (
SELECT p.DsrcName, h.SQN_SrcRelation as SQN, s.SrcRelationName,
s.SrcRelationSchemaName as ShVersion,
s.ChangeType, s.SrcRelationLDTS as LDTS, s.SrcRelationLEDTS as LEDTS
FROM H_SRC_RELATION h INNER JOIN S_SRC_RELATION s ON
h.SQN_SrcRelation=s.H_SRC_REL_SQN_SrcRel
INNER JOIN L_DSRC_REL l ON l.H_SRC_RELATION_SQN_SrcRelation=h.SQN_SrcRelation
INNER JOIN H_DATA_SOURCE d ON d.SQN_DataSource=l.H_DATA_SOURCE_SQN_DataSource
INNER JOIN S_DATA_SOURCE p ON p.H_DATA_SOURCE_SQN_DataSource=d.SQN_DataSource
)
SELECT a.DsrcName, a.SQN, a.SrcRelationName, a.ShVersion, a.ChangeType, a.LDTS,
a.LEDTS
FROM upit a INNER JOIN upit b ON a.SQN=b.SQN WHERE b.ChangeType='Rename relation'

```

The screenshot shows a SQL query window with the following text:

```

WITH upit AS (
SELECT p.DSrcName, h.SQN_SrcRelation as SQN, s.SrcRelationName, s.SrcRelationSchemaName as ShVersion,
s.ChangeType, s.SrcRelationLDTS as LDTS, s.SrcRelationLEDTS as LEDTS
FROM H_SRC_RELATION h INNER JOIN S_SRC_RELATION s ON h.SQN_SrcRelation=s.H_SRC_REL_SQN_SrcRel
INNER JOIN L_DSRC_REL l ON l.H_SRC_RELATION_SQN_SrcRelation=h.SQN_SrcRelation
INNER JOIN H_DATA_SOURCE d ON d.SQN_DataSource=l.H_DATA_SOURCE_SQN_DataSource
INNER JOIN S_DATA_SOURCE p ON p.H_DATA_SOURCE_SQN_DataSource=d.SQN_DataSource
)
SELECT a.DSrcName, a.SQN, a.SrcRelationName, a.ShVersion, a.ChangeType, a.LDTS, a.LEDTS
FROM upit a INNER JOIN upit b ON a.SQN=b.SQN WHERE b.ChangeType='Rename relation'

```

Below the query window, the 'Results' tab is active, displaying a table with the following data:

	DsrcName	SQN	SrcRelationName	ShVersion	ChangeType	LDTS	LEDTS
1	TrainingDB	2	CERTIFICATE_TRAINING	NULL	NULL	2016-10-12 20:42:22.000	2016-10-13 11:02:43.000
2	TrainingDB	2	TRAINING	2	Rename relation	2016-10-13 11:02:43.000	NULL

**Slika 75. Rezultati Upita 1**

Upit 1 nam vraća popis svih izvorišnih tablica (u svim izvorima integriranim u sustav skladišta podataka) nad kojima je provedeno preimenovanje relacije. Iz skupa rezultata možemo vidjeti koje su to tablice, kako su se prije zvala, kako se sada zovu i kada je došlo do te promjene.

Upit 2 nam vraća popis svih tablica koje više ne postoje u izvorima, a u nekome trenutku su postojale (a) ili su postojale samo u izvorišnoj verziji sheme (b), kod početne integracije izvora u sustav skladišta podataka). Možemo vidjeti iz skupa rezultata (a) da je samo jedna tablica izbrisana dosada, a iz skupa rezultata (b) vidimo da nijedna tablica nije izbrisana, a da je bila u početnoj shemi izvora (TRAINER je dodana u izvor tek u 2. verziji sheme).

**Tablica 27. Upit 2 - Koje tablice trenutno ne postoje u izvorima, a postojale su u: a) bilo kojem periodu ili b) početnoj shemi?**

```
-- Koje tablice trenutno ne postoje u izvorima, a postojale su u bilo kojem periodu sustava?
```

```
WITH upit AS (SELECT p.DSrcName, h.SQN_SrcRelation as SQN, s.SrcRelationName,
s.SrcRelationSchemaName as ShVersion,
s.ChangeType, s.SrcRelationLDTS as LDTS, s.SrcRelationLEDTS as LEDTS
FROM H_SRC_RELATION h INNER JOIN S_SRC_RELATION s ON
h.SQN_SrcRelation=s.H_SRC_REL_SQN_SrcRel
INNER JOIN L_DSRC_REL l ON l.H_SRC_RELATION_SQN_SrcRelation=h.SQN_SrcRelation
INNER JOIN H_DATA_SOURCE d ON d.SQN_DataSource=l.H_DATA_SOURCE_SQN_DataSource
INNER JOIN S_DATA_SOURCE p ON p.H_DATA_SOURCE_SQN_DataSource=d.SQN_DataSource)
SELECT a.DSrcName, a.SQN, a.SrcRelationName, a.ShVersion, a.ChangeType, a.LDTS,
a.LEDTS
FROM upit a INNER JOIN upit b ON a.SQN=b.SQN WHERE b.ChangeType='Delete relation'
```

```
--Koje tablice trenutno ne postoje u izvorima, a postojale su u početnoj shemi
(prilikom prvog punjenja)?
```

```
WITH upit AS (SELECT p.DSrcName, h.SQN_SrcRelation as SQN, s.SrcRelationName,
s.SrcRelationSchemaName as ShVersion,
s.ChangeType, s.SrcRelationLDTS as LDTS, s.SrcRelationLEDTS as LEDTS
FROM H_SRC_RELATION h INNER JOIN S_SRC_RELATION s ON
h.SQN_SrcRelation=s.H_SRC_REL_SQN_SrcRel
INNER JOIN L_DSRC_REL l ON l.H_SRC_RELATION_SQN_SrcRelation=h.SQN_SrcRelation
INNER JOIN H_DATA_SOURCE d ON d.SQN_DataSource=l.H_DATA_SOURCE_SQN_DataSource
INNER JOIN S_DATA_SOURCE p ON p.H_DATA_SOURCE_SQN_DataSource=d.SQN_DataSource)
SELECT a.DSrcName, a.SQN, a.SrcRelationName, a.ShVersion, a.ChangeType, a.LDTS,
a.LEDTS
FROM upit a INNER JOIN upit b ON a.SQN=b.SQN WHERE b.ChangeType='Delete relation'
AND b.ShVersion IS NULL
```

The screenshot shows a SQL query window with the following code:

```
--Koje tablice trenutno ne postoje u izvorima, a postojale su u bilo kojem periodu sustava?
WITH upit AS (
SELECT p.DSrcName, h.SQN_SrcRelation as SQN, s.SrcRelationName, s.SrcRelationSchemaName as ShVersion,
s.ChangeType, s.SrcRelationLDTS as LDTS, s.SrcRelationLEDTS as LEDTS
FROM H_SRC_RELATION h INNER JOIN S_SRC_RELATION s ON h.SQN_SrcRelation=s.H_SRC_REL_SQN_SrcRel
INNER JOIN L_DSRC_REL l ON l.H_SRC_RELATION_SQN_SrcRelation=h.SQN_SrcRelation
INNER JOIN H_DATA_SOURCE d ON d.SQN_DataSource=l.H_DATA_SOURCE_SQN_DataSource
INNER JOIN S_DATA_SOURCE p ON p.H_DATA_SOURCE_SQN_DataSource=d.SQN_DataSource
)
SELECT a.DSrcName, a.SQN, a.SrcRelationName, a.ShVersion, a.ChangeType, a.LDTS, a.LEDTS
FROM upit a INNER JOIN upit b ON a.SQN=b.SQN WHERE b.ChangeType='Delete relation'
```

Below the query window, the 'Results' pane shows a table with the following data:

	DSrcName	SQN	SrcRelationName	ShVersion	ChangeType	LDTS	LEDTS
1	TrainingDB	10	TRAINER	NULL	NULL	2016-10-12 13:38:28.000	2016-10-13 14:32:14.000
2	TrainingDB	10	TRAINER	3	Delete relation	2016-10-13 14:32:14.000	2016-10-13 14:32:14.000

**Slika 76. Rezultati Upita 2 a)**



```

ODJ-O422-102.Mdv...o.S_SRC_RELATION  SQLQuery22.sql - O...NIRI\dsubotic (56)* x
--Koje tablice trenutno ne postoje u izvorima, a postojale su u početnoj shemi (prilikom prvog punjenja)?
WITH upit AS (
SELECT p.DSrcName, h.SQN_SrcRelation as SQN, s.SrcRelationName, s.SrcRelationSchemaName as ShVersion,
s.ChangeType, s.SrcRelationLDTs as LDTs, s.SrcRelationLEDTs as LEDTs
FROM H_SRC_RELATION h INNER JOIN S_SRC_RELATION s ON h.SQN_SrcRelation=s.H_SRC_REL_SQN_SrcRel
INNER JOIN L_DSRC_REL l ON l.H_SRC_RELATION_SQN_SrcRelation=h.SQN_SrcRelation
INNER JOIN H_DATA_SOURCE d ON d.SQN_DataSource=l.H_DATA_SOURCE_SQN_DataSource
INNER JOIN S_DATA_SOURCE p ON p.H_DATA_SOURCE_SQN_DataSource=d.SQN_DataSource
)
SELECT a.DSrcName, a.SQN, a.SrcRelationName, a.ShVersion, a.ChangeType, a.LDTs, a.LEDTs
FROM upit a INNER JOIN upit b ON a.SQN=b.SQN WHERE b.ChangeType='Delete relation' AND b.ShVersion IS NULL

```

100 %

Results Messages

DSrcName	SQN	SrcRelationName	ShVersion	ChangeType	LDTs	LEDTs
----------	-----	-----------------	-----------	------------	------	-------

**Slika 77. Rezultati Upita 2 b)**

Upit 3 vraća povijest dodavanja novih hubova u SdvDB shemu, promatrano u odnosu na njenu početnu shemu. Vidimo da je ovaj upit, kao i njegov skup rezultata sličan Upitu 2, a razlika je u tome što Upit 3 vraća skup rezultata iz dijela MdvDB repozitorija koji prati stanje i povijest SdvDB i PmdvDB shema.

**Tablica 28. Upit 3 - Koji hubovi (i kada) su naknadno dodavani u SdvDB u odnosu na početnu shemu SdvDB?**

```

--Koji hubovi (i kada) su naknadno dodavani u SdvDB u odnosu na početnu shemu SdvDB?
SELECT distinct h.DSrcName, e.SrcRelationName, a.HubName, a.HubDefLDTs,
a.HubDefLEDTs, a.ChangeType,
a.HubSchemaName as SchemaVersion
FROM S_HUB_DEF a INNER JOIN H_HUB b ON a.H_HUB_SQN_Hub=b.SQN_Hub INNER JOIN
L_RELATION_HSL c
ON c.H_HUB_SQN_Hub=b.SQN_Hub INNER JOIN H_SRC_RELATION d ON
d.SQN_SrcRelation=c.H_REL_SQNRelation
INNER JOIN S_SRC_RELATION e ON e.H_SRC_REL_SQN_SrcRel=d.SQN_SrcRelation INNER JOIN
L_DSRC_REL f
ON f.H_SRC_RELATION_SQN_SrcRelation=d.SQN_SrcRelation INNER JOIN H_DATA_SOURCE g
ON g.SQN_DataSource=f.H_DATA_SOURCE_SQN_DataSource
INNER JOIN S_DATA_SOURCE h ON h.H_DATA_SOURCE_SQN_DataSource=g.SQN_DataSource
WHERE a.HubSchemaName IS NOT NULL

```

ODJ-O422-102.MdvDB - dbo.S\_HUB\_DEF SQLQuery22.sql - O...NIRI\dsbotoc (56)\* X

```
--Koji hubovi (i kada) su naknadno dodavani u SdvDB u odnosu na početnu shemu SvdDB?
SELECT distinct h.DSrcName, e.SrcRelationName, a.HubName, a.HubDefLDTS, a.HubDefLEDTS, a.ChangeType,
a.HubSchemaName as SchemaVersion
FROM S_HUB_DEF a INNER JOIN H_HUB b ON a.H_HUB_SQN_Hub=b.SQN_Hub INNER JOIN L_RELATION_HSL c
ON c.H_HUB_SQN_Hub=b.SQN_Hub INNER JOIN H_SRC_RELATION d ON d.SQN_SrcRelation=c.H_REL_SQNRelation
INNER JOIN S_SRC_RELATION e ON e.H_SRC_REL_SQN_SrcRel=d.SQN_SrcRelation INNER JOIN L_DSRC_REL f
ON f.H_SRC_RELATION_SQN_SrcRelation=d.SQN_SrcRelation INNER JOIN H_DATA_SOURCE g
ON g.SQN_DataSource=f.H_DATA_SOURCE_SQN_DataSource
INNER JOIN S_DATA_SOURCE h ON h.H_DATA_SOURCE_SQN_DataSource=g.SQN_DataSource
WHERE a.HubSchemaName IS NOT NULL
```

100 %

Results Messages

	DSrcName	SrcRelationName	HubName	HubDefLDTS	HubDefLEDTS	ChangeType	SchemaVersion
1	TrainingDB	TRAINER	H_TRAINER	2016-10-13 12:17:22.000	2016-10-13 14:32:14.000	Add relation	2
2	TrainingDB	TRAINER	H_TRAINER	2016-10-13 14:32:14.000	2016-10-13 14:32:14.000	Delete relation	3

### Slika 78. Rezultati Upita 3

Upit 4 vraća popis hubova i satelita koji su bili aktivni u istom vremenskom periodu. Ovo je temporalni upit za koji smo kreirali dvije pomoćne funkcije, koje vraćaju najraniji i najkasniji datum (minimum i maksimum od dva parametra tipa DATETIME). Iz skupa rezultata možemo vidjeti da primjerice H\_TRAINER i S\_TRAINER imaju isti LDTS pa im je početak važenja isti, ali imaju različit LEDTS (za S\_TRAINER u trenutku provođenja upita nije bio postavljen kraj važenja, već je on i dalje aktivan, dok je H\_TRAINER prestao biti aktivan 13. listopada 2016, u 14:32h) pa im je zajednički kraj važenja manji LEDTS (hubov LEDTS).

**Tablica 29. Upit 4 - Koji hubovi i sateliti su vrijedili (bili aktivni) u istom vremenskom periodu?**

```
CREATE FUNCTION dbo.minDate (@first DATETIME, @second DATETIME)
RETURNS DATETIME
AS BEGIN
RETURN CASE
WHEN @first<@second THEN @first ELSE @second
END;
END;

CREATE FUNCTION dbo.maxDate (@first DATETIME, @second DATETIME)
RETURNS DATETIME
AS BEGIN
RETURN CASE
WHEN @first>@second THEN @first ELSE @second
END;
END;

--Koji hubovi i sateliti su vrijedili (bili aktivni) u istom vremenskom periodu?
```

```

SELECT x.HubName, d.SatName, x.HubDefLDTs, x.HubDefLEDTs, d.SatDefLDTs,
d.SatDefLEDTs,
dbo.maxDate(x.HubDefLDTs, d.SatDefLDTs) as FromDate,
dbo.minDate((ISNULL(x.HubDefLEDTs, '9999-01-01')), (ISNULL(d.SatDefLEDTs, '9999-01-01'))) as ToDate
FROM S_HUB_DEF x INNER JOIN H_HUB a ON a.SQN_Hub=x.H_HUB_SQN_Hub INNER JOIN
L_HUB_SAT b
ON a.SQN_Hub=b.H_HUB_SQN_Hub INNER JOIN H_SATELLITE c ON
c.SQN_Satellite=b.H_SATELLITE_SQN_Satellite
INNER JOIN S_SAT_DEF d ON c.SQN_Satellite=d.H_SAT_SQN_Sat
AND dbo.maxDate(x.HubDefLDTs, d.SatDefLDTs)<dbo.minDate((ISNULL(x.HubDefLEDTs,
'9999-01-01')), (ISNULL(d.SatDefLEDTs, '9999-01-01')))

```

SQLQuery27.sql - O...NIR\dsubotic (57)\* SQLQuery22.sql - O...NIR\dsubotic (56)\* X

```

--Koji hubovi i sateliti su vrijedili (bili aktivne) u istom vremenskom periodu?
SELECT x.HubName, d.SatName, x.HubDefLDTs, x.HubDefLEDTs, d.SatDefLDTs, d.SatDefLEDTs,
dbo.maxDate(x.HubDefLDTs, d.SatDefLDTs) as FromDate,
dbo.minDate((ISNULL(x.HubDefLEDTs, '9999-01-01')), (ISNULL(d.SatDefLEDTs, '9999-01-01'))) as ToDate
FROM S_HUB_DEF x INNER JOIN H_HUB a ON a.SQN_Hub=x.H_HUB_SQN_Hub INNER JOIN L_HUB_SAT b
ON a.SQN_Hub=b.H_HUB_SQN_Hub INNER JOIN H_SATELLITE c ON c.SQN_Satellite=b.H_SATELLITE_SQN_Satellite
INNER JOIN S_SAT_DEF d ON c.SQN_Satellite=d.H_SAT_SQN_Sat
AND dbo.maxDate(x.HubDefLDTs, d.SatDefLDTs)<dbo.minDate((ISNULL(x.HubDefLEDTs, '9999-01-01')), (ISNULL(d.SatDefLEDTs, '9999-01-01')))

```

HubName	SatName	HubDefLDTs	HubDefLEDTs	SatDefLDTs	SatDefLEDTs	FromDate	ToDate
6	H_CERTIFICATE_TRAINING	S_COMPETENCE	2016-10-12 12:59:27.000	NULL	2016-10-12 12:49:59.000	NULL	2016-10-12 12:59:27.000
7	H_EMPLOYEE_MASTER	S_EMPLOYEE_MASTER	2016-10-12 12:59:27.000	NULL	2016-10-12 12:49:59.000	NULL	2016-10-12 12:59:27.000
8	H_EMPLOYEE_MASTER	S_EMPLOYEE_MASTER	2016-10-12 12:59:27.000	NULL	2016-10-12 12:49:59.000	NULL	2016-10-12 12:59:27.000
9	H_EMPLOYEE_MASTER	S_EMPLOYEE_MASTERALL	2016-10-12 12:59:27.000	NULL	2016-10-12 12:49:59.000	NULL	2016-10-12 12:59:27.000
10	H_EMPLOYEE_MASTER	S_EMPLOYEE_MASTERALL	2016-10-12 12:59:27.000	NULL	2016-10-12 12:49:59.000	NULL	2016-10-12 12:59:27.000
11	H_EMPLOYEE_MASTER	S_EMPLOYEE_MASTERALL	2016-10-12 12:59:27.000	NULL	2016-10-12 12:49:59.000	NULL	2016-10-12 12:59:27.000
12	H_EMPLOYEE_MASTER	S_EMPLOYEE_MASTERALL	2016-10-12 12:59:27.000	NULL	2016-10-12 12:49:59.000	NULL	2016-10-12 12:59:27.000
13	H_EMPLOYEE_MASTER	S_EMPLOYEEETRA	2016-10-12 12:59:27.000	NULL	2016-10-12 12:49:59.000	NULL	2016-10-12 12:59:27.000
14	H_EMPLOYEE_MASTER	S_EMPLOYEEETRA	2016-10-12 12:59:27.000	NULL	2016-10-12 12:49:59.000	NULL	2016-10-12 12:59:27.000
15	H_EMPLOYEE_MASTER	S_EMPLOYEEETRA	2016-10-12 12:59:27.000	NULL	2016-10-12 12:49:59.000	NULL	2016-10-12 12:59:27.000
16	H_EMPLOYEE_MASTER	S_EMPLOYEEETRA	2016-10-12 12:59:27.000	NULL	2016-10-12 12:49:59.000	NULL	2016-10-12 12:59:27.000
17	H_EMPLOYEE	S_EMPLOYEEETRA	2016-10-12 12:59:27.000	NULL	2016-10-12 12:49:59.000	NULL	2016-10-12 12:59:27.000
18	H_EMPLOYEE	S_EMPLOYEE	2016-10-12 12:59:27.000	NULL	2016-10-12 12:49:59.000	NULL	2016-10-12 12:59:27.000
19	H_EMPLOYEE	S_EMPLOYEEETRA	2016-10-12 12:59:27.000	NULL	2016-10-12 12:49:59.000	NULL	2016-10-12 12:59:27.000
20	H_EMPLOYEE	S_EMPLOYEE	2016-10-12 12:59:27.000	NULL	2016-10-12 12:49:59.000	NULL	2016-10-12 12:59:27.000
21	H_MARITAL_STATUS_MASTER	S_MARITAL_STATUS_MASTER	2016-10-12 12:59:27.000	NULL	2016-10-12 12:49:59.000	NULL	2016-10-12 12:59:27.000
22	H_MARITAL_STATUS	S_MARITAL_STATUS	2016-10-12 12:59:27.000	NULL	2016-10-12 12:49:59.000	NULL	2016-10-12 12:59:27.000
23	H_PROJECT_MASTER	S_PROJECT_MASTER	2016-10-12 12:59:27.000	NULL	2016-10-12 12:49:59.000	NULL	2016-10-12 12:59:27.000
24	H_PROJECT_MASTER	S_PROJECT_MASTER	2016-10-12 12:59:27.000	NULL	2016-10-12 12:49:59.000	NULL	2016-10-12 12:59:27.000
25	H_PROJECT_MASTER	S_PRO_MAS_STARTERR	2016-10-12 12:59:27.000	NULL	2016-10-12 12:49:59.000	NULL	2016-10-12 12:59:27.000
26	H_PROJECT_MASTER	S_PRO_MAS_STARTERR	2016-10-12 12:59:27.000	NULL	2016-10-12 12:49:59.000	NULL	2016-10-12 12:59:27.000
27	H_PROJECT	S_PROJECT	2016-10-12 12:59:27.000	NULL	2016-10-12 12:49:59.000	NULL	2016-10-12 12:59:27.000
28	H_SUPERVISOR_MASTER	S_SUPERVISOR_MASTER	2016-10-12 12:59:27.000	NULL	2016-10-12 12:49:59.000	NULL	2016-10-12 12:59:27.000
29	H_SUPERVISOR	S_SUPERVISOR	2016-10-12 12:59:27.000	NULL	2016-10-12 12:49:59.000	NULL	2016-10-12 12:59:27.000
30	H_TRAINER_MASTER	S_TRAINER_MASTER	2016-10-13 12:17:22.000	NULL	2016-10-13 12:17:22.000	NULL	2016-10-13 12:17:22.000
31	H_TRAINER	S_TRAINER	2016-10-13 12:17:22.000	2016-10-13 14:32:14.000	2016-10-13 12:17:22.000	NULL	2016-10-13 14:32:14.000
32	H_EMPLOYEE_MASTER	S_DRIVERS_LICENCE_MASTER	2016-10-12 12:59:27.000	NULL	2016-10-13 16:13:12.000	NULL	2016-10-13 16:13:12.000
33	H_EMPLOYEE_MASTER	S_DRIVERS_LICENCE_MASTER	2016-10-12 12:59:27.000	NULL	2016-10-13 16:13:12.000	NULL	2016-10-13 16:13:12.000
34	H_EMPLOYEE	S_DRIVERS_LICENCE	2016-10-12 12:59:27.000	NULL	2016-10-13 16:13:12.000	NULL	2016-10-13 16:13:12.000
35	H_EMPLOYEE	S_DRIVERS_LICENCE	2016-10-12 12:59:27.000	NULL	2016-10-13 16:13:12.000	NULL	2016-10-13 16:13:12.000
36	H_EMPLOYEE	S_DRIVERS_LICENCE	2016-10-12 12:59:27.000	NULL	2016-10-13 16:13:12.000	NULL	2016-10-13 16:13:12.000

Slika 79. Rezultati Upita 4

Upit 5 traži verziju sheme (odnosno vremenski period) u kojem su bili aktivni i određeni hub iz SdvDB i njegova izvorišna tablica iz koje je punjen. Ovo je također temporalni upit koji uspoređuje periode važenja izvorišne tablice i njezinog huba u SdvDB, kako bi se pronašli periodi koji ne odgovaraju. Vidimo iz skupa rezultata da hub H\_CERTIFICATE\_TRAINING zapravo vrijedi od svog početnog LDTs do daljnega (LEDTs mu je NULL), dok je njegova izvorišna tablica u međuvremenu promijenila naziv.

**Tablica 30. Upit 5 - U kojem periodu (u kojoj shemi) su bili aktivni i hub H\_CERTIFICATE\_TRAINING i njegova izvorišna tablica iz koje je hub punjen?**

--U kojem periodu (u kojoj shemi) su bili aktivni i hub H\_CERTIFICATE\_TRAINING i njegova izvorišna tablica iz koje je hub punjen?

```
SELECT x.HubName, d.SrcRelationName, x.HubDefLDTS, x.HubDefLEDTs, d.SrcRelationLDTS,
d.SrcRelationLEDTs,
dbo.maxDate(x.HubDefLDTS, d.SrcRelationLDTS) as FromDate,
dbo.minDate((ISNULL(x.HubDefLEDTs, '9999-01-01')), (ISNULL(d.SrcRelationLEDTs,
'9999-01-01'))) as ToDate
FROM S_HUB_DEF x INNER JOIN H_HUB a ON a.SQN_Hub=x.H_HUB_SQN_Hub INNER JOIN
L_RELATION_HSL b
ON b.H_HUB_SQN_Hub=a.SQN_Hub INNER JOIN H_SRC_RELATION c ON
c.SQN_SrcRelation=b.H_REL_SQNRelation
INNER JOIN S_SRC_RELATION d ON d.H_SRC_REL_SQN_SrcRel=c.SQN_SrcRelation
AND dbo.maxDate(x.HubDefLDTS, d.SrcRelationLDTS)<dbo.minDate((ISNULL(x.HubDefLEDTs,
'9999-01-01')), (ISNULL(d.SrcRelationLEDTs, '9999-01-01')))
AND x.HubName='H_CERTIFICATE_TRAINING'
```

The screenshot shows a SQL query window with the following text:

```
--U kojem periodu (u kojoj shemi) su bili aktivni i hub H_CERTIFICATE_TRAINING i njegova izvorišna tablica iz koje je hub punjen?
SELECT x.HubName, d.SrcRelationName, x.HubDefLDTS, x.HubDefLEDTs, d.SrcRelationLDTS, d.SrcRelationLEDTs,
dbo.maxDate(x.HubDefLDTS, d.SrcRelationLDTS) as FromDate,
dbo.minDate((ISNULL(x.HubDefLEDTs, '9999-01-01')), (ISNULL(d.SrcRelationLEDTs, '9999-01-01'))) as ToDate
FROM S_HUB_DEF x INNER JOIN H_HUB a ON a.SQN_Hub=x.H_HUB_SQN_Hub INNER JOIN L_RELATION_HSL b
ON b.H_HUB_SQN_Hub=a.SQN_Hub INNER JOIN H_SRC_RELATION c ON c.SQN_SrcRelation=b.H_REL_SQNRelation
INNER JOIN S_SRC_RELATION d ON d.H_SRC_REL_SQN_SrcRel=c.SQN_SrcRelation
AND dbo.maxDate(x.HubDefLDTS, d.SrcRelationLDTS)<dbo.minDate((ISNULL(x.HubDefLEDTs, '9999-01-01')), (ISNULL(d.SrcRelationLEDTs, '9999-01-01')))
AND x.HubName='H_CERTIFICATE_TRAINING'
```

Below the query window, the Results window displays the following data:

HubName	SrcRelationName	HubDefLDTS	HubDefLEDTs	SrcRelationLDTS	SrcRelationLEDTs	FromDate	ToDate
H_CERTIFICATE_TRAINING	CERTIFICATE_TRAINING	2016-10-12 12:59:27.000	NULL	2016-10-12 20:42:22.000	2016-10-13 11:02:43.000	2016-10-12 20:42:22.000	2016-10-13 11:02:43.000
H_CERTIFICATE_TRAINING	TRAINING	2016-10-12 12:59:27.000	NULL	2016-10-13 11:02:43.000	NULL	2016-10-13 11:02:43.000	9999-01-01 00:00:00.000

**Slika 80. Rezultati Upita 5**

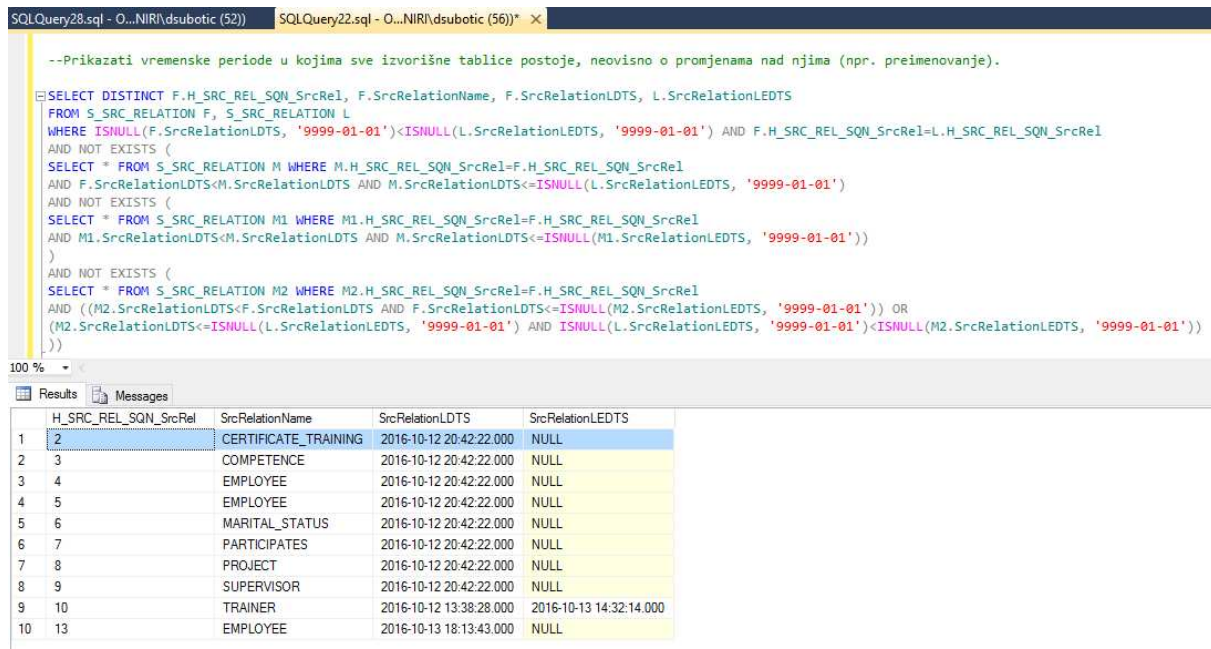
Upit 6 je također temporalni upit – uspoređuje vremenske periode važenja izvorišnih tablica, ne uzimajući u obzir promjene nad njima (dakle, uzima period od početnog punjenja, kada je tablica prvi put došla u izvor, pa do konačnog datuma važenja). Iz skupa rezultata vidimo da je izvorišna tablica CERTIFICATE\_TRAINING (koja je u nekom trenutku promijenila ime u TRAINING, ali u upitu se ta promjena zanemarila) i dalje aktivna (njezin novi zapis TRAINING ima LEDTS NULL pa je u skupu rezultata i za CERTIFICATE\_TRAINING stavljen LEDTS NULL).

I na kraju, upit 7 vraća evoluciju podataka u S\_SRC\_RELATION, koja pohranjuje podatke o izvorišnim tablicama. Iz skupa rezultata možemo vidjeti broj provedenih promjena (dodavanja i preimenovanja tablica) nad izvorišnim tablicama.

**Tablica 31. Upit 6 - Prikazati vremenske periode u kojima svi sateliti iz TrainingDB postoje u bazi, neovisno o promjenama nad njima.**

```
--Prikazati vremenske periode u kojima sve izvorišne tablice postoje, neovisno o
promjenama nad njima (npr. preimenovanje).

SELECT DISTINCT F.H_SRC_REL_SQN_SrcRel, F.SrcRelationName, F.SrcRelationLDTs,
L.SrcRelationLEDTs
FROM S_SRC_RELATION F, S_SRC_RELATION L
WHERE ISNULL(F.SrcRelationLDTs, '9999-01-01')<ISNULL(L.SrcRelationLEDTs, '9999-01-
01') AND F.H_SRC_REL_SQN_SrcRel=L.H_SRC_REL_SQN_SrcRel
AND NOT EXISTS (
SELECT * FROM S_SRC_RELATION M WHERE M.H_SRC_REL_SQN_SrcRel=F.H_SRC_REL_SQN_SrcRel
AND F.SrcRelationLDTs<M.SrcRelationLDTs AND
M.SrcRelationLDTs<=ISNULL(L.SrcRelationLEDTs, '9999-01-01')
AND NOT EXISTS (
SELECT * FROM S_SRC_RELATION M1 WHERE M1.H_SRC_REL_SQN_SrcRel=F.H_SRC_REL_SQN_SrcRel
AND M1.SrcRelationLDTs<M.SrcRelationLDTs AND
M.SrcRelationLDTs<=ISNULL(M1.SrcRelationLEDTs, '9999-01-01'))
)
AND NOT EXISTS (
SELECT * FROM S_SRC_RELATION M2 WHERE M2.H_SRC_REL_SQN_SrcRel=F.H_SRC_REL_SQN_SrcRel
AND ((M2.SrcRelationLDTs<F.SrcRelationLDTs AND
F.SrcRelationLDTs<=ISNULL(M2.SrcRelationLEDTs, '9999-01-01')) OR
(M2.SrcRelationLDTs<=ISNULL(L.SrcRelationLEDTs, '9999-01-01') AND
ISNULL(L.SrcRelationLEDTs, '9999-01-01')<ISNULL(M2.SrcRelationLEDTs, '9999-01-01'))
))
```



**Slika 81. Rezultati Upita 6**

**Tablica 32. Upit 7 - Prikazati evoluciju tablice S\_SRC\_RELATION, odnosno broj tablica nad kojima se izvršila bilo koja vrsta promjene.**

```
-- Prikazati broj tablica nad kojima se izvršila bilo koja vrsta promjene.

