

# DW sistem za praćenje nastavnog procesa

---

SISTEMI SKLADIŠTA PODATAKA

# Sadržaj

Šema OLTP baze podataka

Šema DW baze podataka

Specifikacija ETL procesa

Implementacija ETL procesa

Materijalizovani pogledi

# Šema oltp baze podataka

---

Za izvor podataka ovog Data Warehouse sistema iskorišćen je deo šeme baze podataka studentske službe, koji vodi evidenciju o izvršenim obavezama svakog studenta. Model se sastoji od 23 tabele i sledi kratak opis svake od njih.

- SKOLSKA\_GODINA - U ovoj tabeli se čuvaju školske godine u kojima se realizuju nastavni planovi.
- FAKULTET - Ova tabela čuva informacije o organizacionim jedinicama. Svaka jedinica je opisana svojom šifrom i nazivom.
- DEPARTMAN - U ovoj tabeli se definiše odnos između organizacionih jedinica. Sadrži dve kolone koje čuvaju šifru organizacije i šifru njoj nadređene organizacione jedinice.
- NASTAVNI\_PLAN - Tabela koja čuva podatke o svim nastavnim planovima. Tabela ima dve kolone, šifru organizacione jedinice na kojoj se realizuje i kalendarsku godinu koje je nastavni plan donet.
- REALIZACIJA\_PLANA - U ovoj tabeli se čuvaju podaci za sve realizacije plana. Tabela čuva podatke: godinu realizacije plana, koji se nastavni plan realizuje kao i broj upisanih studenata.
- PREDMET - Tabela čuva informacije o svakom predmetu. Svaki predmet je opisan svojom šifrom, nazivom, fakultetom na kojem se izvodi i godinom nastavnog plana.
- SEMESTAR - Čuva informacije vezane za svaki semestar unutar nekog predmeta. Za svaki predmet se definiše semestar u kojem se predmet izvodi, kao i broj časova predavanja, audiotornih i laboratorijskih vežbi.

# Šema oltp baze podataka

---

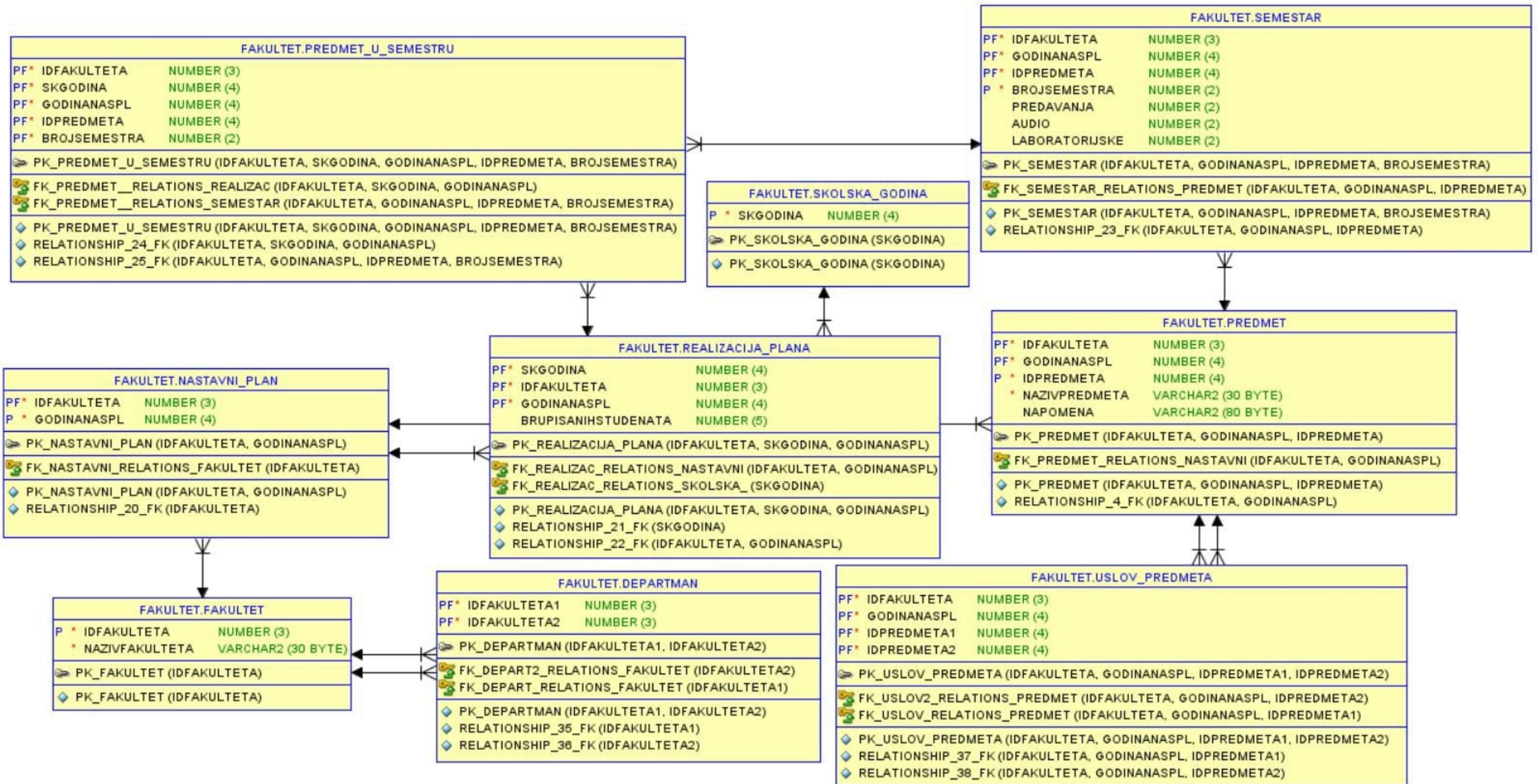
- PREDMET\_U\_SEMESTRU - Ova tabela sadrži informacije o realizaciji predmeta u semestru. Pored predmeta i semestra u kojem se predmet izvodi, čuva se i školska godina u kojoj se predmet realizuje.
- USLOV\_PREDMETA - U ovoj tabeli se čuvaju podaci o međusobnoj zavisnosti predmeta.
- NASTAVNO\_OSoblJE - Tabela koja čuva podatke o nastavnom osoblju (nastavnicima i asistentima). Svaka osoba je okarakterisana šifrom, imenom, prezimenom, zvanjem, podatkom da li je zaposlena.
- NASTAVNIK - Tabela koja čuva nastavnike.
- ASISTENT - U ovoj tabeli se nalaze asistenti.
- GRUPA\_ZA\_PREDAVANJE - Ova tabela čuva podatke o grupama za predavanja. Svaka grupa je opisana rednim brojem, nastavnikom koji drži grupu, predmetom za koji je grupa, danom predavanja, vremenom početka, odnosno završetka termina grupe.
- GRUPA\_ZA\_VEZBE - Tabela čuva podatke o grupama za vežbe. Svaka grupa je opisana rednim brojem, nastavnikom koji drži grupu, predmetom za koji je grupa, danom održavanja vežbi, vremenom početka, odnosno završetka termina grupe.
- STUDENT - U ovoj tabeli se čuvaju podaci o studentima. Svaki student je okarakterisan jedinstvenim rednim brojem, brojem indeksa, imenom, prezimenom, godinom upisa na fakultet, nastavnim planom koji pohađa i rednim brojem semestra koji je upisao.
- POHADJA\_PREDAVANJA - U ovoj tabeli se čuvaju podaci za svakog studenta kojoj grupi predavanja pripada na svakom predmetu koji pohađa.