-- dani u kojima je došlo do promjene tablica
CREATE VIEW TabChanges(Day) AS
SELECT DISTINCT ISNULL(SrcRelationLDTS, '9999-01-01') FROM S_SRC_RELATION
UNION
SELECT DISTINCT ISNULL(SrcRelationLEDTS, '9999-01-01') FROM S_SRC_RELATION;

-- periodi u kojima je došlo do promjene tablica (na temelju prethodno izvučenih dana)
CREATE VIEW TabPeriods(SrcRelationLDTS, SrcRelationLEDTS) AS
SELECT P1.Day, P2.Day
FROM TabChanges P1, TabChanges P2
WHERE P1.Day<P2.Day
AND NOT EXISTS (SELECT * FROM TabChanges P3 WHERE P1.Day<P3.Day AND P3.Day<P2.Day);

-- broj promjena tablica, unutar prethodno izvučenih perioda
CREATE VIEW TabChangeCount(NbChanges, SrcRelationLDTS, SrcRelationLEDTS, SrcRelationName) AS
SELECT COUNT(*) as NumChanges, P.SrcRelationLDTS, P.SrcRelationLEDTS, S.SrcRelationName
FROM S_SRC_RELATION S, TabPeriods P
WHERE S.SrcRelationLDTS<=P.SrcRelationLDTS
AND P.SrcRelationLEDTS<=S.SrcRelationLEDTS
GROUP BY P.SrcRelationLDTS, P.SrcRelationLEDTS, S.SrcRelationName
UNION ALL
SELECT 0, P.SrcRelationLDTS, P.SrcRelationLEDTS, NULL FROM TabPeriods P
WHERE NOT EXISTS (SELECT * FROM S_SRC_RELATION S WHERE S.SrcRelationLDTS<=P.SrcRelationLDTS AND P.SrcRelationLEDTS<=S.SrcRelationLEDTS);

-- sjediniti pogled TabChangeCount

SELECT DISTINCT F.SrcRelationName, F.SrcRelationLDTS, L.SrcRelationLEDTS, F.NbChanges as NumOfChanges
FROM TabChangeCount F, TabChangeCount L
WHERE ISNULL(F.SrcRelationLDTS, '9999-01-01')<ISNULL(L.SrcRelationLEDTS, '9999-01-01') AND F.SrcRelationName=L.SrcRelationName
AND NOT EXISTS (
SELECT * FROM TabChangeCount M WHERE M.SrcRelationName=F.SrcRelationName
AND F.SrcRelationLDTS<M.SrcRelationLDTS AND
M.SrcRelationLDTS<=ISNULL(L.SrcRelationLEDTS, '9999-01-01')
AND NOT EXISTS (
SELECT * FROM TabChangeCount M1 WHERE M1.SrcRelationName=F.SrcRelationName
AND M1.SrcRelationLDTS<M.SrcRelationLDTS AND
M.SrcRelationLDTS<=ISNULL(M1.SrcRelationLEDTS, '9999-01-01'))
)
AND NOT EXISTS (
SELECT * FROM TabChangeCount M2 WHERE M2.SrcRelationName=F.SrcRelationName
AND ((M2.SrcRelationLDTS<F.SrcRelationLDTS AND
F.SrcRelationLDTS<=ISNULL(M2.SrcRelationLEDTS, '9999-01-01')) OR
(M2.SrcRelationLDTS<=ISNULL(L.SrcRelationLEDTS, '9999-01-01') AND
ISNULL(L.SrcRelationLEDTS, '9999-01-01')<ISNULL(M2.SrcRelationLEDTS, '9999-01-01'))
))
))
```

```

SELECT DISTINCT F.SrcRelationName, F.SrcRelationLDTs, L.SrcRelationLEDTs, F.NbChanges as NumOfChanges
FROM TabChangeCount F, TabChangeCount L
WHERE ISNULL(F.SrcRelationLDTs, '9999-01-01')<ISNULL(L.SrcRelationLEDTs, '9999-01-01') AND F.SrcRelationName=L.SrcRelationName
AND NOT EXISTS (
SELECT * FROM TabChangeCount M WHERE M.SrcRelationName=F.SrcRelationName
AND F.SrcRelationLDTs<M.SrcRelationLDTs AND M.SrcRelationLDTs<=ISNULL(L.SrcRelationLEDTs, '9999-01-01')
AND NOT EXISTS (
SELECT * FROM TabChangeCount M1 WHERE M1.SrcRelationName=F.SrcRelationName
AND M1.SrcRelationLDTs<M.SrcRelationLDTs AND M.SrcRelationLDTs<=ISNULL(M1.SrcRelationLEDTs, '9999-01-01'))
)
AND NOT EXISTS (
SELECT * FROM TabChangeCount M2 WHERE M2.SrcRelationName=F.SrcRelationName
AND ((M2.SrcRelationLDTs<F.SrcRelationLDTs AND F.SrcRelationLDTs<=ISNULL(M2.SrcRelationLEDTs, '9999-01-01')) OR
(M2.SrcRelationLDTs<=ISNULL(L.SrcRelationLEDTs, '9999-01-01') AND ISNULL(L.SrcRelationLEDTs, '9999-01-01')<ISNULL(M2.SrcRelationLEDTs, '9999-01-01'))
)
)

```

100 %

Results Messages

	SrcRelationName	SrcRelationLDTs	SrcRelationLEDTs	NumOfChanges
1	CERTIFICATE_TRAINING	2016-10-12 20:42:22.000	2016-10-13 11:02:43.000	1
2	TRAINER	2016-10-12 13:38:28.000	2016-10-13 14:32:14.000	1

**Slika 82. Rezultati Upita 7**

## 6. ZAKLJUČAK

U ovome doktorskom radu predstavljeno je istraživanje čiji cilj je bio razviti model repozitorija meta-podataka (MDV) koji se zasniva na Data Vault (DV) metodi za modeliranje baza podataka, a služi za integraciju skladišta podataka (SP) i sustava za upravljanje matičnim podacima (UMP) te za praćenje i upravljanje promjenama u SP/UMP podacima i meta-podacima, kao i u njihovim shemama. Motivacija za istraživanje proizlazi iz činjenice da u području skladištenja podataka još postoji prostora i mogućnosti za razvoj novih i korisnih rješenja za rješavanje određenih problema evolucije podataka i sheme. Dodatno, iz uviđenih sličnosti SP i UMP sustava (primarno u vidu njihove namjene, načina organizacije podataka te tekućih problema u izgradnji i održavanju) integracija ova dva sustava prirodno se nametnula kao novi smjer za rješavanje problema evolucije podataka i sheme. U radu je detaljno opisan problem evolucije sheme skladišta podataka, kao i neki od dosadašnjih pristupa rješavanju istog. Na temelju analize dosadašnjih istraživanja razvijena je i predstavljena vlastita arhitektura predloženog rješenja, koja se sastoji od više slojeva za čuvanje podataka, uključujući izvore podataka, repozitorij „sirovih“ podataka SdvDB, repozitorij poslovnih i matičnih podataka PmdvDB, repozitorij upravljanja matičnim podacima MdmDB i područno skladište podataka DmDB te na kraju repozitorij meta-podataka MdvDB. Na temelju ovako definirane arhitekture, izgrađen je prototip rješenja. U svrhu izrade prototipa rješenja (kao dodatne teorijske i empirijske validacije opisanog istraživanja) razvijen je niz praktičnih primjera nad kojim su se provele promjene iz definiranog skupa osnovnih promjena, korištenjem postavljenih operacija promjene. U prototipu MdvDB repozitorij ima centralnu ulogu – on predstavlja proširenje sistemskog kataloga korištenog sustava za upravljanje bazama podataka te prikuplja i pohranjuje sve trenutne verzije shema, kao i one prošle – MdvDB pamti povijest razvoja shema, od izvora podataka do centralnog SP. Kroz teorijsku i empirijsku validaciju (definiranog skupa operacija i promjena te samog prototipa rješenja) dokazane su sve tri postavljene hipoteze istraživanja. Dodatno, ostvareni su sljedeći znanstveni doprinosi: a) definiran je trajni model repozitorija meta-podataka za praćenje promjena shema skladišta podataka i sustava za upravljanje matičnim podacima te je njegov podskup razvijen kroz prototip rješenja, b) definiran je model za integraciju skladišta podataka i sustava za upravljanje matičnim podacima te je njegov podskup razvijen kroz prototip rješenja, c) sistematiziran je formalni konačni skup osnovnih promjena nad shemom skladišta



podataka i sustava za upravljanje matičnim podacima, d) definirana je formalna algebra za održavanje sheme skladišta podataka i sustava za upravljanje matičnim podacima koja se sastoji od dvije operacije dovoljne za provedbu evolucije sheme (koristeći sistematizirani skup osnovnih promjena), e) definirana je i opisana arhitektura integriranog skladišta podataka i sustava za upravljanje matičnim podacima i f) razvijena je i empirijski verificirana prva verzija prototipa integriranog rješenja.

U radu je dokazano da se u predloženom rješenju sve promjene (nad svim razinama arhitekture) implementiraju isključivo kroz proširenje postojeće sheme, korištenjem samo dvije definirane operacije te smo kroz upotrebu pripremljenog skupa upita dokazali da u predloženom rješenju nema gubitka informacija (sve se čuva u svako vrijeme) – upiti su vraćali stvarno stanje iz baze. Također, dokazano je da ovo ne vrijedi za standardni relacijski sistemski katalog – promjene se u njemu implementiraju standardnom CRUD (Create-Read-Update-Delete) metodologijom, čime dolazi do gubitka informacija. Dodatno, dokazali smo da je složenost provedbe evolucije sheme skladišta podataka u predloženom pristupu smanjena (ukupni broj potrebnih operacija promjene je manji u odnosu na dosadašnje pristupe verzioniranja sheme koji koriste kopije cijele sheme za izradu nove sheme), jer se u našem pristupu sve verzije sheme zasnivaju na jednoj, trajnoj i razvijajućoj verziji sheme. U radu je dokazano i da je moguće integrirati sustav skladišta podataka sa sustavom za upravljanje matičnim podacima kroz upotrebu MdvDB repozitorija meta-podataka, koji prikuplja i pohranjuje meta-podatke o povezivanju shema SdvDB i PmdvDB. Ovo smo provjerili i dokazali provođenjem skupa upita nad MdvDB repozitorijem, kao i nad MdmDB i DmDB repozitorijima. Korisnost predloženog rješenja dodatno je prikazana kroz provođenje skupa upita nad MdvDB repozitorijem, kako bi se prikazala povijest evolucije navedenih shema.

Obzirom da je područje evolucije skladišta podataka veoma široko, u ovome radu predstavljeni pristup očekivano se planira dalje razvijati. Trenutni nedostaci opisanog pristupa uključuju: a) relativno mali skup definiranih promjena (u budućnosti se planira proširiti skup osnovnih promjena na skup složenih promjena te definirati algoritme za izradu uzoraka provedbe složenih promjena, korištenjem određenih kombinacija osnovnih promjena), b) relativno mali skup praktičnih primjera (u budućnosti se planira razviti opširniji skup praktičnih primjera, koji će bolje oslikavati razne poslovne scenarije), c) ograničenje skupa izvora podataka na strukturirane relacijske izvore (u budućnosti se planiraju uključiti i nestrukturirani izvori, poput

web izvora i NoSQL izvora podataka u MdvDB shemu, kao i u novu verziju prototipa te se planira definirati skup promjena i operacija promjena nad nestrukturiranim izvorima podataka). d) djelomično implementirani prototip rješenja (u MdvDB repozitoriju implementirani su samo dijelovi za praćenje promjena u shemama izvora podataka i SdvDB/PmdvDB repozitorijima, međutim sam MDV model je opsežniji – uključuje i dio za praćenje promjena u shemama područnih SP, kao i aspekt sigurnosti i kontrole pristupa podacima) te e) prototip koji funkcionira na polu-automatskoj osnovi (neki dijelovi MdvDB repozitorija meta-podataka još uvijek se pune ručno, odnosno zapisi za stupce ChangeType i SchemaVersion ručno su kodirani u ETL pakete, što se planira automatizirati u sljedećoj verziji prototipa – uvođenje automatske detekcije tipa promjene u izvoru). Vezano uz prototip rješenja, u budućnosti se planira i proširiti MdvDB repozitorij na aspekt područnih SP i sigurnosti te na aspekt praćenja meta-podataka ETL procesa i paketa. MdmDB repozitorij matičnih podataka trenutno je izgrađen pomoću Microsoft SQL Server Management Studija na način da podupire sporo mijenjajuće dimenzije u tipu 2, međutim u budućnosti se MdmDB planira migrirati na Microsoft Master Data Services (MDS) u kojem će matični podaci biti pohranjeni prema tipu 1 sporo mijenjajućih dimenzija (što postojeća verzija prototipa omogućava, jer SdvDB i PmdvDB ionako čuvaju trenutno stanje i potpunu povijest podataka), a MdmDB će matične podatke propagirati u DmDB i natrag u izvore podataka putem MDS pretplatničkih pogleda. Također, može se razmisliti i o migraciji cjelovitog prototipa na novije (Microsoft SQL Server 2016) okruženje, iz razloga što ono podupire jednostavnije postavljanje i provođenje temporalnih upita nad podacima.

I na kraju, umjesto zaključka kratko treba reći da uistinu vjerujemo da će dosadašnji napori, kao i trenutna verzija rješenja služiti za daljnji razvoj što boljeg, korisnijeg i jednostavnijeg teorijskog i praktičnog rješenja za rješavanje problema evolucije skladišta podataka (kao i upravljanja matičnim podacima).

## Literatura

- [1] Akaichi, J., Oueslati, W. *MAVIE: A Mobile Agents View synchronization system*. In first international conference on the applications of digital information and web technology. Ostravem, pages 145-150, 2008.
- [2] Almazyad, A.S., Siddiqui, M.K. *Incremental View Maintenance: An Algorithmic Approach*. International Journal of Electrical and Computer Sciences (IJECS-IJENS), vol.10, no. 3, 2010.
- [3] Andany, J., Leonard, M., Palisser, C. *Management Of Schema Evolution In Databases*. In Proceedings of the 17th International Conference on Very Large Databases, Barcelona, 1991.
- [4] Banerjee, S., Davis, K.C. *Modeling Data Warehouse Schema Evolution over Extended Hierarchy Semantics*, S.Spaccapietra et.al (EDs): Journal on Data Semantics XIII, LNCS 5530, pp.72-96, Springer- Berlin, Heidelberg, 2009.
- [5] Baralis, E., Paraboschi, S., Teniente, E. *Materialized view selection in a multidimensional database*. Proceedings Conference on Very Large Databases, Athens, Greece, 1997.
- [6] Bebel, B., Eder, J., Koncilia, C., Morzy, T., Wrembel, R. *Creation and Management of Versions in Multiversion Data Warehouse*. In 19th ACM Symposium on Applied Computing (SAC 04), Nicosia, Cyprus, pages 717–723. ACM Press, 2004.
- [7] Bebel, B., Krolinkowski, Z., Wrembel, R. *Formal approach to modeling a multiversion data warehouse*. Bulletin of the Polish academy of sciences, Technical Sciences, vol. 54, no. 1, 2006.
- [8] Bellahsene, Z. *Schema Evolution in Data Warehouses*. Knowledge and Information Systems, 4(3):283–304, 2002.
- [9] Benitez-Guerrero, E., Collet, C., Adiba, M. *The WHES Approach to Data Warehouse Evolution*. e-Gnosis(online), vol.2, 2004.

- [10] Berson, A., Dubbov, L. *Master data management and data governance, 2nd edition*. McGraw Hill, 2011.
- [11] Blaschka, M., Sapia, C., Hofling, G. *On Schema Evolution in Multi-dimensional Databases*. In 1st International Conference on Data Warehousing and Knowledge Discovery (DaWaK 99), Florence, Italy, vol. 1676 of LNCS, pages 153–164. Springer, 1999.
- [12] Body, M. , Miquel, M., Bedard, Y., Tchounikine, A. *A Multidimensional and Multiversion Structure for OLAP Applications*. In 5th ACM International Workshop on Data Warehousing and OLAP (DOLAP 02),McLean, Virginia, USA, pages 1–6. ACM Press, 2002.
- [13] Bouzeghoub, M., Kedad, Z. *A Logical Model for Data Warehouse Design and Evolution*. Proceedings of the 2nd International Conference on Data Warehousing and Knowledge Discovery (DaWaK), London, UK, pages 178-188, 2000.
- [14] Chatterjee, S., Suprateek, S., Fuller, M. A. *A Deontological Approach to Designing Ethical Collaboration*. Journal of the Association for Information Systems, vol.10, Special Issue, pp. 138-169, 2009.
- [15] Chen, J., Chen, S., Rundensteiner, E. *A transactional model for data warehouse maintenance*. In: Spaccapietra, S., March, S.T., Kambayashi, Y. (eds.) ER 2002. LNCS, vol. 2503, pp. 247–262. Springer, Heidelberg, 2002.
- [16] Cui, Y., Widom, J. *Practical Lineage Tracing in Data Warehouses*. In Proceedings of the 16th International Conference on Data Engineering (ICDE'00), San Diego, California, 2000.
- [17] Curino, C.A., Moon, H.J., Zaniolo, C. *Graceful Database Schema Evolution:the PRISM Workbench*. In Proceedings of the Very Large Database Endowment, Auckland, New Zeland, 2008.
- [18] Das, A.K., Tu, S.W., Musen, M.A. *A Historical Relational Data Model for Managing Temporal Data*. Report, Stanford University, Section on Medical Informatics, San Diego, 1992.
- [19] Date, C.J., Darwen, H., Lorentzos, N. *Temporal Data & the Relational Model*. Morgan Kaufmann Publishers, Burlington, USA, 2002.

- [20] Eder, J., Koncilla, C. *Evolution of Dimension Data in Temporal Data Warehouses*. Technical Report, 2000.
- [21] ErWin Data Modeler (15. lipanj 2016). Dostupno na: <http://erwin.com/products/data-modeler>
- [22] Fan, H., Poulouvasilis, A. *Schema Evolution in Data Warehousing Environments – a schema transformation-based approach*. In Proceedings of Conceptual Modeling - ER, 23rd International Conference on Conceptual Modeling, Shanghai, China, 2004.
- [23] Feki, J., Taktak, S. *Impacts of Data Warehouse Evolution on Data Marts*. In Proceedings of the 13th International Arab Conference on Information Technology (ACIT), University of Balamand, Lebanon, 2012.
- [24] FreeDataGenerator (15. lipanj 2016). Dostupno na: <http://www.freedatagenerator.com/>.
- [25] Golfarelli, M., Lechtenbörger, J., Rizzi, S., Vossen, G.: *Schema Versioning in Data Warehouses*. In: Wang, S., Tanaka, K., Zhou, S., Ling, T.-W., Guan, J., Yang, D., Grandi, F., Mangina, E.E., Song, I.-Y., Mayr, H.C. (eds.) ER Workshops 2004. LNCS, vol. 3289, pages 415–428. Springer, Heidelberg 2004.
- [26] Golfarelli, M., Rizzi, S. *X-Time: Schema Versioning and Cross-Version Querying in Data Warehouses*. In Proceedings of the 23rd International Conference on Data Engineering (ICDE 2007), Istanbul, Turkey, pages 1471-1472, 2007.
- [27] Golfarelli, M., Rizzi, S. *Data Warehouse design: Modern principles and methodologies*. McGraw-Hill, Inc, 2009.
- [28] Gregor, S. *The Nature of Theory in Information Systems*. MIS Quaterly, vol.30, no. 3, pp. 611-642, Management Information Systems Research Center, Carlson School of Management, University of Minnesota, 2006.
- [29] Gregor, S., Hevner, A.R. *Positioning and Presenting Design Science Research for Maximum Impact*. MIS Quaterly, vol.37, no. 2, pp. 337-355, Management Information Systems Research Center, Carlson School of Management, University of Minnesota, 2013.
- [30] Gupta, A., Mumick, I. *Maintenance of Materialized Views: Problems, Techniques, and Applications*. Data Engineering Bulletin, 1995.

- [31] Hurtado, C. A., Mendelzon, A. O., Vaisman, A. A. *Maintaining Data Cubes under Dimension Updates*. In Proceedings of the 15th International Conference on Data Engineering (ICDE), Sydney, Australia, pages 346–355. IEEE Computer Society, 1999.
- [32] Hurtado, C. A., Mendelzon, A. O., Vaisman, A. A. *Updating OLAP Dimensions*. In Proceedings of the 2nd International Workshop on Data Warehousing and OLAP, Kansas City, Missouri, USA, 1999.
- [33] Idef1X (11. listopad 2016). Dostupno na: <http://www.idef.com/idef1x-data-modeling-method/>
- [34] Inmon, W.H., Strauss, D., Neushloss, G. *DW 2.0: The Architecture for the Next Generation of Data Warehousing*. Morgan Kaufmann Publishers, Burlington, USA, 2008.
- [35] Inmon, W. H., Linstedt, D. *Data Architecture: A Primer for the Data Scientist: Big Data, Data Warehouse and Data Vault*. Morgan Kaufmann, USA, 2014.
- [36] Jain, H., Gosain, A. *A Comprehensive Study of View Maintenance Approaches in Data Warehousing Evolution*. ACM SIGSOFT Software Engineering Notes, vol.37, iss. 5, New York, USA, 2012.
- [37] Jiang, B. *Constructing data warehouse with metadata-driven generic operators, and more*. DBJ Publishing, 2011.
- [38] Johnston, T., Weis, R. *Managing Time in Relational Databases: How to Design, Update and Query Temporal Data*. Morgan Kaufmann Publishers, Burlington, USA, 2010.
- [39] Jovanović, V., Bojičić, I. *Conceptual Data Vault Model*. In Proceedings of the Southern Association for Information Systems Conference, Atlanta, USA, 2012.
- [40] Jovanović, V., Bojičić, I., Knowles, C., Pavlic, M. *Persistent Staging Area Models For Data Warehouses*. Issues in Information Systems, vol.13, iss. 1, pages 121-132, 2012.
- [41] Kaas, C.E., Pedersen, T.B., Rasmussen, B.D. *Schema Evolution for Stars and Snowflakes*. In Proceedings of the International Conference on Enterprise Information Systems (ICEIS 2004), Portugal, 2004.
- [42] Kimball, R. *The Data Warehouse Lifecycle Toolkit (2nd Edition)*. New York, Wiley, 2008.

- [43] Kimball, R., Ross, M. *The data warehouse toolkit, 3rd edition*. John Wiley, 2013.
- [44] Kuechler, W., Vaishnavi, V. *A Framework for Theory Development in Design Science Research: Multiple Perspectives*. Journal of the Association for Information Systems, vol.13, iss.6, pp. 395-423, 2012.
- [45] Ladley, J. *Data Governance: How to Design, Deploy and Sustain an Effective Data Governance Program*. Morgan Kaufmann Publishers, Burlington, USA, 2012.
- [46] Lakshmanan, L. V. S., Sadri, F., Subramanian, I. N. *Schema SQL: a Language for Interoperability in Relational Multi-Databases Systems*. In Proceedings of the 22nd International Conference on Very Large Databases, pages 239-250, 1996.
- [47] Linstedt, D. *SuperCharge Your Data Warehouse: Invaluable Data Modeling Rules to Implement Your Data Vault*. CreateSpace Independent Publishing Platform, USA, 2011.
- [48] Linstedt, D., Olschimke, M. *Building a Scalable Data Warehouse with Data Vault 2.0: Implementation Guide for Microsoft SQL Server 2014*. Morgan Kaufmann, 2015.
- [49] Loshin, D. *Master Data Management*. Morgan Kaufmann Publishers, Burlington, USA, 2008.
- [50] Malinowski, E., Zimányi, E.: A conceptual model for temporal data warehouses and its transformation to the ER and the object-relational models. *Data & Knowledge Engineering*, vol. 64, pp. 101-133. (2008)
- [51] Marco, D., Jennings, M. *Universal meta data models*. John Willey, 2004.
- [52] Marotta, A., Ruggia, R. *Data Warehouse Design: A schema-transformation approach*. 22nd International Conference of the Chilean Computer Science Society (SCCC), Copiapo, Chile, 2002.
- [53] Meenakshi, A., Gosain, A. *Schema Evolution for Data Warehouse: A Survey*. International Journal of Computer Applications, vol.22, no.6, 2011.
- [54] Microsoft SQL Server 2012 (15. listopad 2016). Dostupno na: <https://www.microsoft.com/en-us/download/details.aspx?id=43351>

- [55] Morzy, T., Wrembel, R. *On Querying Versions of Multiversion Data Warehouse*. In Proceedings of the International Workshop on Data Warehousing and OLAP, DOLAP'04, Washington, USA, 2004.
- [56] Oueslati, W., Akaichi, J. *A survey on Data warehouse evolution*. International Journal of Database Management Systems (IJDBMS), vol.2, no.4, 2010.
- [57] Papastefanatos, G., Vassiliadis, P., Simitsis, A., Vassiliou, Y. *What-if Analysis for Data Warehouse Evolution*. In Proceedings of the 9th International Conference on Data Warehousing and Knowledge Discovery, Regensburg, Germany, 2007.
- [58] Peffers, K., Tuunanen, T., Rothenberger, M.A, Chatterjee, S. *A Design Science Research Methodology for Information Systems Research*. Journal of Management Information Systems, vol.24, iss.3, no.3, pp. 45-77, M. E. Sharpe, Inc. Armonk, NY, USA, winter 2007-2008.
- [59] Quix. *Repository Support for Data Warehouse Evolution*. In Proceedings of the International Workshop DMDW, Heidelberg, Germany 2004.
- [60] Rajaraman, Y. Sagiv, J. D. Ullman. *Answering Queries Using Templates With Binding Patterns*. Proc. ACM Symp. Principles Database System, pages 105-112, 1995.
- [61] Roddick, J. F. *A Survey of Schema Versioning Issues for Database Systems*. Information and Software Technology, 37(7):383–393, 1996.
- [62] Rundensteiner, E. A. , Koeller, A., Zhang, X., Lee, A.J., Nica, A. *Evolvable View Environment EVE: A Data Warehouse System Handling Schema and Data Changes of Distributed Sources*. Proceedings of the International Database Engineering and Application Symposium (IDEAS'99), Montreal, Canada, 1999.
- [63] Rundensteiner, E. A., Koeller, A., Zhang, X. *Maintaining Data Warehouses Over Changing Information Sources*. In Communications of the ACM, vol.43, pages 57-62, New York, USA, 2000.
- [64] Snodgrass, R. T., *The TSQL2 Temporal Query Language*. Kluwer Academic Publishers, 1995.



- [65] Solodovnikova, D. *Data Warehouse Evolution Framework*. In Proceedings of the Spring Young Researcher's Colloquium On Database and Information Systems SYRCoDIS, Moscow, Russia, 2007.
- [66] SQL Data Modeler (15. listopad 2016). Dostupno na: <http://www.oracle.com/technetwork/developer-tools/datamodeler/overview/index.html>
- [67] Talwar, K., Gosain, A. *Implementing Schema Evolution in Data Warehouse through Complex Hierarchy Semantics*. International Journal of Scientific & Engineering Research, vol. 3, iss. 7, 2012.
- [68] Teannenbaum, A. *Metadata solution*. Addison Wesley, 2002.
- [69] Vaisman, A.A., Mendelzon, A.O., Ruaro, W., Cymerman, S.G. *Supporting Dimension Updates in an OLAP Server*. In Proceedings of the CAISE02 Conference, Canada, 2002.
- [70] Zhang, X., Rundensteiner, E.A. *Data Warehouse Maintenance Under Concurrent Schema and Data Updates*. In Proceedings of the International Conference on Data Engineering (ICDE'99), Sydney, 1999.
- [71] Zimányi, E. Temporal Aggregates and Temporal Universal Quantification in Standard SQL. ACM SIGMOD Record, Vol.35, Iss.2, 2006.
- [72] Zouari Turki, I., Ghozzi Jedidi, F., Bouaziz, R. *Constraints to manage consistency in multiversion data warehouse*. In Proceedings of the 35th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia, 2012.

# POPIS SLIKA

SLIKA 1. TIPIČNA TROSLIJSKA ARHITEKTURA SP I UMP SUSTAVA .....	23
SLIKA 2. ARHITEKTURA INTEGRIRANOG SP/UMP SUSTAVA .....	23
SLIKA 3. KONCEPTUALNI MDV MODEL.....	26
SLIKA 4. DIO MDV MODELA KOJI SE ODNOSI NA IZVORE PODATAKA .....	28
SLIKA 5. DIO MDV MODELA KOJI SE ODNOSI NA CENTRALNI REPOZITORIJ PODATAKA (SDV+PMDV) .....	30
SLIKA 6. DIO MDV MODELA KOJI SE ODNOSI NA PODRUČNA SKLADIŠTA PODATAKA .....	31
SLIKA 7. DIO MDV MODELA KOJI SE ODNOSI NA KONTROLU PRISTUPA PODACIMA .....	31
SLIKA 8. DIO MDV MODELA KOJI SE ODNOSI NA INTEGRACIJU SP/UMP.....	33
SLIKA 9. JOBDB SCHEMA.....	48
SLIKA 10. TRAININGDB SCHEMA .....	49
SLIKA 11. POČETNA INTEGRIRANA SDVDB I PMDVDB.....	49
SLIKA 12. DODAVANJE NOVE RELACIJE, $D_R$ I DODAVANJE VEZE IZMEĐU RELACIJA, $D_V$ U TRAININGDB.....	50
SLIKA 13. DODAVANJE NOVE RELACIJE, $D_R$ I DODAVANJE VEZE IZMEĐU RELACIJA, $D_V$ U SDVDB .....	51
SLIKA 14. PREIMENOVANJE RELACIJE, $I_R$ U TRAININGDB SCHEMI I EFEKT U MDV REPOZITORIJU.....	51
SLIKA 15. BRISANJE RELACIJE, $B_R$ I BRISANJE VEZE IZMEĐU RELACIJA, $B_V$ U MDV REPOZITORIJU.....	52
SLIKA 16. DODAVANJE ATRIBUTA U RELACIJU $D_A$ U TRAININGDB SCHEMU (A) I U SDVDB SCHEMU (B).....	53
SLIKA 17. DODAVANJE ATRIBUTA U RELACIJU $D_A$ U TRAININGDB NA PODATKOVNOJ RAZINI .....	53
SLIKA 18. DODAVANJE ATRIBUTA U RELACIJU $D_A$ U SDVDB NA PODATKOVNOJ RAZINI .....	54
SLIKA 19. PREIMENOVANJE ATRIBUTA $I_{AN}$ U MDV NA PODATKOVNOJ RAZINI .....	54
SLIKA 20. PROMJENA TIPA PODATAKA ATRIBUTA $I_{AT}$ U MDV NA PODATKOVNOJ RAZINI.....	56
SLIKA 21. BRISANJE ATRIBUTA IZ RELACIJE $B_A$ U MDV NA PODATKOVNOJ RAZINI .....	56
SLIKA 22. PRETVORBA RELACIJE U VEZU $I_{RV}$ U JOBDB SCHEMI .....	57
SLIKA 23. PRETVORBA RELACIJE U VEZU $I_{RV}$ U SDV SCHEMI .....	58
SLIKA 24. PRETVORBA VEZE U RELACIJU $I_{VR}$ U JOBDB SCHEMI .....	60
SLIKA 25. PRETVORBA VEZE U RELACIJU $I_{VR}$ U SDV SCHEMI .....	61
SLIKA 26. PRETVORBA ATRIBUTA U RELACIJU $I_{AR}$ U JOBDB SCHEMI.....	62
SLIKA 27. PRETVORBA ATRIBUTA U RELACIJU $I_{AR}$ U SDV SCHEMI.....	63
SLIKA 28. PRETVORBA RELACIJE U ATRIBUT $I_{RA}$ U JOBDB SCHEMI.....	64
SLIKA 29. PRETVORBA RELACIJE U ATRIBUT $I_{RA}$ U JOBDB SCHEMI.....	65
SLIKA 30. POČETNA CENTRALNA 3NF SCHEMA ZA INTEGRIRANI JOBDB I TRAININGDB.....	67
SLIKA 31. DIJAGRAM ARHITEKTURE PROTOTIPA.....	80
SLIKA 32. SSIS TOK KONTROLE ZA PUNJENJE SDVDB IZ IZVORA .....	82
SLIKA 33. SSIS TOK PODATAKA ZA PUNJENJE SDVDB HUBA (PRIMJER H_MARITAL_STATUS) .....	83
SLIKA 34. SSIS TOK PODATAKA ZA PUNJENJE SDVDB HUBA (PRIMJER H_EMPLOYEE).....	83
SLIKA 35. SSIS TOK PODATAKA ZA PUNJENJE SDVDB LINKA (PRIMJER L_PARTICIPATES).....	84
SLIKA 36. SSIS TOK PODATAKA ZA PUNJENJE SDVDB LINKA (PRIMJER L_SUPERVISES) .....	85
SLIKA 37. SSIS TOK PODATAKA ZA PUNJENJE I AŽURIRANJE SDVDB SATELITA .....	86
SLIKA 38. SSIS TOK PODATAKA ZA AŽURIRANJE OBRISANIH REDAKA SDVDB SATELITA.....	86
SLIKA 39. SSIS TOK PODATAKA ZA ELIMINACIJU DUPLIH REDAKA U PMDVDB .....	87
SLIKA 40. SSIS TOK PODATAKA ZA INTEGRACIJU VIŠE TABLICA U JEDNU U PMDVDB.....	88
SLIKA 41. SSIS TOK PODATAKA ZA AŽURIRANJE SATELITA U PMDVDB.....	89
SLIKA 42. SSIS TOK PODATAKA ZA PUNJENJE SATELITA S POSLOVNIM PRAVILOM U PMDVDB .....	90
SLIKA 43. SSIS TOK KONTROLE ZA PUNJENJE MDMDB MATIČNIH PODATAKA O EMPLOYEE I TRAINING.....	91
SLIKA 44. SSIS TOK PODATAKA ZA PUNJENJE EMPLOYEEMDM .....	91
SLIKA 45. SSIS TOK KONTROLE ZA PUNJENJE DMDB DIMENZIJA EMPLOYEE I TRAINING I TABLICE ČINJENICA PARTICIPATES.....	92
SLIKA 46. SSIS TOK PODATAKA ZA PUNJENJE DIMENZIJE EMPLOYEE.....	92
SLIKA 47. SSIS TOK PODATAKA ZA PUNJENJE MDVDB HUBOVA.....	93
SLIKA 48. SSIS TOK PODATAKA ZA PUNJENJE MDVDB LINKOVA.....	94
SLIKA 49. SSIS TOK PODATAKA ZA PUNJENJE MDVDB SATELITA .....	94
SLIKA 50. MDVDB STANJE U IZVORIMA PRIJE DODAVANJA RELACIJE .....	99

SLIKA 51. MDVDB STANJE U IZVORIMA NAKON DODAVANJA RELACIJE .....	99
SLIKA 52. MDVDB STANJE U SDV/PMDV HUB-SATELIT SHEMAMA NAKON DODAVANJA RELACIJE.....	100
SLIKA 53. MDVDB STANJE U SDV/PMDV SATELIT-ATRIBUT SHEMAMA NAKON DODAVANJA RELACIJE .....	101
SLIKA 54. MDVDB STANJE U PMDV SHEMI NAKON DODAVANJA RELACIJE .....	101
SLIKA 55. MDVDB STANJE U IZVORIŠNOJ SHEMI NAKON PREIMENOVANJA RELACIJE.....	102
SLIKA 56. STANJE U SISTEMSKOM KATALOGU NAKON PREIMENOVANJA RELACIJE.....	102
SLIKA 57. PODATKOVNO STANJE U SdvDB I PMdvDB NAKON BRISANJA RELACIJE .....	103
SLIKA 58. MDVDB STANJE U IZVORIŠNOJ SHEMI TABLICA NAKON BRISANJA RELACIJE .....	103
SLIKA 59. MDVDB STANJE U IZVORIŠNOJ SHEMI STUPACA NAKON BRISANJA RELACIJE .....	104
SLIKA 60. MDVDB STANJE U SDV/PMDV SHEMAMA NAKON BRISANJA RELACIJE – HUBOVI .....	104
SLIKA 61. MDVDB STANJE U SDV/PMDV SHEMAMA NAKON BRISANJA RELACIJE - ATRIBUTI.....	105
SLIKA 62. STANJE U SISTEMSKOM KATALOGU NAKON BRISANJA RELACIJE .....	105
SLIKA 63. PODATKOVNO STANJE U SdvDB I PMdvDB NAKON DODAVANJA ATRIBUTA.....	107
SLIKA 64. MDVDB STANJE U SDV/PMDV SHEMI NAKON DODAVANJA ATRIBUTA .....	107
SLIKA 65. STANJE U MDvDB I SISTEMSKOM KATALOGU NAKON PREIMENOVANJA ATRIBUTA .....	108
SLIKA 66. STANJE U MDvDB I SISTEMSKOM KATALOGU NAKON PROMJENE TIPA PODATKA ATRIBUTA.....	109
SLIKA 67. STANJE U MDVDB I SISTEMSKOM KATALOGU NAKON BRISANJA ATRIBUTA .....	109
SLIKA 68. INTEGRACIJA SdvDB I PMdvDB HUBOVA U MDVDB .....	112
SLIKA 69. INTEGRACIJA SdvDB I PMdvDB SATELITA U MDVDB .....	112
SLIKA 70. INTEGRACIJA IZVORA I SdvDB HUBOVA U MDVDB .....	113
SLIKA 71. STANJE U SdvDB I PMdvDB NAKON PODATKOVNE PROMJENE U JOBDB IZVORU .....	114
SLIKA 72. STANJE U MDMDB POSLIJE PODATKOVNE PROMJENE .....	114
SLIKA 73. STANJE U DMDB (I MDMDB) POSLIJE PODATKOVNE PROMJENE.....	115
SLIKA 74. BROJ REDAKA U MDMDB I DMDB POSLIJE PODATKOVNE PROMJENE.....	115
SLIKA 75. REZULTATI UPITA 1 .....	116
SLIKA 76. REZULTATI UPITA 2 A).....	117
SLIKA 77. REZULTATI UPITA 2 B).....	118
SLIKA 78. REZULTATI UPITA 3 .....	119
SLIKA 79. REZULTATI UPITA 4 .....	120
SLIKA 80. REZULTATI UPITA 5 .....	121
SLIKA 81. REZULTATI UPITA 6 .....	122
SLIKA 82. REZULTATI UPITA 7 .....	124

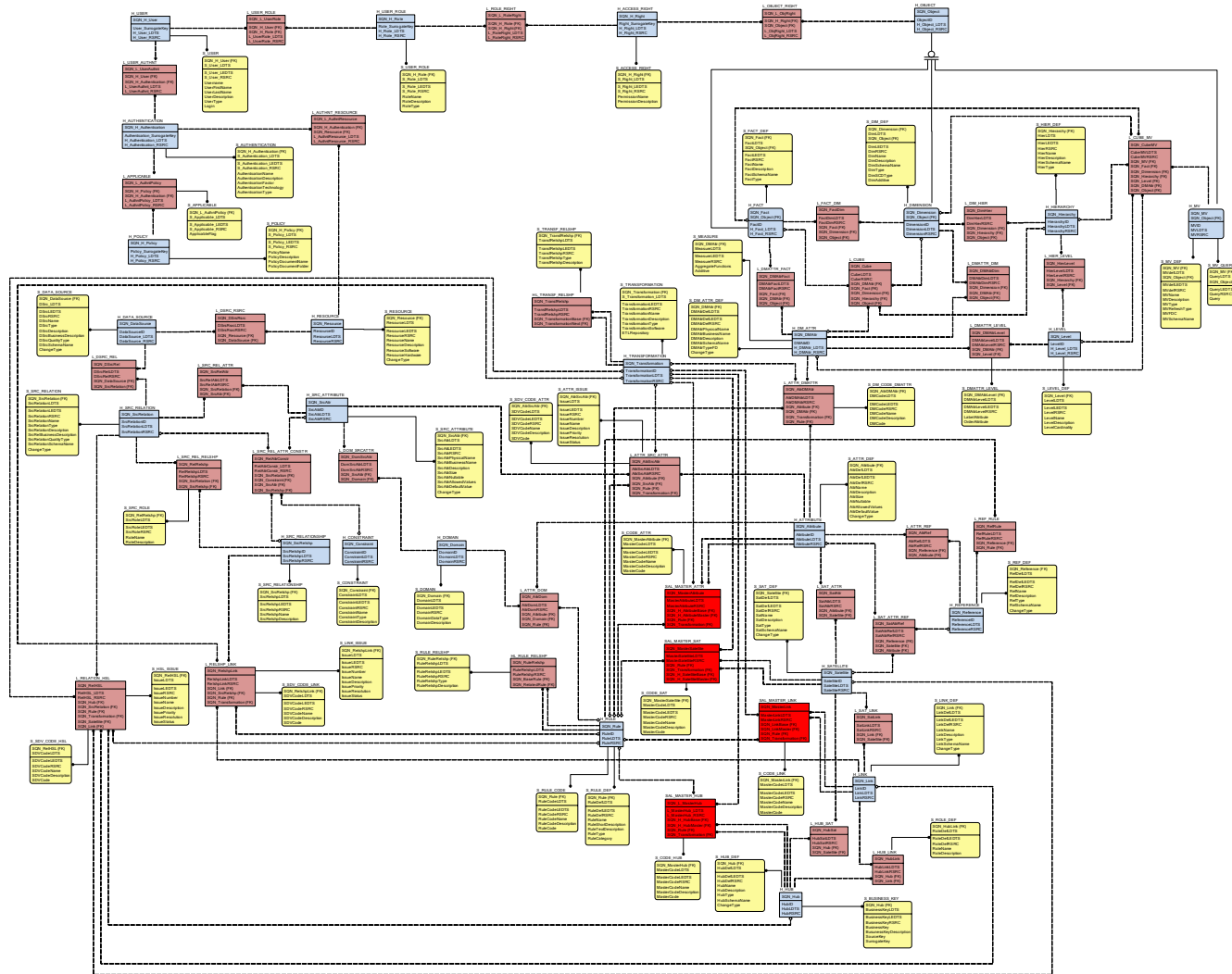
# POPIS TABLICA

TABLICA 1. PRIKAZ PREDLOŽENIH OPERACIJA EVOLUCIJE PO ISTRAŽIVANJIMA .....	16
TABLICA 2. PRIKAZ PREDLOŽENIH PRISTUPA ODRŽAVANJA POGLEDA PO ISTRAŽIVANJIMA.....	19
TABLICA 3. UČINAK PROMJENA U SHEMI IZVORA PODATAKA NA SDV/PMDV I MDV GRAF.....	41
TABLICA 4. UČINAK PROMJENE U SDV/PMDV GRAFU NA MDV GRAF.....	45
TABLICA 5. BROJ OPERACIJA ZA D <sub>R</sub> I D <sub>V</sub> NAD 3NF REPOZITORIJEM.....	67
TABLICA 6. BROJ OPERACIJA ZA D <sub>R</sub> I D <sub>V</sub> NAD SDV/PMDV REPOZITORIJEM .....	68
TABLICA 7. BROJ OPERACIJA ZA B <sub>R</sub> I B <sub>V</sub> NAD 3NF REPOZITORIJEM .....	69
TABLICA 8. BROJ OPERACIJA ZA B <sub>R</sub> I B <sub>V</sub> NAD SDV/PMDV REPOZITORIJEM .....	69
TABLICA 9. BROJ OPERACIJA ZA D <sub>A</sub> NAD 3NF REPOZITORIJEM.....	70
TABLICA 10. BROJ OPERACIJA ZA D <sub>A</sub> NAD SDV/PMDV REPOZITORIJEM.....	70
TABLICA 11. BROJ OPERACIJA ZA I <sub>AT</sub> NAD 3NF REPOZITORIJEM.....	71
TABLICA 12. BROJ OPERACIJA ZA I <sub>AT</sub> NAD SDV/PMDV REPOZITORIJEM .....	72
TABLICA 13. BROJ OPERACIJA ZA B <sub>A</sub> NAD 3NF REPOZITORIJEM .....	72
TABLICA 14. BROJ OPERACIJA ZA B <sub>A</sub> NAD SDV/PMDV REPOZITORIJEM .....	73
TABLICA 15. BROJ OPERACIJA ZA I <sub>RV</sub> NAD 3NF REPOZITORIJEM .....	74
TABLICA 16. BROJ OPERACIJA ZA I <sub>RV</sub> NAD SDV/PMDV REPOZITORIJEM.....	74
TABLICA 17. BROJ OPERACIJA ZA I <sub>VR</sub> NAD 3NF REPOZITORIJEM .....	75
TABLICA 18. BROJ OPERACIJA ZA I <sub>VR</sub> NAD SDV/PMDV REPOZITORIJEM.....	75
TABLICA 19. BROJ OPERACIJA ZA I <sub>AR</sub> NAD 3NF REPOZITORIJEM .....	76
TABLICA 20. BROJ OPERACIJA ZA I <sub>AR</sub> NAD SDV/PMDV REPOZITORIJEM.....	77
TABLICA 21. BROJ OPERACIJA ZA I <sub>RA</sub> NAD 3NF REPOZITORIJEM .....	78
TABLICA 22. BROJ OPERACIJA ZA I <sub>RA</sub> NAD SDV/PMDV REPOZITORIJEM.....	78
TABLICA 23. DODAVANJE NOVE RELACIJE, D <sub>R</sub> .....	95
TABLICA 24. SQL NAREDBE ZA DODAVANJE NOVOG SATELITA U SdVDB I PMDVB .....	106
TABLICA 25. REDOSLIJED PROMJENA U IZVORIMA ZA SLOŽENE PROMJENE .....	110
TABLICA 26. UPIT 1 – KOJE TABLICE SU I KADA U SVIM IZVORIMA PROMIJENILE NAZIV I KAKO SU SE PRIJE ZVALE? .....	116
TABLICA 27. UPIT 2 - KOJE TABLICE TRENUTNO NE POSTOJE U IZVORIMA, A POSTOJALE SU U: A) BILO KOJEM PERIODU ILI B) POČETNOJ SHEMI? .....	117
TABLICA 28. UPIT 3 - KOJI HUBOVI (I KADA) SU NAKNADNO DODAVANI U SdVDB U ODNOSU NA POČETNU SHEMU SdVDB? .....	118
TABLICA 29. UPIT 4 - KOJI HUBOVI I SATELITI SU VRIJEDILI (BILI AKTIVNI) U ISTOM VREMENSKOM PERIODU? .....	119
TABLICA 30. UPIT 5 - U KOJEM PERIODU (U KOJOJ SHEMI) SU BILI AKTIVNI I HUB H_CERTIFICATE_TRAINING I NJEGOVA IZVORIŠNA TABLICA IZ KOJE JE HUB PUNJEN? .....	121
TABLICA 31. UPIT 6 - PRIKAZATI VREMENSKE PERIODE U KOJIMA SVI SATELITI IZ TRAININGDB POSTOJE U BAZI, NEOVISNO O PROMJENAMA NAD NJIMA. ....	122
TABLICA 32. UPIT 7 - PRIKAZATI EVOLUCIJU TABLICE S_SRC_RELATION, ODNOSNO BROJ TABLICA NAD KOJIMA SE IZVRŠILA BILO KOJA VRSTA PROMJENE. ....	123

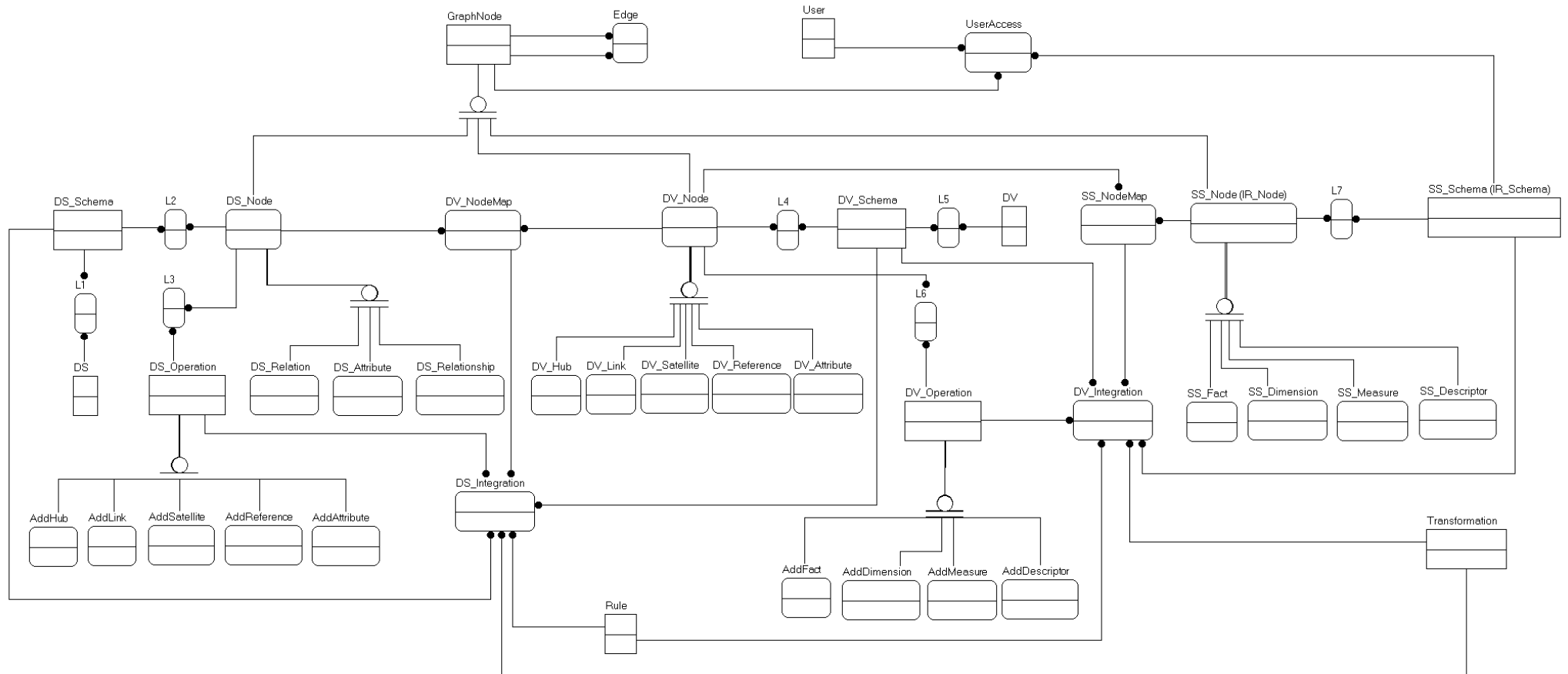
## POPIS PRIVITAKA

PRIVITAK 1. CJELOVITI MDV MODEL REPOZITORIJA METAPODATAKA.....	139
PRIVITAK 2. KONCEPTUALNI MDV MODEL REPOZITORIJA METAPODATAKA .....	140
PRIVITAK 3. POSLOVNA DEFINICIJA MDV MODELA .....	141
PRIVITAK 4. SQL DDL SKRIPTE ZA IZRADU PROTOTIPA.....	199

# Privitak 1. Cjeloviti MDV model repozitorija metapodataka



## Privitak 2. Konceptualni MDV model repozitorija metapodataka



### **Privitak 3. Poslovna definicija MDV modela**

Izvori podataka:

<b>NAZIV ENTITETA</b>	<b>DEFINICIJA ENTITETA</b>
H_RESOURCE	Poslovni ključevi resursa (hardverskih ili softverskih) – hub entitet.
S_RESOURCE	Opisni podaci za resurse – satelit entitet.
H_DATA_SOURCE	Poslovni ključevi izvora podataka - hub entitet.
S_DATA_SOURCE	Opisni podaci za izvore podataka – satelit entitet.
L_DSRC_RSRC	Veze između izvora podataka i resursa – link entitet.
H_SRC_RELATION	Poslovni ključevi izvorišnih entiteta (tablica) - hub entitet.
S_SRC_RELATION	Opisni podaci izvorišnih entiteta (tablica) – satelit entitet.
L_DSRC_REL	Veze između izvora podataka i izvorišnih tablica – link entitet.
H_SRC_ATTRIBUTE	Poslovni ključevi izvorišnih atributa (stupaca) - hub entitet.
S_SRC_ATTR	Opisni podaci izvorišnih atributa (stupaca) – satelit entitet.
L_SRC_REL_ATTR	Veze između izvorišnih entiteta (tablica) i izvorišnih atributa (stupaca) – link entitet.
H_SRC_RELATIONSHIP	Poslovni ključevi izvorišnih veza - hub entitet.
S_SRC_RELATIONSHIP	Opisni podaci izvorišnih veza – satelit entitet.
L_SRC_REL_RELSHP	Pridruživanje izvorišnih tablica i izvorišnih veza preko uloge veze – link entitet.
S_SRC_ROLE	Opis uloge veze iz L_SRC_REL_RELSHP
H_CONSTRAINT	Poslovni ključevi ograničenja - hub entitet.
S_CONSTRAINT	Opisni podaci za ograničenja – satelit entitet.
L_SRC_REL_ATTR_CONSTR	Pridruživanje ograničenja na izvorišnu tablicu i/ili izvorišni atribut – link entitet.

<b>NAZIV ENTITETA</b>	<b>NAZIV ATRIBUTA</b>	<b>DEFINICIJA ATRIBUTA</b>
H_RESOURCE		
	SQLN_Resource	Primarni umjetni ključ (sekvenca) hub-entiteta.



	ResourceID	Poslovni ključ hub-entiteta.
	ResourceLDTS	Datum punjenja zapisa o resursu u meta-bazu.
	ResourceRSRC	Izvor iz kojeg je zapis o resursu došao u meta-bazu.
S_RESOURCE		
	SQN_Resource	Dio primarnog ključa satelit-entiteta (ujedno i primarni ključ njegovog hub-entiteta H_RESOURCE).
	ResourceLDTS	Dio primarnog ključa satelit-entiteta – datum punjenja zapisa o resursu u meta-bazu.
	ResourceLEDTS	Datum završetka punjenja zapisa o resursu u meta-bazu.
	ResourceRSRC	Izvor iz kojeg je zapis o resursu došao u meta-bazu.
	ResourceName	Naziv resursa.
	ResourceDescription	Tekstualni opis resursa.
	ResourceSoftware	Naziv softvera resursa.
	ResourceHardware	Naziv hardvera resursa.
	ChangeType	Vrsta primjenjene promjene
H_DATA_SOURCE		
	SQN_DataSource	Primarni umjetni ključ (sekvenca) hub-entiteta.
	DataSourceID	Poslovni ključ hub-entiteta.
	DataSourceLDTS	Datum punjenja zapisa o izvorišnom sustavu u meta-bazu.
	DataSourceRSRC	Izvor iz kojeg je zapis o izvorišnom sustavu došao u meta-bazu.
S_DATA_SOURCE		
	SQN_DataSource	Dio primarnog ključa satelit-entiteta (ujedno i primarni ključ

		njegovog hub-entiteta H_DATA_SOURCE).
	DSrcLDTS	Dio primarnog ključa satelit-entiteta – datum punjenja zapisa o izvorišnom sustavu u meta-bazu.
	DSrcLEDTS	Datum završetka punjenja zapisa o izvorišnom sustavu u meta-bazu.
	DSrcRSRC	Izvor iz kojeg je zapis o izvorišnom sustavu došao u meta-bazu.
	DSrcName	Naziv izvorišnog sustava.
	DSrcType	Tip izvorišnog sustava (može biti relacijska baza ili flat file).
	DSrcDescription	Opis izvorišnog sustava.
	DSrcBusinessDescription	Poslovni opis izvorišnog sustava.
	ChangeType	Vrsta primjenjene promjene
	SchemaName	Verzija sheme zapisa (npr. 1, 2, 3...)
	DsrcQualityType	Tip kvalitete izvorišnog sustava (može biti niska, prosječna ili visoka).
L_DSRC_RSRC		
	SQN_DSsrcRsrc	Primarni umjetni ključ (sekvenca) link-entiteta.
	SQN_Resource	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta).
	SQN_DataSource	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta).
	DSrcRsrcLDTS	Datum punjenja zapisa o vezi između resursa i izvorišnog sustava u meta-bazu.
	DSrcRsrcRSRC	Izvor iz kojeg je zapis o vezi između resursa i izvorišnog sustava došao u meta-bazu.

H_SRC_RELATION		
	SQN_SrcRelation	Primarni umjetni ključ (sekvenca) hub-entiteta.
	SrcTableID	Poslovni ključ hub-entiteta.
	SrcRelationLDTS	Datum punjenja zapisa o izvorišnoj tablici u meta-bazu.
	SrcRelationRSRC	Izvor iz kojeg je zapis o izvorišnoj tablici došao u meta-bazu.
S_SRC_RELATION		
	SQN_SrcRelation	Dio primarnog ključa satelit-entiteta (ujedno i primarni ključ njegovog hub-entiteta H_SRC_TABLE).
	SrcRelationLDTS	Dio primarnog ključa satelit-entiteta – datum punjenja zapisa o izvorišnoj tablici u meta-bazu.
	SrcRelationLEDTS	Datum završetka punjenja zapisa o izvorišnoj tablici u meta-bazu.
	SrcRelationRSRC	Izvor iz kojeg je zapis o izvorišnoj tablici došao u meta-bazu.
	SrcRelName	Naziv izvorišne tablice.
	SrcRelDescription	Kratki opis izvorišne tablice.
	SrcRelBusinessDescription	Poslovni opis izvorišne tablice.
	SrcRelType	Tip izvorišne tablice (može biti nezavisna ili zavisna)
	ChangeType	Vrsta primjenjene promjene
	SchemaName	Verzija sheme zapisa (npr. 1, 2, 3...)
	SrcRelQualityType	Tip kvalitete izvorišne tablice (može biti niska, prosječna ili visoka).
L_DSRC_REL		
	SQN_DSRCREL	Primarni umjetni ključ (sekvenca) link-entiteta.

	SQLN_DataSource	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_DATA_SOURCE).
	SQLN_SrcRelation	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_SRC_TABLE).
	DSrcRelLDTS	Datum punjenja zapisa o vezi između izvorišnog sustava i izvorišne tablice u meta-bazu.
	DSrcRelRSRC	Izvor iz kojeg je zapis o vezi između izvorišnog sustava i izvorišne tablice došao u meta-bazu.
H_SRC_ATTRIBUTE		
	SQLN_SrcAttr	Primarni umjetni ključ (sekvenca) hub-entiteta.
	SrcAttrID	Poslovni ključ hub-entiteta.
	SrcAttrLDTS	Datum punjenja zapisa o izvorišnom atributu u meta-bazu.
	SrcAttrRSRC	Izvor iz kojeg je zapis o izvorišnom atributu došao u meta-bazu.
S_SRC_ATTR		
	SQLN_SrcAttr	Dio primarnog ključa satelit-entiteta (ujedno i primarni ključ njegovog hub-entiteta H_SRC_ATTRIBUTE).
	SrcAttrLDTS	Dio primarnog ključa satelit-entiteta – datum punjenja zapisa o izvorišnom atributu (stupcu) u meta-bazu.
	SrcAttrLEDTS	Datum završetka punjenja zapisa o izvorišnom atributu (stupcu) u meta-bazu.

	SrcAttrRSRC	Izvor iz kojeg je zapis o izvorišnom atributu (stupcu) došao u meta-bazu.
	SrcAttrPhysicalName	Fizički naziv izvorišnog atributa (stupca).
	SrcAttrBusinessName	Poslovni naziv izvorišnog atributa (stupca).
	SrcAttrDescription	Tekstualni opis izvorišnog atributa (stupca).
	ChangeType	Vrsta primjenjene promjene
	SrcAttrSize	Veličina izvorišnog atributa (stupca).
	SrcAttrNullable	Oznaka da je izvorišni atribut (stupac) NULL.
L_SRC_REL_ATTR		
	SQL_SrcRelAttr	Primarni umjetni ključ (sekvenca) link-entiteta.
	SQL_SrcRelation	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_SRC_TABLE).
	SQL_SrcAttr	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_SRC_ATTRIBUTE).
	SrcRelAttrLDTS	Datum punjenja zapisa o vezi između izvorišne tablice i izvorišnog resursa u meta-bazu.
	SrcRelAttrRSRC	Izvor iz kojeg je zapis o vezi između izvorišne tablice i izvorišnog resursa došao u meta-bazu.
H_SRC_RELATIONSHIP		
	SQL_SrcRelshp	Primarni umjetni ključ (sekvenca) hub-entiteta.
	SrcRelshpID	Poslovni ključ hub-entiteta.

	SrcRelLDTS	Datum punjenja zapisa o izvorišnoj vezi u meta-bazu.
	SrcRelRSRC	Izvor iz kojeg je zapis o izvorišnoj vezi došao u meta-bazu.
S_SRC_RELATIONSHIP		
	SQN_SrcRelshp	Dio primarnog ključa satelit-entiteta (ujedno i primarni ključ njegovog hub-entiteta H_SRC_RELATIONSHIP).
	SrcRelshpLDTS	Dio primarnog ključa satelit-entiteta – datum punjenja zapisa o izvorišnoj vezi u meta-bazu.
	SrcRelshpLEDTS	Datum završetka punjenja zapisa o izvorišnoj vezi u meta-bazu.
	SrcRelshpRSRC	Izvor iz kojeg je zapis o izvorišnoj vezi došao u meta-bazu.
	RelshpName	Naziv izvorišne veze.
	RelshpDescription	Tekstualni opis izvorišne veze.
L_SRC_REL_RELSHP		
	SQN_RelRelshp	Primarni umjetni ključ (sekvenca) link-entiteta.
	SQN_SrcRelation	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_SRC_RELATION).
	SQN_SrcRelshp	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_SRC_RELATIONSHIP).
	RelRelshpLDTS	Datum punjenja zapisa o vezi između izvorišne tablice i izvorišne veze u meta-bazu.
	RelRelshpRSRC	Izvor iz kojeg je zapis o vezi između izvorišne tablice i izvorišne veze došao u meta-bazu.
S_SRC_ROLE		

	SQL_RelRelshp	Dio primarnog ključa satelit-entiteta (ujedno i primarni ključ njegovog link-entiteta L_SRC_REL_RELSHP).
	SrcRoleLDTS	Dio primarnog ključa satelit-entiteta – datum punjenja zapisa o ulozi u meta-bazu.
	SrcRoleLEDTS	Datum završetka punjenja zapisa o ulozi u meta-bazu.
	SrcRoleRSRC	Izvor iz kojeg je zapis o ulozi došao u meta-bazu
	RoleName	Naziv uloge
	RoleDescription	Opis uloge
H_CONSTRAINT		
	SQL_SrcConstr	Primarni umjetni ključ (sekvenca) hub-entiteta.
	SrcConstrID	Poslovni ključ hub-entiteta.
	SrcConstrLDTS	Datum punjenja zapisa o izvorišnom ograničenju u meta-bazu.
	SrcConstrRSRC	Izvor iz kojeg je zapis o izvorišnom ograničenju došao u meta-bazu.
S_SRC_CONSTRAINT		
	SQL_SrcConstr	Dio primarnog ključa satelit-entiteta (ujedno i primarni ključ njegovog hub-entiteta H_CONSTRAINT).
	SrcConstrLDTS	Dio primarnog ključa satelit-entiteta – datum punjenja zapisa o izvorišnom ograničenju u meta-bazu.
	SrcConstrLEDTS	Datum završetka punjenja zapisa o izvorišnom ograničenju u meta-bazu.

	SrcConstrRSRC	Izvor iz kojeg je zapis o izvorišnom ograničenju došao u meta-bazu.
	ConstraintName	Naziv izvorišnog ograničenja.
	ConstraintType	Tip ograničenja (PK, FK, UNIQUE; CHECK)
	ConstraintDescription	Tekstualni opis izvorišnog ograničenja.
L_SRC_REL_ATTR_CONSTR		
	SQN_RelAttrConstr	Primarni umjetni ključ (sekvenca) link-entiteta.
	SQN_SrcRelation	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_SRC_TABLE).
	SQN_SrcRelshp	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_SRC_RELATIONSHIP).
	SQN_SrcConstr	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_CONSTRAINT).
	SQN_SrcAttr	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_SRC_ATTRIBUTE).
	RelAttrConstrLDTs	Datum punjenja zapisa o vezi između izvorišne tablice, izvorišnog atributa i izvorišnog ograničenja u meta-bazu.
	RelAttrConstrRSRC	Izvor iz kojeg je zapis o vezi između izvorišne tablice, izvorišnog atributa i izvorišnog ograničenja došao u meta-bazu.



Skladište podataka:

<b>NAZIV ENTITETA</b>	<b>DEFINICIJA ENTITETA</b>
H_HUB	Poslovni ključevi hubova – hub entitet.
S_HUB_DEF	Opisni podaci za hubove – satelit entitet.
S_BUSINESS_KEY	Opisni podaci o poslovnim ključevima hubova – satelit entitet.
SAL_MASTER_HUB	Zapisi o vezama između hubova – koji hubovi su master hubovi – link entitet.
S_CODE_HUB	Opisni podaci o master hubovima, odnosno kod koji se koristi za transformaciju huba u (master) hub – satelit entitet.
H_SATELLITE	Poslovni ključevi satelita – hub entitet.
S_SAT_DEF	Opisni podaci za satelite – satelit entitet.
SAL_MASTER_SATELLITE	Zapisi o vezama između satelita – koji sateliti su master sateliti – link entitet.
S_CODE_SAT	Opisni podaci o master satelitima, odnosno kod koji se koristi za transformaciju satelita u (master) satelit – satelit entitet.
H_LINK	Poslovni ključevi linkova – hub entitet.
S_LINK_DEF	Opisni podaci za linkove – satelit entitet.
SAL_MASTER_LINK	Zapisi o vezama između linkova – koji linkovi su master linkovi – link entitet.
S_CODE_LINK	Opisni podaci o master linkovima, odnosno kod koji se koristi za transformaciju linka u (master) link – satelit entitet.
H_ATTRIBUTE	Poslovni ključevi atributa – hub entitet.
S_ATTR_DEF	Opisni podaci za attribute – satelit entitet.
SAL_MASTER_ATTR	Zapisi o vezama između atributa – koji atributi su master atributi – link entitet.
S_CODE_ATTR	Opisni podaci o master atributima, odnosno kod koji se koristi za transformaciju atributa u (master) atribut – satelit entitet.
H_REFERENCE	Poslovni ključevi referenci – hub entitet.

S_REF_DEF	Opisni podaci za reference – satelit entitet.
H_RULE	Poslovni ključevi za poslovna pravila – hub entitet.
S_RULE_DEF	Opći opisni podaci za poslovna pravila – satelit entitet.
S_RULE_CODE	Kod za provedbu poslovnih pravila – satelit entitet.
HL_RULE_RELSHP	Veze između poslovnih pravila – link entitet.
S_RULE_RELSHP	Opisni podaci o vezama između poslovnih pravila – satelit entitet.
H_TRANSFORMATION	Poslovni ključevi za transformacije – hub entitet.
S_TRANSFORMATION	Opisni podaci za transformacije – satelit entitet.
HL_TRANSF_RELATIONSHIP	Veze između transformacija i grupiranja u hijerarhiju ili slijed transformacija – link entitet.
S_TRANSF_RELSHP	Opisni podaci o vezama između transformacija – satelit entitet.
H_DOMAIN	Poslovni ključevi za domene – hub entitet.
S_DOM_DEF	Opći opisni podaci za domene – satelit entitet.
L_DOM_SRCATTR	Veze između domene i izvorišnog atributa – link entitet.
L_ATTR_DOM	Veze između domene i atributa – link entitet.
L_HUB_SAT	Veze između huba i satelita – link entitet.
L_HUB_LINK	Veze između huba i linka – link entitet.
S_ROLE_DEF	Opisni podaci za uloge – satelit entitet.
L_SAT_LINK	Veze između satelita i linka – link entitet.
L_SAT_ATTR	Veze između satelita i atributa – link entitet.
L_ATTR_REF	Veze između atributa i reference – link entitet.
L_SAT_ATTR_REF	Veze između satelita, atributa i reference – link entitet.
L_REF_RULE	Veze između reference i poslovnog pravila – link entitet.
L_RELATION_HSL	Veze između izvorišne tablice i hubova, linkova i satelita (mapiranje u SDV iz izvora) – link entitet.
S_SDV_CODE_HSL	Opisni podaci za veze između izvorišne tablice i hubova, linkova i satelita – satelit entitet.
S_HSL_ISSUE	Podaci o greškama kod povezivanja izvora i SDV – satelit entitet
L_RELSHP_LINK	Veze između izvorišne veze i linka (mapiranje u SDV iz izvora) – link entitet.

S_SDV_CODE_LINK	Opisni podaci za veze između izvorišne veze i linka – satelit entitet.
L_ATTR_SRC_ATTR	Veze između izvorišnjog atributa i atributa (mapiranje u SDV iz izvora) – link entitet.
L_ATTR_DMATTR	Veze između atributa i atributa tržišta podataka (mapiranje u tržište podataka iz SDV/PMDV) – link entitet.
S_DM_CODE_DMATTR	Opisni podaci za veze između atributa i atributa tržišta podataka – satelit entitet.

NAZIV ENTITETA	NAZIV ATRIBUTA	DEFINICIJA ATRIBUTA
H_HUB		
	SQN_Hub	Primarni umjetni ključ (sekvenca) hub-entiteta.
	HubID	Poslovni ključ hub-entiteta.
	HubLDTS	Datum punjenja zapisa o hubu u meta-bazu.
	HubRSRC	Izvor iz kojeg je zapis o hubu došao u meta-bazu.
S_HUB_DEF		
	SQN_Hub	Dio primarnog ključa satelit-entiteta (ujedno i primarni ključ njegovog hub-entiteta H_HUB).
	HubDefLDTS	Dio primarnog ključa satelit-entiteta – datum punjenja zapisa o hubu u meta-bazu.
	HubDefLEDTS	Datum završetka punjenja zapisa o hubu u meta-bazu.
	HubDefRSRC	Izvor iz kojeg je zapis o hubu došao u meta-bazu.
	HubName	Naziv huba.
	HubDescription	Tekstualni opis huba.
	HubSchemaName	Verzija sheme kojoj hub pripada.
	HubType	Tip huba.

	ChangeType	Tip primijenjene promjene.
S_BUSINESS_KEY		
	SQN_Hub	Dio primarnog ključa satelit-entiteta (ujedno i primarni ključ njegovog hub-entiteta H_HUB).
	BusinessKeyLDTS	Dio primarnog ključa satelit-entiteta – datum punjenja zapisa o hubu u meta-bazu.
	BusinessKeyLEDTS	Datum završetka punjenja zapisa o hubu u meta-bazu.
	BusinessKeyRSRC	Izvor iz kojeg je zapis o hubu došao u meta-bazu.
	BusinessKey	Naziv poslovnog ključa huba.
	BusinessKeyDescription	Tekstualni opis poslovnog ključa huba.
	SourceKey	Naziv izvorišnog poslovnog ključa.
	SurrogateKey	Naziv surogat ključa huba.
SAL_MASTER_HUB		
	SQN_MasterHub	Primarni umjetni ključ (sekvenca) link-entiteta.
	SQN_HubBase	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_HUB).
	SQN_HubMaster	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_HUB).
	SQN_Rule	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_RULE).
	SQN_Transformation	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_TRANSFORMATION).
	MasterHubLDTS	Datum punjenja zapisa o vezi između osnovnog i master huba u meta-bazu.

	MasterHubRSRC	Izvor iz kojeg je zapis o vezi između osnovnog i master huba došao u meta-bazu.
S_CODE_HUB		
	SQN_MasterHub	Dio primarnog ključa satelit-entiteta (ujedno i primarni ključ njegovog link-entiteta SAL_MASTER_HUB).
	MasterCodeLDTs	Dio primarnog ključa satelit-entiteta – datum punjenja zapisa o master hubu u meta-bazu.
	MasterCodeLEDTS	Datum završetka punjenja zapisa o master hubu u meta-bazu.
	MasterCodeRSRC	Izvor iz kojeg je zapis o master hubu došao u meta-bazu.
	MasterCodeName	Naziv koda za transformaciju u master hub.
	MasterCodeDescription	Tekstualni opis koda za transformaciju u master hub.
	MasterCode	Kod za transformaciju u master hub.
H_SATELLITE		
	SQN_Satellite	Primarni umjetni ključ (sekvenca) hub-entiteta.