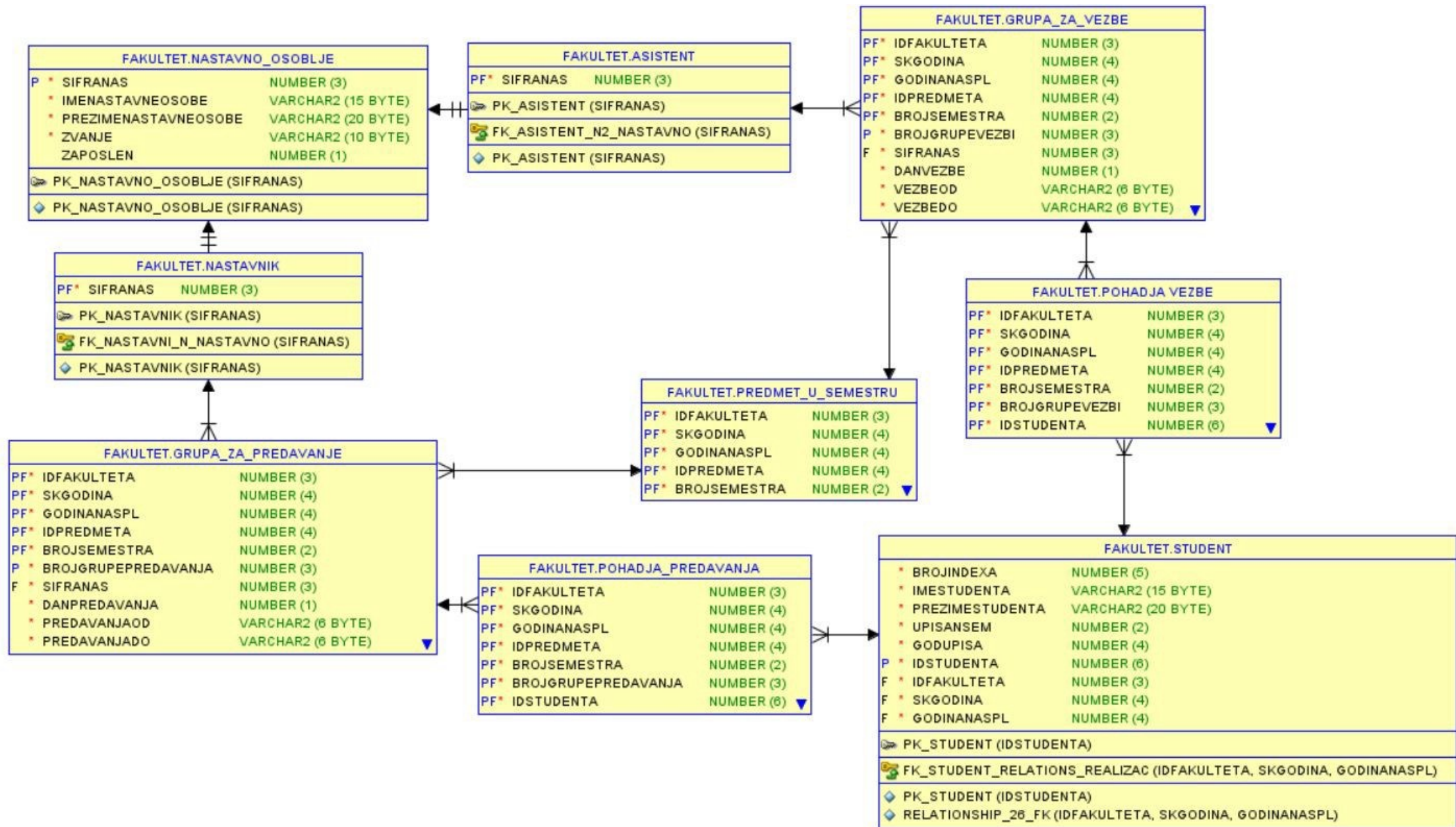
# Šema oltp baze podataka

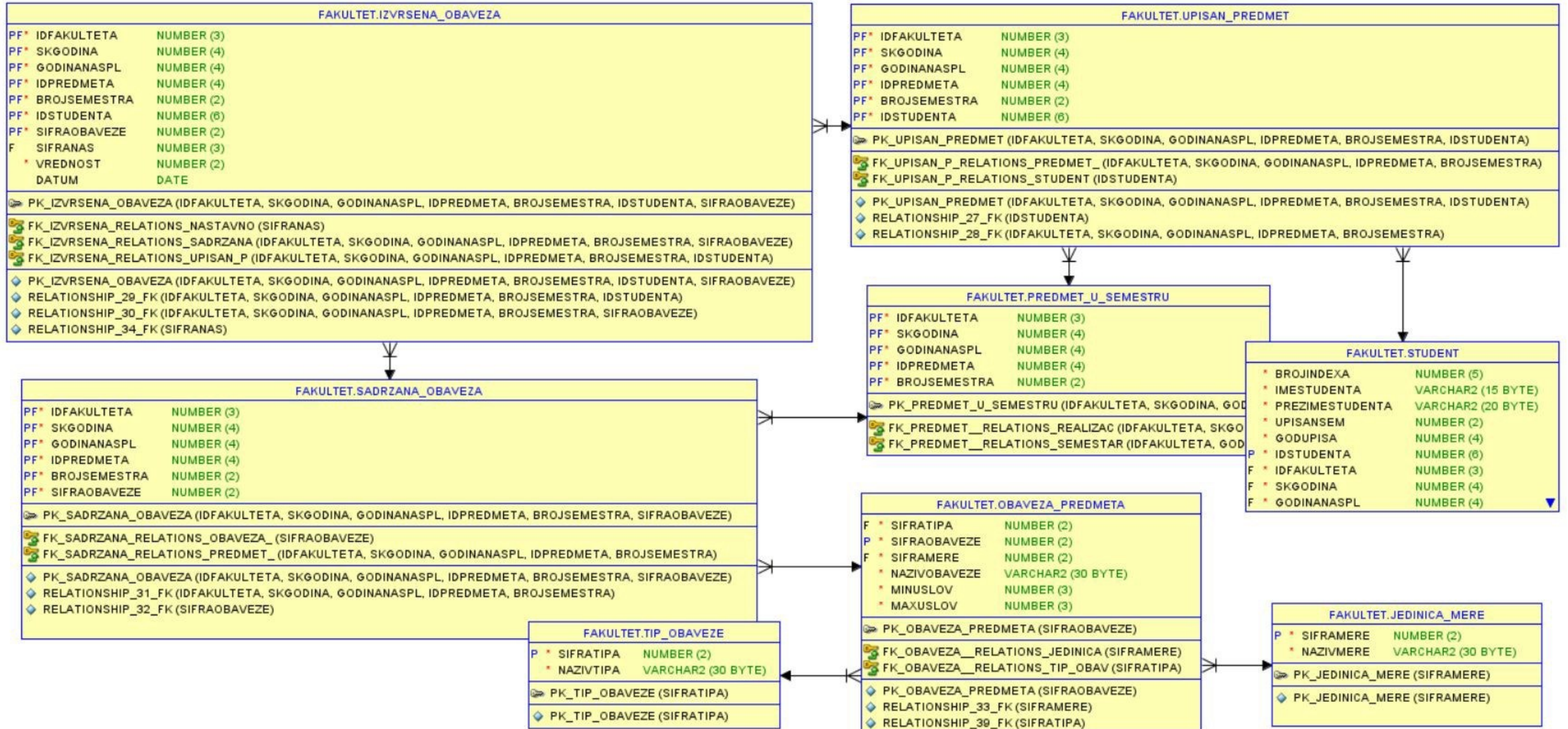
---

- POHADJA\_VEZBE - Tabela čuva podatke za svakog studenta kojoj grupi vežbi pripada na svakom predmetu koji pohađa.
- UPISAN\_PREDMET - Tabela koja čuva podatke koje sve predmete svaki student sluša.
- TIP\_OBAVEZE - U ovoj tabeli se definišu tipovi obaveze na predmetima koje student treba da izvrši.
- JEDINICA\_MERE - U ovoj tabeli se čuvaju osnovni podaci o jedinicama mere koje se koriste kao kriterijum za izvršavanje obaveza studenata.
- OBAVEZA\_PREDMETA - U ovoj tabeli se čuvaju podaci o opštenastavnim obavezama predmeta. Svaka obaveza je opisana jedinstvenom šifrom, nazivom, tipom obaveze kojoj pripada, jedinicom mere, kao i minimalnim i maksimalnim uslovom izvršenja obaveze.
- SADRZANA\_OBAVEZA - Tabela sadrži podatke koje obaveze svaki predmet sadrži.
- IZVRSENA\_OBAVEZA - Ova tabela čuva evidenciju o svim izvršenim obavezama svakog studenta. Pored podataka o studentu, predmetu i vrsti obaveze koja je izvršena evidentira se i nastavnik koji je ocenio studenta, kao i vrednost koju je student postigao. Ova tabela je centralana tabela ovog modela i predstavlja zapravo izvor činjenica posmatranog Data Warehouse sistema.