	SatelliteID	Poslovni ključ hub-entiteta.
	SatelliteLDTs	Datum punjenja zapisa o satelitu u meta-bazu.
	SatelliteRSRC	Izvor iz kojeg je zapis o satelitu došao u meta-bazu.
S_SAT_DEF		
	SQN_Satellite	Dio primarnog ključa satelit-entiteta (ujedno i primarni ključ njegovog hub-entiteta H_SATELLITE).
	SatDefLDTs	Dio primarnog ključa satelit-entiteta – datum punjenja zapisa o satelitu u meta-bazu.

	SatDefLEDTS	Datum završetka punjenja zapisa o satelitu u meta-bazu.
	SatDefRSRC	Izvor iz kojeg je zapis o satelitu došao u meta-bazu.
	SatName	Naziv satelita.
	SatDescription	Tekstualni opis satelita.
	SatSchemaName	Naziv sheme kojoj satelit pripada.
	SatType	Tip satelita.
	ChangeType	Tip primijenjene promjene
SAL_MASTER_SATELLITE		
	SQLN_MasterSatellite	Primarni umjetni ključ (sekvenca) link-entiteta.
	SQLN_SatelliteBase	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_SATELLITE).
	SQLN_SatelliteMaster	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_SATELLITE).
	SQLN_Rule	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_RULE).
	SQLN_Transformation	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_TRANSFORMATION).
	MasterSatelliteLDTS	Datum punjenja zapisa o vezi između osnovnog i master satelita u meta-bazu.
	MasterSatelliteRSRC	Izvor iz kojeg je zapis o vezi između osnovnog i master satelita došao u meta-bazu.
S_CODE_SAT		
	SQLN_MasterSatellite	Dio primarnog ključa satelit-entiteta (ujedno i primarni ključ njegovog link-entiteta SAL_MASTER_SATELLITE).

	MasterCodeLDTS	Dio primarnog ključa satelit-entiteta – datum punjenja zapisa o master satelitu u meta-bazu.
	MasterCodeLEDTS	Datum završetka punjenja zapisa o master satelitu u meta-bazu.
	MasterCodeRSRC	Izvor iz kojeg je zapis o master satelitu došao u meta-bazu.
	MasterCodeName	Naziv koda za transformaciju u master satelit.
	MasterCodeDescription	Tekstualni opis koda za transformaciju u master satelit.
	MasterCode	Kod za transformaciju u master satelit.
H_LINK		
	SQN_Link	Primarni umjetni ključ (sekvenca) hub-entiteta.
	LinkID	Poslovni ključ hub-entiteta.
	LinkLDTS	Datum punjenja zapisa o linku u meta-bazu.
	LinkRSRC	Izvor iz kojeg je zapis o linku došao u meta-bazu.
S_LINK_DEF		
	SQN_Link	Dio primarnog ključa satelit-entiteta (ujedno i primarni ključ njegovog hub-entiteta H_LINK).
	LinkDefLDTS	Dio primarnog ključa satelit-entiteta – datum punjenja zapisa o linku u meta-bazu.
	LinkDefLEDTS	Datum završetka punjenja zapisa o linku u meta-bazu.
	LinkDefRSRC	Izvor iz kojeg je zapis o linku došao u meta-bazu.
	LinkName	Naziv linka.
	LinkDescription	Tekstualni opis linka.

	LinkSchemaName	Naziv sheme kojoj link pripada.
	LinkType	Tip linka.
	ChangeType	Tip primijenjene promjene.
SAL_MASTER_LINK		
	SQN_MasterLink	Primarni umjetni ključ (sekvenca) link-entiteta.
	SQN_LinkBase	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_LINK).
	SQN_LinkMaster	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_LINK).
	SQN_Rule	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_RULE).
	SQN_Transformation	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_TRANSFORMATION).
	MasterLinkLDTS	Datum punjenja zapisa o vezi između osnovnog i master linka u meta-bazu.
	MasterLinkRSRC	Izvor iz kojeg je zapis o vezi između osnovnog i master linka došao u meta-bazu.
S_CODE_LINK		
	SQN_MasterLink	Dio primarnog ključa satelit-entiteta (ujedno i primarni ključ njegovog link-entiteta SAL_MASTER_LINK).
	MasterCodeLDTS	Dio primarnog ključa satelit-entiteta – datum punjenja zapisa o master linku u meta-bazu.
	MasterCodeLEDTS	Datum završetka punjenja zapisa o master linku u meta-bazu.
	MasterCodeRSRC	Izvor iz kojeg je zapis o master linku došao u meta-bazu.



	MasterCodeName	Naziv koda za transformaciju u master link.
	MasterCodeDescription	Tekstualni opis koda za transformaciju u master link.
	MasterCode	Kod za transformaciju u master link.
H_ATTRIBUTE		
	SQN_Attribute	Primarni umjetni ključ (sekvenca) hub-entiteta.
	AttributeID	Poslovni ključ hub-entiteta.
	Attribute LDTS	Datum punjenja zapisa o atributu u meta-bazu.
	Attribute RSRC	Izvor iz kojeg je zapis o atributu došao u meta-bazu.
S_ATTR_DEF		
	SQN_Attribute	Dio primarnog ključa satelit-entiteta (ujedno i primarni ključ njegovog hub-entiteta H_ATTRIBUTE).
	AttrDefLDTS	Dio primarnog ključa satelit-entiteta – datum punjenja zapisa o atributu u meta-bazu.
	AttrDefLEDTS	Datum završetka punjenja zapisa o atributu u meta-bazu.
	AttrDefRSRC	Izvor iz kojeg je zapis o atributu došao u meta-bazu.
	AttrName	Naziv atributa.
	AttrDescription	Tekstualni opis atributa.
	AttrSize	Veličina atributa.
	AttrNullable	Može li imati NULL vrijednosti.
	AttrAllowedValues	Dozvoljene vrijednosti.
	AttrDefaultValue	Početna standardna vrijednost.
	ChangeTypa	Tip primijenjene promjene.
SAL_MASTER_ATTRIBUTE		
	SQN_MasterAttribute	Primarni umjetni ključ (sekvenca) link-entiteta.

	SQN_AttributeBase	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_ATTRIBUTE).
	SQN_AttributeMaster	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_ATTRIBUTE).
	SQN_Rule	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_RULE).
	SQN_Transformation	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_TRANSFORMATION).
	MasterAttributeLDS	Datum punjenja zapisa o vezi između osnovnog i master atributa u meta-bazu.
	MasterAttributeRSRC	Izvor iz kojeg je zapis o vezi između osnovnog i master atributa došao u meta-bazu.
S_CODE_ATTR		
	SQN_MasterAttribute	Dio primarnog ključa satelit-entiteta (ujedno i primarni ključ njegovog link-entiteta SAL_MASTER_ATTRIBUTE).
	MasterCodeLDS	Dio primarnog ključa satelit-entiteta – datum punjenja zapisa o master atributu u meta-bazu.
	MasterCodeLEDTS	Datum završetka punjenja zapisa o master atributu u meta-bazu.
	MasterCodeRSRC	Izvor iz kojeg je zapis o master atributu došao u meta-bazu.
	MasterCodeName	Naziv koda za transformaciju u master atribut.
	MasterCodeDescription	Tekstualni opis koda za transformaciju u master atribut.

	MasterCode	Kod za transformaciju u master atribut.
H_REFERENCE		
	SQN_Reference	Primarni umjetni ključ (sekvenca) hub-entiteta.
	ReferenceID	Poslovni ključ hub-entiteta.
	ReferenceLDTS	Datum punjenja zapisa o referenci u meta-bazu.
	ReferenceRSRC	Izvor iz kojeg je zapis o referenci došao u meta-bazu.
S_REF_DEF		
	SQN_Reference	Dio primarnog ključa satelit-entiteta (ujedno i primarni ključ njegovog hub-entiteta H_REFERENCE).
	RefDefLDTS	Dio primarnog ključa satelit-entiteta – datum punjenja zapisa o referenci u meta-bazu.
	RefDefLEDTS	Datum završetka punjenja zapisa o referenci u meta-bazu.
	RefDefRSRC	Izvor iz kojeg je zapis o referenci došao u meta-bazu.
	RefName	Naziv reference.
	RefDescription	Tekstualni opis reference.
	RefSchemaName	Verzija sheme kojoj referenca pripada.
	RefType	Tip reference.
	ChangeType	Vrsta primijenjene promjene.
H_RULE		
	SQN_Rule	Primarni umjetni ključ (sekvenca) hub-entiteta.
	RuleID	Poslovni ključ hub-entiteta.
	RuleLDTS	Datum punjenja zapisa o poslovnom pravilu u meta-bazu.
	RuleRSRC	Izvor iz kojeg je zapis o poslovnom pravilu došao u meta-bazu.

S_RULE_DEF		
	SQN_Rule	Dio primarnog ključa satelit-entiteta (ujedno i primarni ključ njegovog hub-entiteta H_RULE).
	RuleDefLDTS	Dio primarnog ključa satelit-entiteta – datum punjenja zapisa o poslovnom pravilu u meta-bazu.
	RuleDefLEDTS	Datum završetka punjenja zapisa o poslovnom pravilu u meta-bazu.
	RuleDefRSRC	Izvor iz kojeg je zapis o poslovnom pravilu došao u meta-bazu.
	RuleName	Naziv poslovnog pravila.
	RuleShortDescription	Kratki tekstualni opis poslovnog pravila.
	RuleTextDescription	Duži tekstualni opis poslovnog pravila.
	RuleType	Tip poslovnog pravila.
	RuleCategory	Kategorija poslovnog pravila.
S_HSL_ISSUE		
	SQN_RelHSL	Dio primarnog ključa satelit-entiteta (ujedno i primarni ključ njegovog link-entiteta L_RELATION_HSL).
	IssueLDTS	Dio primarnog ključa satelit-entiteta – datum punjenja zapisa o problemu u meta-bazu.
	IssueLEDTS	Datum završetka punjenja zapisa o problemu u meta-bazu.
	IssueRSRC	Izvor iz kojeg je zapis o problemu došao u meta-bazu.
	IssueNumber	Broj problema/greške.
	IssueName	Naziv problema.
	IssueDescription	Tekstualni opis problema.
	IssuePriority	Razina prioriteta za rješavanje problema (može biti niska, srednja, visoka).

	IssueResolution	Opis na koji način se rješava problem.
	IssueStatus	Status (stanje) problema (nije riješen ili riješen).
S_RULE_CODE		
	SQN_Rule	Dio primarnog ključa satelit-entiteta (ujedno i primarni ključ njegovog hub-entiteta H_RULE).
	RuleCodeLDTS	Dio primarnog ključa satelit-entiteta – datum punjenja zapisa o kodu poslovnog pravila u meta-bazu.
	RuleCodeLEDTS	Datum završetka punjenja zapisa o o kodu poslovnog pravila u meta-bazu.
	RuleCodeRSRC	Izvor iz kojeg je zapis o kodu poslovnog pravila došao u meta-bazu.
	RuleCodeName	Naziv koda poslovnog pravila.
	RuleCodeDescription	Tekstualni opis koda poslovnog pravila.
	RuleCode	Kod poslovnog pravila.
HL_RULE_RELSHP		
	SQN_RuleRelshp	Primarni umjetni ključ (sekvenca) link-entiteta.
	SQN_BaseRule	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_RULE).
	SQN_RelatedRule	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_RULE).
	RuleRelshpLDTS	Datum punjenja zapisa o vezi između osnovnog i vezanog poslovnog pravila u meta-bazu.
	RuleRelshpRSRC	Izvor iz kojeg je zapis o vezi između osnovnog i vezanog poslovnog pravila došao u meta-bazu.
S_RULE_RELSHP		

	SQL_RuleRelshp	Dio primarnog ključa satelit-entiteta (ujedno i primarni ključ njegovog link-entiteta HL_RULE_RELATIONSHIP).
	RuleRelshpLDTS	Dio primarnog ključa satelit-entiteta – datum punjenja zapisa vezi između osnovnog i vezanog poslovnog pravila u meta-bazu.
	RuleRelshpLEDTS	Datum završetka punjenja zapisa o vezi između osnovnog i vezanog poslovnog pravila u meta-bazu.
	RuleRelshpRSRC	Izvor iz kojeg je zapis o vezi između osnovnog i vezanog poslovnog pravila došao u meta-bazu.
	RuleRelshpType	Tip veze između osnovnog i vezanog poslovnog pravila.
	RuleRelshpDescription	Tekstualni opis tipa veze između osnovnog i vezanog poslovnog pravila.
H_TRANSFORMATION		
	SQL_Transformation	Primarni umjetni ključ (sekvenca) hub-entiteta.
	TransformationID	Poslovni ključ hub-entiteta.
	TransformationLDTS	Datum punjenja zapisa o transformaciji u meta-bazu.
	TransformationRSRC	Izvor iz kojeg je zapis o transformaciji došao u meta-bazu.
S_TRANSFORMATION		
	SQL_Transformation	Dio primarnog ključa satelit-entiteta (ujedno i primarni ključ njegovog hub-entiteta H_TRANSFORMATION).
	TransformationLDTS	Dio primarnog ključa satelit-entiteta – datum punjenja zapisa o transformaciji u meta-bazu.
	TransformationLEDTS	Datum završetka punjenja zapisa o transformaciji u meta-bazu.

	TransformationRSRC	Izvor iz kojeg je zapis o transformaciji došao u meta-bazu.
	TransformationName	Naziv transformacije.
	TransformationDescription	Kratki tekstualni opis transformacije.
	TransformationType	Tip transformacije (npr. agregacija, filtriranje, spajanje, unija, presjek, pivot, ...).
	TransformationSoftware	Naziv softvera koji vrši transformaciju.
	ETLRepository	Naziv repozitorija ETL procesa.
HL_TRANSF_RELSHP		
	SQL_TransfRelationship	Primarni umjetni ključ (sekvenca) link-entiteta.
	SQL_TransformationBase	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_TRANSFORMATION).
	SQL_TransformationNext	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_TRANSFORMATION).
	TransfRelshpLDTS	Datum punjenja zapisa o vezi između osnovne i vezane transformacije u meta-bazu.
	TransfRelshpRSRC	Izvor iz kojeg je zapis o vezi između osnovne i vezane transformacije došao u meta-bazu.
HL_TRANSF_RELSHP		
	SQL_TransfRelationship	Dio primarnog ključa satelita iz link-entiteta.
	TransfRelshpLDTS	Datum punjenja zapisa o vezi između osnovne i vezane transformacije u meta-bazu – dio primarnog ključa.
	TransfRelshpLEDTS	Datum završetka punjenja zapisa u meta-bazu.

	TransfRelshpRSRC	Izvor iz kojeg je zapis o vezi između osnovne i vezane transformacije došao u meta-bazu.
	TransfRelshpType	Tip veze između osnovne i vezane transformacije.
	TransfRelshpDescription	Tekstualni opis tipa veze između osnovne i vezane transformacije.
H_DOMAIN		
	SQN_Domain	Primarni umjetni ključ (sekvenca) hub-entiteta.
	DomainID	Poslovni ključ hub-entiteta.
	DomainLDTS	Datum punjenja zapisa o domeni u meta-bazu.
	DomainRSRC	Izvor iz kojeg je zapis o domeni došao u meta-bazu.
S_DOM_DEF		
	SQN_Domain	Dio primarnog ključa satelit-entiteta (ujedno i primarni ključ njegovog hub-entiteta H_DOMAIN).
	DomDefLDTS	Dio primarnog ključa satelit-entiteta – datum punjenja zapisa o domeni u meta-bazu.
	DomDefLEDTS	Datum završetka punjenja zapisa o domeni u meta-bazu.
	DomDefRSRC	Izvor iz kojeg je zapis o domeni došao u meta-bazu.
	DomDescription	Tekstualni opis domene.
	DataType	Podatkovni tip domene.
L_DOM_SRCATTR		
	SQN_DomSrcAttr	Primarni umjetni ključ (sekvenca) link-entiteta.
	SQN_Domain	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_DOMAIN).



	SQL_SrcAttr	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_SRC_ATTRIBUTE).
	DomSrcAttrLDTS	Datum punjenja zapisa o vezi između izvorišnog atributa i domene u meta-bazu.
	DomSrcAttrRSRC	Izvor iz kojeg je zapis o vezi između izvorišnog atributa i domene došao u meta-bazu.
L_ATTR_DOM		
	SQL_AttrDom	Primarni umjetni ključ (sekvenca) link-entiteta.
	SQL_Domain	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_DOMAIN).
	SQL_Attribute	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_ATTRIBUTE).
	SQL_Rule	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_RULE).
	AttrDomLDTS	Datum punjenja zapisa o vezi između atributa i domene u meta-bazu.
	AttrDomRSRC	Izvor iz kojeg je zapis o vezi između atributa i domene došao u meta-bazu.
L_HUB_SAT		
	SQL_HubSat	Primarni umjetni ključ (sekvenca) link-entiteta.
	SQL_Hub	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_HUB).
	SQL_Satellite	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_SATELLITE).

	HubSatLDTS	Datum punjenja zapisa o vezi između huba i satelita u meta-bazu.
	HubSatRSRC	Izvor iz kojeg je zapis o vezi između huba i satelita došao u meta-bazu.
L_HUB_LINK		
	SQN_HubLink	Primarni umjetni ključ (sekvenca) link-entiteta.
	SQN_Hub	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_HUB).
	SQN_Link	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_LINK).
	SQN_Role	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_ROLE).
	HubLinkLDTS	Datum punjenja zapisa o vezi između huba i linka u meta-bazu.
	HubLinkRSRC	Izvor iz kojeg je zapis o vezi između huba i linka došao u meta-bazu.
S_ROLE_DEF		
	SQN_HubLink	Dio primarnog ključa satelit-entiteta (ujedno i primarni ključ njegovog link-entiteta L_HUB_LINK).
	RoleDefLDTS	Dio primarnog ključa satelit-entiteta – datum punjenja zapisa o ulozi veze u meta-bazu.
	RoleDefLEDTS	Datum završetka punjenja zapisa o ulozi veze u meta-bazu.
	RoleDefRSRC	Izvor iz kojeg je zapis o ulozi veze došao u meta-bazu.
	RoleName	Naziv uloge veze.
	RoleDescription	Tekstualni opis uloge veze.
L_SAT_LINK		

	SQL_SatLink	Primarni umjetni ključ (sekvenca) link-entiteta.
	SQL_Link	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_LINK).
	SQL_Satellite	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_SATELLITE).
	SatLinkLDTS	Datum punjenja zapisa o vezi između satelita i linka u meta-bazu.
	SatLinkRSRC	Izvor iz kojeg je zapis o vezi između satelita i linka došao u meta-bazu.
L_SATELLITE_ATTRIBUTE		
	SQL_SatAttr	Primarni umjetni ključ (sekvenca) link-entiteta.
	SQL_Attribute	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_ATTRIBUTE).
	SQL_Satellite	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_SATELLITE).
	SatAttrLDTS	Datum punjenja zapisa o vezi između satelita i atributa u meta-bazu.
	SatAttrRSRC	Izvor iz kojeg je zapis o vezi između satelita i atributa došao u meta-bazu.
L_ATTR_REF		
	SQL_AttrRef	Primarni umjetni ključ (sekvenca) link-entiteta.
	SQL_Attribute	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_ATTRIBUTE).
	SQL_Reference	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_REFERENCE).

	AttrRefLDTS	Datum punjenja zapisa o vezi između reference i atributa u meta-bazu.
	AttrRefRSRC	Izvor iz kojeg je zapis o vezi između reference i atributa došao u meta-bazu.
L_ SAT_ATTR_REF		
	SQN_SatAttrRef	Primarni umjetni ključ (sekvenca) link-entiteta.
	SQN_Attribute	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_ATTRIBUTE).
	SQN_Reference	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_REFERENCE).
	SQN_Satellite	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_SATELLITE).
	SatAttrRefLDTS	Datum punjenja zapisa o vezi između satelita, reference i atributa u meta-bazu.
	SatAttrRefRSRC	Izvor iz kojeg je zapis o vezi između satelita, reference i atributa došao u meta-bazu.
L_REF_RULE		
	SQN_RefRule	Primarni umjetni ključ (sekvenca) link-entiteta.
	SQN_Rule	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_RULE).
	SQN_Reference	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_REFERENCE).

	RefRuleLDTS	Datum punjenja zapisa o vezi između reference i poslovnog pravila u meta-bazu.
	RefRuleRSRC	Izvor iz kojeg je zapis o vezi između reference i poslovnog pravila došao u meta-bazu.
L_RELATION_HSL		
	SQN_RelHSL	Primarni umjetni ključ (sekvenca) link-entiteta.
	SQN_Link	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_LINK).
	SQN_Hub	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_HUB).
	SQN_Satellite	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_SATELLITE).
	SQN_SrcRelation	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_SRC_RELATION).
	SQN_Rule	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_RULE).
	SQN_Transformation	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_TRANSFORMATION).
	RelHSL_LDTS	Datum punjenja zapisa o vezi između izvorišne tablice i SDV tablicau meta-bazu.
	RelHSL_RSRC	Izvor iz kojeg je zapis o vezi između izvorišne tablice i SDV tablica došao u meta-bazu.
S_SDV_CODE_HSL		

	SQN_RelHSL	Dio primarnog ključa satelit-entiteta (ujedno i primarni ključ njegovog link-entiteta L_RELATION_HSL).
	SDVCodeLDTS	Dio primarnog ključa satelit-entiteta – datum punjenja zapisa o vezi izvorišne tablice i SDV tablica u meta-bazu.
	SDVCode LEDTS	Datum završetka punjenja zapisa o vezi izvorišne tablice i SDV tablica u meta-bazu.
	SDVCode RSRC	Izvor iz kojeg je zapis o vezi izvorišne tablice i SDV tablica došao u meta-bazu.
	SDVCodeName	Naziv koda za punjenje izvorišne tablice u SDV tablice.
	SDVCodeDescription	Tekstualni opis koda za punjenje izvorišne tablice u hub.
	SDVCode	Kod za punjenje izvorišne tablice u SDV tablice.
L_RELSHP_LINK		
	SQN_RelshpLink	Primarni umjetni ključ (sekvenca) link-entiteta.
	SQN_Link	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_LINK).
	SQN_SrcRelshp	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_SRC_RELATIONSHIP).
	SQN_Rule	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_RULE).
	SQN_Transformation	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_TRANSFORMATION).

	RelatLinkLDTS	Datum punjenja zapisa o vezi između izvorišne veze i linka u meta-bazu.
	RelatLinkRSRC	Izvor iz kojeg je zapis o vezi između izvorišne veze i linka došao u meta-bazu.
S_SDV_CODE_LINK		
	SQN_RelshpLink	Dio primarnog ključa satelit-entiteta (ujedno i primarni ključ njegovog link-entiteta L_RELAT_LINK).
	SDVCodeLDTS	Dio primarnog ključa satelit-entiteta – datum punjenja zapisa o vezi izvorišne veze i linka u meta-bazu.
	SDVCode LEDTS	Datum završetka punjenja zapisa o vezi izvorišne veze i linka u meta-bazu.
	SDVCode RSRC	Izvor iz kojeg je zapis o vezi izvorišne veze i linka došao u meta-bazu.
	SDVCodeName	Naziv koda za punjenje izvorišne veze u link.
	SDVCodeDescription	Tekstualni opis koda za punjenje izvorišne veze u link.
	SDVCode	Kod za punjenje izvorišne veze u link.
S_LINK_ISSUE		
	SQN_RelshpLink	Dio primarnog ključa satelit-entiteta (ujedno i primarni ključ njegovog link-entiteta L_RELSHP_LINK).
	IssueLDTS	Dio primarnog ključa satelit-entiteta – datum punjenja zapisa o problemu u meta-bazu.
	IssueLEDTS	Datum završetka punjenja zapisa o problemu u meta-bazu.
	IssueRSRC	Izvor iz kojeg je zapis o problemu došao u meta-bazu.
	IssueNumber	Broj problema/greške.
	IssueName	Naziv problema.

	IssueDescription	Tekstualni opis problema.
	IssuePriority	Razina prioriteta za rješavanje problema (može biti niska, srednja, visoka).
	IssueResolution	Opis na koji način se rješava problem.
	IssueStatus	Status (stanje) problema (nije riješen ili riješen).
L_ATTR_SRC_ATTR		
	SQN_AttrSrcAttr	Primarni umjetni ključ (sekvenca) link-entiteta.
	SQN_Attribute	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_ATTRIBUTE).
	SQN_SrcAttr	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_SRC_ATTRIBUTE).
	SQN_Rule	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_RULE).
	SQN_Transformation	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_TRANSFORMATION).
	AttrSrcAttrLDTS	Datum punjenja zapisa o vezi između izvorišnog atributa i atributa u meta-bazu.
	AttrSrcAttrRSRC	Izvor iz kojeg je zapis o vezi između izvorišnog atributa i atributa došao u meta-bazu.
S_SDV_CODE_ATTR		
	SQN_AttrSrcAttr	Dio primarnog ključa satelit-entiteta (ujedno i primarni ključ njegovog link-entiteta L_ATTR_SRC_ATTR).
	CodeLDTS	Dio primarnog ključa satelit-entiteta – datum punjenja zapisa o vezi



		izvorišnog atributa i atributa u meta-bazu.
	Code LEDTS	Datum završetka punjenja zapisa o vezi izvorišnog atributa i atributa u meta-bazu.
	Code RSRC	Izvor iz kojeg je zapis o vezi izvorišnog atributa i atributa došao u meta-bazu.
	CodeName	Naziv koda za punjenje izvorišnog atributa u atribut.
	CodeDescription	Tekstualni opis koda za punjenje izvorišnog atributa u atribut.
	Code	Kod za punjenje izvorišnog atributa u atribut.
L_ATTR_DMATTR		
	SQL_AttrDMAttr	Primarni umjetni ključ (sekvenca) link-entiteta.
	SQL_Attribute	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_ATTRIBUTE).
	SQL_DMAttr	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_DM_ATTRIBUTE).
	SQL_Rule	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_RULE).
	SQL_Transformation	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_TRANSFORMATION).
	AttrDMAttrLDTS	Datum punjenja zapisa o vezi između atributa i atributa tržišta podataka u meta-bazu.
	AttrDMAttrRSRC	Izvor iz kojeg je zapis o vezi između atributa i atributa tržišta podataka došao u meta-bazu.

S_DM_CODE_DMATTR		
	SQL_AttrDMAttr	Dio primarnog ključa satelit-entiteta (ujedno i primarni ključ njegovog link-entiteta L_ATTR_DMATTR).
	DMCodeLDTS	Dio primarnog ključa satelit-entiteta – datum punjenja zapisa o vezi atributa i atributa tržišta podataka u meta-bazu.
	DMCode LEDTS	Datum završetka punjenja zapisa o vezi atributa i atributa tržišta podataka u meta-bazu.
	DMCode RSRC	Izvor iz kojeg je zapis o vezi atributa i atributa tržišta podataka došao u meta-bazu.
	DMCodeName	Naziv koda za punjenje atributa u atribut tržišta podataka.
	DMCodeDescription	Tekstualni opis koda za punjenje atributa u atribut tržišta podataka.
	DMCode	Kod za punjenje atributa u atribut tržišta podataka.

Područno skladište podataka:

NAZIV ENTITETA	DEFINICIJA ENTITETA
H_OBJECT	Poslovni ključevi za objekte u tržištu podataka (objekti mogu biti činjenice, dimenzije ili materijalizirani pogledi) – hub entitet.
H_FACT	Poslovni ključevi za tablice činjenica u tržištu podataka – hub entitet.
S_FACT_DEF	Opisni podaci za tablicu činjenica – satelit entitet.
H_DIMENSION	Poslovni ključevi za tablicu dimenzija – hub entitet.
S_DIM_DEF	Opisni podaci za tablicu dimenzija – satelit entitet.
L_FACT_DIM	Veze između činjenica i dimenzija – link entitet.
H_HIERARCHY	Poslovni ključevi za dimenzijske hijerarhije – hub entitet.
S_HIER_DEF	Opisni podaci za dimenzijske hijerarhije – satelit entitet.
L_DIM_HIER	Pridruživanje dimenzijskih hijerarhija dimenzijama – link entitet.
H_LEVEL	Poslovni ključevi za hijerarhijsku razinu – hub entitet.
S_LEVEL_DEF	Opisni podaci za hijerarhijsku razinu – satelit entitet.
L_HIER_LEVEL	Pridruživanje hijerarhijske razine hijerarhiji – link entitet.
H_DM_ATTR	Poslovni ključevi za attribute kocke (tržišta podataka) – hub entitet.
S_DM_ATTR_DEF	Opisni podaci za attribute kocke (tržišta podataka) – satelit entitet.
S_MEASURE	Opisni podaci za činjenične mjere – satelit entitet.
L_DMATTR_LEVEL	Pridruživanje atributa tržišta podataka hijerarhijskoj razini – link entitet.
S_DMATTR_LEVEL	Opisni podaci za pridruživanje atributa tržišta podataka hijerarhijskoj razini – satelit entitet.
L_DMATTR_FACT	Pridruživanje atributa tržišta podataka činjenici – link entitet.
L_DMATTR_DIM	Pridruživanje atributa tržišta podataka dimenziji – link entitet.
L_CUBE	Kreiranje podatkovne kocke - pridruživanje atributa tržišta podataka, činjenice, dimenzije i hijerarhije – link entitet.
H_MV	Poslovni ključevi za materijalizirani pogled (koji predstavlja podatkovnu kocku) – hub entitet.

S_MV_DEF	Opisni podaci za materijalizirani pogled – satelit entitet.
S_MV_QUERY	Opisni podaci za upit materijaliziranog pogleda– satelit entitet.
L_CUBE_MV	Kreiranje podatkovne kocke kroz materijalizirani pogled - pridruživanje materijaliziranog pogleda, atributa tržišta podataka, činjenice, dimenzije, hijerarhije i razine – link entitet.

NAZIV ENTITETA	NAZIV ATRIBUTA	DEFINICIJA ATRIBUTA
H_OBJECT		
	SQN_Object	Primarni umjetni ključ (sekvenca) hub-entiteta.
	ObjectID	Poslovni ključ hub-entiteta.
	ObjectLDTS	Datum punjenja zapisa o objektu u meta-bazu.
	ObjectRSRC	Izvor iz kojeg je zapis o objektu došao u meta-bazu.
H_FACT		
	SQN_Fact	Primarni umjetni ključ (sekvenca) hub-entiteta.
	SQN_Object	Primarni umjetni ključ (sekvenca) hub-entiteta.
	FactID	Poslovni ključ hub-entiteta.
	FactLDTS	Datum punjenja zapisa o činjenici u meta-bazu.
	FactRSRC	Izvor iz kojeg je zapis o činjenici došao u meta-bazu.
S_FACT_DEF		
	SQN_Fact	Dio primarnog ključa satelit-entiteta (ujedno i primarni ključ njegovog hub-entiteta H_FACT).
	SQN_Object	Dio primarnog ključa satelit-entiteta (ujedno i primarni ključ njegovog hub-entiteta H_OBJECT).

	FactLDTS	Dio primarnog ključa satelit-entiteta – datum punjenja zapisa o činjenici u meta-bazu.
	FactLEDTS	Datum završetka punjenja zapisa o činjenici u meta-bazu.
	FactRSRC	Izvor iz kojeg je zapis o činjenici došao u meta-bazu.
	FactName	Naziv činjenice.
	FactDescription	Tekstualni opis činjenice.
	SchemaName	Naziv sheme kojoj činjenica pripada.
	FactType	Tip činjenice (može biti transakcijska tablica, periodični snimak ili akumulirajući snimak).
H_DIMENSION		
	SQLN_Dimension	Primarni umjetni ključ (sekvenca) hub-entiteta.
	SQLN_Object	Primarni umjetni ključ (sekvenca) hub-entiteta.
	DimensionID	Poslovni ključ hub-entiteta.
	DimensionLDTS	Datum punjenja zapisa o dimenziji u meta-bazu.
	DimensionRSRC	Izvor iz kojeg je zapis o dimenziji došao u meta-bazu.
S_DIM_DEF		
	SQLN_Dimension	Dio primarnog ključa satelit-entiteta (ujedno i primarni ključ njegovog hub-entiteta H_DIMENSION).
	SQLN_Object	Dio primarnog ključa satelit-entiteta (ujedno i primarni ključ njegovog hub-entiteta H_OBJECT).
	DimensionLDTS	Dio primarnog ključa satelit-entiteta – datum punjenja zapisa o dimenziji u meta-bazu.

	DimensionLEDTS	Datum završetka punjenja zapisa o dimenziji u meta-bazu.
	DimensionRSRC	Izvor iz kojeg je zapis o dimenziji došao u meta-bazu.
	DimensionName	Naziv dimenzije.
	DimensionDescription	Tekstualni opis dimenzije.
	SchemaName	Naziv sheme kojoj dimenzija pripada.
	DimensionType	Tip dimenzije (može biti usklađena, nepotrebna, degenerirana ili višenamjenska).
	DimSCDType	Tip sporo mijenjajuće dimenzije (može biti 0, 1, 2, 3, 4, 5, 6 ili 7)
	DimAdditive	Govori je li dimenzija zbrojiva, poluzbrojiva ili nezbroyiva.
L_FACT_DIM		
	SQN_FactDim	Primarni umjetni ključ (sekvenca) link-entiteta.
	SQN_Fact	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_FACT).
	SQN_Dimension	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_DIMENSION).
	SQN_Object	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_OBJECT).
	FactDimLDTS	Datum punjenja zapisa o vezi između činjenice i dimenzije u meta-bazu.
	FactDimRSRC	Izvor iz kojeg je zapis o vezi između činjenice i dimenzije došao u meta-bazu.
H_HIERARCHY		

	SQL_Hierarchy	Primarni umjetni ključ (sekvenca) hub-entiteta.
	HierarchyID	Poslovni ključ hub-entiteta.
	HierarchyLDTS	Datum punjenja zapisa o hijerarhiji u meta-bazu.
	HierarchyRSRC	Izvor iz kojeg je zapis o hijerarhiji došao u meta-bazu.
S_HIER_DEF		
	SQL_Hierarchy	Dio primarnog ključa satelit-entiteta (ujedno i primarni ključ njegovog hub-entiteta H_HIERARCHY).
	HierarchyLDTS	Dio primarnog ključa satelit-entiteta – datum punjenja zapisa o hijerarhiji u meta-bazu.
	HierarchyLEDTS	Datum završetka punjenja zapisa o hijerarhiji u meta-bazu.
	HierarchyRSRC	Izvor iz kojeg je zapis o hijerarhiji došao u meta-bazu.
	HierarchyName	Naziv hijerarhije.
	HierarchyDescription	Tekstualni opis hijerarhije.
	SchemaName	Naziv sheme kojoj hijerarhija pripada.
	HierarchyType	Tip hijerarhije (može biti uravnotežena, neuravnotežena ili pocijepana).
L_DIM_HIER		
	SQL_DimHier	Primarni umjetni ključ (sekvenca) link-entiteta.
	SQL_Hierarchy	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_HIERARCHY).
	SQL_Dimension	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_DIMENSION).

	SQL_Object	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_OBJECT).
	DimHierLDTS	Datum punjenja zapisa o vezi između dimenzije i hijerarhije u meta-bazu.
	DimHierRSRC	Izvor iz kojeg je zapis o vezi između dimenzije i hijerarhije došao u meta-bazu.
H_LEVEL		
	SQL_Level	Primarni umjetni ključ (sekvenca) hub-entiteta.
	LevelID	Poslovni ključ hub-entiteta.
	LevelLDTS	Datum punjenja zapisa o razini u meta-bazu.
	LevelRSRC	Izvor iz kojeg je zapis o razini došao u meta-bazu.
S_LEVEL_DEF		
	SQL_Level	Dio primarnog ključa satelit-entiteta (ujedno i primarni ključ njegovog hub-entiteta H_LEVEL).
	LevelLDTS	Dio primarnog ključa satelit-entiteta – datum punjenja zapisa o razini u meta-bazu.
	LevelLEDTS	Datum završetka punjenja zapisa o razini u meta-bazu.
	LevelRSRC	Izvor iz kojeg je zapis o razini došao u meta-bazu.
	LevelName	Naziv razine.
	Level Description	Tekstualni opis razine.
	SchemaName	Naziv sheme kojoj razina pripada.
	LevelCardinality	Simbolička aproksimacija broja članova razine.
L_HIER_LEVEL		



	SQLN_HierLevel	Primarni umjetni ključ (sekvenca) link-entiteta.
	SQLN_Hierarchy	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_HIERARCHY).
	SQLN_Level	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_LEVEL).
	HierLevelLDTS	Datum punjenja zapisa o vezi između hijerarhije i razine u meta-bazu.
	HierLevelRSRC	Izvor iz kojeg je zapis o vezi između hijerarhije i razine došao u meta-bazu.
H_DM_ATTR		
	SQLN_DMAttr	Primarni umjetni ključ (sekvenca) hub-entiteta.
	DMAttrID	Poslovni ključ hub-entiteta.
	DMAttrLDTS	Datum punjenja zapisa o atributu tržišta podataka u meta-bazu.
	DMAttrRSRC	Izvor iz kojeg je zapis o atributu tržišta podataka došao u meta-bazu.
S_DM_ATTR_DEF		
	SQLN_DMAttr	Dio primarnog ključa satelit-entiteta (ujedno i primarni ključ njegovog hub-entiteta H_DM_ATTRIBUTE).
	DMAttrDefLDTS	Dio primarnog ključa satelit-entiteta – datum punjenja zapisa o atributu tržišta podataka u meta-bazu.
	DMAttrDefLEDTS	Datum završetka punjenja zapisa o atributu tržišta podataka u meta-bazu.

	DMAAttrDefRSRC	Izvor iz kojeg je zapis o atributu tržišta podataka došao u meta-bazu.
	DMAAttrPhysicalName	Fizički naziv atributa tržišta podataka.
	DMAAttrBusinessName	Fizički naziv atributa tržišta podataka.
	DMAAttrDescription	Tekstualni opis atributa tržišta podataka.
	SchemaName	Naziv sheme kojoj atribut tržišta podataka pripada.
	FactAttributeFlag	Oznaka pripada li atribut tržišta podataka činjenici.
	DimensionAttributeFlag	Oznaka pripada li atribut tržišta podataka dimenziji.
S_MEASURE		
	SQN_DMAAttr	Dio primarnog ključa satelit-entiteta (ujedno i primarni ključ njegovog hub-entiteta H_DM_ATTRIBUTE).
	MeasureLDTS	Dio primarnog ključa satelit-entiteta – datum punjenja zapisa o mjeri u meta-bazu.
	MeasureLEDTS	Datum završetka punjenja zapisa o mjeri u meta-bazu.
	MeasureRSRC	Izvor iz kojeg je zapis o mjeri došao u meta-bazu.
	AggregateFunctions	Popis funkcija po kojima se mjera može agregirati (npr. sum, avg, min, max, count).
	Additive	Je li mjera zbrojiva, poluzbrojiva ili nezbrojiva.
L_DMATTR_LEVEL		
	SQN_DMAAttrLevel	Primarni umjetni ključ (sekvenca) link-entiteta.

	SQL_DMAttr	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_DM_ATTRIBUTE).
	SQL_Level	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_LEVEL).
	DMAAttrLevelDTS	Datum punjenja zapisa o vezi između atributa tržišta podataka i razine u meta-bazu.
	DMAAttrLevelSRC	Izvor iz kojeg je zapis o vezi između atributa tržišta podataka i razine došao u meta-bazu.
S_DMATTR_LEVEL		
	SQL_DMAttrLevel	Dio primarnog ključa satelit-entiteta (ujedno i primarni ključ njegovog link-entiteta L_DMATTR_LEVEL).
	DMAAttrLevelDTS	Dio primarnog ključa satelit-entiteta – datum punjenja zapisa o vezi između atributa tržišta podataka i razine u meta-bazu.
	DMAAttrLevelEDTS	Datum završetka punjenja zapisa o vezi između atributa tržišta podataka i razine u meta-bazu.
	DMAAttrLevelSRC	Izvor iz kojeg je zapis o vezi između atributa tržišta podataka i razine došao u meta-bazu.
	LabelAttribute	Naziv razine koji će se prikazivati.
	OrderAttribute	Naziv atributa tržišta podataka koji se koristi za sortiranje.
L_DMATTR_FACT		
	SQL_DMAttrFact	Primarni umjetni ključ (sekvenca) link-entiteta.

	SQN_DMAttr	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_DM_ATTRIBUTE).
	SQN_Fact	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_FACT).
	SQN_Object	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_OBJECT).
	DMAAttrFactLDTS	Datum punjenja zapisa o vezi između atributa tržišta podataka i činjenice u meta-bazu.
	DMAAttrFactRSRC	Izvor iz kojeg je zapis o vezi između atributa tržišta podataka i činjenice došao u meta-bazu.
L_DMATTR_DIMENSION		
	SQN_DMAttrDim	Primarni umjetni ključ (sekvenca) link-entiteta.
	SQN_DMAttr	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_DM_ATTRIBUTE).
	SQN_Dimension	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_DIMENSION).
	SQN_Object	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_OBJECT).
	DMAAttrDimLDTS	Datum punjenja zapisa o vezi između atributa tržišta podataka i dimenzije u meta-bazu.
	DMAAttrDimRSRC	Izvor iz kojeg je zapis o vezi između atributa tržišta podataka i dimenzije došao u meta-bazu.
L_CUBE		

	SQL_Cube	Primarni umjetni ključ (sekvenca) link-entiteta.
	SQL_DMAAttr	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_DM_ATTRIBUTE).
	SQL_Dimension	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_DIMENSION).
	SQL_Fact	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_FACT).
	SQL_Object	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_OBJECT).
	SQL_Hierarchy	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_HIERARCHY).
	CubeLDTS	Datum punjenja zapisa o podatkovnoj kocki - veze između atributa tržišta podataka, činjenice, dimenzije i hijerarhije.
	CubeRSRC	Izvor iz kojeg je zapis o podatkovnoj kocki - vezi između atributa tržišta podataka, činjenice, dimenzije i hijerarhije došao u meta-bazu.
H_MV		
	SQL_MV	Primarni umjetni ključ (sekvenca) hub-entiteta.
	SQL_Object	Primarni umjetni ključ (sekvenca) hub-entiteta.
	MVID	Poslovni ključ hub-entiteta.
	MVLDTS	Datum punjenja zapisa o materijaliziranom pogledu u meta-bazu.

	MVRSRC	Izvor iz kojeg je zapis o materijaliziranom pogledu došao u meta-bazu.
S_MV_DEF		
	SQL_MV	Dio primarnog ključa satelit-entiteta (ujedno i primarni ključ njegovog hub-entiteta H_MV).
	SQL_Object	Dio primarnog ključa satelit-entiteta (ujedno i primarni ključ njegovog hub-entiteta H_OBJECT).
	MVdefLDTS	Dio primarnog ključa satelit-entiteta – datum punjenja zapisa o materijaliziranom pogledu u meta-bazu.
	MVdefLEDTS	Datum završetka punjenja zapisa o materijaliziranom pogledu u meta-bazu.
	MVdefRSRC	Izvor iz kojeg je zapis o materijaliziranom pogledu došao u meta-bazu.
	MVName	Naziv materijaliziranog pogleda.
	MVDescription	Tekstualni opis materijaliziranog pogleda.
	MVType	Tip materijaliziranog pogleda (može biti osvježen ili star).
	MVRefreshType	Definira kako i kada će se materijalizirani pogled osvježavati.
	MVDimensionFlag	Oznaka predstavlja li materijalizirani pogled dimenziju.
	MVFactFlag	Oznaka predstavlja li materijalizirani pogled činjenicu.
	MVCubeFlag	Oznaka predstavlja li materijalizirani pogled podatkovnu kocku (zvjezdanu shemu).

S_MV_QUERY		
	SQN_MV	Dio primarnog ključa satelit-entiteta (ujedno i primarni ključ njegovog hub-entiteta H_MV).
	SQN_Object	Dio primarnog ključa satelit-entiteta (ujedno i primarni ključ njegovog hub-entiteta H_OBJECT).
	QueryLDTS	Dio primarnog ključa satelit-entiteta – datum punjenja zapisa o hijerarhiji u meta-bazu.
	QueryLEDTS	Datum završetka punjenja zapisa o razini u meta-bazu.
	QueryRSRC	Izvor iz kojeg je zapis o razini došao u meta-bazu.
	Query	Naziv materijaliziranog pogleda.
L_CUBE_MV		
	SQN_CubeMV	Primarni umjetni ključ (sekvenca) link-entiteta.
	SQN_DMAttr	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_DM_ATTRIBUTE).
	SQN_Dimension	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_DIMENSION).
	SQN_Fact	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_FACT).
	SQN_Object	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_OBJECT).
	SQN_Hierarchy	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_HIERARCHY).

	SQL_Level	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_LEVEL).
	SQL_MV	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_MV).
	CubeLDS	Datum punjenja zapisa o vezi materijaliziranog pogleda i podatkovne kocke - veze između atributa tržišta podataka, činjenice, dimenzije, hijerarhije, razine i materijaliziranog pogleda.
	CubeRSRC	Izvor iz kojeg je zapis o vezi materijaliziranog pogleda i podatkovne kocke - veze između atributa tržišta podataka, činjenice, dimenzije, hijerarhije, razine i materijaliziranog pogleda.



Sigurnost podataka:

NAZIV ENTITETA	DEFINICIJA ENTITETA
H_ACCESS_RIGHT	Poslovni ključevi za prava pristupa – hub entitet.
S_ACCESS_RIGHT	Opisni podaci za prava pristupa – satelit entitet.
L_OBJECT-RIGHT	Veze između objekta baze podataka (može biti činjenica, dimenzija ili materijalizirani pogled) i prava pristupa – link entitet.
H_ROLE	Poslovni ključevi za korisničku ulogu – hub entitet.
S_ROLE	Opisni podaci za korisničku ulogu – satelit entitet.
L_ROLE_RIGHT	Pridruživanje prava pristupa korisničkoj ulozi – link entitet.
H_USER	Poslovni ključevi za korisnika – hub entitet.
S_USER	Opisni podaci za korisnika – satelit entitet.
L_USER_ROLE	Pridruživanje korisnika i korisničke uloge – link entitet.
H_AUTHENTICATION	Poslovni ključevi za način autentifikacije – hub entitet.
S_AUTHENTICATION	Opisni podaci za način autentifikacije – satelit entitet.
L_USER_AUTHNT	Pridruživanje korisnika i načina korisničke autentifikacije – link entitet.
L_AUTHNT_RESOURCE	Pridruživanje resursa i načina korisničke autentifikacije – link entitet.
H_POLICY	Poslovni ključevi za politike i standarde sigurnosti – hub entitet.
S_POLICY	Opisni podaci za politike i standarde sigurnosti – satelit entitet.
L_APPLICABLE	Pridruživanje politike sigurnosti i načina korisničke autentifikacije – link entitet.
S_APPLICABLE	Opisni podaci za način primjenjivost načina autentifikacije u okviru politike sigurnosti – satelit entitet.

NAZIV ENTITETA	NAZIV ATRIBUTA	DEFINICIJA ATRIBUTA
H_ACCESS_RIGHT		
	SQL_Right	Primarni umjetni ključ (sekvenca) hub-entiteta.
	RightID	Poslovni ključ hub-entiteta.

	H_RightLDTS	Datum punjenja zapisa o pravu pristupa u meta-bazu.
	H_RightRSRC	Izvor iz kojeg je zapis o pravu pristupa došao u meta-bazu.
S_ACCESS_RIGHT		
	SQL_Right	Dio primarnog ključa satelit-entiteta (ujedno i primarni ključ njegovog hub-entiteta H_ACCESS_RIGHT).
	S_RightLDTS	Dio primarnog ključa satelit-entiteta – datum punjenja zapisa o pravu pristupa u meta-bazu.
	S_RightLEDTS	Datum završetka punjenja zapisa o pravu pristupa u meta-bazu.
	S_RightRSRC	Izvor iz kojeg je zapis o pravu pristupa došao u meta-bazu.
	PermissionName	Naziv prava pristupa.
	PermissionDescription	Tekstualni opis prava pristupa.
L_OBJECT-RIGHT		
	SQL_ObjRight	Primarni umjetni ključ (sekvenca) link-entiteta.
	SQL_Right	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_RIGHT).
	SQL_Object	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_OBJECT).
	L_ObjRightLDTS	Datum punjenja zapisa o vezi između prava pristupa i objekta (dimenzije, činjenice ili materijaliziranog pogleda) u meta-bazu.
	L_ObjRightRSRC	Izvor iz kojeg je zapis o vezi između prava pristupa i objekta (dimenzije,

		činjenice ili materijaliziranog pogleda) došao u meta-bazu.
H_ROLE		
	SQL_Role	Primarni umjetni ključ (sekvenca) hub-entiteta.
	RoleID	Poslovni ključ hub-entiteta.
	H_RoleLDTS	Datum punjenja zapisa o korisničkoj ulozi u meta-bazu.
	H_RoleRSRC	Izvor iz kojeg je zapis o korisničkoj ulozi došao u meta-bazu.
S_ROLE		
	SQL_Role	Dio primarnog ključa satelit-entiteta (ujedno i primarni ključ njegovog hub-entiteta H_ROLE).
	S_RoleLDTS	Dio primarnog ključa satelit-entiteta – datum punjenja zapisa o korisničkoj ulozi u meta-bazu.
	S_RoleLEDTS	Datum završetka punjenja zapisa o korisničkoj ulozi u meta-bazu.
	S_RoleRSRC	Izvor iz kojeg je zapis o korisničkoj ulozi došao u meta-bazu.
	RoleName	Naziv korisničke uloge.
	RoleDescription	Tekstualni opis korisničke uloge.
	RoleType	Tip korisničke uloge (niska, srednja ili visoka razina pristupa).
L_ROLE_RIGHT		
	SQL_RoleRight	Primarni umjetni ključ (sekvenca) link-entiteta.
	SQL_Right	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_RIGHT).
	SQL_Role	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_ROLE).

	L_RoleRightLDTS	Datum punjenja zapisa o vezi između prava pristupa i korisničke uloge u meta-bazu.
	L_RoleRightRSRC	Izvor iz kojeg je zapis o vezi između prava pristupa i korisničke uloge došao u meta-bazu.
H_USER		
	SQN_User	Primarni umjetni ključ (sekvenca) hub-entiteta.
	UserID	Poslovni ključ hub-entiteta.
	H_User LDTS	Datum punjenja zapisa o korisniku u meta-bazu.
	H_UserRSRC	Izvor iz kojeg je zapis o korisniku došao u meta-bazu.
S_USER		
	SQN_User	Dio primarnog ključa satelit-entiteta (ujedno i primarni ključ njegovog hub-entiteta H_USER).
	S_UserLDTS	Dio primarnog ključa satelit-entiteta – datum punjenja zapisa o korisniku u meta-bazu.
	S_UserLEDTS	Datum završetka punjenja zapisa o korisniku u meta-bazu.
	S_UserRSRC	Izvor iz kojeg je zapis o korisniku došao u meta-bazu.
	Username	Korisničko ime korisnika za login.
	UserFirstName	Ime korisnika.
	UserLastName	Prezime korisnika.
	UserDescription	Tekstualni opis korisnika.
	UserType	Tip korisnika (radno mjesto unutar organizacije).
	Login	Podaci za login korisnika.
L_USER_ROLE		

	SQLN_UserRole	Primarni umjetni ključ (sekvenca) link-entiteta.
	SQLN_User	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_USER).
	SQLN_Role	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_ROLE).
	L_UserRoleLDTS	Datum punjenja zapisa o vezi između korisničke uloge i korisnika u meta-bazu.
	L_UserRoleRSRC	Izvor iz kojeg je zapis o vezi između korisničke uloge i korisnika došao u meta-bazu.
H_AUTHENTICATION		
	SQLN_Authentication	Primarni umjetni ključ (sekvenca) hub-entiteta.
	AuthenticationID	Poslovni ključ hub-entiteta.
	H_AuthenticationLDTS	Datum punjenja zapisa o načinu autentifikacije u meta-bazu.
	H_AuthenticationRSRC	Izvor iz kojeg je zapis o načinu autentifikacije došao u meta-bazu.
S_AUTHENTICATION		
	SQLN_Authentication	Dio primarnog ključa satelit-entiteta (ujedno i primarni ključ njegovog hub-entiteta H_AUTHENTICATION).
	S_AuthenticationLDTS	Dio primarnog ključa satelit-entiteta – datum punjenja zapisa o načinu autentifikacije u meta-bazu.
	S_AuthenticationLEDTS	Datum završetka punjenja zapisa o načinu autentifikacije u meta-bazu.
	S_AuthenticationRSRC	Izvor iz kojeg je zapis o načinu autentifikacije došao u meta-bazu.
	AuthenticationName	Naziv načina autentifikacije.

	AuthenticationDescription	Tekstualni opis načina autentifikacije.
	AuthenticationFactor	Faktor načina autentifikacije (kategorije autentifikacije – mogu biti faktor znanja, faktor posjedovanja i faktor postojanja).
	AuthenticationTechnology	Tehnologija načina autentifikacije (sigurnosni tokeni, mobilna autentifikacija, biometrijska autentifikacija).
	AuthenticationType	Tip načina autentifikacije (korisničko ime, lozinka, PIN, odgovor na tajno pitanje; token, zaposlenička ID kartica, SIM kartica, ključ; otisak prsta, sken šarenice oka, glasovno prepoznavanje, prepoznavanje lica, geometrija ruke).
L_USER_AUTHNT		
	SQN_UserAuthnt	Primarni umjetni ključ (sekvenca) link-entiteta.
	SQN_User	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_USER).
	SQN_Authentication	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_AUTHENTICATION).
	L_UserAuthntLDTS	Datum punjenja zapisa o vezi između korisnika i načina autentifikacije u meta-bazu.
	L_UserAuthntRSRC	Izvor iz kojeg je zapis o vezi između korisnika i načina autentifikacije došao u meta-bazu.
L_AUTHNT_RESOURCE		

	SQL_AuthntResource	Primarni umjetni ključ (sekvenca) link-entiteta.
	SQL_Resource	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_RESOURCE).
	SQL_Authentication	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_AUTHENTICATION).
	L_AuthntResourceLDTS	Datum punjenja zapisa o vezi između resursa i načina autentifikacije u meta-bazu.
	L_AuthntResourceRSRC	Izvor iz kojeg je zapis o vezi između resursa i načina autentifikacije došao u meta-bazu.
H_POLICY		
	SQL_Policy	Primarni umjetni ključ (sekvenca) hub-entiteta.
	PolicyID	Poslovni ključ hub-entiteta.
	H_Policy LDTS	Datum punjenja zapisa o politici sigurnosti u meta-bazu.
	H_PolicyRSRC	Izvor iz kojeg je zapis o politici sigurnosti došao u meta-bazu.
S_POLICY		
	SQL_Policy	Dio primarnog ključa satelit-entiteta (ujedno i primarni ključ njegovog hub-entiteta H_POLICY).
	S_PolicyLDTS	Dio primarnog ključa satelit-entiteta – datum punjenja zapisa o politici sigurnosti u meta-bazu.
	S_PolicyLEDTS	Datum završetka punjenja zapisa o politici sigurnosti u meta-bazu.
	S_PolicyRSRC	Izvor iz kojeg je zapis o politici sigurnosti došao u meta-bazu.
	PolicyName	Naziv politike sigurnosti.

	PolicyDescription	Tekstualni opis politike sigurnosti.
	PolicyDocumentName	Naziv dokumenta politike sigurnosti.
	PolicyDocumentFolder	Naziv mape u kojoj se nalazi dokument politike sigurnosti.
L_APPLICABLE		
	SQL_AuthntPolicy	Primarni umjetni ključ (sekvenca) link-entiteta.
	SQL_Authentication	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_AUTHENTICATION).
	SQL_Policy	Vanjski ključ link-entiteta (ujedno i primarni ključ vezanog hub-entiteta H_POLICY).
	L_AuthntPolicyLDTS	Datum punjenja zapisa o vezi između politike sigurnosti i načina autentifikacije u meta-bazu.
	L_AuthntPolicyRSRC	Izvor iz kojeg je zapis o vezi između politike sigurnosti i načina autentifikacije došao u meta-bazu.
S_APPLICABLE		
	SQL_AuthntPolicy	Dio primarnog ključa satelit-entiteta (ujedno i primarni ključ njegovog link-entiteta L_APPLICABLE).
	S_ApplicableLDTS	Dio primarnog ključa satelit-entiteta – datum punjenja zapisa o primjenjivosti načina autentifikacije unutar politike sigurnosti u meta-bazu.
	S_ApplicableLEDTS	Datum završetka punjenja zapisa o primjenjivosti načina autentifikacije unutar politike sigurnosti u meta-bazu.



	S_ApplicableRSRC	Izvor iz kojeg je zapis o primjenjivosti načina autentifikacije unutar politike sigurnosti došao u meta-bazu.
	ApplicableFlag	Oznaka primjenjivosti načina autentifikacije unutar politike sigurnosti (primjenjivo ili nije primjenjivo).

## Privitak 4. SQL DDL Skripte za izradu prototipa

### JobDB

```
-- Kreiranje tablica u bazi podataka JobDB

CREATE
TABLE EMPLOYEE
(
  EmployeeID      NUMERIC (5) NOT NULL ,
  Name            NVARCHAR (30) NOT NULL ,
  Surname         NVARCHAR (30) NOT NULL ,
  Title           NVARCHAR (10) ,
  Gender          NVARCHAR (10) NOT NULL ,
  BirthDate       DATE NOT NULL ,
  StreetNumber    NVARCHAR (5) NOT NULL ,
  StreetName      NVARCHAR (40) NOT NULL ,
  ZIP             NUMERIC (20) NOT NULL ,
  City            NVARCHAR (50) ,
  County          NVARCHAR (50) ,
  Country         NVARCHAR (50) ,
  EmailAddress    NVARCHAR (35) ,
  PhoneNumber     NVARCHAR (20) ,
  ShortBiography  NVARCHAR (1500) ,
  SUPERVISOR_SupervisorID  NUMERIC (5) ,
  MARITAL_STATUS_MaritalStatusID NUMERIC (2) NOT NULL ,
  PROJECT_ProjectID    NUMERIC (5) NOT NULL
)
ON "default"
GO
ALTER TABLE EMPLOYEE ADD CONSTRAINT EMPLOYEE_PK PRIMARY KEY CLUSTERED (
EmployeeID)
WITH
(
  ALLOW_PAGE_LOCKS = ON ,
  ALLOW_ROW_LOCKS = ON
)
ON "default"
GO
```

```

CREATE
TABLE MARITAL_STATUS
(
    MaritalStatusID NUMERIC (2) NOT NULL ,
    MaritalStatus NVARCHAR (20) NOT NULL
)
ON "default"
GO
ALTER TABLE MARITAL_STATUS ADD CONSTRAINT MARITAL_STATUS_PK PRIMARY KEY
CLUSTERED (MaritalStatusID)
WITH
(
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE PROJECT
(
    ProjectID NUMERIC (5) NOT NULL ,
    ProjectName NVARCHAR (50) NOT NULL ,
    ProjectGoalShort NVARCHAR (500) NOT NULL ,
    ProjectDescription NVARCHAR (2000) ,
    ProjectStartDate DATE NOT NULL ,
    ProjectEndDate DATE ,
    PlannedEndDate DATE NOT NULL ,
    BudgetWorkHours NUMERIC (7) NOT NULL ,
    BudgetSalaryBonus NUMERIC (9,2) NOT NULL
)
ON "default"
GO
ALTER TABLE PROJECT ADD CONSTRAINT PROJECT_PK PRIMARY KEY CLUSTERED (ProjectID)
WITH
(
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON
)

```

```

ON "default"
GO

CREATE
TABLE SUPERVISOR
(
SupervisorID NUMERIC (5) NOT NULL ,
Name NVARCHAR (30) NOT NULL ,
Surname NVARCHAR (30) NOT NULL ,
EmailAddress NVARCHAR (35) ,
PhoneNumber NVARCHAR (20) ,
ShortBiography NVARCHAR (1000)
)
ON "default"
GO
ALTER TABLE SUPERVISOR ADD CONSTRAINT SUPERVISOR_PK PRIMARY KEY CLUSTERED (
SupervisorID)
WITH
(
ALLOW_PAGE_LOCKS = ON ,
ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

ALTER TABLE EMPLOYEE
ADD CONSTRAINT EMPLOYEE_MARITAL_STATUS_FK FOREIGN KEY
(MARITAL_STATUS_MaritalStatusID) REFERENCES MARITAL_STATUS (
MaritalStatusID) ON DELETE NO ACTION ON UPDATE NO ACTION
GO

ALTER TABLE EMPLOYEE
ADD CONSTRAINT EMPLOYEE_PROJECT_FK FOREIGN KEY(
PROJECT_ProjectID) REFERENCES PROJECT (ProjectID) ON
DELETE NO ACTION ON UPDATE NO ACTION
GO

ALTER TABLE EMPLOYEE
ADD CONSTRAINT EMPLOYEE_SUPERVISOR_FK FOREIGN KEY(SUPERVISOR_SupervisorID)
REFERENCES SUPERVISOR(SupervisorID) ON DELETE NO ACTION ON UPDATE NO ACTION

```

GO

-- Punjenje tablica u bazi podataka JobDB izvršeno je preko FreeDataGenerator.com (u radu je priložen primjer zapisa po svakoj tablici)

```
INSERT INTO PROJECT VALUES (1020, 'Taciti parturient diam elementum sodales.', 'Lorem ipsum massa erat lectus iaculis non luctus pharetra ridiculus non dis.\nTincidunt ipsum id magnis gravida pharetra at metus, vehicula et.\nSodales mauris lacus congue, eros porta neque dictum.\nSemper lectus nascetur porta ante tellus a senectus metus sociis tortor.\n', 'Lorem ipsum maecenas curabitur lobortis porttitor montes praesent placerat parturient cum. Lorem ipsum adipiscing mi, per sagittis quis pharetra non augue class. Lorem ipsum fermentum primis fames non iaculis elementum metus semper feugiat. Lorem ipsum ac facilisi donec laoreet orci dui vivamus ultrices eros. \nLeo semper dui feugiat fermentum nostra ipsum placerat dapibus. Iaculis nam vehicula aliquet nibh vitae, nisl urna tellus. Ultrices ultricies justo integer facilisi platea id tempus sociis. Posuere eleifend donec condimentum faucibus bibendum litora ante faucibus. Malesuada cum bibendum taciti donec feugiat elit tristique purus. Class arcu himenaeos malesuada ac auctor varius sagittis commodo. \nA justo id elit himenaeos felis bibendum viverra, lacinia dui elementum. Aenean sagittis proin pulvinar, fames felis eleifend vulputate himenaeos consectetur adipiscing. Nam luctus fames iaculis etiam iaculis fringilla proin dui lectus mollis! Fermentum suscipit habitasse convallis vulputate lacinia posuere leo lacinia faucibus quam. Tempor, mauris quis sodales at odio mattis condimentum class dui tempor. Turpis nostra pharetra gravida praesent augue non malesuada elementum augue senectus. \n', convert(datetime, '2000-03-13'), null, convert(datetime, '2016-11-24'), 281300, 68976.57);
INSERT INTO SUPERVISOR VALUES (100, 'Easton', 'Townsend', 'Easto.TOWNSE8995@hushmail.com', '(240) 961-9089', 'Lorem ipsum id ut auctor commodo amet facilisi sollicitudin lobortis rutrum nisi mus. Lorem ipsum nullam habitasse auctor adipiscing magna amet dui leo justo enim ut. Lorem ipsum ad dis elit ut habitant a maecenas pharetra lorem quis ornare. \nLeo ante, morbi dui litora congue euismod class cursus. Aliquam nascetur fames at lectus aenean enim lacinia ultricies. Iaculis curae; dignissim himenaeos mauris viverra mattis mauris nibh. Felis laoreet vel magna vulputate dolor sed in tristique. Justo dui eu imperdiet ante pulvinar in primis est. Sollicitudin aliquet, tincidunt sem condimentum donec litora molestie malesuada. \n');
INSERT INTO EMPLOYEE VALUES (100, 'Lexie', 'Stokes', 'Mr.', 'm', convert(datetime, '1987-08-21'), 'h8', 'Clarence', '68879', 'Omaha', 'Nebraska', 'United States', 'Lex.STOKE1246@yahoo.com', '(229) 106-1446', 'Lorem ipsum varius fringilla etiam vel porta nostra auctor parturient senectus? Lorem ipsum lobortis ipsum condimentum donec aptent curabitur ullamcorper senectus cubilia! Lorem ipsum commodo metus enim sociis commodo nec at cubilia tristique? \nRisus parturient dictumst habitant nullam amet magna aliquam condimentum montes habitasse penatibus. Molestie lectus cubilia vestibulum nunc aenean mauris sapien congue, nam mollis phasellus. Nec dictumst malesuada condimentum elit platea ligula in porta mi volutpat dapibus. Faucibus interdum auctor cursus dis posuere suspendisse mollis enim, sociosqu aliquam malesuada. \n', 123, 1, 1055);
INSERT INTO MARITAL_STATUS VALUES (1, 'Single');
```

## TrainingDB

```
-- Kreiranje tablica u bazi podataka TrainingDB

CREATE
TABLE CERTIFICATE_TRAINING
(
  TrainingID      NUMERIC (5) NOT NULL ,
  TrainingName    NVARCHAR (100) NOT NULL ,
  TrainingDescription NVARCHAR (1500) ,
  TrainingLevel   NVARCHAR (30) ,
  TrainingDays    NUMERIC (5) ,
  TrainingHours   NUMERIC (5) ,
  TrainingCertificateName NVARCHAR (100) ,
  TrainingCostPerEmployee NUMERIC (9,2) NOT NULL
)
ON "default"
GO
ALTER TABLE CERTIFICATE_TRAINING ADD CONSTRAINT CERTIFICATE_TRAINING_PK PRIMARY
KEY CLUSTERED (TrainingID)
WITH
(
  ALLOW_PAGE_LOCKS = ON ,
  ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE COMPETENCE
(
  CompetenceID      NUMERIC (5) NOT NULL ,
  CompetenceName    NVARCHAR (500) NOT NULL ,
  CERTIFICATE_TRAINING_TrainingID NUMERIC (5) NOT NULL
)
ON "default"
GO
ALTER TABLE COMPETENCE ADD CONSTRAINT COMPETENCE_PK PRIMARY KEY CLUSTERED (
```

```

CompetenceID, CERTIFICATE_TRAINING_TrainingID)
WITH
(
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE EMPLOYEE
(
    EmployeeID NUMERIC (5) NOT NULL ,
    Name NVARCHAR (30) NOT NULL ,
    Surname NVARCHAR (30) NOT NULL ,
    Title NVARCHAR (10) ,
    Gender NVARCHAR (10) NOT NULL ,
    MaritalStatus NVARCHAR (20) NOT NULL ,
    BirthDate DATE NOT NULL ,
    StreetNumber NVARCHAR (5) NOT NULL ,
    StreetName NVARCHAR (40) NOT NULL ,
    ZIP NUMERIC (20) NOT NULL ,
    City NVARCHAR (50) ,
    County NVARCHAR (50) ,
    Country NVARCHAR (50) ,
    EmailAddress NVARCHAR (35) ,
    PhoneNumber NVARCHAR (20)
)
ON "default"
GO
ALTER TABLE EMPLOYEE ADD CONSTRAINT EMPLOYEE_PK PRIMARY KEY CLUSTERED (
EmployeeID)
WITH
(
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

```

```

CREATE
TABLE PARTICIPATES
(
  TrainingStartDate DATE NOT NULL ,
  TrainingEndDate DATE ,
  Passed BIT ,
  Grade NUMERIC (6,2) ,
  FirstTime BIT ,
  EMPLOYEE_EmployeeID NUMERIC (5) NOT NULL ,
  CERTIFICATE_TRAINING_TrainingID NUMERIC (5) NOT NULL
)
ON "default"
GO
ALTER TABLE PARTICIPATES ADD CONSTRAINT PARTICIPATES_PK PRIMARY KEY CLUSTERED (
EMPLOYEE_EmployeeID, CERTIFICATE_TRAINING_TrainingID)
WITH
(
  ALLOW_PAGE_LOCKS = ON ,
  ALLOW_ROW_LOCKS = ON
)
ON "default"
GO
ALTER TABLE COMPETENCE
ADD CONSTRAINT COMPETENCE_CERTIFICATE_TRAINING_FK FOREIGN KEY (CERTIFICATE_TRAINING_TrainingID
) REFERENCES CERTIFICATE_TRAINING (TrainingID) ON DELETE NO ACTION ON UPDATE NO ACTION
GO
ALTER TABLE PARTICIPATES
ADD CONSTRAINT PARTICIPATES_CERTIFICATE_TRAINING_FK FOREIGN KEY (CERTIFICATE_TRAINING_TrainingID
)REFERENCES CERTIFICATE_TRAINING (TrainingID) ON DELETE NO ACTION ON UPDATE NO ACTION
GO
ALTER TABLE PARTICIPATES
ADD CONSTRAINT PARTICIPATES_EMPLOYEE_FK FOREIGN KEY (EMPLOYEE_EmployeeID) REFERENCES EMPLOYEE
(EmployeeID) ON DELETE NO ACTION ON UPDATE NO ACTION
GO
-- Punjenje tablica u bazi podataka TrainingDB (primjer jednog retka po tablici s FreeDataGenerator.com)

```



```

INSERT INTO TrainingDB.dbo.EMPLOYEE SELECT employeeid, name, surname, title, gender, maritalstatus,
birthdate, streetnumber, streetname, zip, city, county, country, emailaddress, phonenumber
FROM JobDB.dbo.employee INNER JOIN JobDB.dbo.marital_status on
JobDB.dbo.employee.marital_status_maritalstatusid=JobDB.dbo.marital_status.maritalstatusid AND
JobDB.dbo.EMPLOYEE.employeeid BETWEEN 170 AND 199;

INSERT INTO EMPLOYEE VALUES (200, 'Zachary', 'Lowery', 'Mrs.', 'm', 'Married', '19751125', '54a', 'State', '92624',
'Irvine', 'California', 'United States', 'Zachary.LOWERY1894@live.com', '(919) 587-8901');
INSERT INTO CERTIFICATE_TRAINING VALUES (10, 'Lorem ipsum donec placerat justo natoque nisi fames erat
suscipit in.\n', 'Lorem ipsum dictumst dis posuere vel litora conubia ac vulputate aliquet. Lorem ipsum dui nulla
dolor egestas mauris nulla porttitor aenean faucibus. Lorem ipsum magna aliquet nunc tempus rhoncus interdum
facilisis class molestie. Lorem ipsum iaculis viverra, commodo habitasse augue vel interdum gravida morbi! \n', '2',
74, 510, 'Lorem ipsum sodales eu massa convallis viverra vehicula sociosqu vehicula aliquet volutpat leo.\n',
8310.31);
INSERT INTO COMPETENCE VALUES (1, 'Lorem ipsum diam ultrices integer feugiat malesuada elementum
senectus ligula dolor bibendum?\n', 10);
INSERT INTO PARTICIPATES VALUES ('20130421', null, 1, 7.33, 0, 181, 25);

```

## SdvDB

```

-- Kreiranje tablica u bazi podataka SdvDB

CREATE
TABLE H_CERTIFICATE_TRAINING
(
SQN_Training NUMERIC (5) NOT NULL ,
TrainingID NUMERIC (5) NOT NULL ,
TrainingLDTS DATETIME ,
TrainingRSRC NVARCHAR (100)
)
ON "default"
GO
ALTER TABLE H_CERTIFICATE_TRAINING ADD CONSTRAINT H_CERTIFICATE_TRAINING_PK
PRIMARY KEY CLUSTERED (SQN_Training)
WITH
(
ALLOW_PAGE_LOCKS = ON ,
ALLOW_ROW_LOCKS = ON

```

```

)
ON "default"
GO

CREATE
TABLE H_EMPLOYEE
(
    SQN_Employee NUMERIC (5) NOT NULL ,
    EmployeeID NUMERIC (5) NOT NULL ,
    EmployeeLDTS DATETIME ,
    EmployeeRSRC NVARCHAR (100)
)
ON "default"
GO
ALTER TABLE H_EMPLOYEE ADD CONSTRAINT H_EMPLOYEE_PK PRIMARY KEY CLUSTERED (
SQN_Employee)
WITH
(
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE H_EMPLOYEE_MASTER
(
    SQN_EmployeeMaster NUMERIC (5) NOT NULL ,
    EmployeeID NUMERIC (5) NOT NULL ,
    EmployeeLDTS DATETIME ,
    EmployeeRSRC NVARCHAR (100)
)
ON "default"
GO
ALTER TABLE H_EMPLOYEE_MASTER ADD CONSTRAINT H_EMPLOYEE_MASTER_PK PRIMARY KEY
CLUSTERED (SQN_EmployeeMaster)
WITH
(
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON
)

```

```

)
ON "default"
GO

CREATE
TABLE H_MARITAL_STATUS
(
SQN_MaritalStatus NUMERIC (5) NOT NULL ,
MaritalStatusID NUMERIC (2) NOT NULL ,
MaritalStatusLDTS DATETIME ,
MaritalStatusRSRC NVARCHAR (100)
)
ON "default"
GO
ALTER TABLE H_MARITAL_STATUS ADD CONSTRAINT H_MARITAL_STATUS_PK PRIMARY KEY
CLUSTERED (SQN_MaritalStatus)
WITH
(
ALLOW_PAGE_LOCKS = ON ,
ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE H_PROJECT
(
SQN_Project NUMERIC (5) NOT NULL ,
ProjectID NUMERIC (5) NOT NULL ,
ProjectLDTS DATETIME ,
ProjectRSRC NVARCHAR (100)
)
ON "default"
GO
ALTER TABLE H_PROJECT ADD CONSTRAINT H_PROJECT_PK PRIMARY KEY CLUSTERED (
SQN_Project)
WITH
(
ALLOW_PAGE_LOCKS = ON ,
ALLOW_ROW_LOCKS = ON

```

```

)
ON "default"
GO

CREATE
TABLE H_SUPERVISOR
(
SQN_Supervisor NUMERIC (5) NOT NULL ,
SupervisorID NUMERIC (5) NOT NULL ,
SupervisorLDTS DATETIME ,
SupervisorRSRC NVARCHAR (100)
)
ON "default"
GO
ALTER TABLE H_SUPERVISOR ADD CONSTRAINT H_SUPERVISOR_PK PRIMARY KEY CLUSTERED (
SQN_Supervisor)
WITH
(
ALLOW_PAGE_LOCKS = ON ,
ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE L_EMPLOYEE_MASTER
(
SQN_EmpMaster NUMERIC (5) NOT NULL ,
EmployeeMasterLDTS DATETIME ,
EmployeeMasterRSRC NVARCHAR (100) ,
H_EMPLOYEE_MASTER_SQN_EmployeeMaster NUMERIC (5) NOT NULL ,
H_EMPLOYEE_SQN_Employee NUMERIC (5) NOT NULL
)
ON "default"
GO
ALTER TABLE L_EMPLOYEE_MASTER ADD CONSTRAINT L_EMPLOYEE_MASTER_PK PRIMARY KEY
CLUSTERED (SQN_EmpMaster)
WITH
(
ALLOW_PAGE_LOCKS = ON ,

```

```

ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE L_EMP_MARITAL
(
SQN_EmpMarital          NUMERIC (5) NOT NULL ,
EmpMaritalLDTS         DATETIME ,
EmpMaritalRSRC         NVARCHAR (100) ,
H_MARITAL_STATUS_SQN_MaritalStatus NUMERIC (5) NOT NULL ,
H_EMPLOYEE_SQN_Employee   NUMERIC (5) NOT NULL
)
ON "default"
GO
ALTER TABLE L_EMP_MARITAL ADD CONSTRAINT L_EMP_MARITAL_PK PRIMARY KEY CLUSTERED
(SQN_EmpMarital)
WITH
(
ALLOW_PAGE_LOCKS = ON ,
ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE L_PARTICIPATES
(
SQN_Participates        NUMERIC (5) NOT NULL ,
ParticipatesLDTS       DATETIME ,
ParticipatesRSRC       NVARCHAR (100) ,
H_EMPLOYEE_SQN_Employee   NUMERIC (5) NOT NULL ,
H_CERTIFICATE_TRAINING_SQN_Training NUMERIC (5) NOT NULL
)
ON "default"
GO
ALTER TABLE L_PARTICIPATES ADD CONSTRAINT L_PARTICIPATES_PK PRIMARY KEY
CLUSTERED (SQN_Participates)
WITH

```

```

(
  ALLOW_PAGE_LOCKS = ON ,
  ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE L_SUPERVISES
(
  SQN_Supervises      NUMERIC (5) NOT NULL ,
  SupervisesLDTS      DATETIME ,
  SupervisesRSRC      NVARCHAR (100) ,
  H_SUPERVISOR_SQN_Supervisor NUMERIC (5) NOT NULL ,
  H_EMPLOYEE_SQN_Employee  NUMERIC (5) NOT NULL
)
ON "default"
GO

ALTER TABLE L_SUPERVISES ADD CONSTRAINT L_SUPERVISES_PK PRIMARY KEY CLUSTERED (
SQN_Supervises)
WITH
(
  ALLOW_PAGE_LOCKS = ON ,
  ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE L_WORKS_ON
(
  SQN_WorksOn      NUMERIC (5) NOT NULL ,
  WorksOnLDTS      DATETIME ,
  WorksOnRSRC      NVARCHAR (100) ,
  H_EMPLOYEE_SQN_Employee NUMERIC (5) NOT NULL ,
  H_PROJECT_SQN_Project  NUMERIC (5) NOT NULL
)
ON "default"
GO

ALTER TABLE L_WORKS_ON ADD CONSTRAINT L_WORKS_ON_PK PRIMARY KEY CLUSTERED (

```

```

SQN_WorksOn)
WITH
(
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE S_CERTIFICATE_TRAINING
(
    TrainingLDTS          DATETIME NOT NULL ,
    TrainingName          NVARCHAR (100) NOT NULL ,
    TrainingDescription    NVARCHAR (1500) ,
    TrainingLevel         NVARCHAR (30) ,
    TrainingDays          NUMERIC (5) ,
    TrainingHours         NUMERIC (5) ,
    TrainingCertificateName NVARCHAR (100) ,
    TrainingCostPerEmployee NUMERIC (9,2) NOT NULL ,
    TrainingLEDTS         DATETIME ,
    TrainingRSRC          NVARCHAR (100) ,
    H_CERTIFICATE_TRAINING_SQN_Training NUMERIC (5) NOT NULL
)
ON "default"
GO
ALTER TABLE S_CERTIFICATE_TRAINING ADD CONSTRAINT S_CERTIFICATE_TRAINING_PK
PRIMARY KEY CLUSTERED (TrainingLDTS, H_CERTIFICATE_TRAINING_SQN_Training)
WITH
(
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE S_COMPETENCE
(
    CompetenceLDTS        DATETIME NOT NULL ,

```

```

CompetenceID          NUMERIC (5) NOT NULL ,
CompetenceName        NVARCHAR (500) NOT NULL ,
CompetenceLEDTS       DATETIME ,
CompetenceRSRC        NVARCHAR (100) ,
H_CERTIFICATE_TRAINING_SQN_Training NUMERIC (5) NOT NULL
)
ON "default"
GO
ALTER TABLE S_COMPETENCE ADD CONSTRAINT S_COMPETENCE_PK PRIMARY KEY CLUSTERED (
CompetenceLEDTS, H_CERTIFICATE_TRAINING_SQN_Training)
WITH
(
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE S_EMPLOYEE
(
    EmployeeLEDTS      DATETIME NOT NULL ,
    Name               NVARCHAR (30) NOT NULL ,
    Surname            NVARCHAR (30) NOT NULL ,
    Title              NVARCHAR (10) ,
    Gender             NVARCHAR (10) NOT NULL ,
    BirthDate          DATE NOT NULL ,
    StreetNumber       NVARCHAR (5) NOT NULL ,
    StreetName         NVARCHAR (40) NOT NULL ,
    ZIP                NUMERIC (20) NOT NULL ,
    City               NVARCHAR (50) ,
    County             NVARCHAR (50) ,
    Country            NVARCHAR (50) ,
    EmailAddress       NVARCHAR (35) ,
    PhoneNumber        NVARCHAR (20) ,
    ShortBiography     NVARCHAR (1500) ,
    EmployeeLEDTS      DATETIME ,
    EmployeeRSRC       NVARCHAR (100) ,
    H_EMPLOYEE_SQN_Employee NUMERIC (5) NOT NULL
)

```



```

ON "default"
GO
ALTER TABLE S_EMPLOYEE ADD CONSTRAINT S_EMPLOYEE_PK PRIMARY KEY CLUSTERED (
EmployeeLDTS, H_EMPLOYEE_SQN_Employee)
WITH
(
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE S_EMPLOYEE_MASTER
(
    EmployeeLDTS          DATETIME NOT NULL ,
    Name                  NVARCHAR (30) NOT NULL ,
    Surname               NVARCHAR (30) NOT NULL ,
    Title                 NVARCHAR (10) ,
    Gender                NVARCHAR (10) NOT NULL ,
    BirthDate             DATE NOT NULL ,
    StreetNumber          NVARCHAR (5) NOT NULL ,
    StreetName            NVARCHAR (40) NOT NULL ,
    ZIP                   NUMERIC (20) NOT NULL ,
    City                  NVARCHAR (50) ,
    County                NVARCHAR (50) ,
    Country               NVARCHAR (50) ,
    EmailAddress          NVARCHAR (35) ,
    PhoneNumber           NVARCHAR (20) ,
    ShortBiography        NVARCHAR (1500) ,
    EmployeeLEDTS         DATETIME ,
    EmployeeRSRC          NVARCHAR (100) ,
    H_EMPLOYEE_MASTER_SQN_EmployeeMaster NUMERIC (5) NOT NULL
)
ON "default"
GO
ALTER TABLE S_EMPLOYEE_MASTER ADD CONSTRAINT S_EMPLOYEE_MASTER_PK PRIMARY KEY
CLUSTERED (EmployeeLDTS, H_EMPLOYEE_MASTER_SQN_EmployeeMaster)
WITH
(

```

```

ALLOW_PAGE_LOCKS = ON ,
ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE S_MARITAL_STATUS
(
    MaritalStatusLDTS          DATETIME NOT NULL ,
    MaritalStatus              NVARCHAR (20) NOT NULL ,
    MaritalStatusLEDTS        DATETIME ,
    MaritalStatusRSRC          NVARCHAR (100) ,
    H_MARITAL_STATUS_SQN_MaritalStatus NUMERIC (5) NOT NULL
)
ON "default"
GO
ALTER TABLE S_MARITAL_STATUS ADD CONSTRAINT S_MARITAL_STATUS_PK PRIMARY KEY
CLUSTERED (MaritalStatusLDTS, H_MARITAL_STATUS_SQN_MaritalStatus)
WITH
(
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE S_PARTICIPATES
(
    ParticipatesLDTS DATETIME NOT NULL ,
    TrainingStartDate DATE NOT NULL ,
    TrainingEndDate  DATE ,
    Passed BIT ,
    Grade NUMERIC (6,2) ,
    FirstTime BIT ,
    ParticipatesLEDTS          DATETIME ,
    ParticipatesRSRC           NVARCHAR (100) ,
    L_PARTICIPATES_SQN_Participates NUMERIC (5) NOT NULL
)

```

```

ON "default"
GO
ALTER TABLE S_PARTICIPATES ADD CONSTRAINT S_PARTICIPATES_PK PRIMARY KEY
CLUSTERED (ParticipatesLDTS, L_PARTICIPATES_SQN_Participates)
WITH
(
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE S_PROJECT
(
    ProjectLDTS      DATETIME NOT NULL ,
    ProjectName      NVARCHAR (50) NOT NULL ,
    ProjectGoalShort NVARCHAR (500) NOT NULL ,
    ProjectDescription NVARCHAR (2000) ,
    ProjectStartDate DATE NOT NULL ,
    ProjectEndDate   DATE ,
    PlannedEndDate   DATE NOT NULL ,
    BudgetWorkHours  NUMERIC (7) NOT NULL ,
    BudgetSalaryBonus NUMERIC (9,2) NOT NULL ,
    ProjectLEDTS     DATETIME ,
    ProjectRSRC      NVARCHAR (100) ,
    H_PROJECT_SQN_Project NUMERIC (5) NOT NULL
)
ON "default"
GO
ALTER TABLE S_PROJECT ADD CONSTRAINT S_PROJECT_PK PRIMARY KEY CLUSTERED (
ProjectLDTS, H_PROJECT_SQN_Project)
WITH
(
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

```

```

CREATE TABLE [dbo].[S_EMPLOYEEETRA](
    [EmployeeLDTS] [datetime] NOT NULL,
    [Name] [NVARCHAR](30) NOT NULL,
    [Surname] [NVARCHAR](30) NOT NULL,
    [Title] [NVARCHAR](10) NULL,
    [Gender] [NVARCHAR](10) NOT NULL,
    [MaritalStatus] [NVARCHAR](20) NOT NULL,
    [BirthDate] [date] NOT NULL,
    [StreetNumber] [NVARCHAR](5) NOT NULL,
    [StreetName] [NVARCHAR](40) NOT NULL,
    [ZIP] [numeric](20, 0) NOT NULL,
    [City] [NVARCHAR](50) NULL,
    [County] [NVARCHAR](50) NULL,
    [Country] [NVARCHAR](50) NULL,
    [EmailAddress] [NVARCHAR](35) NULL,
    [PhoneNumber] [NVARCHAR](20) NULL,
    [EmployeeLEDTS] [datetime] NULL,
    [EmployeeRSRC] [NVARCHAR](100) NULL,
    [H_EMPLOYEE_SQN_Employee] [numeric](5, 0) NOT NULL,
CONSTRAINT [S_EMPLOYEEETra_PK] PRIMARY KEY CLUSTERED
(
    [EmployeeLDTS] ASC,
    [H_EMPLOYEE_SQN_Employee] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS =
ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO

SET ANSI_PADDING OFF
GO

ALTER TABLE [dbo].[S_EMPLOYEEETRA] WITH CHECK ADD CONSTRAINT [S_EMPLOYEEETRA_EMPLOYEE_FK]
FOREIGN KEY([H_EMPLOYEE_SQN_Employee])
REFERENCES [dbo].[H_EMPLOYEE] ([SQN_Employee])
GO
ALTER TABLE [dbo].[S_EMPLOYEEETRA] CHECK CONSTRAINT [S_EMPLOYEEETRA_EMPLOYEE_FK]
GO
CREATE
TABLE S_SUPERVISOR

```

```

(
  SupervisorLDTS      DATETIME NOT NULL ,
  Name                NVARCHAR (30) NOT NULL ,
  Surname             NVARCHAR (30) NOT NULL ,
  EmailAddress        NVARCHAR (35) ,
  PhoneNumber         NVARCHAR (20) ,
  ShortBiography      NVARCHAR (1000) ,
  SupervisorLEDTS     DATETIME ,
  SupervisorRSRC      NVARCHAR (100) ,
  H_SUPERVISOR_SQN_Supervisor NUMERIC (5) NOT NULL
)
ON "default"
GO
ALTER TABLE S_SUPERVISOR ADD CONSTRAINT S_SUPERVISOR_PK PRIMARY KEY CLUSTERED (
SupervisorLDTS, H_SUPERVISOR_SQN_Supervisor)
WITH
(
  ALLOW_PAGE_LOCKS = ON ,
  ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

ALTER TABLE L_EMPLOYEE_MASTER
ADD CONSTRAINT L_EMPLOYEE_MASTER_EMPLOYEE_FK FOREIGN KEY
(
H_EMPLOYEE_SQN_Employee
)
REFERENCES H_EMPLOYEE
(
SQN_Employee
)

ALTER TABLE L_EMPLOYEE_MASTER
ADD CONSTRAINT L_EMPLOYEE_MASTER_EMPLOYEE_MASTER_FK FOREIGN KEY
(
H_EMPLOYEE_MASTER_SQN_EmployeeMaster
)
REFERENCES H_EMPLOYEE_MASTER
(

```