Pored ove baze podataka, kao izvor je iskorišćena i jedna datoteka, kandidati.txt koja čuva podatke o kandidatima prilikom upisa na fakultet. Svaki kandidat je okarakterisan brojem indeksa, imenom, prezimenom i uspehom koji je postigao u srednjoj školi, u obliku prosečne ocene.







# Sadržaj

---

Šema OLTP baze podataka

Šema DW baze podataka

Specifikacija ETL procesa

Implementacija ETL procesa

Materijalizovani pogledi

# Šema dw baze podataka

---

Model baze Data Warehouse sistema se sastoji od osam tabela od kojih su dve tabele činjenica, a šest tabela su dimenzije. Ova šema, iako jednostavnija po strukturi, spada u šeme tipa sazvežđa jer postoji više od jedne tabele činjenica sa zajedničkom dimenzijom. Dve tabele ovog tipa su potrebne jer se prati proces izvršavanja obaveza studenata i deo procesa oko upisa novih studenata na fakultet. Sledi kratak opis svake od ovih tabela.

- Dimenzije koje ovaj sistem sadrži su: DEPARTMANIDIM, STUDENTIDIM, NASTAVNICIDIM, PREDMETIDIM, TIPOBAVEZEDIM i VREMEDIM. Tabele činjenica koje se nalaze u ovom sistemu su IZVRSENE\_OBAVEZE\_FACT i KANDIDATI\_FACT. U svim dimenzijama osim u DEPARTMANIDIM ažuriranje podataka se obavlja bez praćenja istorije izmena.
- DEPARTMANIDIM je dimenzija koja čuva podatke o departmanima (organizacionim jednicama) fakulteta. Podaci za ovu tabelu se dobavljaju iz tabele FAKULTET izvorne baze podataka. Pored kolona u kojima se čuvaju šifra i naziv departmana, ova dimenzija sadrži još dva dodatna polja startdate i enddate. Ova polja služe za praćenje evidencije istorije izmene podataka.

# Šema dw baze podataka

---

- U trenutku učitavanja nove torke u dimenziju polje startdate se postavlja na trenutni datum, a polje enddate na vrednost null. Prilikom zatvaranja određene torke polje enddate se postavlja na datum zatvaranja i samim tim ona više nije aktuelna. Primarni ključ u ovoj dimenziji je posebna kolona i za svaku novu torku njegova vrednost se generiše. Ona je povezana sa tabelom činjenica IZVRSENE\_OBAVEZE\_FACT, kako bi se omogućilo praćenje izvršenih obaveza po departmanima.
- STUDENTIDIM je tabela koja vodi evidenciju studenata upisanih na fakultet. Podaci se učitavaju iz tabele STUDENT izvorne baze podataka. Međutim kako se traži da ova dimenzija sadrži i naziv departmana na koji je student upisan, potrebno je ovaj podatak potražiti u tabeli FAKULTET baze podataka. Iako na ovaj način dolazi do redundantnosti podataka, jer se podaci o departmanima čuvaju na dva mesta, sa stanovišta performansi sistema mnogo je bolje smanjiti broj operacija spojeva (join) prilikom upita. Pored ove kolone svaka torka studenta sadrži podatke o njegovom imenu, prezimenu, broju indeksa, godini upisa na fakultet i šifi koja je i primarni ključ u izvornoj tabeli. Prilikom ažuriranja podataka evidencija o izmeni podataka se ne prati. S toga je primarni ključ prirodan, nasleđuje se iz baze podataka, jer se sastoji samo od jednog polja. Ova dimenzija je povezana sa obe tabele činjenica jer ne bi imalo smisla da se čuvaju podaci o izvršenim obavezama, a da se ne zna ko je zapravo izvršio tu obavezu. Takođe je neophodno znati koji kandidat je ostvario određeni uspeh u srednjoj školi. Pretpostavka je da će se ova dimenzija najčešće menjati, tj. da će se u nju najčešće dodavati novi redovi.

# Šema dw baze podataka

---

- Dimenzija NASTAVNICIDIM je tabela u kojoj se čuvaju nastavnici i asistenti fakulteta za koji je ovaj sistem napravljen. Podaci o njima se preuzimaju iz tabele NASTAVNO\_OSOBLJE izvorne baze podataka. Kako se svi potrebni podaci nalaze u ovoj tabeli izvornoj tabeli, preuzimanje podataka se vrši direktno bez spajanja sa nekom drugom tabelom. Za svaku nastavnu osobu se čuva njegovo ime, prezime, zvanje i šifra koja je i primarni ključ u izvornoj tabeli. Evidencija o izmeni podataka prilikom ažuriranja se ne vodi. Primarni ključ u ovoj dimenziji je nasleđen iz izvorne tabele, a jedan od razloga je upravo to što se ne vodi evidencija prilikom ažuriranja. Takođe je i primarni ključ šifra nastavne osobe, koja se sastoji od samo jednog atributa. Ova dimenzija je povezana sa tabelom činjenica IZVRSENE\_OBAVEZE\_FACT, kako bi se moglo utvrditi ko je ocenio studenta, prosek ocena kod određene nastavne osobe i mnoge druge analize.
- PREDMETIDIM je dimenzija u kojoj se čuvaju podaci o predmetima na fakultetu, organizovani po studijskim programima. Kako se u ovoj dimenziji čuvaju podaci o predmetima i studijskim programima kojima pripadaju, napravljena je hijerarhija u dva nivoa. Shodno tome kakvi se podaci u njima nalaze, naziv prvog nivoa je PREDMET, a drugog STUDIJSKI PROGRAM. Podaci za ovu dimenziju se uzimaju iz tabela PREDMET, PREDMET\_U\_SEMESTRU, SEMESTAR i REALIZACIJA\_PLANA. Za svaku torku predmet se beleži naziv, identifikacioni broj, semestar u kojem se sluša, broj časova predavanja, audiotornih i laboratorijskih vežbi i studijski program kojem predmet pripada.

# Šema dw baze podataka

---

- Svaki studijski program je opisan sa dva atributa: godinom kada je studijski program proglašen i godinom realizacije studijskog programa. Pored ova dva atributa svaki studijski program ima svoju jedinstvenu šifru, koju je potrebno generisati jer ne postoji u izvornoj tabeli. Evidencija o izmenama prilikom ažuriranja podataka ove dimenzije, nije podržana. Za svaku unetu novu torku se generiše vrednost njenog primarnog ključa. Odlučeno je da se koristi tip veštačkog primarnog ključa, jer se primarni ključ u izvornoj tabeli sastoji od nekoliko atributa, a u ovakvim sistemima je poželjno da on bude što jednostavniji. Dimenzija PREDMETIDIM je povezana sa tabelom činjenica IZVRSENE\_OBAVEZE\_FACT, kako bi se podržala mogućnost praćenja izvršenih obaveza po predmetima.
- TIPOBAVEZEDIM je dimenzija koja čuva podatke o tipovima obaveza, koje student ima na predmetima. Neki od tipova koji se mogu javiti su: predispitna, ispitna, krajnja ali i neke druge propisane od strane fakulteta. Podaci za ovu dimenziju se dobijaju iz tabela TIP\_OBAVEZE, JEDINICA\_MERE i izborne baze podataka. Spajanje nekoliko izvornih tabela je neophodno jer se za svaki tip obaveze čuva: njegov naziv, naziv jedinice mere, maksimalan i minimalan broj bodova i šifra koja je jedinstvena. Prilikom ažuriranja podataka ne vodi se evidencija o promenama. Primarni ključ u ovoj tabeli se preuzima iz izvorne jer je prost, jedinstven, te nema potrebe za generisanjem veštačkog ključa. Dimenzija TIPOBAVEZEDIM je povezana sa tabelom činjenica IZVRSENE\_OBAVEZE\_FACT jer je jedan od zahteva ovog sistema praćnje obaveza studenata po tipu.

# Šema dw baze podataka

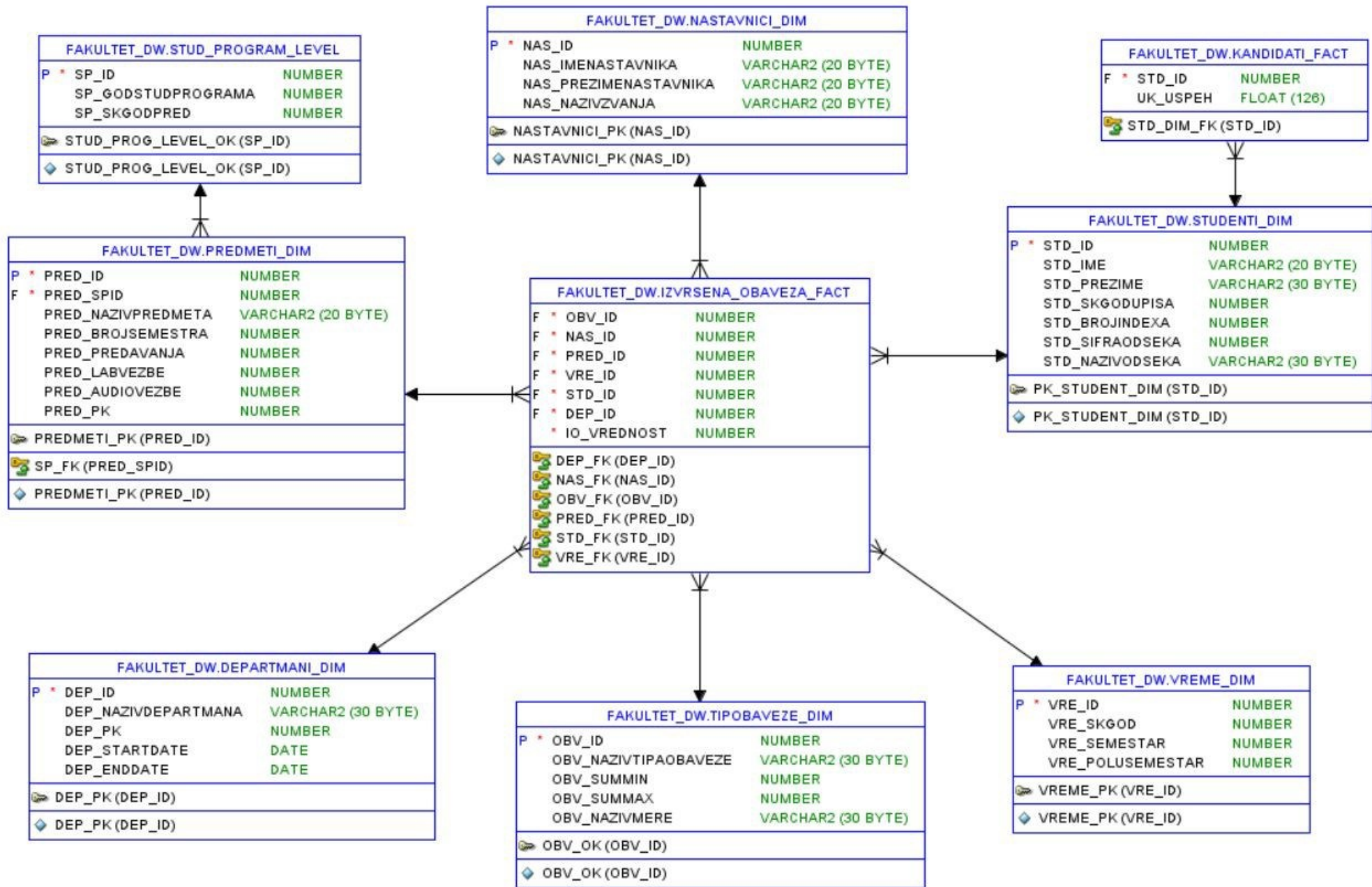
---

- Shodno tome podaci u ovoj tabeli činjenica, nalaze se upravo na nivou granularnosti tipa. Kako je jedan od zahteva analiza prisutnosti, bilo je neophodno ručno dodati prisutnost kao novi tip obaveze, jer se ona nalazi u izvornom sistemu kao obaveza, što je nivo finije granularnosti od one koja je tražena.
- VREMEDIM je dimenzija koja je popunjena podacima do 2020. godine. Svaki Data Warehouse sistem mora imati ovu dimenziju, jer je neophodno vremensko praćenje. Vreme u ovo sistemu nije kalendarsko, tj. nema obeležja tipa mesec, dan, datum. Ovakav oblik ove dimenzije ne bi bio od velike koristi jer u sistemu ne treba da se vodi evidencija o tačnom datumu izvršene obaveze. Međutim korisnicima ovakvog sistema bi bilo interesantno da imaju uvid u pregled izvršenih obaveza po godinama, semestrima i polusemestrima. Zbog toga ova dimenzija sadrži upravo ta tri atributa, koji na sasvim zadovoljavajućem nivou detaljnosti pokazuju vreme izvršenog tipa obaveze. Kako ove tri kolone određuju jedinstvenost torke, bilo je neophodno uvesti još jednu, koja predstavlja šifru. Ova kolona je zapravo veštački generisan primarni ključ i uvedena je zbog jednostovanijeg referenciranja određene torke. Podaci u ovoj tabeli su nepromenljivog tipa, tj. nije podržan bilo kakav oblik izmene. Ona je povezana sa tabelom činjenica IZVRSENE\_OBAVEZE\_FACT jer je jedan od zahteva da se neke analize vrše po vremenu izvršavanja.

# Šema dw baze podataka

---

- Prva tabela činjenica je tabela IZVRSENE\_OBAVEZE\_FACT, u kojoj se beleže sve izvršene obaveze studenta, na nivou detaljnosti tipa obaveze. Ova tabela činjenica je povezana sa dimenzijama DEPARTMANIDIM, STUDENTIDIM, NASTAVNICIDIM, PREDMETIDIM, TIPOBAVEZEDIM, VREMEDIM. Sadrži jednu kolonu koja predstavlja meru i u njoj se čuva vrednost izvršene obaveze studenta po tipovima obaveze. Ova tabela se popunjava iz tabele IZVRSENA\_OBAVEZA iz izvorne baze podataka. Međutim podaci se u nju ne prepisuju direktno, već se prvo sumiraju za svakog studenta vrednosti onih izvršenih obaveza koje su istog tipa. Mera činjenica ove tabele je neaditivna jer se ne može sabirati ni po jednoj dimenziji, ali se iz nje mogu računati razne srednje vrednosti. Ona je takođe izvedenog tipa, jer nije direktno prepisana iz izvorne tabele. Ostali podaci se direktno preuzimaju sa izvora, kako bi se lako moglo utvrditi koji student je kada položio koji deo predmeta, ko ga je ocenio i za koji predmet je ta obaveza izvršena. Primarni ključ ove tabele činjenica je složen, i njegove delove predstavljaju strani ključevi dimenzija. Ova tabela činjenica će vremenom sadržati najveći broj redova, jer se operacija izvršavanja obaveza najčešće obavlja.
- Druga tabela činjenica je KANDIDATI\_FACT, koja čuva za svakog studenta uspeh u srednjoj školi. Ona je povezana samo sa dimenzijom STUDENTIDIM jer se podaci za ovu tabelu preuzimaju iz datoteke kandidati.txt, koja sadrži samo osnovne podatke studenta i njegov uspeh. Meru u ovoj tabeli činjenica predstavlja upravo uspeh studenta u srednjoj školi. Ona je neaditivna jer se ne može sabirati po studentima. S druge strane posmatrano prema izvoru generisanja, ona je osnovna jer je direktno preuzeta iz izvora.