```

SQN_EmployeeMaster
)
ALTER TABLE L_EMP_MARITAL
ADD CONSTRAINT L_EMP_MARITAL_EMPLOYEE_FK FOREIGN KEY
(
H_EMPLOYEE_SQN_Employee
)
REFERENCES H_EMPLOYEE
(
SQN_Employee
)
ALTER TABLE L_EMP_MARITAL
ADD CONSTRAINT L_EMP_MARITAL_MARITAL_STATUS_FK FOREIGN KEY
(
H_MARITAL_STATUS_SQN_MaritalStatus
)
REFERENCES H_MARITAL_STATUS
(
SQN_MaritalStatus
)

ALTER TABLE L_PARTICIPATES
ADD CONSTRAINT L_PARTICIPATES_CERTIFICATE_TRAINING_FK FOREIGN KEY
(
H_CERTIFICATE_TRAINING_SQN_Training
)
REFERENCES H_CERTIFICATE_TRAINING
(
SQN_Training
)

ALTER TABLE L_PARTICIPATES
ADD CONSTRAINT L_PARTICIPATES_EMPLOYEE_FK FOREIGN KEY
(
H_EMPLOYEE_SQN_Employee
)
REFERENCES H_EMPLOYEE
(
SQN_Employee
)

```

```

ALTER TABLE L_SUPERVISES
ADD CONSTRAINT L_SUPERVISES_EMPLOYEE_FK FOREIGN KEY
(
H_EMPLOYEE_SQN_Employee
)
REFERENCES H_EMPLOYEE
(
SQN_Employee
)

ALTER TABLE L_SUPERVISES
ADD CONSTRAINT L_SUPERVISES_SUPERVISOR_FK FOREIGN KEY
(
H_SUPERVISOR_SQN_Supervisor
)
REFERENCES H_SUPERVISOR
(
SQN_Supervisor
)

ALTER TABLE L_WORKS_ON
ADD CONSTRAINT L_WORKS_ON_EMPLOYEE_FK FOREIGN KEY
(
H_EMPLOYEE_SQN_Employee
)
REFERENCES H_EMPLOYEE
(
SQN_Employee
)

ALTER TABLE L_WORKS_ON
ADD CONSTRAINT L_WORKS_ON_PROJECT_FK FOREIGN KEY
(
H_PROJECT_SQN_Project
)
REFERENCES H_PROJECT
(
SQN_Project
)

ALTER TABLE S_CERTIFICATE_TRAINING
ADD CONSTRAINT S_CERTIFICATE_TRAINING_CERTIFICATE_TRAINING_FK FOREIGN KEY
(

```

```

H_CERTIFICATE_TRAINING_SQN_Training
)
REFERENCES H_CERTIFICATE_TRAINING
(
SQN_Training
)
ALTER TABLE S_COMPETENCE
ADD CONSTRAINT S_COMPETENCE_CERTIFICATE_TRAINING_FK FOREIGN KEY
(
H_CERTIFICATE_TRAINING_SQN_Training
)
REFERENCES H_CERTIFICATE_TRAINING
(
SQN_Training
)
ALTER TABLE S_EMPLOYEE
ADD CONSTRAINT S_EMPLOYEE_EMPLOYEE_FK FOREIGN KEY
(
H_EMPLOYEE_SQN_Employee
)
REFERENCES H_EMPLOYEE
(
SQN_Employee
)
ALTER TABLE S_EMPLOYEE_MASTER
ADD CONSTRAINT S_EMPLOYEE_MASTER_EMPLOYEE_MASTER_FK FOREIGN KEY
(
H_EMPLOYEE_MASTER_SQN_EmployeeMaster
)
REFERENCES H_EMPLOYEE_MASTER
(
SQN_EmployeeMaster
)
ALTER TABLE S_MARITAL_STATUS
ADD CONSTRAINT S_MARITAL_STATUS_MARITAL_STATUS_FK FOREIGN KEY
(H_MARITAL_STATUS_SQN_MaritalStatus)
REFERENCES H_MARITAL_STATUS(
SQN_MaritalStatus)
ALTER TABLE S_PARTICIPATES
ADD CONSTRAINT S_PARTICIPATES_PARTICIPATES_FK FOREIGN KEY

```



```

(
L_PARTICIPATES_SQN_Participates
)
REFERENCES L_PARTICIPATES
(
SQN_Participates
)
ALTER TABLE S_PROJECT
ADD CONSTRAINT S_PROJECT_PROJECT_FK FOREIGN KEY
(
H_PROJECT_SQN_Project
)
REFERENCES H_PROJECT
(
SQN_Project
)
ALTER TABLE S_SUPERVISOR
ADD CONSTRAINT S_SUPERVISOR_SUPERVISOR_FK FOREIGN KEY
(
H_SUPERVISOR_SQN_Supervisor
)
REFERENCES H_SUPERVISOR
(
SQN_Supervisor
)
ALTER TABLE dbo.H_EMPLOYEE
ADD CONSTRAINT DF_HubEmpSQN DEFAULT (NEXT VALUE FOR dbo.SeqEmployeeSQN)
FOR SQN_Employee;
GO
ALTER TABLE dbo.H_SUPERVISOR
ADD CONSTRAINT DF_HubSuprSQN DEFAULT (NEXT VALUE FOR dbo.SeqSupervisorSQN)
FOR SQN_Supervisor;
GO
ALTER TABLE dbo.H_PROJECT
ADD CONSTRAINT DF_HubSuprSQN DEFAULT (NEXT VALUE FOR dbo.SeqProjectSQN)
FOR SQN_Project;
GO

ALTER TABLE dbo.H_EMPLOYEE_MASTER
ADD CONSTRAINT DF_HubSuprSQN DEFAULT (NEXT VALUE FOR dbo.SeqEmpMastSQN)

```

```

FOR SQN_EmployeeMaster;
GO
ALTER TABLE dbo.H_CERTIFICATE_TRAINING
ADD CONSTRAINT DF_HubSuprSQN DEFAULT (NEXT VALUE FOR dbo.SeqTrainingSQN)
FOR SQN_Training;
GO
ALTER TABLE dbo.H_MARITAL_STATUS
ADD CONSTRAINT DF_HubSuprSQN DEFAULT (NEXT VALUE FOR dbo.SeqMaritalSQN)
FOR SQN_MaritalStatus;
GO
ALTER TABLE dbo.L_SUPERVISES
ADD CONSTRAINT DF_HubSuprSQN DEFAULT (NEXT VALUE FOR dbo.SeqSupervisesSQN)
FOR SQN_Supervises;
GO
ALTER TABLE dbo.L_WORKS_ON
ADD CONSTRAINT DF_HubSuprSQN DEFAULT (NEXT VALUE FOR dbo.SeqWorksOnSQN)
FOR SQN_WorksOn;
GO
ALTER TABLE dbo.L_EMPLOYEE_MASTER
ADD CONSTRAINT DF_HubSuprSQN DEFAULT (NEXT VALUE FOR dbo.SeqEmpMastLinkSQN)
FOR SQN_EmpMasterEmp;
GO
ALTER TABLE dbo.L_PARTICIPATES
ADD CONSTRAINT DF_HubSuprSQN DEFAULT (NEXT VALUE FOR dbo.SeqParticipatesSQN)
FOR SQN_Participates;
GO
ALTER TABLE dbo.L_EMP_MARITAL
ADD CONSTRAINT DF_HubSuprSQN DEFAULT (NEXT VALUE FOR dbo.SeqEmpMaritalSQN)
FOR SQN_EmpMarital;
GO

```

## PmdvDB

```

-- Kreiranje tablica u bazi podataka PmdvDB
CREATE
TABLE H_CERTIFICATE_TRAINING_MASTER
(
SQN_Training NUMERIC (5) NOT NULL ,
TrainingID NUMERIC (5) NOT NULL ,

```

```

TrainingLDTS DATETIME ,
TrainingRSRC NVARCHAR (100)
)
ON "default"
GO
ALTER TABLE H_CERTIFICATE_TRAINING_MASTER ADD CONSTRAINT H_CERT_TRA_MAST_PK
PRIMARY KEY CLUSTERED (SQN_Training)
WITH
(
ALLOW_PAGE_LOCKS = ON ,
ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE H_EMPLOYEE_MASTER
(
SQN_Employee NUMERIC (5) NOT NULL ,
EmployeeID NUMERIC (5) NOT NULL ,
EmployeeLDTS DATETIME ,
EmployeeRSRC NVARCHAR (100)
)
ON "default"
GO
ALTER TABLE H_EMPLOYEE_MASTER ADD CONSTRAINT H_EMPLOYEE_MASTER_PK PRIMARY KEY
CLUSTERED (SQN_Employee)
WITH
(
ALLOW_PAGE_LOCKS = ON ,
ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE H_MARITAL_STATUS_MASTER
(
SQN_MaritalStatus NUMERIC (5) NOT NULL ,
MaritalStatusID NUMERIC (2) NOT NULL ,

```

```

MaritalStatusLDTS DATETIME ,
MaritalStatusRSRC NVARCHAR (100)
)
ON "default"
GO
ALTER TABLE H_MARITAL_STATUS_MASTER ADD CONSTRAINT H_MARITAL_STATUS_MASTER_PK
PRIMARY KEY CLUSTERED (SQN_MaritalStatus)
WITH
(
ALLOW_PAGE_LOCKS = ON ,
ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE H_PROJECT_MASTER
(
SQN_Project NUMERIC (5) NOT NULL ,
ProjectID NUMERIC (5) NOT NULL ,
ProjectLDTS DATETIME ,
ProjectRSRC NVARCHAR (100)
)
ON "default"
GO
ALTER TABLE H_PROJECT_MASTER ADD CONSTRAINT H_PROJECT_MASTER_PK PRIMARY KEY
CLUSTERED (SQN_Project)
WITH
(
ALLOW_PAGE_LOCKS = ON ,
ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE H_SUPERVISOR_MASTER
(
SQN_Supervisor NUMERIC (5) NOT NULL ,
SupervisorID NUMERIC (5) NOT NULL ,

```

```

SupervisorLDTS DATETIME ,
SupervisorRSRC NVARCHAR (100)
)
ON "default"
GO
ALTER TABLE H_SUPERVISOR_MASTER ADD CONSTRAINT H_SUPERVISOR_MASTER_PK PRIMARY
KEY CLUSTERED (SQN_Supervisor)
WITH
(
ALLOW_PAGE_LOCKS = ON ,
ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE L_EMP_MARITAL_MASTER
(
SQN_EmpMarital    NUMERIC (5) NOT NULL ,
EmpMaritalLDTS   DATETIME ,
EmpMaritalRSRC   NVARCHAR (100) ,
H_MASTAMAS_SQN_MarlSta NUMERIC (5) NOT NULL ,
H_EMPMAS_SQN_Emp    NUMERIC (5) NOT NULL
)
ON "default"
GO
ALTER TABLE L_EMP_MARITAL_MASTER ADD CONSTRAINT L_EMPMARMAS_PK PRIMARY KEY
CLUSTERED (SQN_EmpMarital)
WITH
(
ALLOW_PAGE_LOCKS = ON ,
ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE L_PARTICIPATES_MASTER
(
SQN_Participates  NUMERIC (5) NOT NULL ,

```

```

ParticipatesLDTS DATETIME ,
ParticipatesRSRC NVARCHAR (100) ,
H_EMP_MAS_SQN_Emp NUMERIC (5) NOT NULL ,
H_CERTRAMAS_SQN_Tra NUMERIC (5) NOT NULL
)
ON "default"
GO
ALTER TABLE L_PARTICIPATES_MASTER ADD CONSTRAINT L_PARTIC_MASTER_PK PRIMARY KEY
CLUSTERED (SQN_Participates)
WITH
(
ALLOW_PAGE_LOCKS = ON ,
ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE L_SUPERVISES_MASTER
(
SQN_Supervises NUMERIC (5) NOT NULL ,
SupervisesLDTS DATETIME ,
SupervisesRSRC NVARCHAR (100) ,
H_SUPERMAS_SQN_Super NUMERIC (5) NOT NULL ,
H_EMP_MAS_SQN_Emp NUMERIC (5) NOT NULL
)
ON "default"
GO
ALTER TABLE L_SUPERVISES_MASTER ADD CONSTRAINT L_SUPER_MAS_PK PRIMARY KEY
CLUSTERED (SQN_Supervises)
WITH
(
ALLOW_PAGE_LOCKS = ON ,
ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE L_WORKS_ON_MASTER

```

```

(
  SQN_WorksOn      NUMERIC (5) NOT NULL ,
  WorksOnLDTs     DATETIME ,
  WorksOnRSRC     NVARCHAR (100) ,
  H_EMPMAS_SQN_Emp  NUMERIC (5) NOT NULL ,
  H_PROJ_MAS_SQN_Project NUMERIC (5) NOT NULL
)
ON "default"
GO
ALTER TABLE L_WORKS_ON_MASTER ADD CONSTRAINT L_WORKS_ON_MASTER_PK PRIMARY KEY
CLUSTERED (SQN_WorksOn)
WITH
(
  ALLOW_PAGE_LOCKS = ON ,
  ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE S_CERTIFICATE_TRAINING_MASTER
(
  TrainingLDTs     DATETIME NOT NULL ,
  TrainingName     NVARCHAR (100) NOT NULL ,
  TrainingDescription NVARCHAR (1500) ,
  TrainingLevel    NVARCHAR (30) ,
  TrainingDays     NUMERIC (5) ,
  TrainingHours    NUMERIC (5) ,
  TrainingCertificateName NVARCHAR (100) ,
  TrainingCostPerEmployee NUMERIC (9,2) NOT NULL ,
  TrainingLEDTS    DATETIME ,
  TrainingRSRC     NVARCHAR (100) ,
  H_CER_TRA_MAS_SQN_Tra NUMERIC (5) NOT NULL
)
ON "default"
GO
ALTER TABLE S_CERTIFICATE_TRAINING_MASTER ADD CONSTRAINT S_CER_TRA_MAS_PK
PRIMARY KEY CLUSTERED (TrainingLDTs, H_CER_TRA_MAS_SQN_Tra)
WITH
(

```

```

ALLOW_PAGE_LOCKS = ON ,
ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE S_COMPETENCE_MASTER
(
CompetenceLDTS    DATETIME NOT NULL ,
CompetenceID     NUMERIC (5) NOT NULL ,
CompetenceName   NVARCHAR (500) NOT NULL ,
CompetenceLEDTS  DATETIME ,
CompetenceRSRC   NVARCHAR (100) ,
H_CERTTRAMAS_SQN_Tra NUMERIC (5) NOT NULL
)
ON "default"
GO
ALTER TABLE S_COMPETENCE_MASTER ADD CONSTRAINT S_COMP_MAS_PK PRIMARY KEY
CLUSTERED (CompetenceLDTS, H_CERTTRAMAS_SQN_Tra)
WITH
(
ALLOW_PAGE_LOCKS = ON ,
ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE S_EMPLOYEE_MASTER
(
EmployeeLDTS    DATETIME NOT NULL ,
Name           NVARCHAR (30) NOT NULL ,
Surname        NVARCHAR (30) NOT NULL ,
Title         NVARCHAR (10) ,
Gender        NVARCHAR (10) NOT NULL ,
BirthDate     DATE NOT NULL ,
StreetNumber  NVARCHAR (5) NOT NULL ,
StreetName    NVARCHAR (40) NOT NULL ,
ZIP          NUMERIC (20) NOT NULL ,

```



```

City          NVARCHAR (50) ,
County        NVARCHAR (50) ,
Country       NVARCHAR (50) ,
EmailAddress  NVARCHAR (35) ,
PhoneNumber   NVARCHAR (20) ,
ShortBiography NVARCHAR (1500) ,
EmployeeLEDTs DATETIME ,
EmployeeRSRC  NVARCHAR (100) ,
H_EMP_MAS_SQN_Emp NUMERIC (5) NOT NULL
)
ON "default"
GO
ALTER TABLE S_EMPLOYEE_MASTER ADD CONSTRAINT S_EMPLOYEE_MASTER_PK PRIMARY KEY
CLUSTERED (EmployeeLEDTs, H_EMP_MAS_SQN_Emp)
WITH
(
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE S_MARITAL_STATUS_MASTER
(
    MaritalStatusLEDTs DATETIME NOT NULL ,
    MaritalStatus      NVARCHAR (20) NOT NULL ,
    MaritalStatusLEDTs DATETIME ,
    MaritalStatusRSRC  NVARCHAR (100) ,
    H_MARSTAMAS_SQN_Mar NUMERIC (5) NOT NULL
)
ON "default"
GO
ALTER TABLE S_MARITAL_STATUS_MASTER ADD CONSTRAINT S_MAR_STA_MAS_PK PRIMARY KEY
CLUSTERED (MaritalStatusLEDTs, H_MARSTAMAS_SQN_Mar)
WITH
(
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON
)

```

```

ON "default"
GO

CREATE
TABLE S_PARTICIPATES_MASTER
(
    ParticipatesLDTS DATETIME NOT NULL ,
    TrainingStartDate DATE NOT NULL ,
    TrainingEndDate DATE ,
    Passed BIT ,
    Grade NUMERIC (6,2) ,
    FirstTime BIT ,
    ParticipatesLEDTS DATETIME ,
    ParticipatesRSRC NVARCHAR (100) ,
    L_PART_MAS_SQN_Part NUMERIC (5) NOT NULL
)
ON "default"
GO

ALTER TABLE S_PARTICIPATES_MASTER ADD CONSTRAINT S_PARTICIPATES_MASTER_PK
PRIMARY KEY CLUSTERED (ParticipatesLDTS, L_PART_MAS_SQN_Part)
WITH
(
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE S_PROJECT_MASTER
(
    ProjectLDTS DATETIME NOT NULL ,
    ProjectName NVARCHAR (50) NOT NULL ,
    ProjectGoalShort NVARCHAR (500) NOT NULL ,
    ProjectDescription NVARCHAR (2000) ,
    ProjectStartDate DATE NOT NULL ,
    ProjectEndDate DATE ,
    PlannedEndDate DATE NOT NULL ,
    BudgetWorkHours NUMERIC (7) NOT NULL ,
    BudgetSalaryBonus NUMERIC (9,2) NOT NULL ,

```

```

ProjectLEDTS    DATETIME ,
ProjectRSRC     NVARCHAR (100) ,
H_PROJ_MAS_SQN_Project NUMERIC (5) NOT NULL
)
ON "default"
GO
ALTER TABLE S_PROJECT_MASTER ADD CONSTRAINT S_PROJECT_MASTER_PK PRIMARY KEY
CLUSTERED (ProjectLDTS, H_PROJ_MAS_SQN_Project)
WITH
(
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE S_SUPERVISOR_MASTER
(
    SupervisorLDTS    DATETIME NOT NULL ,
    Name              NVARCHAR (30) NOT NULL ,
    Surname           NVARCHAR (30) NOT NULL ,
    EmailAddress      NVARCHAR (35) ,
    PhoneNumber       NVARCHAR (20) ,
    ShortBiography    NVARCHAR (1000) ,
    SupervisorLEDTS   DATETIME ,
    SupervisorRSRC    NVARCHAR (100) ,
    H_SUPERMAS_SQN_Super NUMERIC (5) NOT NULL
)
ON "default"
GO
ALTER TABLE S_SUPERVISOR_MASTER ADD CONSTRAINT S_SUPERVISOR_MASTER_PK PRIMARY
KEY CLUSTERED (SupervisorLDTS, H_SUPERMAS_SQN_Super)
WITH
(
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

```

```

ALTER TABLE L_EMP_MARITAL_MASTER
ADD CONSTRAINT L_EMPMARMAS_EMPMAS_FK FOREIGN KEY
(
H_EMPMAS_SQN_Emp
)
REFERENCES H_EMPLOYEE_MASTER
(
SQN_Employee
)
ALTER TABLE L_EMP_MARITAL_MASTER
ADD CONSTRAINT L_EMPMARMAS_MARSTAMAS_FK FOREIGN KEY
(
H_MASTAMAS_SQN_MarlSta
)
REFERENCES H_MARITAL_STATUS_MASTER
(
SQN_MaritalStatus
)
ALTER TABLE L_PARTICIPATES_MASTER
ADD CONSTRAINT L_PARTMAS_CERTRAMAS_FK FOREIGN KEY
(
H_CERTRAMAS_SQN_Tra
)
REFERENCES H_CERTIFICATE_TRAINING_MASTER
(
SQN_Training
)
ALTER TABLE L_PARTICIPATES_MASTER
ADD CONSTRAINT L_PARTMAS_EMPMAS_FK FOREIGN KEY
(
H_EMP_MAS_SQN_Emp
)
REFERENCES H_EMPLOYEE_MASTER
(
SQN_Employee
)
ALTER TABLE L_SUPERVISES_MASTER
ADD CONSTRAINT L_SUPERMAS_EMPMAS_FK FOREIGN KEY
(

```

```

H_EMP_MAS_SQN_Emp
)
REFERENCES H_EMPLOYEE_MASTER
(
SQN_Employee
)
ALTER TABLE L_SUPERVISES_MASTER
ADD CONSTRAINT L_SUPERMAS_SUPERMAS_FK FOREIGN KEY
(
H_SUPERMAS_SQN_Super
)
REFERENCES H_SUPERVISOR_MASTER
(
SQN_Supervisor
)
ALTER TABLE L_WORKS_ON_MASTER
ADD CONSTRAINT L_WORMAS_EMP_MAS_FK FOREIGN KEY
(
H_EMPMAS_SQN_Emp
)
REFERENCES H_EMPLOYEE_MASTER
(
SQN_Employee
)
ALTER TABLE L_WORKS_ON_MASTER
ADD CONSTRAINT L_WORMAS_PROJ_MAS_FK FOREIGN KEY
(
H_PROJ_MAS_SQN_Project
)
REFERENCES H_PROJECT_MASTER
(
SQN_Project
)
ALTER TABLE S_CERTIFICATE_TRAINING_MASTER
ADD CONSTRAINT S_CERTRAMAS_CERTRAMAS_FK FOREIGN KEY
(
H_CER_TRA_MAS_SQN_Tra
)
REFERENCES H_CERTIFICATE_TRAINING_MASTER
(

```

```

SQN_Training
)
ALTER TABLE S_COMPETENCE_MASTER
ADD CONSTRAINT S_COMPMAS_CERTRAMAS_FK FOREIGN KEY
(
H_CERTTRAMAS_SQN_Tra
)
REFERENCES H_CERTIFICATE_TRAINING_MASTER
(
SQN_Training
)
ALTER TABLE S_EMPLOYEE_MASTER
ADD CONSTRAINT S_EMP_MAS_EMP_MAS_FK FOREIGN KEY
(
H_EMP_MAS_SQN_Emp
)
REFERENCES H_EMPLOYEE_MASTER
(
SQN_Employee
)
ALTER TABLE S_MARITAL_STATUS_MASTER
ADD CONSTRAINT S_MARSTAMAS_MARSTAMAS_FK FOREIGN KEY
(
H_MARSTAMAS_SQN_Mar
)
REFERENCES H_MARITAL_STATUS_MASTER
(
SQN_MaritalStatus
)
ALTER TABLE S_PARTICIPATES_MASTER
ADD CONSTRAINT S_PARTMAS_PARTMAS_FK FOREIGN KEY
(
L_PART_MAS_SQN_Part
)
REFERENCES L_PARTICIPATES_MASTER
(
SQN_Participates
)
ALTER TABLE S_PROJECT_MASTER
ADD CONSTRAINT S_PROJ_MAS_PROJ_MAS_FK FOREIGN KEY

```

```

(
H_PROJ_MAS_SQN_Project
)
REFERENCES H_PROJECT_MASTER
(
SQN_Project
)
ALTER TABLE S_SUPERVISOR_MASTER
ADD CONSTRAINT S_SUPERMAS_SUPERMAS_FK FOREIGN KEY
(
H_SUPERMAS_SQN_Super
)
REFERENCES H_SUPERVISOR_MASTER
(
SQN_Supervisor
)

```

## MdmDB

```

-- Kreiranje tablica u bazi podataka MdmDB
CREATE
TABLE MDM_TRAINING
(
TrainingID      NUMERIC (5) NOT NULL ,
TrainingName    VARCHAR (100) NOT NULL ,
TrainingDescription  VARCHAR (1500) ,
TrainingLevel   VARCHAR (30) ,
TrainingDays    NUMERIC (5) ,
TrainingHours   NUMERIC (5) ,
TrainingCertificateName VARCHAR (100) ,
TrainingCostPerEmployee NUMERIC (9,2) NOT NULL ,
RecordStartDateT  DATE NOT NULL ,
RecordEndDate    DATE ,
ETL_LoadJob      NVARCHAR (500) ,
RecordSource     NVARCHAR (100) ,
SQN_mdm_Training INTEGER NOT NULL
)
ON "default"
GO

```

```

ALTER TABLE MDM_TRAINING ADD CONSTRAINT DIM_TRAINING_PK PRIMARY KEY CLUSTERED (
SQN_mdm_Training, RecordStartDateT)
WITH
(
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE FACT_PARTICIPATES
(
    Passed BIT ,
    Grade NUMERIC (6,2) ,
    FirstTime BIT ,
    MDM_EMPLOYEE_SQN_mdm_Employee INTEGER NOT NULL ,
    MDM_EMPLOYEE_RecordStartDateE DATE NOT NULL ,
    DIM_TRAINING_SQN_mdm_Training INTEGER NOT NULL ,
    DIM_TRAINING_RecordStartDateT DATE NOT NULL
)
ON "default"
GO

ALTER TABLE FACT_PARTICIPATES ADD CONSTRAINT FACT_PARTICIPATES_PK PRIMARY KEY
CLUSTERED (MDM_EMPLOYEE_SQN_mdm_Employee, MDM_EMPLOYEE_RecordStartDateE,
MDM_TRAINING_SQN_mdm_Training, MDM_TRAINING_RecordStartDateT)
WITH
(
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE FACT_WORKS_ON
(
    WorkedHours          NUMERIC (9,2) ,
    HourlySalaryBonus    NUMERIC (9,2) ,
    ProjectBonusSalary   NUMERIC (9,2) ,

```



```

MDM_EMPLOYEE_SQN_mdm_Employee INTEGER NOT NULL ,
MDM_EMPLOYEE_RecordStartDateE DATE NOT NULL ,
MDM_PROJECT_SQN_mdm_Project  INTEGER NOT NULL ,
MDM_PROJECT_RecordStartDateP DATE NOT NULL
)
ON "default"
GO
ALTER TABLE FACT_WORKS_ON ADD CONSTRAINT FACT_WORKS_ON_PK PRIMARY KEY CLUSTERED
(MDM_EMPLOYEE_SQN_mdm_Employee, MDM_EMPLOYEE_RecordStartDateE,
MDM_PROJECT_SQN_mdm_Project, MDM_PROJECT_RecordStartDateP)
WITH
(
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE MDM_EMPLOYEE
(
    EmployeeID    NUMERIC (5) NOT NULL ,
    Name          VARCHAR (30) NOT NULL ,
    Surname       VARCHAR (30) NOT NULL ,
    Title         VARCHAR (10) ,
    Gender        VARCHAR (10) NOT NULL ,
    MaritalStatus VARCHAR (20) ,
    BirthDate     DATE NOT NULL ,
    StreetNumber  VARCHAR (5) NOT NULL ,
    StreetName    VARCHAR (40) NOT NULL ,
    ZIP           NUMERIC (20) NOT NULL ,
    City          VARCHAR (50) ,
    County        VARCHAR (50) ,
    Country       VARCHAR (50) ,
    EmailAddress  VARCHAR (35) ,
    PhoneNumber   VARCHAR (20) ,
    SupervisorID  NUMERIC (5) ,
    SupervisorName VARCHAR (30) ,
    RecordStartDateE DATE NOT NULL ,
    RecordEndDate DATE ,

```

```

    ETL_LoadJob NVARCHAR (500) ,
    RecordSource NVARCHAR (100) ,
    SQN_mdm_Employee INTEGER NOT NULL
)
ON "default"
GO
ALTER TABLE MDM_EMPLOYEE ADD CONSTRAINT MDM_EMPLOYEE_PK PRIMARY KEY CLUSTERED (
SQN_mdm_Employee, RecordStartDateE)
WITH
(
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE MDM_PROJECT
(
    ProjectID    NUMERIC (5) NOT NULL ,
    ProjectName  VARCHAR (50) NOT NULL ,
    ProjectGoalShort  VARCHAR (500) NOT NULL ,
    ProjectStartDate  DATE NOT NULL ,
    ProjectEndDate   DATE ,
    PlannedEndDate   DATE NOT NULL ,
    BudgetWorkHours  NUMERIC (7) NOT NULL ,
    BudgetSalaryBonus NUMERIC (9,2) NOT NULL ,
    RecordStartDateP  DATE NOT NULL ,
    RecordEndDate    DATE ,
    ETL_LoadJob NVARCHAR (500) ,
    RecordSource NVARCHAR (100) ,
    SQN_mdm_Project INTEGER NOT NULL
)
ON "default"
GO
ALTER TABLE MDM_PROJECT ADD CONSTRAINT MDM_PROJECT_PK PRIMARY KEY CLUSTERED (
SQN_mdm_Project, RecordStartDateP)
WITH
(
    ALLOW_PAGE_LOCKS = ON ,

```

```

ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

ALTER TABLE FACT_PARTICIPATES
ADD CONSTRAINT FACT_PARTICIPATES_DIM_TRAINING_FK FOREIGN KEY
(
DIM_TRAINING_SQN_mdm_Training,
DIM_TRAINING_RecordStartDateT
)
REFERENCES DIM_TRAINING
(
SQN_mdm_Training ,
RecordStartDateT
)

ALTER TABLE FACT_PARTICIPATES
ADD CONSTRAINT FACT_PARTICIPATES_MDM_EMPLOYEE_FK FOREIGN KEY
(
MDM_EMPLOYEE_SQN_mdm_Employee,
MDM_EMPLOYEE_RecordStartDateE
)
REFERENCES MDM_EMPLOYEE
(
SQN_mdm_Employee ,
RecordStartDateE
)

ALTER TABLE FACT_WORKS_ON
ADD CONSTRAINT FACT_WORKS_ON_MDM_EMPLOYEE_FK FOREIGN KEY
(
MDM_EMPLOYEE_SQN_mdm_Employee,
MDM_EMPLOYEE_RecordStartDateE
)
REFERENCES MDM_EMPLOYEE
(
SQN_mdm_Employee ,
RecordStartDateE
)

```

```

ALTER TABLE FACT_WORKS_ON
ADD CONSTRAINT FACT_WORKS_ON_MDM_PROJECT_FK FOREIGN KEY
(
MDM_PROJECT_SQN_mdm_Project,
MDM_PROJECT_RecordStartDateP
)
REFERENCES MDM_PROJECT
(
SQN_mdm_Project ,
RecordStartDateP
)

```

## DmDB

```

-- Kreiranje tablica u bazi podataka DmDB

CREATE
TABLE DIM_EMPLOYEE
(
EmployeeID NUMERIC (5) NOT NULL ,
Name NVARCHAR (30) NOT NULL ,
Surname NVARCHAR (30) NOT NULL ,
Title NVARCHAR (10) ,
Gender NVARCHAR (10) NOT NULL ,
MaritalStatus NVARCHAR (20) ,
BirthDate DATE NOT NULL ,
StreetNumber NVARCHAR (5) NOT NULL ,
StreetName NVARCHAR (40) NOT NULL ,
ZIP NUMERIC (20) NOT NULL ,
City NVARCHAR (50) ,
County NVARCHAR (50) ,
Country NVARCHAR (50) ,
EmailAddress NVARCHAR (35) ,
PhoneNumber NVARCHAR (20) ,
SupervisorID NUMERIC (5) ,
SupervisorName NVARCHAR (30) ,
RecordStartDate DATE NOT NULL ,
RecordEndDate DATE ,
ETL_LoadJob NNVARCHAR (500) ,

```

```

RecordSource NNVARCHAR (100) ,
SQN_Dim_Employee INTEGER NOT NULL
)
ON "default"
GO
ALTER TABLE DIM_EMPLOYEE ADD CONSTRAINT DIM_EMPLOYEE_PK PRIMARY KEY CLUSTERED (
SQN_Dim_Employee, RecordStartDate)
WITH
(
ALLOW_PAGE_LOCKS = ON ,
ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE DIM_PROJECT
(
ProjectID NUMERIC (5) NOT NULL ,
ProjectName NVARCHAR (50) NOT NULL ,
ProjectGoalShort NVARCHAR (500) NOT NULL ,
ProjectStartDate DATE NOT NULL ,
ProjectEndDate DATE ,
PlannedEndDate DATE NOT NULL ,
BudgetWorkHours NUMERIC (7) NOT NULL ,
BudgetSalaryBonus NUMERIC (9,2) NOT NULL ,
RecordStartDate DATE NOT NULL ,
RecordEndDate DATE ,
ETL_LoadJob NNVARCHAR (500) ,
RecordSource NNVARCHAR (100) ,
SQN_Dim_Project INTEGER NOT NULL
)
ON "default"
GO
ALTER TABLE DIM_PROJECT ADD CONSTRAINT DIM_PROJECT_PK PRIMARY KEY CLUSTERED (
SQN_Dim_Project, RecordStartDate)
WITH
(
ALLOW_PAGE_LOCKS = ON ,
ALLOW_ROW_LOCKS = ON

```

```

)
ON "default"
GO

CREATE
TABLE DIM_TRAINING
(
    TrainingID          NUMERIC (5) NOT NULL ,
    TrainingName        NVARCHAR (200) NOT NULL ,
    TrainingDescription NVARCHAR (1500) ,
    TrainingLevel       NVARCHAR (30) ,
    TrainingDays        NUMERIC (5) ,
    TrainingHours       NUMERIC (5) ,
    TrainingCertificateName NVARCHAR (200) ,
    TrainingCostPerEmployee NUMERIC (9,2) NOT NULL ,
    RecordStartDate     DATE NOT NULL ,
    RecordEndDate       DATE ,
    ETL_LoadJob         NNVARCHAR (500) ,
    RecordSource        NNVARCHAR (100) ,
    SQN_Dim_Training   INTEGER NOT NULL
)
ON "default"
GO

ALTER TABLE DIM_TRAINING ADD CONSTRAINT DIM_TRAINING_PK PRIMARY KEY CLUSTERED (
SQN_Dim_Training, RecordStartDate)
WITH
(
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE FACT_PARTICIPATES
(
    Passed BIT ,
    Grade NUMERIC (6,2) ,
    FirstTime BIT ,
    DIM_TRA_SQN_DimTra    INTEGER NOT NULL ,

```

```

    DIM_EMP_SQN_DimEm      INTEGER NOT NULL ,
    DIM_EMPLOYEE_RecordStartDate DATE NOT NULL ,
    DIM_TRAINING_RecordStartDate DATE NOT NULL
)
ON "default"
GO
ALTER TABLE FACT_PARTICIPATES ADD CONSTRAINT FACT_PARTICIPATES_PK PRIMARY KEY
CLUSTERED (DIM_TRA_SQN_DimTra, DIM_EMP_SQN_DimEm, DIM_EMPLOYEE_RecordStartDate,
DIM_TRAINING_RecordStartDate)
WITH
(
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE FACT_WORKS_ON
(
    WorkedHours          NUMERIC (9,2) ,
    HourlySalaryBonus    NUMERIC (9,2) ,
    ProjectBonusSalary   NUMERIC (9,2) ,
    DIM_EMP_SQN_DimEmp    INTEGER NOT NULL ,
    DIM_PROJ_SQN_DimProj  INTEGER NOT NULL ,
    DIM_EMPLOYEE_RecordStartDate DATE NOT NULL ,
    DIM_PROJECT_RecordStartDate DATE NOT NULL
)
ON "default"
GO
ALTER TABLE FACT_WORKS_ON ADD CONSTRAINT FACT_WORKS_ON_PK PRIMARY KEY CLUSTERED
(DIM_EMP_SQN_DimEmp, DIM_PROJ_SQN_DimProj, DIM_EMPLOYEE_RecordStartDate,
DIM_PROJECT_RecordStartDate)
WITH
(
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

```

```

ALTER TABLE FACT_PARTICIPATES
ADD CONSTRAINT FACT_PART_DIM_EMP_FK FOREIGN KEY
(
DIM_EMP_SQN_DimEm,
DIM_EMPLOYEE_RecordStartDate
)
REFERENCES DIM_EMPLOYEE
(
SQN_Dim_Employee ,
RecordStartDate
)

ALTER TABLE FACT_PARTICIPATES
ADD CONSTRAINT FACT_PART_DIM_TRA_FK FOREIGN KEY
(
DIM_TRA_SQN_DimTra,
DIM_TRAINING_RecordStartDate
)
REFERENCES DIM_TRAINING
(
SQN_Dim_Training ,
RecordStartDate
)

ALTER TABLE FACT_WORKS_ON
ADD CONSTRAINT FACT_WORKSON_DIMEMP_FK FOREIGN KEY
(
DIM_EMP_SQN_DimEmp,
DIM_EMPLOYEE_RecordStartDate
)
REFERENCES DIM_EMPLOYEE
(
SQN_Dim_Employee ,
RecordStartDate
)

ALTER TABLE FACT_WORKS_ON
ADD CONSTRAINT FACT_WORKSON_DIMPRO_FK FOREIGN KEY
(
DIM_PROJ_SQN_DimProj,

```



```

DIM_PROJECT_RecordStartDate
)
REFERENCES DIM_PROJECT
(
SQN_Dim_Project ,
RecordStartDate
)
ALTER TABLE [dbo].[FACT_PARTICIPATES] ADD CONSTRAINT [FACT_PART_DIM_DAT_FK] FOREIGN
KEY([DIM_DAT_SQN_Key])
REFERENCES [dbo].[DT_DATE] ([DateKey])
GO

ALTER TABLE [dbo].[FACT_WORKS_ON] ADD CONSTRAINT [FACT_WORKS_DIM_DAT_FK] FOREIGN
KEY([DIM_DAT_SQN_Key])
REFERENCES [dbo].[DT_DATE] ([DateKey])
GO

CREATE TABLE [dbo].[DT_Date]
( [DateKey] BIGINT NOT NULL
, [FullDate] DATETIME NULL
, [DateName] CHAR(11) NULL
, [DayOfWeek] TINYINT NULL
, [DayNameOfWeek] CHAR(10) NULL
, [DayOfMonth] TINYINT NULL
, [DayOfYear] SMALLINT NULL
, [WeekdayWeekend] CHAR(7) NULL
, [WeekOfYear] TINYINT NULL
, [MonthName] CHAR(10) NULL
, [MonthOfYear] TINYINT NULL
, [IsLastDayOfMonth] CHAR(1) NULL
, [CalendarQuarter] TINYINT NULL
, [CalendarYear] SMALLINT NULL
, [CalendarYearMonth] CHAR(7) NULL
, [CalendarYearQtr] CHAR(7) NULL
, [FiscalMonthOfYear] TINYINT NULL
, [FiscalQuarter] TINYINT NULL
, [FiscalYear] INT NULL
, [FiscalYearMonth] CHAR(9) NULL
, [FiscalYearQtr] CHAR(8) NULL
, [AuditKey] BIGINT IDENTITY NOT NULL

```