# Sadržaj

---

Šema OLTP baze podataka

Šema DW baze podataka

Specifikacija ETL procesa

Implementacija ETL procesa

Materijalizovani pogledi

# Specifikacija ETL procesa

---

- **Dimenzija NASTAVNICI\_DIM**
  - Dimenzija NASTAVNICIDIM se popunjava kopiranjem podataka iz izvorne tabele NASTAVNO\_OSoblJE. Kako se svi potrebni podaci nalaze u ovoj izvornoj tabeli nije potrebno bilo kakvo spajanje se nekom drugom tabelom.
- **Dimenzija DEPARTMANI\_DIM**
  - Za dimenziju DEPARTMANIDIM podaci se preuzimaju iz tabele FAKULTET izvorne baze podataka. Međutim za razliku od prethodne dimenzije koja ne prati evidenciju o izmenama podataka, u ovoj je prema zahtevima korisnika to neophodno uraditi. S toga je proces učitavanja podataka dosta kompleksniji.
  - Najpre je potrebno spojiti izvornu tabelu i ovu dimenziju tako da se ne izgube podaci koji još nisu uneti u projektovani Data Warehouse sistem. Zatim se vrši filtriranje podataka jer posmatrajući dobijene torke razlikuju se nekoliko slučajeva.

# Specifikacija ETL procesa

---

- **Dimenzija DEPARTMANI\_DIM**

- U prvi slučaj spadaju podaci koji se ne nalaze u dimenziji i njih je potrebno direktno ubaciti postavljanjem polja *startdate* na trenutni datum.
- Drugi slučaj su podaci kod kojih su kolone u izvornoj i ciljnoj bazi jednake i njih ne treba uzimati u obzir.
- Treći slučaj su redovi kod kojih je došlo do promene naziva departmana. To su torke kod kojih kolona nazivdepartmana i nazivfakulteta nemaju istu vrednost.
  - Za ovakav slučaj je potrebno prvo odraditi zatvaranje važećeg reda u dimenziji koji se pronalazi po primarnom ključu iz izvorne tabele. Sve što je potrebno uraditi jeste da se vrednost kolone postavi na trenutni datum.
  - Zatim se vrši dodavanje novog reda u dimenziju sa najnovijim vrednostima te torke iz izvorne tabele, postavljanjem kolone *startdate* na trenutni datum.

-

# Specifikacija ETL procesa

---

- **Dimenzija STUDENTI\_DIM**

- Podaci o studentima za dimenziju STUDENTIDIM se dobijaju iz tabele STUDENT izvorne baze. Međutim, kako je zahtev korisnika da pored šifre departmana na koji je student upisan stoji podatak o nazivu departmana, potrebna je i izvorna tabela FAKULTET.

- **Dimenzija TIPOBAVEZE\_DIM**

- Dimenzija TIPOBAVEZEDIM popunjava se podacima iz tabele TIP\_OBAVEZE izvorne baze podataka. Međutim, korisniku je neophodno da ima podatke o ukupnoj maksimalnoj i minimalnoj vrednosti koju student može da ostvari na obavezama tog tipa (Primjer: Tip obaveze je predispitna ali se ona sastoji od više obaveza predmeta-Test1, Test2...), kao i o nazivu mere kojom se ta vrednost izražava. Te podatke treba potražiti u izvornim tabelama OBAVEZA\_PREDMETA i JEDINICA\_MERE.

# Specifikacija ETL procesa

---

- **Dimenzija TIPOBAVEZE\_DIM**
  - Jedan od zahteva korisnika je analiza prisutnosti koja ne pripada tipu obaveze, već je jedna od vrsta obaveze. Maksimalne i minimalne vrednosti treba potražiti u izvornoj tabeli OBAVEZA\_PREDMETA, a naziv mere u tabeli JEDINICA\_MERE. Ova vrsta obaveze se u dw tretira kao tip obaveze.
- **Dimenzija VREME\_DIM**
  - VREMEDIM je dimenzija za koju ne postoji izvor podataka. S toga je potrebno napisati uskladištenu proceduru koja generiše podatke za ovu dimenziju. Kreiraju se podaci za školske godine u rasponu od 2000. do 2020. godine. Svaka školska godina ima deset semestara, a svaki semestar dva polusemestra.

# Specifikacija ETL procesa

---

- **Dimenzija PREDMETI\_DIM**

- Za dimenziju PREDMETIDIM, podaci se moraju potražiti u nekoliko izvornih tabela. Tabela od koje se polazi je PREDMET\_U\_SEMESTRU, gde se nalaze osnovne činjenice o predmetima. Kako ovaj izvor podataka nije dovoljan mora se izvršiti spoj ove izvorne tabele sa tabelama PREDMET i SEMESTAR. Ovim su prikupljeni skoro svi podaci.
- Pre učitavanja podataka o predmetima potrebno je izvršiti kopiranje podataka o studijskim programima i formiranje dimenzije STUD\_PROGRAM\_LEVEL radi hijerarhijske organizacije ove dimenzije.
- Moguće je koristiti pomoćne *stage* tabele.

# Specifikacija ETL procesa

---

- **Tabela činjenica IZVRSENE\_OBAVEZE\_FACT**

- Kao izvor podataka koristi tabelu IZVRSENA\_OBAVEZA izvorne baze podataka. Kako su podaci u ovoj tabeli na nivou obaveze predmeta, a u ciljnoj tabeli su na nivou tipa obavaze, neophodno je izvršiti operaciju spoja sa tabelama SADRZANA\_OBAVEZA i OBAVEZA\_PREDMETA.
- Da bi se dobile odgovarajuće torke koje treba smestiti u ovu tabelu činjenica (odgovarajući strani ključevi), potrebno je izvršiti operacije spoja i sa dimenzijama DEPARTMANIDIM, STUDENTIDIM, NASTAVNICIDIM, PREDMETIDIM, TIPOBAVEZEDIM, VREMEDIM.
- Poseban tretman je potreban za torke koje imaju prisutnost kao obavezu. Moguće je izvršiti operaciju unije skupova.

# Specifikacija ETL procesa

---

- **Tabela činjenica KANDIDATI\_FACT**
  - Za drugu tabelu činjenica KANDIDATI\_FACT kao izvor podataka se koristi datoteka kandidati.txt kao i dimenzija STUDENT.