```

    , CONSTRAINT [PK_DT_Date] PRIMARY KEY CLUSTERED ( [DateKey] )
  )
ON [PRIMARY]
GO

```

## MdvDB

```

CREATE
  TABLE HL_RULE_RELSHP
  (
    SQN_RuleRelshp NUMERIC (10) NOT NULL ,
    RuleRelshpLDTS DATETIME ,
    RuleRelshpRSRC NVARCHAR (100) ,
    H_RULE_SQN_Rule NUMERIC (10) NOT NULL ,
    H_RULE_SQN_Rule1 NUMERIC (10) NOT NULL
  )
ON "default"
GO
ALTER TABLE HL_RULE_RELSHP ADD CONSTRAINT HL_RULE_RELSHP_PK PRIMARY KEY
CLUSTERED (SQN_RuleRelshp)
WITH
  (
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON
  )
ON "default"
GO

CREATE
  TABLE HL_TRANSF_RELSHP
  (
    SQN_TransfRelshp NUMERIC (10) NOT NULL ,
    TransfRelshpLDTS DATETIME ,
    TransfRelshpRSRC NVARCHAR (100) ,
    H_TRNSF_SQN_TrnsfMaster NUMERIC (10) NOT NULL ,
    H_TRNSF_SQN_TrnsfBase NUMERIC (10) NOT NULL
  )
ON "default"

```

```

GO
ALTER TABLE HL_TRANSF_RELSHP ADD CONSTRAINT HL_TRANSF_RELSHP_PK PRIMARY KEY
CLUSTERED (SQN_TransfRelshp)
WITH
(
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE H_ATTRIBUTE
(
    SQN_Attribute NUMERIC (10) NOT NULL ,
    AttributeID NVARCHAR (50) ,
    AttributeLDTS DATETIME ,
    AttributeRSRC NVARCHAR (100)
)
ON "default"
GO

ALTER TABLE H_ATTRIBUTE ADD CONSTRAINT H_ATTRIBUTE_PK PRIMARY KEY CLUSTERED (
SQN_Attribute)
WITH
(
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE H_CONSTRAINT
(
    SQN_Constraint NUMERIC (10) NOT NULL ,
    ConstraintID NVARCHAR (50) ,
    ConstraintLDTS DATETIME ,
    ConstraintRSRC NVARCHAR (100)
)
ON "default"

```

```

GO
ALTER TABLE H_CONSTRAINT ADD CONSTRAINT H_CONSTRAINT_PK PRIMARY KEY CLUSTERED (
SQN_Constraint)
WITH
(
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE H_DATA_SOURCE
(
    SQN_DataSource NUMERIC (10) NOT NULL ,
    DataSourceID NVARCHAR (50) ,
    DataSourceLDTS DATETIME ,
    DataSourceRSRC NVARCHAR (100)
)
ON "default"
GO

ALTER TABLE H_DATA_SOURCE ADD CONSTRAINT H_DATA_SOURCE_PK PRIMARY KEY CLUSTERED
(SQN_DataSource)
WITH
(
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE H_DOMAIN
(
    SQN_Domain NUMERIC (10) NOT NULL ,
    DomainID NVARCHAR (50) ,
    DomainLDTS DATETIME ,
    DomainRSRC NVARCHAR (100)
)
ON "default"

```

```

GO
ALTER TABLE H_DOMAIN ADD CONSTRAINT H_DOMAIN_PK PRIMARY KEY CLUSTERED (
SQN_Domain)
WITH
(
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE H_HUB
(
    SQN_Hub NUMERIC (10) NOT NULL ,
    HubID NVARCHAR (50) ,
    HubLDTS DATETIME ,
    HubRSRC NVARCHAR (100)
)
ON "default"
GO
ALTER TABLE H_HUB ADD CONSTRAINT H_HUB_PK PRIMARY KEY CLUSTERED (SQN_Hub)
WITH
(
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE H_LINK
(
    SQN_Link NUMERIC (10) NOT NULL ,
    LinkID NVARCHAR (50) ,
    LinkLDTS DATETIME ,
    LinkRSRC NVARCHAR (100)
)
ON "default"
GO

```

```

ALTER TABLE H_LINK ADD CONSTRAINT H_LINK_PK PRIMARY KEY CLUSTERED (SQN_Link)
WITH
(
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE H_REFERENCE
(
    SQN_Reference NUMERIC (10) NOT NULL ,
    ReferenceID NVARCHAR (50) ,
    ReferenceLDTS DATETIME ,
    ReferenceRSRC NVARCHAR (100)
)
ON "default"
GO
ALTER TABLE H_REFERENCE ADD CONSTRAINT H_REFERENCE_PK PRIMARY KEY CLUSTERED (
SQN_Reference)
WITH
(
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE H_RESOURCE
(
    SQN_Resource NUMERIC (10) NOT NULL ,
    ResourceID NVARCHAR (50) ,
    ResourceLDTS DATETIME ,
    ResourceRSRC NVARCHAR (100) ,
    ChangeType NVARCHAR (100)
)
ON "default"
GO

```

```

ALTER TABLE H_RESOURCE ADD CONSTRAINT H_RESOURCE_PK PRIMARY KEY CLUSTERED (
SQN_Resource)
WITH
(
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE H_RULE
(
    SQN_Rule NUMERIC (10) NOT NULL ,
    RuleID NVARCHAR (50) ,
    RuleLDTS DATETIME ,
    RuleRSRC NVARCHAR (100)
)
ON "default"
GO

ALTER TABLE H_RULE ADD CONSTRAINT H_RULE_PK PRIMARY KEY CLUSTERED (SQN_Rule)
WITH
(
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE H_SATELLITE
(
    SQN_Satellite NUMERIC (10) NOT NULL ,
    SatelliteID NVARCHAR (50) ,
    SatelliteLDTS DATETIME ,
    SatelliteRSRC NVARCHAR (100)
)
ON "default"
GO

ALTER TABLE H_SATELLITE ADD CONSTRAINT H_SATELLITE_PK PRIMARY KEY CLUSTERED (

```

```

SQN_Satellite)
WITH
(
  ALLOW_PAGE_LOCKS = ON ,
  ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE H_SRC_ATTRIBUTE
(
  SQN_SrcAttr NUMERIC (10) NOT NULL ,
  SrcAttrID NVARCHAR (50) ,
  SrcAttrLDTS DATETIME ,
  SrcAttrRSRC NVARCHAR (100)
)
ON "default"
GO
ALTER TABLE H_SRC_ATTRIBUTE ADD CONSTRAINT H_SRC_ATTRIBUTE_PK PRIMARY KEY
CLUSTERED (SQN_SrcAttr)
WITH
(
  ALLOW_PAGE_LOCKS = ON ,
  ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE H_SRC_RELATION
(
  SQN_SrcRelation NUMERIC (10) NOT NULL ,
  SrcRelationID NVARCHAR (50) ,
  SrcRelationLDTS DATETIME ,
  SrcRelationRSRC NVARCHAR (100)
)
ON "default"
GO
ALTER TABLE H_SRC_RELATION ADD CONSTRAINT H_SRC_RELATION_PK PRIMARY KEY

```



```

CLUSTERED (SQN_SrcRelation)
WITH
(
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE H_SRC_RELATIONSHIP
(
    SQN_SrcRelshp NUMERIC (10) NOT NULL ,
    SrcRelshpID NVARCHAR (50) ,
    SrcRelshpLDTs DATETIME ,
    SrcRelshpRSRC NVARCHAR (100)
)
ON "default"
GO
ALTER TABLE H_SRC_RELATIONSHIP ADD CONSTRAINT H_SRC_RELATIONSHIP_PK PRIMARY KEY
CLUSTERED (SQN_SrcRelshp)
WITH
(
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE H_TRANSFORMATION
(
    SQN_Transformation NUMERIC (10) NOT NULL ,
    TransformationID NVARCHAR (50) ,
    TransformationLDTs DATETIME ,
    TransformationRSRC NVARCHAR (100)
)
ON "default"
GO
ALTER TABLE H_TRANSFORMATION ADD CONSTRAINT H_TRANSFORMATION_PK PRIMARY KEY

```

```

CLUSTERED (SQN_Transformation)
WITH
(
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE L_ATTR_DOM
(
    SQN_AttrDom          NUMERIC (10) NOT NULL ,
    AttrDomLDTS         DATETIME ,
    AttrDomRSRC         NVARCHAR (100) ,
    H_RULE_SQN_Rule     NUMERIC (10) ,
    H_ATTRIBUTE_SQN_Attribute NUMERIC (10) NOT NULL ,
    H_DOMAIN_SQN_Domain NUMERIC (10) NOT NULL
)
ON "default"
GO

ALTER TABLE L_ATTR_DOM ADD CONSTRAINT L_ATTR_DOM_PK PRIMARY KEY CLUSTERED (
SQN_AttrDom)
WITH
(
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE L_ATTR_REF
(
    SQN_AttrRef          NUMERIC (10) NOT NULL ,
    AttrRefLDTS         DATETIME ,
    AttrRefRSRC         NVARCHAR (100) ,
    H_ATTRIBUTE_SQN_Attribute NUMERIC (10) NOT NULL ,
    H_REFERENCE_SQN_Reference NUMERIC (10) NOT NULL
)

```

```

ON "default"
GO
ALTER TABLE L_ATTR_REF ADD CONSTRAINT L_ATTR_REF_PK PRIMARY KEY CLUSTERED (
SQN_AttrRef)
WITH
(
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE L_ATTR_SRC_ATTR
(
    SQN_AttrSrcAttr    NUMERIC (10) NOT NULL ,
    AttrSrcAttrLDTs    DATETIME ,
    AttrSrcAttrRSRC    NVARCHAR (100) ,
    H_ATTR_SQN_Attribute NUMERIC (10) NOT NULL ,
    H_SRC_ATTR_SQN_SrcAttr NUMERIC (10) NOT NULL ,
    H_TRNSF_SQN_Trnsf    NUMERIC (10) ,
    H_RULE_SQN_Rule    NUMERIC (10)
)
ON "default"
GO
ALTER TABLE L_ATTR_SRC_ATTR ADD CONSTRAINT L_ATTR_SRC_ATTR_PK PRIMARY KEY
CLUSTERED (SQN_AttrSrcAttr)
WITH
(
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE L_DOM_SRCATTR
(
    SQN_DomSrcAttr    NUMERIC (10) NOT NULL ,
    DomSrcAttrLDTs    DATETIME ,

```

```

DomSrcAttrRSRC    NVARCHAR (100) ,
H_DOMAIN_SQN_Domain  NUMERIC (10) NOT NULL ,
H_SRC_ATTR_SQN_SrcAttr NUMERIC (10) NOT NULL
)
ON "default"
GO
ALTER TABLE L_DOM_SRCATTR ADD CONSTRAINT L_DOM_SRCATTR_PK PRIMARY KEY CLUSTERED
(SQN_DomSrcAttr)
WITH
(
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE L_DSRC_REL
(
    SQN_DSrcRel        NUMERIC (10) NOT NULL ,
    DSrcRelLDTS        DATETIME ,
    DSrcRelRSRC        NVARCHAR (100) ,
    H_DATA_SOURCE_SQN_DataSource NUMERIC (10) NOT NULL ,
    H_SRC_RELATION_SQN_SrcRelation NUMERIC (10) NOT NULL
)
ON "default"
GO
ALTER TABLE L_DSRC_REL ADD CONSTRAINT L_DSRC_REL_PK PRIMARY KEY CLUSTERED (
SQN_DSrcRel)
WITH
(
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE L_DSRC_RSRC
(

```

```

SQN_DSrcRsrc          NUMERIC (10) NOT NULL ,
DSrcRsrcLDTS         DATETIME ,
DSrcRsrcRSRC         NVARCHAR (100) ,
H_DATA_SOURCE_SQN_DataSource NUMERIC (10) NOT NULL ,
H_RESOURCE_SQN_Resource  NUMERIC (10) NOT NULL
)
ON "default"
GO
ALTER TABLE L_DSRC_RSRC ADD CONSTRAINT L_DSRC_RSRC_PK PRIMARY KEY CLUSTERED (
SQN_DSrcRsrc)
WITH
(
  ALLOW_PAGE_LOCKS = ON ,
  ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE L_HUB_LINK
(
  SQN_HubLink  NUMERIC (10) NOT NULL ,
  HubLinkLDTS  DATETIME ,
  HubLinkRSRC  NVARCHAR (100) ,
  H_HUB_SQN_Hub  NUMERIC (10) NOT NULL ,
  H_LINK_SQN_Link NUMERIC (10) NOT NULL
)
ON "default"
GO
ALTER TABLE L_HUB_LINK ADD CONSTRAINT L_HUB_LINK_PK PRIMARY KEY CLUSTERED (
SQN_HubLink)
WITH
(
  ALLOW_PAGE_LOCKS = ON ,
  ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE

```

```

TABLE L_HUB_SAT
(
  SQN_HubSat      NUMERIC (10) NOT NULL ,
  HubSatLDTS     DATETIME ,
  HubSatRSRC     NVARCHAR (100) ,
  H_HUB_SQN_Hub  NUMERIC (10) NOT NULL ,
  H_SATELLITE_SQN_Satellite NUMERIC (10) NOT NULL
)
ON "default"
GO
ALTER TABLE L_HUB_SAT ADD CONSTRAINT L_HUB_SAT_PK PRIMARY KEY CLUSTERED (
SQN_HubSat)
WITH
(
  ALLOW_PAGE_LOCKS = ON ,
  ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE L_REF_RULE
(
  SQN_RefRule     NUMERIC (10) NOT NULL ,
  RefRuleLDTS    DATETIME ,
  RefRuleRSRC    NVARCHAR (100) ,
  H_REFERENCE_SQN_Reference NUMERIC (10) NOT NULL ,
  H_RULE_SQN_Rule  NUMERIC (10) NOT NULL
)
ON "default"
GO
ALTER TABLE L_REF_RULE ADD CONSTRAINT L_REF_RULE_PK PRIMARY KEY CLUSTERED (
SQN_RefRule)
WITH
(
  ALLOW_PAGE_LOCKS = ON ,
  ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

```

```

CREATE
TABLE L_RELATION_HSL
(
SQN_RelHSL      NUMERIC (10) NOT NULL ,
RelHSL_LDTS    DATETIME ,
RelHSL_RSRC    NVARCHAR (100) ,
H_RULE_SQN_Rule  NUMERIC (10) ,
H_TRNSF_SQN_Trnsf  NUMERIC (10) ,
H_HUB_SQN_Hub   NUMERIC (10) ,
H_LINK_SQN_Link  NUMERIC (10) ,
H_SAT_SQN_Satellite NUMERIC (10)
)
ON "default"
GO
ALTER TABLE L_RELATION_HSL ADD CONSTRAINT L_RELATION_HSL_PK PRIMARY KEY
CLUSTERED (SQN_RelHSL)
WITH
(
ALLOW_PAGE_LOCKS = ON ,
ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE L_RELSHP_LINK
(
SQN_RelshpLink    NUMERIC (10) NOT NULL ,
RelshpLinkLDTS    DATETIME ,
RelshpLinkRSRC    NVARCHAR (100) ,
H_SRC_RLSHP_SQN_SrcRlshp NUMERIC (10) NOT NULL ,
H_LINK_SQN_Link    NUMERIC (10) NOT NULL ,
H_RULE_SQN_Rule    NUMERIC (10) ,
H_TRNSF_SQN_Trnsf  NUMERIC (10)
)
ON "default"
GO
ALTER TABLE L_RELSHP_LINK ADD CONSTRAINT L_RELSHP_LINK_PK PRIMARY KEY CLUSTERED
(SQN_RelshpLink)

```

```

WITH
(
  ALLOW_PAGE_LOCKS = ON ,
  ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE L_SAT_ATTR
(
  SQN_SatAttr      NUMERIC (10) NOT NULL ,
  SatAttrLDTS     DATETIME ,
  SatAttrRSRC     NVARCHAR (100) ,
  H_ATTRIBUTE_SQN_Attribute NUMERIC (10) NOT NULL ,
  H_SATELLITE_SQN_Satellite NUMERIC (10) NOT NULL
)
ON "default"
GO
ALTER TABLE L_SAT_ATTR ADD CONSTRAINT L_SAT_ATTR_PK PRIMARY KEY CLUSTERED (
SQN_SatAttr)
WITH
(
  ALLOW_PAGE_LOCKS = ON ,
  ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE L_SAT_ATTR_REF
(
  SQN_SatAttrRef   NUMERIC (10) NOT NULL ,
  SatAttrRefLDTS   DATETIME ,
  SatAttrRefRSRC   NVARCHAR (100) ,
  H_SATELLITE_SQN_Satellite NUMERIC (10) ,
  H_ATTRIBUTE_SQN_Attribute NUMERIC (10) ,
  H_REFERENCE_SQN_Reference NUMERIC (10)
)
ON "default"

```



```

GO
ALTER TABLE L_SAT_ATTR_REF ADD CONSTRAINT L_SAT_ATTR_REF_PK PRIMARY KEY
CLUSTERED (SQN_SatAttrRef)
WITH
(
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE L_SAT_LINK
(
    SQN_SatLink      NUMERIC (10) NOT NULL ,
    SatLinkLDTS     DATETIME ,
    SatLinkRSRC     NVARCHAR (100) ,
    H_SATELLITE_SQN_Satellite NUMERIC (10) NOT NULL ,
    H_LINK_SQN_Link  NUMERIC (10) NOT NULL
)
ON "default"
GO
ALTER TABLE L_SAT_LINK ADD CONSTRAINT L_SAT_LINK_PK PRIMARY KEY CLUSTERED (
SQN_SatLink)
WITH
(
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE L_SRC_REL_ATTR
(
    SQN_SrcRelAttr   NUMERIC (10) NOT NULL ,
    SrcRelAttrLDTS   DATETIME ,
    SrcRelAttrRSRC   NVARCHAR (100) ,
    H_SRC_ATTR_SQN_SrcAttr NUMERIC (10) NOT NULL ,
    H_SRC_REL_SQN_SrcRel  NUMERIC (10) NOT NULL
)

```

```

)
ON "default"
GO
ALTER TABLE L_SRC_REL_ATTR ADD CONSTRAINT L_SRC_REL_ATTR_PK PRIMARY KEY
CLUSTERED (SQN_SrcRelAttr)
WITH
(
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE L_SRC_REL_ATTR_CONSTR
(
    SQN_RelAttrConstr    NUMERIC (10) NOT NULL ,
    RelAttrConstrLDTS    DATETIME ,
    RelAttrConstrRSRC    NVARCHAR (100) ,
    H_SRC_REL_SQN_SrcRel    NUMERIC (10) ,
    H_SRC_ATTR_SQN_SrcAttr    NUMERIC (10) ,
    H_CONSTR_SQN_Constraint    NUMERIC (10) ,
    H_SRC_RLSHP_SQN_SrcRelshp    NUMERIC (10)
)
ON "default"
GO
ALTER TABLE L_SRC_REL_ATTR_CONSTR ADD CONSTRAINT L_SRC_REL_ATTR_CONSTR_PK
PRIMARY KEY CLUSTERED (SQN_RelAttrConstr)
WITH
(
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE L_SRC_REL_RELSHP
(
    SQN_RelRelshp        NUMERIC (10) NOT NULL ,

```

```

RelRelshpLDTS      DATETIME ,
RelRelshpRSRC      NVARCHAR (100) ,
H_SRC_REL_SQN_SrcRel  NUMERIC (10) NOT NULL ,
H_SRC_RLSHP_SQN_SrcRlshp NUMERIC (10) NOT NULL
)
ON "default"
GO
ALTER TABLE L_SRC_REL_RELSHP ADD CONSTRAINT L_SRC_REL_RELSHP_PK PRIMARY KEY
CLUSTERED (SQN_RelRelshp)
WITH
(
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE SAL_MASTER_ATTR
(
    SQN_MasterAttr    NUMERIC (10) NOT NULL ,
    MasterAttrLDTS    DATETIME ,
    MasterAttrRSRC    NVARCHAR (100) ,
    H_ATTR_SQN_AttrBase NUMERIC (10) NOT NULL ,
    H_ATTR_SQN_AttrMaster NUMERIC (10) NOT NULL ,
    H_TRANSF_SQN_Transf NUMERIC (10) ,
    H_RULE_SQN_Rule    NUMERIC (10)
)
ON "default"
GO
ALTER TABLE SAL_MASTER_ATTR ADD CONSTRAINT SAL_MASTER_ATTR_PK PRIMARY KEY
CLUSTERED (SQN_MasterAttr)
WITH
(
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

```

```

CREATE
TABLE SAL_MASTER_HUB
(
SQN_MasterHub    NUMERIC (10) NOT NULL ,
MasterHubLDTS    DATETIME ,
MasterHubRSRC    NVARCHAR (100) ,
H_HUB_SQN_HubBase NUMERIC (10) NOT NULL ,
H_HUB_SQN_HubMaster NUMERIC (10) NOT NULL ,
H_RULE_SQN_Rule  NUMERIC (10) ,
H_TRANSF_SQN_Transf NUMERIC (10)
)
ON "default"
GO
ALTER TABLE SAL_MASTER_HUB ADD CONSTRAINT SAL_MASTER_HUB_PK PRIMARY KEY
CLUSTERED (SQN_MasterHub)
WITH
(
ALLOW_PAGE_LOCKS = ON ,
ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE SAL_MASTER_LINK
(
SQN_MasterLink    NUMERIC (10) NOT NULL ,
MasterLinkLDTS    DATETIME ,
MasterLinkRSRC    NVARCHAR (100) ,
H_LINK_SQN_LinkBas NUMERIC (10) NOT NULL ,
H_LINK_SQN_LinkMas NUMERIC (10) NOT NULL ,
H_TRANSF_SQN_Transf NUMERIC (10) ,
H_RULE_SQN_Rule  NUMERIC (10)
)
ON "default"
GO
ALTER TABLE SAL_MASTER_LINK ADD CONSTRAINT SAL_MASTER_LINK_PK PRIMARY KEY
CLUSTERED (SQN_MasterLink)
WITH
(

```

```

ALLOW_PAGE_LOCKS = ON ,
ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE SAL_MASTER_SAT
(
SQN_MasterSat    NUMERIC (10) NOT NULL ,
MasterSatLDTS    DATETIME ,
MasterSatRSRC    NVARCHAR (100) ,
H_SAT_SQN_SatBase NUMERIC (10) NOT NULL ,
H_SAT_SQN_SatMaster NUMERIC (10) NOT NULL ,
H_TRANSF_SQN_Transf NUMERIC (10) ,
H_RULE_SQN_Rule  NUMERIC (10)
)
ON "default"
GO
ALTER TABLE SAL_MASTER_SAT ADD CONSTRAINT SAL_MASTER_SAT_PK PRIMARY KEY
CLUSTERED (SQN_MasterSat)
WITH
(
ALLOW_PAGE_LOCKS = ON ,
ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE S_ATTR_DEF
(
AttrDefLDTS    DATETIME NOT NULL ,
AttrDefLEDTS   DATETIME ,
AttrDefRSRC    NVARCHAR (100) ,
AttrName       NVARCHAR (100) ,
AttrDescription NVARCHAR (1000) ,
AttrSize       NVARCHAR (30) ,
AttrNullable   BIT ,
AttrAllowedValues NVARCHAR (1000) ,

```

```

AttrDefaultValue      NVARCHAR (1000) ,
H_ATTRIBUTE_SQN_Attribute NUMERIC (10) NOT NULL ,
ChangeType           NVARCHAR (100)
)
ON "default"
GO
ALTER TABLE S_ATTR_DEF ADD CONSTRAINT S_ATTR_DEF_PK PRIMARY KEY CLUSTERED (
AttrDefLDTS, H_ATTRIBUTE_SQN_Attribute)
WITH
(
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE S_ATTR_ISSUE
(
    IssueLDTS      DATETIME NOT NULL ,
    IssueLEDTS     DATETIME ,
    IssueRSRC      NVARCHAR (100) ,
    IssueNumber    NVARCHAR (10) ,
    IssueName      NVARCHAR (100) ,
    IssueDescription NVARCHAR (1000) ,
    IssuePriority  NVARCHAR (50) ,
    IssueResolution NVARCHAR (1000) ,
    IssueStatus    NVARCHAR (50) ,
    L_AT_SRC_AT_SQN_ASA NUMERIC (10) NOT NULL
)
ON "default"
GO
ALTER TABLE S_ATTR_ISSUE ADD CONSTRAINT S_ATTR_ISSUE_PK PRIMARY KEY CLUSTERED (
IssueLDTS, L_AT_SRC_AT_SQN_ASA)
WITH
(
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON
)
ON "default"

```

```

GO

CREATE
TABLE S_BUSINESS_KEY
(
    BusinessKeyLDTS    DATETIME NOT NULL ,
    BusinessKeyLEDTS   DATETIME ,
    BusinessKeyRSRC    NVARCHAR (100) ,
    BusinessKey        NVARCHAR (100) ,
    BusinessKeyDescription NVARCHAR (1000) ,
    SourceKey          NVARCHAR (100) ,
    SurogateKey        NVARCHAR (100) ,
    H_HUB_SQN_Hub      NUMERIC (10) NOT NULL
)
ON "default"
GO
ALTER TABLE S_BUSINESS_KEY ADD CONSTRAINT S_BUSINESS_KEY_PK PRIMARY KEY
CLUSTERED (BusinessKeyLDTS, H_HUB_SQN_Hub)
WITH
(
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE S_CODE_ATTR
(
    MasterCodeLDTS      DATETIME NOT NULL ,
    MasterCodeLEDTS     DATETIME ,
    MasterCodeRSRC      NVARCHAR (100) ,
    MasterCodeName      NVARCHAR (100) ,
    MasterCodeDescription NVARCHAR (1000) ,
    MasterCode          NVARCHAR (7000) ,
    SAL_MASTER_ATTR_SQN_MasterAttr NUMERIC (10) NOT NULL
)
ON "default"
GO
ALTER TABLE S_CODE_ATTR ADD CONSTRAINT S_CODE_ATTR_PK PRIMARY KEY CLUSTERED (

```

```

MasterCodeLDTS, SAL_MASTER_ATTR_SQN_MasterAttr)
WITH
(
  ALLOW_PAGE_LOCKS = ON ,
  ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE S_CODE_HUB
(
  MasterCodeLDTS      DATETIME NOT NULL ,
  MasterCodeLEDTS     DATETIME ,
  MasterCodeRSRC      NVARCHAR (100) ,
  MasterCodeName      NVARCHAR (100) ,
  MasterCodeDescription NVARCHAR (1000) ,
  MasterCode          NVARCHAR (7000) ,
  SAL_MASTER_HUB_SQN_MasterHub NUMERIC (10) NOT NULL
)
ON "default"
GO
ALTER TABLE S_CODE_HUB ADD CONSTRAINT S_CODE_HUB_PK PRIMARY KEY CLUSTERED (
MasterCodeLDTS, SAL_MASTER_HUB_SQN_MasterHub)
WITH
(
  ALLOW_PAGE_LOCKS = ON ,
  ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE S_CODE_LINK
(
  MasterCodeLDTS      DATETIME NOT NULL ,
  MasterCodeLEDTS     DATETIME ,
  MasterCodeRSRC      NVARCHAR (100) ,
  MasterCodeName      NVARCHAR (100) ,
  MasterCodeDescription NVARCHAR (1000) ,

```



```

MasterCode          NVARCHAR (7000) ,
SAL_MASTER_LINK_SQN_MasterLink NUMERIC (10) NOT NULL
)
ON "default"
GO
ALTER TABLE S_CODE_LINK ADD CONSTRAINT S_CODE_LINK_PK PRIMARY KEY CLUSTERED (
MasterCodeLDTs, SAL_MASTER_LINK_SQN_MasterLink)
WITH
(
ALLOW_PAGE_LOCKS = ON ,
ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE S_CODE_SAT
(
MasterCodeLDTs      DATETIME NOT NULL ,
MasterCodeLEDTs     DATETIME ,
MasterCodeRSRC      NVARCHAR (100) ,
MasterCodeName      NVARCHAR (100) ,
MasterCodeDescription NVARCHAR (1000) ,
MasterCode          NVARCHAR (7000) ,
SAL_MASTER_SAT_SQN_MasterSat NUMERIC (10) NOT NULL
)
ON "default"
GO
ALTER TABLE S_CODE_SAT ADD CONSTRAINT S_CODE_SAT_PK PRIMARY KEY CLUSTERED (
MasterCodeLDTs, SAL_MASTER_SAT_SQN_MasterSat)
WITH
(
ALLOW_PAGE_LOCKS = ON ,
ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE S_CONSTRAINT

```

```

(
  ConstraintLDTS      DATETIME NOT NULL ,
  ConstraintLEDTS     DATETIME ,
  ConstraintRSRC      NVARCHAR (100) ,
  ConstraintName      NVARCHAR (100) ,
  ConstraintType      NVARCHAR (100) ,
  ConstraintDescription NVARCHAR (1000) ,
  H_CONSTRAINT_SQN_Constraint NUMERIC (10) NOT NULL
)
ON "default"
GO
ALTER TABLE S_CONSTRAINT ADD CONSTRAINT S_CONSTRAINT_PK PRIMARY KEY CLUSTERED (
ConstraintLDTS, H_CONSTRAINT_SQN_Constraint)
WITH
(
  ALLOW_PAGE_LOCKS = ON ,
  ALLOW_ROW_LOCKS = ON
)
ON "default"
GO
CREATE
TABLE S_DATA_SOURCE
(
  DSrcLDTS          DATETIME NOT NULL ,
  DSrcLEDTS         DATETIME ,
  DSrcRSRC          NVARCHAR (100) ,
  DSrcName          NVARCHAR (100) ,
  DSrcType          NVARCHAR (100) ,
  DSrcDescription   NVARCHAR (1000) ,
  DSrcBusinessDescription NVARCHAR (1000) ,
  DSrcQualityType   NVARCHAR (100) ,
  DSrcSchemaName    NVARCHAR (200) ,
  DSrcChangeType    NVARCHAR (100) ,
  H_DATA_SOURCE_SQN_DataSource NUMERIC (10) NOT NULL
)
ON "default"
GO
ALTER TABLE S_DATA_SOURCE ADD CONSTRAINT S_DATA_SOURCE_PK PRIMARY KEY CLUSTERED
(DSrcLDTS, H_DATA_SOURCE_SQN_DataSource)

```

```

WITH
(
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE S_DOMAIN
(
    DomainLDTS    DATETIME NOT NULL ,
    DomainLEDTS   DATETIME ,
    DomainRSRC    NVARCHAR (100) ,
    DomainDataType NVARCHAR (100) ,
    DomainDescription NVARCHAR (1000) ,
    H_DOMAIN_SQN_Domain NUMERIC (10) NOT NULL
)
ON "default"
GO

ALTER TABLE S_DOMAIN ADD CONSTRAINT S_DOMAIN_PK PRIMARY KEY CLUSTERED (
DomainLDTS, H_DOMAIN_SQN_Domain)
WITH
(
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE S_HSL_ISSUE
(
    IssueLDTS      DATETIME NOT NULL ,
    IssueLEDTS     DATETIME ,
    IssueRSRC      NVARCHAR (100) ,
    IssueNumber    NVARCHAR (10) ,
    IssueName      NVARCHAR (100) ,
    IssueDescription NVARCHAR (1000) ,
    IssuePriority  NVARCHAR (50) ,

```

```

IssueResolution    NVARCHAR (1000) ,
IssueStatus        NVARCHAR (50) ,
L_RELATION_HSL_SQN_ReIHSL NUMERIC (10) NOT NULL
)
ON "default"
GO
ALTER TABLE S_HSL_ISSUE ADD CONSTRAINT S_HSL_ISSUE_PK PRIMARY KEY CLUSTERED (
IssueLDTS, L_RELATION_HSL_SQN_ReIHSL)
WITH
(
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE S_HUB_DEF
(
    HubDefLDTS    DATETIME NOT NULL ,
    HubDefLEDTS   DATETIME ,
    HubDefRSRC    NVARCHAR (100) ,
    HubName       NVARCHAR (100) ,
    HubDescription NVARCHAR (1000) ,
    HubType       NVARCHAR (100) ,
    HubSchemaName NVARCHAR (200) ,
    H_HUB_SQN_Hub NUMERIC (10) NOT NULL ,
    ChangeType    NVARCHAR (100)
)
ON "default"
GO
ALTER TABLE S_HUB_DEF ADD CONSTRAINT S_HUB_DEF_PK PRIMARY KEY CLUSTERED (
HubDefLDTS, H_HUB_SQN_Hub)
WITH
(
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

```

```

CREATE
TABLE S_LINK_DEF
(
    LinkDefLDTS    DATETIME NOT NULL ,
    LinkDefLEDTS   DATETIME ,
    LinkDefRSRC    NVARCHAR (100) ,
    LinkName       NVARCHAR (100) ,
    LinkDescription NVARCHAR (1000) ,
    LinkType       NVARCHAR (100) ,
    LinkSchemaName NVARCHAR (200) ,
    H_LINK_SQN_Link NUMERIC (10) NOT NULL ,
    ChangeType     NVARCHAR (100)
)
ON "default"
GO
ALTER TABLE S_LINK_DEF ADD CONSTRAINT S_LINK_DEF_PK PRIMARY KEY CLUSTERED (
LinkDefLDTS, H_LINK_SQN_Link)
WITH
(
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE S_LINK_ISSUE
(
    IssueLDTS      DATETIME NOT NULL ,
    IssueLEDTS     DATETIME ,
    IssueRSRC      NVARCHAR (100) ,
    IssueNumber    NVARCHAR (10) ,
    IssueName      NVARCHAR (100) ,
    IssueDescription NVARCHAR (1000) ,
    IssuePriority   NVARCHAR (50) ,
    IssueResolution NVARCHAR (1000) ,
    IssueStatus    NVARCHAR (50) ,
    L_RELSHP_LINK_SQN_RelshpLink NUMERIC (10) NOT NULL
)

```

```

ON "default"
GO
ALTER TABLE S_LINK_ISSUE ADD CONSTRAINT S_LINK_ISSUE_PK PRIMARY KEY CLUSTERED (
IssueLDTS, L_RELSHP_LINK_SQN_RelshpLink)
WITH
(
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE S_REF_DEF
(
    RefDefLDTS          DATETIME NOT NULL ,
    RefDefLEDTS        DATETIME ,
    RefDefRSRC          NVARCHAR (100) ,
    RefName             NVARCHAR (100) ,
    RefDescription      NVARCHAR (1000) ,
    RefType             NVARCHAR (100) ,
    RefSchemaName      NVARCHAR (200) ,
    H_REFERENCE_SQN_Reference NUMERIC (10) NOT NULL ,
    ChangeType         NVARCHAR (100)
)
ON "default"
GO
ALTER TABLE S_REF_DEF ADD CONSTRAINT S_REF_DEF_PK PRIMARY KEY CLUSTERED (
RefDefLDTS, H_REFERENCE_SQN_Reference)
WITH
(
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE S_RESOURCE
(

```

```

ResourceLDTS      DATETIME NOT NULL ,
ResourceLEDTS     DATETIME ,
ResourceRSRC      NVARCHAR (100) ,
ResourceName      NVARCHAR (100) ,
ResourceDescription NVARCHAR (1000) ,
ResourceSoftware  NVARCHAR (500) ,
ResourceHardware  NVARCHAR (500) ,
H_RESOURCE_SQN_Resource NUMERIC (10) NOT NULL
)
ON "default"
GO
ALTER TABLE S_RESOURCE ADD CONSTRAINT S_RESOURCE_PK PRIMARY KEY CLUSTERED (
ResourceLDTS, H_RESOURCE_SQN_Resource)
WITH
(
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE S_ROLE_DEF
(
    RoleDefLDTS      DATETIME NOT NULL ,
    RoleDefLEDTS     DATETIME ,
    RoleDefRSRC      NVARCHAR (100) ,
    RoleName         NVARCHAR (100) ,
    RoleDescription  NVARCHAR (1000) ,
    L_HUB_LINK_SQN_HubLink NUMERIC (10) NOT NULL
)
ON "default"
GO
ALTER TABLE S_ROLE_DEF ADD CONSTRAINT S_ROLE_DEF_PK PRIMARY KEY CLUSTERED (
RoleDefLDTS, L_HUB_LINK_SQN_HubLink)
WITH
(
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON
)

```

```

ON "default"
GO

CREATE
TABLE S_RULE_CODE
(
RuleCodeLDTS    DATETIME NOT NULL ,
RuleCodeLEDTS  DATETIME ,
RuleCodeRSRC   NVARCHAR (100) ,
RuleCodeName   NVARCHAR (100) ,
RuleCodeDescription NVARCHAR (1000) ,
RuleCode       NVARCHAR (7000) ,
H_RULE_SQN_Rule NUMERIC (10) NOT NULL
)
ON "default"
GO
ALTER TABLE S_RULE_CODE ADD CONSTRAINT S_RULE_CODE_PK PRIMARY KEY CLUSTERED (
RuleCodeLDTS, H_RULE_SQN_Rule)
WITH
(
ALLOW_PAGE_LOCKS = ON ,
ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE S_RULE_DEF
(
RuleDefLDTS    DATETIME NOT NULL ,
RuleDefLEDTS  DATETIME ,
RuleDefRSRC   NVARCHAR (100) ,
RuleName      NVARCHAR (100) ,
RuleShortDescription NVARCHAR (500) ,
RuleTextDescription NVARCHAR (1000) ,
RuleType      NVARCHAR (100) ,
RuleCategory  NVARCHAR (100) ,
H_RULE_SQN_Rule NUMERIC (10) NOT NULL
)
ON "default"

```



```

GO
ALTER TABLE S_RULE_DEF ADD CONSTRAINT S_RULE_DEF_PK PRIMARY KEY CLUSTERED (
RuleDefLDTS, H_RULE_SQN_Rule)
WITH
(
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE S_RULE_RELSHP
(
    RuleRelshpLDTS    DATETIME NOT NULL ,
    RuleRelshpLEDTS   DATETIME ,
    RuleRelshpRSRC    NVARCHAR (100) ,
    RuleRelshpType     NVARCHAR (100) ,
    RuleRelshpDescription NVARCHAR (1000) ,
    HL_RUL_RLSHP_SQNRuRlshp NUMERIC (10) NOT NULL
)
ON "default"
GO
ALTER TABLE S_RULE_RELSHP ADD CONSTRAINT S_RULE_RELSHP_PK PRIMARY KEY CLUSTERED
(RuleRelshpLDTS, HL_RUL_RLSHP_SQNRuRlshp)
WITH
(
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE S_SAT_DEF
(
    SatDefLDTS    DATETIME NOT NULL ,
    SatDefLEDTS   DATETIME ,
    SatDefRSRC    NVARCHAR (100) ,
    SatName       NVARCHAR (100) ,

```