# Sadržaj

---

Šema OLTP baze podataka

Šema DW baze podataka

Specifikacija ETL procesa

Implementacija ETL procesa

Materijalizovani pogledi

- mapiranje nastavnici\_dim



- **mapiranje nastavnici\_dim**
  - jednostavno mapiranje gdje je moguće preslikati atribute po poziciji

Attribute Matching ✕

Finds matching attributes in source and target and creates maps. Optionally creates attributes if match is not found

---

**Connection Path**

Connect from NAS\_SRC to NAS\_DIM

---

**Match Options**

By Name

By Position

Ignore Case

---

**Create Actions**

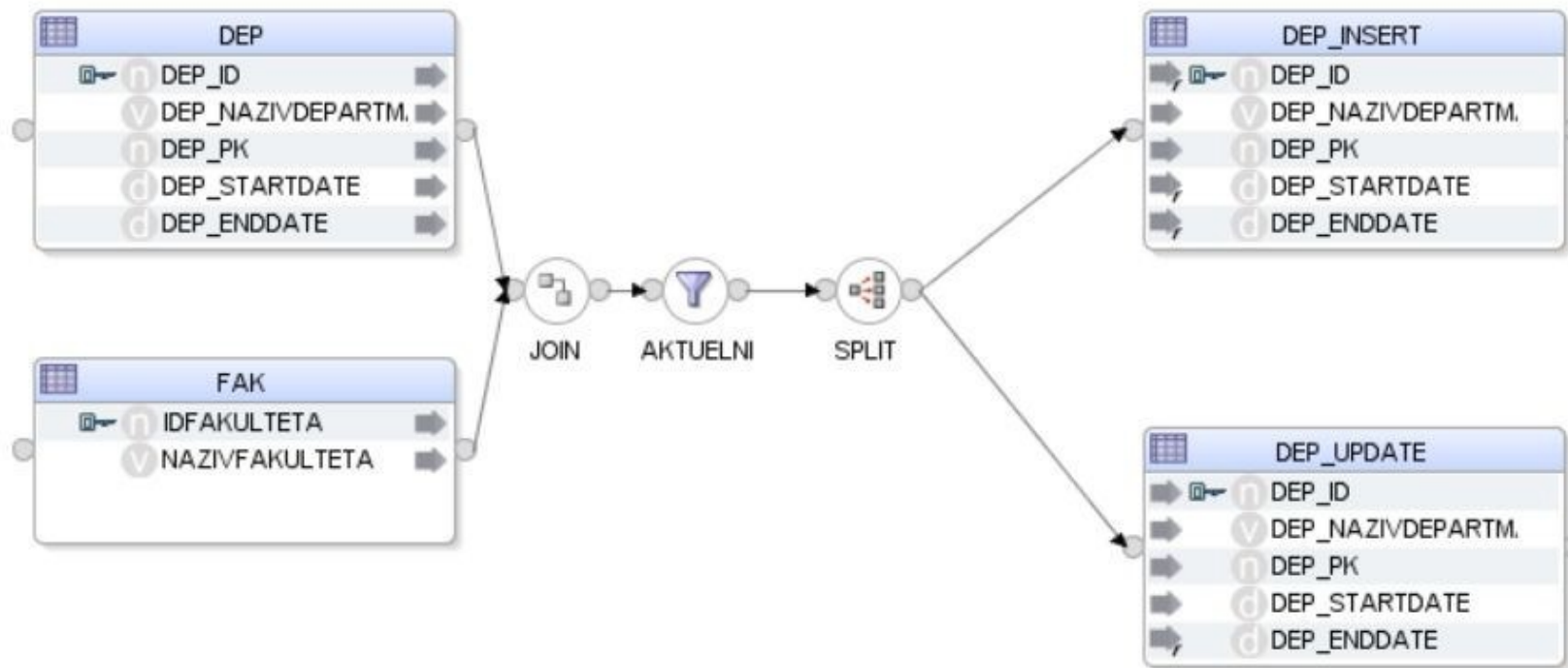
Create Attributes On Source

Create Attributes On Target

Auto Map

Preserve Existing Expressions

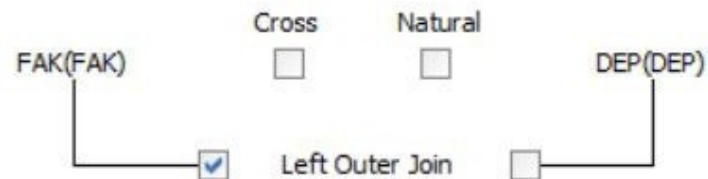
- mapiranje departmani\_dim



- **mapiranje departmani\_dim**

- *JOIN*: Spoj dimenzije DEPARTMANI\_DIM (ova dimenzija čuva ključ iz izvorne baze podataka) i tabele FAKULTET. Left Outer Join - dobiju se torke koje nemaju svog para u dimenziji (nove torke koje još nisu insertovane)

FAK.IDFAKULTETA = DEP.DEP\_PK



All rows of FAK(FAK) including the rows unpaired with DEP(DEP) rows

- **mapiranje departmani\_dim**

- *AKTUELNE*: Zanimaju nas samo aktuelne torke iz dimenzije (kod onih koje čuvaju istorijske podatke *dep\_enddate* je postavljen na datum iz prošlosti)

Filter Condition: `DEP.DEP_ENDDATE IS NULL`

- *SPLIT*: *OUTPUT1* daje nove torke koje je potrebno insertovati u tabelu dimenzije. To su torke iz tabele *FAKULTET* koje nemaju para u dimenziji *DEPARTMANI\_DIM* ili torke kod kojih je došlo do promene naziva u tabeli *FAKULTET*. *OUTPUT2* su torke koje imaju svog para ali je došlo do promene naziva.

Name	De...	Bo...	Splitter Condition	Is Remainder	Connected To
OUTPUT1			DEP.DEP_NAZIVDEPARTMANA is null or DEP.DEP_NAZIVDEPARTMANA != FAK.NAZIVFAKULTETA	<input type="checkbox"/>	DEP_INS...
OUTPUT2			FAK.NAZIVFAKULTETA!= DEP.DEP_NAZIVDEPARTMANA AND DEP.DEP_NAZIVDEPARTMANA IS NOT NULL	<input type="checkbox"/>	DEP_UP...

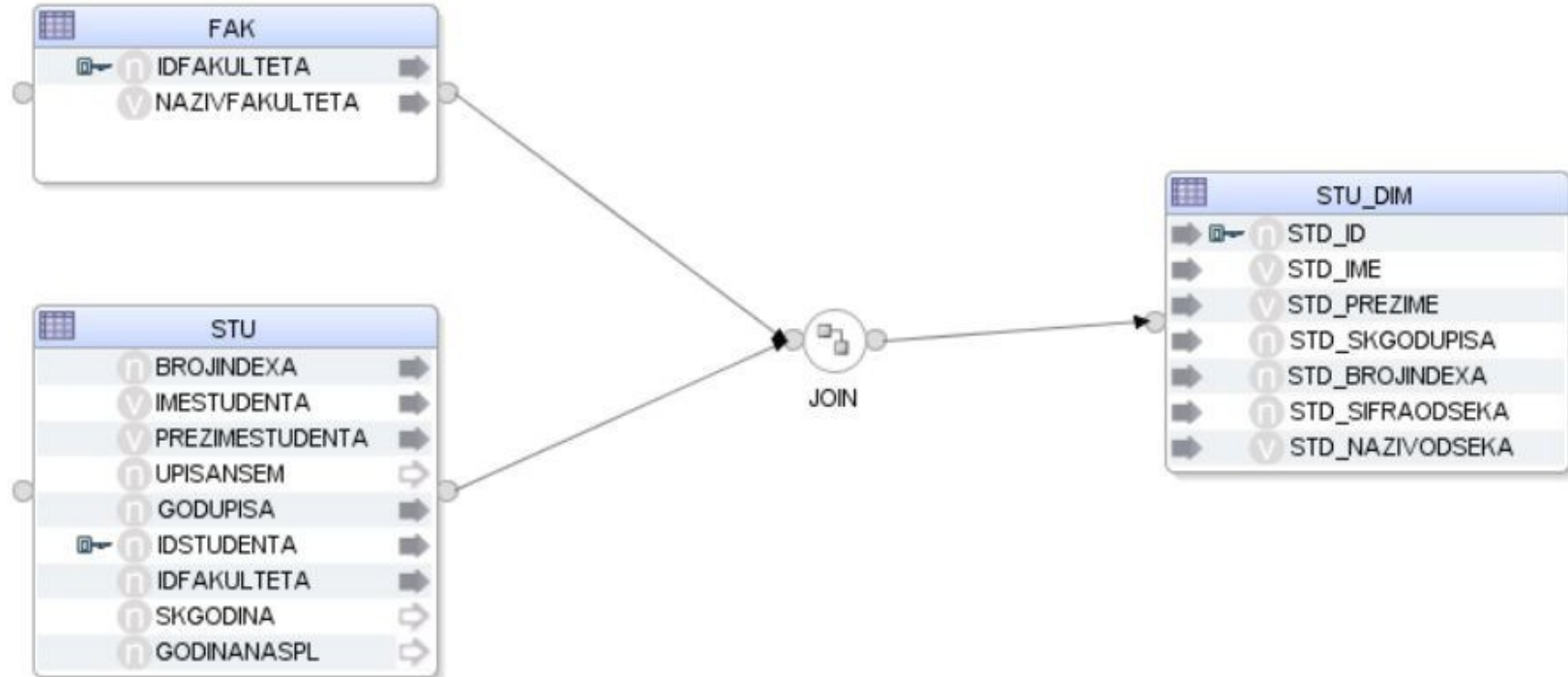
- mapiranje departmani\_dim

- *DEP\_UPDATE* zahteva dodatna podešavanja:

Attributes	Integration Type:	Incremental Update
General	Update Key:	DEP_PK
<b>Target</b>	Maximum Number of Errors Allowed:	
Journalizing	Integration Errors as Percentage:	<input type="checkbox"/>
Constraints		
Connector Points		

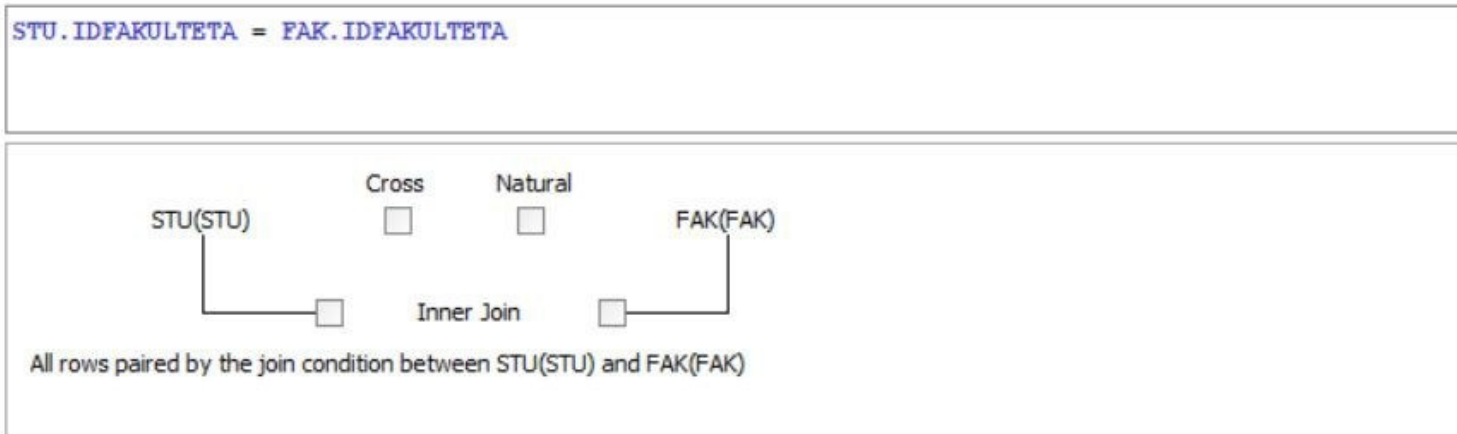
- *Integration Type* je potrebno postaviti na Incremental Update, umesto podrazumevane vrednosti *Slowly Changing Dimension*
- Sve vrednosti *DEP\_UPDATE* su preuzete iz *DEP* osim *DEP\_ENDDATE* -a koji je postavljen na *sysdate*
- Vrednost *DEP\_ID* kod *DEP\_INSERT*-a je postavljena na sledeću vrednost kreiranog sekvencera