```

SatDescription NVARCHAR (1000) ,
SatType      NVARCHAR (100) ,
SatSchemaName NVARCHAR (200) ,
H_SAT_SQN_Sat NUMERIC (10) NOT NULL ,
ChangeType   NVARCHAR (100)
)
ON "default"
GO
ALTER TABLE S_SAT_DEF ADD CONSTRAINT S_SAT_DEF_PK PRIMARY KEY CLUSTERED (
SatDefLDTS, H_SAT_SQN_Sat)
WITH
(
ALLOW_PAGE_LOCKS = ON ,
ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE S_SDV_CODE_ATTR
(
SDVCodeLDTS    DATETIME NOT NULL ,
SDVCodeLEDTS   DATETIME ,
SDVCodeRSRC    NVARCHAR (100) ,
SDVCodeName    NVARCHAR (100) ,
SDVCodeDescription NVARCHAR (1000) ,
SDVCode        NVARCHAR (7000) ,
L_AT_SRCAT_SQN_ASA NUMERIC (10) NOT NULL
)
ON "default"
GO
ALTER TABLE S_SDV_CODE_ATTR ADD CONSTRAINT S_SDV_CODE_ATTR_PK PRIMARY KEY
CLUSTERED (SDVCodeLDTS, L_AT_SRCAT_SQN_ASA)
WITH
(
ALLOW_PAGE_LOCKS = ON ,
ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

```

```

CREATE
TABLE S_SDV_CODE_HSL
(
SDVCodeLDTS    DATETIME NOT NULL ,
SDVCodeLEDTS   DATETIME ,
SDVCodeRSRC    NVARCHAR (100) ,
SDVCodeName    NVARCHAR (100) ,
SDVCodeDescription NVARCHAR (1000) ,
SDVCode        NVARCHAR (7000) ,
L_REL_HSL_SQN_ReIHSL NUMERIC (10) NOT NULL
)
ON "default"
GO
ALTER TABLE S_SDV_CODE_HSL ADD CONSTRAINT S_SDV_CODE_HSL_PK PRIMARY KEY
CLUSTERED (SDVCodeLDTS, L_REL_HSL_SQN_ReIHSL)
WITH
(
ALLOW_PAGE_LOCKS = ON ,
ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE S_SDV_CODE_LINK
(
SDVCodeLDTS    DATETIME NOT NULL ,
SDVCodeLEDTS   DATETIME ,
SDVCodeRSRC    NVARCHAR (100) ,
SDVCodeName    NVARCHAR (100) ,
SDVCodeDescription NVARCHAR (1000) ,
SDVCode        NVARCHAR (7000) ,
L_RLSHP_LINK_SQN_RlshpLink NUMERIC (10) NOT NULL
)
ON "default"
GO
ALTER TABLE S_SDV_CODE_LINK ADD CONSTRAINT S_SDV_CODE_LINK_PK PRIMARY KEY
CLUSTERED (SDVCodeLDTS, L_RLSHP_LINK_SQN_RlshpLink)
WITH

```

```

(
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE S_SRC_ATTRIBUTE
(
    SrcAttrLDTS    DATETIME NOT NULL ,
    SrcAttrLEDTS   DATETIME ,
    SrcAttrRSRC    NVARCHAR (100) ,
    SrcAttrPhysicalName NVARCHAR (100) ,
    SrcAttrBusinessName NVARCHAR (100) ,
    SrcAttrDescription NVARCHAR (1000) ,
    SrcAttrSize    NVARCHAR (30) ,
    SrcAttrNullable BIT ,
    SrcAttrAllowedValues NVARCHAR (1000) ,
    SrcAttrDefaultValue NVARCHAR (1000) ,
    SrcAttrChangeType NVARCHAR (100) ,
    H_SRC_ATTR_SQN_SrcAttr NUMERIC (10) NOT NULL
)
ON "default"
GO
ALTER TABLE S_SRC_ATTRIBUTE ADD CONSTRAINT S_SRC_ATTRIBUTE_PK PRIMARY KEY
CLUSTERED (SrcAttrLDTS, H_SRC_ATTR_SQN_SrcAttr)
WITH
(
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE S_SRC_RELATION
(
    SrcRelationLDTS    DATETIME NOT NULL ,
    SrcRelationLEDTS   DATETIME ,

```

```

SrcRelationRSRC      NVARCHAR (100) ,
SrcRelationName      NVARCHAR (100) ,
SrcRelationType      NVARCHAR (100) ,
SrcRelationDescription NVARCHAR (1000) ,
SrcRelBusinessDescription NVARCHAR (1000) ,
SrcRelationQualityType NVARCHAR (100) ,
SrcRelationSchemaName NVARCHAR (200) ,
H_SRC_REL_SQN_SrcRel  NUMERIC (10) NOT NULL ,
ChangeType          NVARCHAR (100)
)
ON "default"
GO
ALTER TABLE S_SRC_RELATION ADD CONSTRAINT S_SRC_RELATION_PK PRIMARY KEY
CLUSTERED (SrcRelationLDTS, H_SRC_REL_SQN_SrcRel)
WITH
(
  ALLOW_PAGE_LOCKS = ON ,
  ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE S_SRC_RELATIONSHIP
(
  SrcRelshpLDTS      DATETIME NOT NULL ,
  SrcRelshpLEDTS     DATETIME ,
  SrcRelshpRSRC      NVARCHAR (100) ,
  SrcRelshpName      NVARCHAR (100) ,
  SrcRelshpDescription NVARCHAR (1000) ,
  H_SRCRLSHP_SQNSrcRelshp NUMERIC (10) NOT NULL
)
ON "default"
GO
ALTER TABLE S_SRC_RELATIONSHIP ADD CONSTRAINT S_SRC_RELATIONSHIP_PK PRIMARY KEY
CLUSTERED (SrcRelshpLDTS, H_SRCRLSHP_SQNSrcRelshp)
WITH
(
  ALLOW_PAGE_LOCKS = ON ,
  ALLOW_ROW_LOCKS = ON
)

```

```

)
ON "default"
GO

CREATE
TABLE S_SRC_ROLE
(
SrcRoleLDTS          DATETIME NOT NULL ,
SrcRoleLEDTS         DATETIME ,
SrcRoleRSRC          NVARCHAR (100) ,
SrcRoleName          NVARCHAR (100) ,
SrcRoleDescription   NVARCHAR (1000) ,
L_SRC_REL_RELSHP_SQN_RelRelshp NUMERIC (10) NOT NULL
)
ON "default"
GO
ALTER TABLE S_SRC_ROLE ADD CONSTRAINT S_SRC_ROLE_PK PRIMARY KEY CLUSTERED (
SrcRoleLDTS, L_SRC_REL_RELSHP_SQN_RelRelshp)
WITH
(
ALLOW_PAGE_LOCKS = ON ,
ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE S_TRANSFORMATION
(
TransformationLDTS   DATETIME NOT NULL ,
TransformationLEDTS  DATETIME ,
TransformationRSRC   NVARCHAR (100) ,
TransformationName   NVARCHAR (100) ,
TransformationDescription NVARCHAR (1000) ,
TransformationType   NVARCHAR (100) ,
TransformationSoftware NVARCHAR (100) ,
ETLRepository        NVARCHAR (1000) ,
H_TRANSF_SQN_Transf  NUMERIC (10) NOT NULL
)
ON "default"

```

```

GO
ALTER TABLE S_TRANSFORMATION ADD CONSTRAINT S_TRANSFORMATION_PK PRIMARY KEY
CLUSTERED (TransformationLDTS, H_TRANSF_SQN_Transf)
WITH
(
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

CREATE
TABLE S_TRANSF_RELSHP
(
    TransfRelshpLDTS    DATETIME NOT NULL ,
    TransfRelshpLEDTS   DATETIME ,
    TransfRelshpRSRC    NVARCHAR (100) ,
    TransfRelshpType    NVARCHAR (100) ,
    TransfRelshpDescription  NVARCHAR (1000) ,
    HL_TRSF_RLSHP_SQN_TrshRlshp NUMERIC (10) NOT NULL
)
ON "default"
GO

ALTER TABLE S_TRANSF_RELSHP ADD CONSTRAINT S_TRANSF_RELSHP_PK PRIMARY KEY
CLUSTERED (TransfRelshpLDTS, HL_TRSF_RLSHP_SQN_TrshRlshp)
WITH
(
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON
)
ON "default"
GO

ALTER TABLE HL_RULE_RELSHP
ADD CONSTRAINT HL_RULE_RELSHP_RULE_FK FOREIGN KEY
(
    H_RULE_SQN_Rule
)
REFERENCES H_RULE
(

```

```

SQN_Rule
)

ALTER TABLE HL_RULE_RELSHP
ADD CONSTRAINT HL_RULE_RELSHP_RULE_FKv1 FOREIGN KEY
(
H_RULE_SQN_Rule1
)
REFERENCES H_RULE
(
SQN_Rule
)

ALTER TABLE HL_TRANSF_RELSHP
ADD CONSTRAINT HL_TRANSF_RELSHP_TRNSF_FK FOREIGN KEY
(
H_TRNSF_SQN_TrnsfBase
)
REFERENCES H_TRANSFORMATION
(
SQN_Transformation
)

ALTER TABLE HL_TRANSF_RELSHP
ADD CONSTRAINT HL_TRANSF_RELSHP_TRNSF_FKv1 FOREIGN KEY
(
H_TRNSF_SQN_TrnsfMaster
)
REFERENCES H_TRANSFORMATION
(
SQN_Transformation
)

ALTER TABLE L_ATTR_DOM
ADD CONSTRAINT L_ATTR_DOM_ATTRIBUTE_FK FOREIGN KEY
(
H_ATTRIBUTE_SQN_Attribute
)
REFERENCES H_ATTRIBUTE
(
SQN_Attribute
)

ALTER TABLE L_ATTR_DOM

```



```

ADD CONSTRAINT L_ATTR_DOM_DOMAIN_FK FOREIGN KEY
(
H_DOMAIN_SQN_Domain
)
REFERENCES H_DOMAIN
(
SQN_Domain
)
ALTER TABLE L_ATTR_DOM
ADD CONSTRAINT L_ATTR_DOM_RULE_FK FOREIGN KEY
(
H_RULE_SQN_Rule
)
REFERENCES H_RULE
(
SQN_Rule
)
ALTER TABLE L_ATTR_REF
ADD CONSTRAINT L_ATTR_REF_ATTRIBUTE_FK FOREIGN KEY
(
H_ATTRIBUTE_SQN_Attribute
)
REFERENCES H_ATTRIBUTE
(
SQN_Attribute
)
ALTER TABLE L_ATTR_REF
ADD CONSTRAINT L_ATTR_REF_REFERENCE_FK FOREIGN KEY
(
H_REFERENCE_SQN_Reference
)
REFERENCES H_REFERENCE
(
SQN_Reference
)
ALTER TABLE L_ATTR_SRC_ATTR
ADD CONSTRAINT L_ATTR_SRC_ATTR_ATTR_FK FOREIGN KEY
(
H_ATTR_SQN_Attribute
)

```

```

REFERENCES H_ATTRIBUTE
(
SQN_Attribute
)
ALTER TABLE L_ATTR_SRC_ATTR
ADD CONSTRAINT L_ATTR_SRC_ATTR_RULE_FK FOREIGN KEY
(
H_RULE_SQN_Rule
)
REFERENCES H_RULE
(
SQN_Rule
)
ALTER TABLE L_ATTR_SRC_ATTR
ADD CONSTRAINT L_ATTR_SRC_ATTR_SRC_ATTR_FK FOREIGN KEY
(
H_SRC_ATTR_SQN_SrcAttr
)
REFERENCES H_SRC_ATTRIBUTE
(
SQN_SrcAttr
)
ALTER TABLE L_ATTR_SRC_ATTR
ADD CONSTRAINT L_ATTR_SRC_ATTR_TRNSF_FK FOREIGN KEY
(
H_TRNSF_SQN_Trnsf
)
REFERENCES H_TRANSFORMATION
(
SQN_Transformation
)
ALTER TABLE L_DOM_SRCATTR
ADD CONSTRAINT L_DOM_SRCATTR_DOMAIN_FK FOREIGN KEY
(
H_DOMAIN_SQN_Domain
)
REFERENCES H_DOMAIN
(
SQN_Domain
)

```

```

ALTER TABLE L_DOM_SRCATTR
ADD CONSTRAINT L_DOM_SRCATTR_SRCATTR_FK FOREIGN KEY
(
H_SRC_ATTR_SQN_SrcAttr
)
REFERENCES H_SRC_ATTRIBUTE
(
SQN_SrcAttr
)
ALTER TABLE L_DSRC_REL
ADD CONSTRAINT L_DSRC_REL_DATA_SOURCE_FK FOREIGN KEY
(
H_DATA_SOURCE_SQN_DataSource
)
REFERENCES H_DATA_SOURCE
(
SQN_DataSource
)
ALTER TABLE L_DSRC_REL
ADD CONSTRAINT L_DSRC_REL_SRC_RELATION_FK FOREIGN KEY
(
H_SRC_RELATION_SQN_SrcRelation
)
REFERENCES H_SRC_RELATION
(
SQN_SrcRelation
)
ALTER TABLE L_DSRC_RSRC
ADD CONSTRAINT L_DSRC_RSRC_DATA_SOURCE_FK FOREIGN KEY
(
H_DATA_SOURCE_SQN_DataSource
)
REFERENCES H_DATA_SOURCE
(
SQN_DataSource
)
ALTER TABLE L_DSRC_RSRC
ADD CONSTRAINT L_DSRC_RSRC_RESOURCE_FK FOREIGN KEY
(
H_RESOURCE_SQN_Resource

```

```

)
REFERENCES H_RESOURCE
(
SQN_Resource
)
ALTER TABLE L_HUB_LINK
ADD CONSTRAINT L_HUB_LINK_HUB_FK FOREIGN KEY
(
H_HUB_SQN_Hub
)
REFERENCES H_HUB
(
SQN_Hub
)
ALTER TABLE L_HUB_LINK
ADD CONSTRAINT L_HUB_LINK_LINK_FK FOREIGN KEY
(
H_LINK_SQN_Link
)
REFERENCES H_LINK
(
SQN_Link
)
ALTER TABLE L_HUB_SAT
ADD CONSTRAINT L_HUB_SAT_HUB_FK FOREIGN KEY
(
H_HUB_SQN_Hub
)
REFERENCES H_HUB
(
SQN_Hub
)
ALTER TABLE L_HUB_SAT
ADD CONSTRAINT L_HUB_SAT_SATELLITE_FK FOREIGN KEY
(
H_SATELLITE_SQN_Satellite
)
REFERENCES H_SATELLITE
(
SQN_Satellite

```

```

)
ALTER TABLE L_REF_RULE
ADD CONSTRAINT L_REF_RULE_REFERENCE_FK FOREIGN KEY
(
H_REFERENCE_SQN_Reference
)
REFERENCES H_REFERENCE
(
SQN_Reference
)
ALTER TABLE L_REF_RULE
ADD CONSTRAINT L_REF_RULE_RULE_FK FOREIGN KEY
(
H_RULE_SQN_Rule
)
REFERENCES H_RULE
(
SQN_Rule
)
ALTER TABLE L_RELSHP_LINK
ADD CONSTRAINT L_RELSHP_LINK_LINK_FK FOREIGN KEY
(
H_LINK_SQN_Link
)
REFERENCES H_LINK
(
SQN_Link
)
ALTER TABLE L_RELSHP_LINK
ADD CONSTRAINT L_RELSHP_LINK_RULE_FK FOREIGN KEY
(
H_RULE_SQN_Rule
)
REFERENCES H_RULE
(
SQN_Rule
)
ALTER TABLE L_RELATION_HSL
ADD CONSTRAINT L_REL_HSL_HUB_FK FOREIGN KEY
(

```

```

H_HUB_SQN_Hub
)
REFERENCES H_HUB
(
SQN_Hub
)
ALTER TABLE L_RELATION_HSL
ADD CONSTRAINT L_REL_HSL_LINK_FK FOREIGN KEY
(
H_LINK_SQN_Link
)
REFERENCES H_LINK
(
SQN_Link
)
ALTER TABLE L_RELATION_HSL
ADD CONSTRAINT L_REL_HSL_RULE_FK FOREIGN KEY
(
H_RULE_SQN_Rule
)
REFERENCES H_RULE
(
SQN_Rule
)
ALTER TABLE L_RELATION_HSL
ADD CONSTRAINT L_REL_HSL_SATELLITE_FK FOREIGN KEY
(
H_SAT_SQN_Satellite
)
REFERENCES H_SATELLITE
(
SQN_Satellite
)
ALTER TABLE L_RELATION_HSL
ADD CONSTRAINT L_REL_HSL_TRNSF_FK FOREIGN KEY
(
H_TRNSF_SQN_Trnsf
)
REFERENCES H_TRANSFORMATION
(

```

```

SQN_Transformation
)
ALTER TABLE L_RELSHP_LINK
ADD CONSTRAINT L_RLSHP_LINK_SRC_RLSHP_FK FOREIGN KEY
(
H_SRC_RLSHP_SQN_SrcRlshp
)
REFERENCES H_SRC_RELATIONSHIP
(
SQN_SrcRelshp
)
ALTER TABLE L_RELSHP_LINK
ADD CONSTRAINT L_RLSHP_LINK_TRNSF_FK FOREIGN KEY
(
H_TRNSF_SQN_Trnsf
)
REFERENCES H_TRANSFORMATION
(
SQN_Transformation
)
ALTER TABLE L_SAT_ATTR
ADD CONSTRAINT L_SAT_ATTR_ATTRIBUTE_FK FOREIGN KEY
(
H_ATTRIBUTE_SQN_Attribute
)
REFERENCES H_ATTRIBUTE
(
SQN_Attribute
)
ALTER TABLE L_SAT_ATTR
ADD CONSTRAINT L_SAT_ATTR_SATELLITE_FK FOREIGN KEY
(
H_SATELLITE_SQN_Satellite
)
REFERENCES H_SATELLITE
(
SQN_Satellite
)
ALTER TABLE L_SAT_ATTR_REF
ADD CONSTRAINT L_SAT_ATTR_REF_ATTRIBUTE_FK FOREIGN KEY

```

```

(
H_ATTRIBUTE_SQN_Attribute
)
REFERENCES H_ATTRIBUTE
(
SQN_Attribute
)
ALTER TABLE L_SAT_ATTR_REF
ADD CONSTRAINT L_SAT_ATTR_REF_REFERENCE_FK FOREIGN KEY
(
H_REFERENCE_SQN_Reference
)
REFERENCES H_REFERENCE
(
SQN_Reference
)
ALTER TABLE L_SAT_ATTR_REF
ADD CONSTRAINT L_SAT_ATTR_REF_SATELLITE_FK FOREIGN KEY
(
H_SATELLITE_SQN_Satellite
)
REFERENCES H_SATELLITE
(
SQN_Satellite
)
ALTER TABLE L_SAT_LINK
ADD CONSTRAINT L_SAT_LINK_LINK_FK FOREIGN KEY
(
H_LINK_SQN_Link
)
REFERENCES H_LINK
(
SQN_Link
)
ALTER TABLE L_SAT_LINK
ADD CONSTRAINT L_SAT_LINK_SATELLITE_FK FOREIGN KEY
(
H_SATELLITE_SQN_Satellite
)
REFERENCES H_SATELLITE

```



```

(
SQN_Satellite
)
ALTER TABLE L_SRC_REL_ATTR_CONSTR
ADD CONSTRAINT L_SRCREL_AT_CON_CON_FK FOREIGN KEY
(
H_CONSTR_SQN_Constraint
)
REFERENCES H_CONSTRAINT
(
SQN_Constraint
)
ALTER TABLE L_SRC_REL_ATTR_CONSTR
ADD CONSTRAINT L_SRCREL_AT_CON_SRCAT_FK FOREIGN KEY
(
H_SRC_ATTR_SQN_SrcAttr
)
REFERENCES H_SRC_ATTRIBUTE
(
SQN_SrcAttr
)
ALTER TABLE L_SRC_REL_ATTR_CONSTR
ADD CONSTRAINT L_SRCREL_AT_CON_SRCREL_FK FOREIGN KEY
(
H_SRC_REL_SQN_SrcRel
)
REFERENCES H_SRC_RELATION
(
SQN_SrcRelation
)
ALTER TABLE L_SRC_REL_ATTR_CONSTR
ADD CONSTRAINT L_SRCREL_AT_CON_SRCRLSHP_FK FOREIGN KEY
(
H_SRC_RLSHP_SQN_SrcRelshp
)
REFERENCES H_SRC_RELATIONSHIP
(
SQN_SrcRelshp
)
ALTER TABLE L_SRC_REL_ATTR

```

```

ADD CONSTRAINT L_SRCREL_AT_SRCAT_FK FOREIGN KEY
(
H_SRC_ATTR_SQN_SrcAttr
)
REFERENCES H_SRC_ATTRIBUTE
(
SQN_SrcAttr
)
ALTER TABLE L_SRC_REL_ATTR
ADD CONSTRAINT L_SRCREL_AT_SRCREL_FK FOREIGN KEY
(
H_SRC_REL_SQN_SrcRel
)
REFERENCES H_SRC_RELATION
(
SQN_SrcRelation
)
ALTER TABLE L_SRC_REL_RELSHP
ADD CONSTRAINT L_SRC_REL_RLSHP_SRC_REL_FK FOREIGN KEY
(
H_SRC_REL_SQN_SrcRel
)
REFERENCES H_SRC_RELATION
(
SQN_SrcRelation
)
ALTER TABLE L_SRC_REL_RELSHP
ADD CONSTRAINT L_SRC_REL_RLSHP_SRC_RLSHP_FK FOREIGN KEY
(
H_SRC_RLSHP_SQN_SrcRlshp
)
REFERENCES H_SRC_RELATIONSHIP
(
SQN_SrcRelshp
)
ALTER TABLE SAL_MASTER_ATTR
ADD CONSTRAINT SAL_MASTER_ATTR_ATTRB_FK FOREIGN KEY
(
H_ATTR_SQN_AttrBase
)

```

```

REFERENCES H_ATTRIBUTE
(
SQN_Attribute
)
ALTER TABLE SAL_MASTER_ATTR
ADD CONSTRAINT SAL_MASTER_ATTR_ATTRM_FK FOREIGN KEY
(
H_ATTR_SQN_AttrMaster
)
REFERENCES H_ATTRIBUTE
(
SQN_Attribute
)
ALTER TABLE SAL_MASTER_ATTR
ADD CONSTRAINT SAL_MASTER_ATTR_RULE_FK FOREIGN KEY
(
H_RULE_SQN_Rule
)
REFERENCES H_RULE
(
SQN_Rule
)
ALTER TABLE SAL_MASTER_ATTR
ADD CONSTRAINT SAL_MASTER_ATTR_TRNSF_FK FOREIGN KEY
(
H_TRANSF_SQN_Transf
)
REFERENCES H_TRANSFORMATION
(
SQN_Transformation
)
ALTER TABLE SAL_MASTER_HUB
ADD CONSTRAINT SAL_MASTER_HUB_HUBB_FK FOREIGN KEY
(
H_HUB_SQN_HubBase
)
REFERENCES H_HUB
(
SQN_Hub
)

```

```

ALTER TABLE SAL_MASTER_HUB
ADD CONSTRAINT SAL_MASTER_HUB_HUBM_FK FOREIGN KEY
(
H_HUB_SQN_HubMaster
)
REFERENCES H_HUB
(
SQN_Hub
)
ALTER TABLE SAL_MASTER_HUB
ADD CONSTRAINT SAL_MASTER_HUB_RULE_FK FOREIGN KEY
(
H_RULE_SQN_Rule
)
REFERENCES H_RULE
(
SQN_Rule
)
ALTER TABLE SAL_MASTER_HUB
ADD CONSTRAINT SAL_MASTER_HUB_TRNSF_FK FOREIGN KEY
(
H_TRANSF_SQN_Transf
)
REFERENCES H_TRANSFORMATION
(
SQN_Transformation
)
ALTER TABLE SAL_MASTER_LINK
ADD CONSTRAINT SAL_MASTER_LINK_LINKB_FK FOREIGN KEY
(
H_LINK_SQN_LinkBas
)
REFERENCES H_LINK
(
SQN_Link
)
ALTER TABLE SAL_MASTER_LINK
ADD CONSTRAINT SAL_MASTER_LINK_LINKM_FK FOREIGN KEY
(
H_LINK_SQN_LinkMas

```

```

)
REFERENCES H_LINK
(
SQN_Link
)
ALTER TABLE SAL_MASTER_LINK
ADD CONSTRAINT SAL_MASTER_LINK_RULE_FK FOREIGN KEY
(
H_RULE_SQN_Rule
)
REFERENCES H_RULE
(
SQN_Rule
)
ALTER TABLE SAL_MASTER_LINK
ADD CONSTRAINT SAL_MASTER_LINK_TRNSF_FK FOREIGN KEY
(
H_TRANSF_SQN_Transf
)
REFERENCES H_TRANSFORMATION
(
SQN_Transformation
)
ALTER TABLE SAL_MASTER_SAT
ADD CONSTRAINT SAL_MASTER_SAT_RULE_FK FOREIGN KEY
(
H_RULE_SQN_Rule
)
REFERENCES H_RULE
(
SQN_Rule
)
ALTER TABLE SAL_MASTER_SAT
ADD CONSTRAINT SAL_MASTER_SAT_SATB_FK FOREIGN KEY
(
H_SAT_SQN_SatBase
)
REFERENCES H_SATELLITE
(
SQN_Satellite

```

```

)
ALTER TABLE SAL_MASTER_SAT
ADD CONSTRAINT SAL_MASTER_SAT_SATM_FK FOREIGN KEY
(
H_SAT_SQN_SatMaster
)
REFERENCES H_SATELLITE
(
SQN_Satellite
)
ALTER TABLE SAL_MASTER_SAT
ADD CONSTRAINT SAL_MASTER_SAT_TRNSF_FK FOREIGN KEY
(
H_TRANSF_SQN_Transf
)
REFERENCES H_TRANSFORMATION
(
SQN_Transformation
)
ALTER TABLE S_ATTR_DEF
ADD CONSTRAINT S_ATTR_DEF_ATTRIBUTE_FK FOREIGN KEY
(
H_ATTRIBUTE_SQN_Attribute
)
REFERENCES H_ATTRIBUTE
(
SQN_Attribute
)
ALTER TABLE S_ATTR_ISSUE
ADD CONSTRAINT S_ATT_ISS_ATT_SRCATT_FK FOREIGN KEY
(
L_AT_SRC_AT_SQN_ASA
)
REFERENCES L_ATTR_SRC_ATTR
(
SQN_AttrSrcAttr
)
ALTER TABLE S_BUSINESS_KEY
ADD CONSTRAINT S_BUSINESS_KEY_HUB_FK FOREIGN KEY
(

```

```

H_HUB_SQN_Hub
)
REFERENCES H_HUB
(
SQN_Hub
)
ALTER TABLE S_CODE_ATTR
ADD CONSTRAINT S_CODE_ATTR_SAL_MASTER_ATTR_FK FOREIGN KEY
(
SAL_MASTER_ATTR_SQN_MasterAttr
)
REFERENCES SAL_MASTER_ATTR
(
SQN_MasterAttr
)
ALTER TABLE S_CODE_HUB
ADD CONSTRAINT S_CODE_HUB_SAL_MASTER_HUB_FK FOREIGN KEY
(
SAL_MASTER_HUB_SQN_MasterHub
)
REFERENCES SAL_MASTER_HUB
(
SQN_MasterHub
)
ALTER TABLE S_CODE_LINK
ADD CONSTRAINT S_CODE_LINK_SAL_MASTER_LINK_FK FOREIGN KEY
(
SAL_MASTER_LINK_SQN_MasterLink
)
REFERENCES SAL_MASTER_LINK
(
SQN_MasterLink
)
ALTER TABLE S_CODE_SAT
ADD CONSTRAINT S_CODE_SAT_SAL_MASTER_SAT_FK FOREIGN KEY
(
SAL_MASTER_SAT_SQN_MasterSat
)
REFERENCES SAL_MASTER_SAT
(

```

```

SQN_MasterSat
)
ALTER TABLE S_CONSTRAINT
ADD CONSTRAINT S_CONSTRAINT_CONSTRAINT_FK FOREIGN KEY
(
H_CONSTRAINT_SQN_Constraint
)
REFERENCES H_CONSTRAINT
(
SQN_Constraint
)
ALTER TABLE S_DATA_SOURCE
ADD CONSTRAINT S_DATA_SOURCE_DATA_SOURCE_FK FOREIGN KEY
(
H_DATA_SOURCE_SQN_DataSource
)
REFERENCES H_DATA_SOURCE
(
SQN_DataSource
)
ALTER TABLE S_DOMAIN
ADD CONSTRAINT S_DOMAIN_DOMAIN_FK FOREIGN KEY
(
H_DOMAIN_SQN_Domain
)
REFERENCES H_DOMAIN
(
SQN_Domain
)
ALTER TABLE S_HSL_ISSUE
ADD CONSTRAINT S_HSL_ISSUE_RELATION_HSL_FK FOREIGN KEY
(
L_RELATION_HSL_SQN_ReIHSL
)
REFERENCES L_RELATION_HSL
(
SQN_ReIHSL
)
ALTER TABLE S_HUB_DEF
ADD CONSTRAINT S_HUB_DEF_HUB_FK FOREIGN KEY

```



```

(
H_HUB_SQN_Hub
)
REFERENCES H_HUB
(
SQN_Hub
)
ALTER TABLE S_LINK_DEF
ADD CONSTRAINT S_LINK_DEF_LINK_FK FOREIGN KEY
(
H_LINK_SQN_Link
)
REFERENCES H_LINK
(
SQN_Link
)
ALTER TABLE S_LINK_ISSUE
ADD CONSTRAINT S_LINK_ISSUE_RELSHP_LINK_FK FOREIGN KEY
(
L_RELSHP_LINK_SQN_RelshpLink
)
REFERENCES L_RELSHP_LINK
(
SQN_RelshpLink
)
ALTER TABLE S_REF_DEF
ADD CONSTRAINT S_REF_DEF_REFERENCE_FK FOREIGN KEY
(
H_REFERENCE_SQN_Reference
)
REFERENCES H_REFERENCE
(
SQN_Reference
)
ALTER TABLE S_RESOURCE
ADD CONSTRAINT S_RESOURCE_RESOURCE_FK FOREIGN KEY
(
H_RESOURCE_SQN_Resource
)
REFERENCES H_RESOURCE

```

```

(
SQN_Resource
)
ALTER TABLE S_ROLE_DEF
ADD CONSTRAINT S_ROLE_DEF_HUB_LINK_FK FOREIGN KEY
(
L_HUB_LINK_SQN_HubLink
)
REFERENCES L_HUB_LINK
(
SQN_HubLink
)
ALTER TABLE S_RULE_CODE
ADD CONSTRAINT S_RULE_CODE_RULE_FK FOREIGN KEY
(
H_RULE_SQN_Rule
)
REFERENCES H_RULE
(
SQN_Rule
)
ALTER TABLE S_RULE_DEF
ADD CONSTRAINT S_RULE_DEF_RULE_FK FOREIGN KEY
(
H_RULE_SQN_Rule
)
REFERENCES H_RULE
(
SQN_Rule
)
ALTER TABLE S_RULE_RELSHP
ADD CONSTRAINT S_RUL_RLSHP_HL_RULRLSHP_FK FOREIGN KEY
(
HL_RUL_RLSHP_SQNRuRlshp
)
REFERENCES HL_RULE_RELSHP
(
SQN_RuleRelshp
)
ALTER TABLE S_SAT_DEF

```

```

ADD CONSTRAINT S_SAT_DEF_SAT_FK FOREIGN KEY
(
H_SAT_SQN_Sat
)
REFERENCES H_SATELLITE
(
SQN_Satellite
)
ALTER TABLE S_SDV_CODE_ATTR
ADD CONSTRAINT S_SDVCODAT_ATSRCAT_FK FOREIGN KEY
(
L_AT_SRCAT_SQN_ASA
)
REFERENCES L_ATTR_SRC_ATTR
(
SQN_AttrSrcAttr
)
ALTER TABLE S_SDV_CODE_LINK
ADD CONSTRAINT S_SDVCODLNK_RLSHPLINK_FK FOREIGN KEY
(
L_RLSHP_LINK_SQN_RlshpLink
)
REFERENCES L_RELSHP_LINK
(
SQN_RelshpLink
)
ALTER TABLE S_SDV_CODE_HSL
ADD CONSTRAINT S_SDVCOD_HSL_REL_HSL_FK FOREIGN KEY
(
L_REL_HSL_SQN_ReIHSL
)
REFERENCES L_RELATION_HSL
(
SQN_ReIHSL
)
ALTER TABLE S_SRC_RELATIONSHIP
ADD CONSTRAINT S_SRCRLSHP_SRCRLSHP_FK FOREIGN KEY
(
H_SRCRLSHP_SQNsrcRelshp
)

```

```

REFERENCES H_SRC_RELATIONSHIP
(
SQN_SrcRelshp
)
ALTER TABLE S_SRC_ATTRIBUTE
ADD CONSTRAINT S_SRC_ATTR_SRC_ATTR_FK FOREIGN KEY
(
H_SRC_ATTR_SQN_SrcAttr
)
REFERENCES H_SRC_ATTRIBUTE
(
SQN_SrcAttr
)
ALTER TABLE S_SRC_RELATION
ADD CONSTRAINT S_SRC_REL_SRCREL_FK FOREIGN KEY
(
H_SRC_REL_SQN_SrcRel
)
REFERENCES H_SRC_RELATION
(
SQN_SrcRelation
)
ALTER TABLE S_SRC_ROLE
ADD CONSTRAINT S_SRC_ROLE_SRC_REL_RELSHP_FK FOREIGN KEY
(
L_SRC_REL_RELSHP_SQN_RelRelshp
)
REFERENCES L_SRC_REL_RELSHP
(
SQN_RelRelshp
)
ALTER TABLE S_TRANSFORMATION
ADD CONSTRAINT S_TRNSF_TRNSF_FK FOREIGN KEY
(
H_TRANSF_SQN_Transf
)
REFERENCES H_TRANSFORMATION
(
SQN_Transformation
)

```

```
ALTER TABLE S_TRANSF_RELSHP
ADD CONSTRAINT S_TRSF_RLSHP_HL_TRSF_RLSHP_FK FOREIGN KEY
(
HL_TRSF_RLSHP_SQN_TrsfRlshp
)
REFERENCES HL_TRANSF_RELSHP
(
SQN_TransfRelshp
)
```

## ŽIVOTOPIS

Danijela Jakšić (rođena Subotić) rođena je 6. ožujka 1987. godine u Rijeci. Nakon završetka osnovnoškolskog i srednješkolskog obrazovanja (Prva riječka hrvatska gimnazija, opći smjer) upisuje preddiplomski studij na Filozofskom fakultetu u Rijeci (danas: Odjel za informatiku Sveučilišta u Rijeci), smjer: informatika (jednopedmetna). 2008. godine brani završni rad pod nazivom „Mali priručnik za SQL“ (mentor: izv. prof. dr. sc. Patrizia Pošćić) te upisuje diplomski studij na Odjelu za informatiku Sveučilišta u Rijeci, smjer:nastavnički. Diplomirala je u rujnu 2010. godine obranom diplomskog rada pod nazivom „Izrada aplikacije za evidenciju primki i otpremioca poslovnog poduzeća“ (mentor: izv. prof. dr. sc. Patrizia Pošćić). Od studenog 2010. zaposlena je na Odjelu za informatiku Sveučilišta u Rijeci, na radnom mjestu asistenta. Aktivno priprema i izvodi vježbe iz kolegija na preddiplomskom studiju (Uvod u Baze podataka, Baze podataka) te na diplomskom studiju (Odabrana poglavlja iz baza podataka, Menadžment i upravljanje, Poslovna komunikacija).

Poslijediplomski doktorski studij informacijskih znanosti upisala je 2010 godine na Filozofskom fakultetu u Zagrebu te ga nastavlja 2012. godine na Odjelu za informatiku Sveučilišta u Rijeci.

Sudjelovala je u nekoliko nacionalnih znanstvenih te međunarodnih stručnih projekata. Sudjelovala je na brojnim međunarodnim znanstvenim konferencijama, te samostalno i u suautorstvu objavila više znanstvenih radova. Znanstveni interesi obuhvaćaju područje baza podataka, skladišta podataka te poslovne inteligencije. Ima iskustva u pisanju i prijavi nacionalnih (UNIRI potpora) znanstvenih projekata.

U nastavku slijedi popis objavljenih znanstvenih radova.

1. Pavkov, S., Pošćić, P., Jakšić, D.: Business Intelligence Systems Yesterday, Today and Tomorrow - an Overview // Zbornik Sveučilišta u Rijeci, Rijeka, Croatia, 2016.
2. Šubić, T., Pošćić, P., Jakšić, D.: Big Data in Data Warehouses // 5th International Conference The Future of Information Sciences (INFuture2015), Zagreb, Croatia, 2015.
3. Jakšić, D., Pošćić, P.: Data Warehouse Models in Higher Education Courses // International Conference on Advanced Technology & Sciences (ICAT'15), Antalya, Turkey, 2015.

4. Jakšić, Danijela; Pošćić, Patrizia. Reasons for academic dishonesty in a traditional and a partially online environment: A survey // 16th International Conference on Computer Systems and Technologies CompSysTech'15. Dublin, Ireland, 2015.
5. Subotic, D.: Data Warehouse Schema Evolution Perspectives // Advances in Intelligent Systems and Computing / Bassiliades, Nick ; Ivanovic, Mirjana ; Kon-Popovska, Margita ; Manolopoulos, Yannis ; Palpanas, Themis ; Trajcevski, Goce ; Vakali, Athena (ur.). // 18th East-European Conference on Advances in Databases and Information Systems ADBIS 2014 Ohrid : Springer, 2014. 333–338.
6. Subotić, D., Jovanović, V., Pošćić, P.: Data Warehouse Schema Evolution: State of the Art. In: Proceedings of 25th Central European Conference on Information and Intelligent Systems CECIS 2014. Varaždin, Croatia, 2014.
7. Subotić, D., Jovanović, V., Pošćić, P.: Data Warehouse and Master Data Management Evolution - A Meta-Data\_vault Approach. Issues in Information Systems. DeLorenzo, Gary (ed.), International Association of Computer Information Systems, 2014. 14-23.
8. Jovanović, V., Subotić, D., Mrdalj, S.: Data Modeling Styles in Data Warehousing. In: Proceedings of 37th International convention on Information and Communication Technology, Electronics and Microelectronics MIPRO 2014. Opatija, Croatia, 2014.
9. Subotić, D., Pošćić, P.: Academic dishonesty in a partially online environment: A survey. 15th International Conference on Computer Systems and Technologies CompSysTech'14. Ruse, Bulgaria, 2014.
10. Pošćić, P., Subotić, D., Ivašić-Kos, M.: Developing the course Database systems to respond to market requirements. In: Proceedings of Jubilee 35th International Convention on Information and Communication Technology, Electronics and Microelectronics / Biljanović, Petar (ur.). Rijeka : Croatian Society for Information and Communication Technology, Electronics and Microelectronics - MIPRO, 2012. 1327-1331.
11. Subotić, D., Pošćić, P., Slavuj, V.: OLAP Tools in Education. Book of Manuscripts, Society and Technology 2012 - Dr. Juraj Plenković / Mustić, Darija ; Plenković, Mario (ur.). Zagreb : International Federation of Communication Associations - IFCA ; Croatian Communication Association - CCA, 2012. 207-217.