- mapiranje studenti\_dim

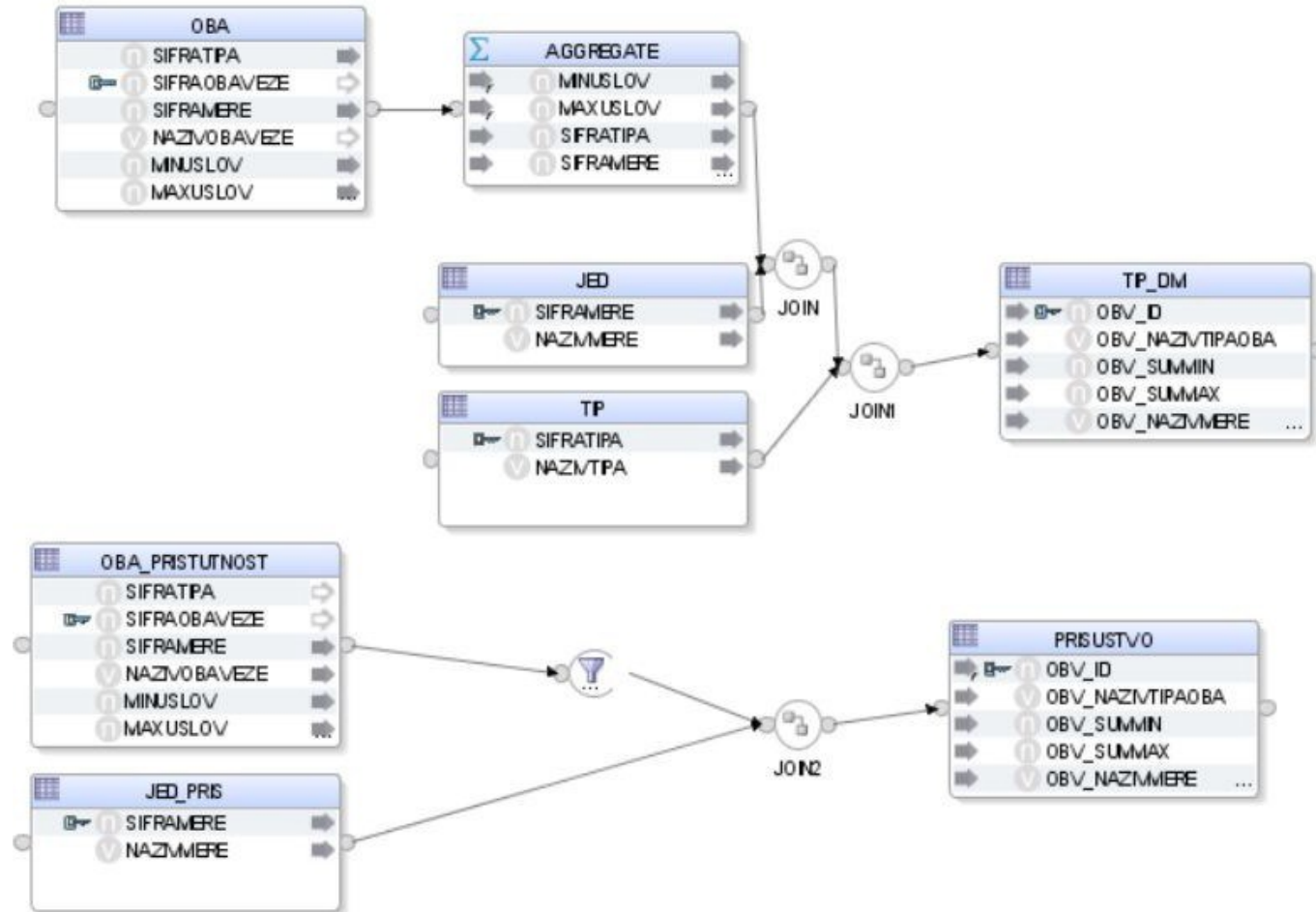


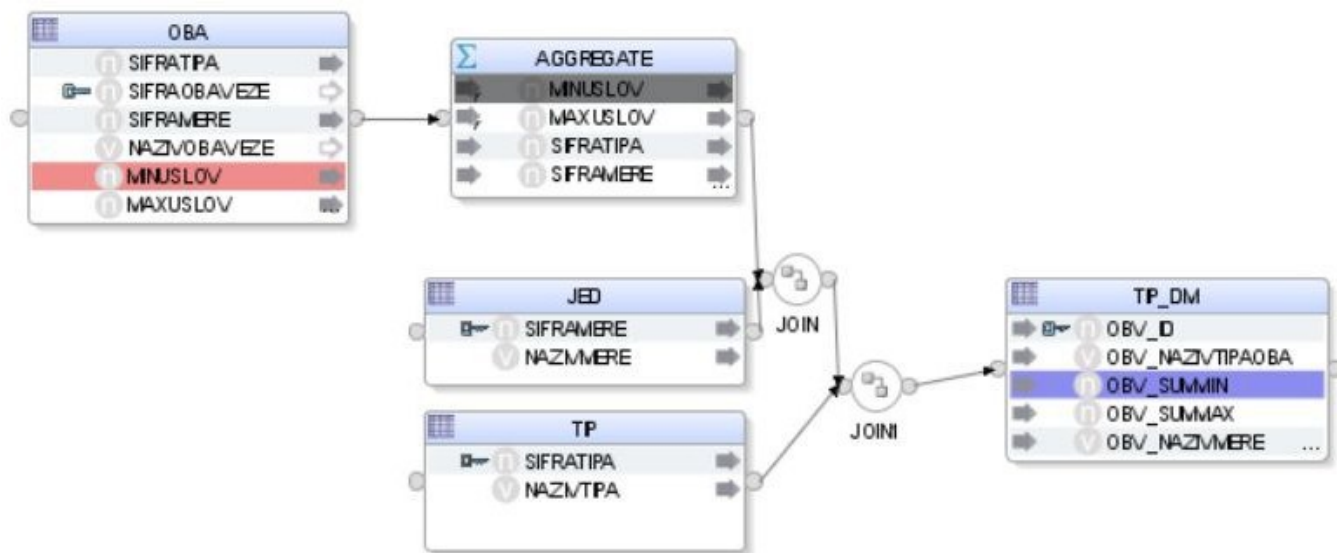
- **mapiranje studenti\_dim**

- *JOIN*: tabele student i fakultet su spojene po koloni *IDFAKULTETA* (Može se odabrati opcija *Natural join*) i *STU\_DIM* preuzima naziv odseka iz tabele fakultet (*NAZIVFAKULTETA*). Ostale vrednosti preuzima iz tabele studenti



- mapiranje tipobaveze\_dim





- **mapiranje tipobaveze\_dim**
  - **AGGREGATE:** Unutar ove komponente potrebno je postaviti da se grupisanje obavlja po vrednostima sifre tipa i sifre mere. Vrednost minuslov-a i maxuslov-a se sumira na nivou te grupe. Ta vrednost se preuzima u dimenziji TIP\_OBAVEZE\_DIM
  - Identifikator tipa obaveze se preuzima iz izvorne table TIP\_OBAVEZE (SIFRATIPA)
  - Da bi se dobio naziv tipa obaveze i naziv mere urađen je spoj torki uz pomoć komponenti *JOIN* i *JOIN1* (redosled je proizvoljan)

Name	Data Type	Length	Scale	Data format	Expression	Execute on Hint	Fixed Executio...	Is Group By
MINUSLOV	NUMERIC	3	0		SUM(OBA.MIN...	No Hint		No
MAXUSLOV	NUMERIC	3	0		SUM(OBA.MA...	No Hint		No
SIFRATIPA	NUMERIC	2	0		OBA.SIFRATIPA	No Hint		Yes
SIFRAMERE	NUMERIC	2	0		OBA.SIFRAME...	No Hint		Yes

**Target**

General

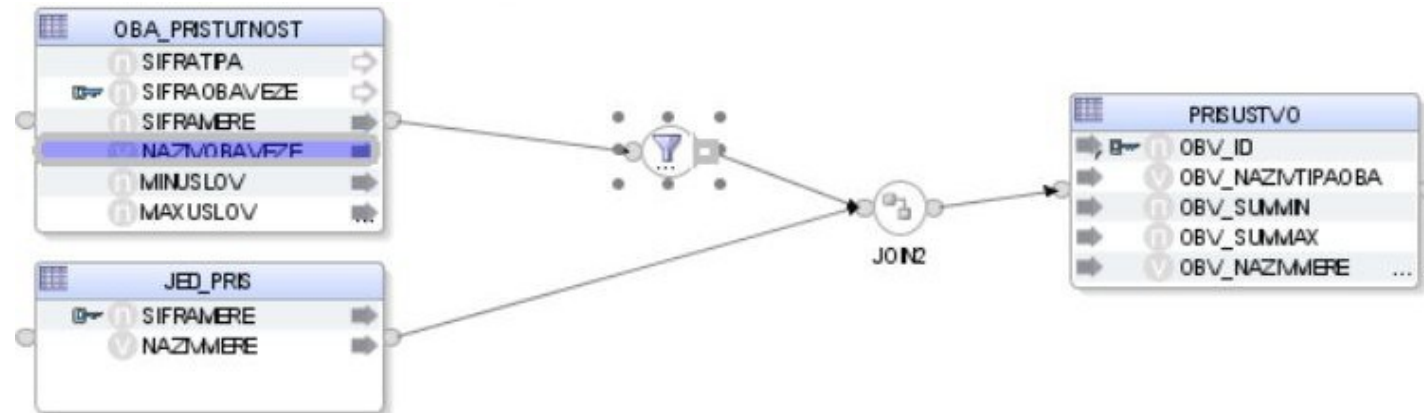
Expression: `SUM(OBA.MINUSLOV)`

Execute on Hint: No Hint

Fixed Execution Location:

- **mapiranje tipobaveze\_dim**

- Specijalni tip obaveze za prisustvo je dobijen filtriranjem tabele OBAVEZA\_PREDMETA po nazivu obaveze (Osim komponente JOIN, mogla se koristiti i komponenta LOOKUP)
- Spoj sa tabelom JEDINICA\_MERE uz pomoć komponente JOIN2 urađen je u cilju dobijanja naziva mere (prirodni spoj po koloni SIFRA\_MERE)



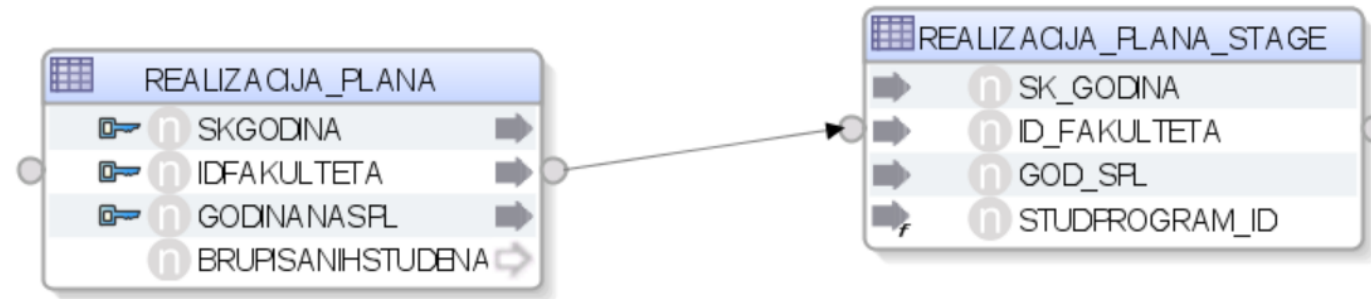
Condition	
General	
Connector Points	
Filter Condition:	<code>OBA_PRISTUTNOST.NAZVOBAVEZE = 'prisutnost'</code>
Technical Description:	<code>[OBAVEZA_PREDMETA (OBA_PRISTUTNOST)].[NAZVOBAVEZE] = 'prisutnost'</code>
Execute on Hint:	No Hint

- **mapiranje tipobaveze\_dim**

- Za vrednost ID specijalnog tipa obaveze prisutnost postavljena je varijabla definisana na nivou projekta. Ovaj način je bolji od postavljanja vrednosti prostim unosom u okviru expression-a jer je ova vrednost relevantna i u ostalim mapiranjima a na ovaj način joj je moguće pristupiti.

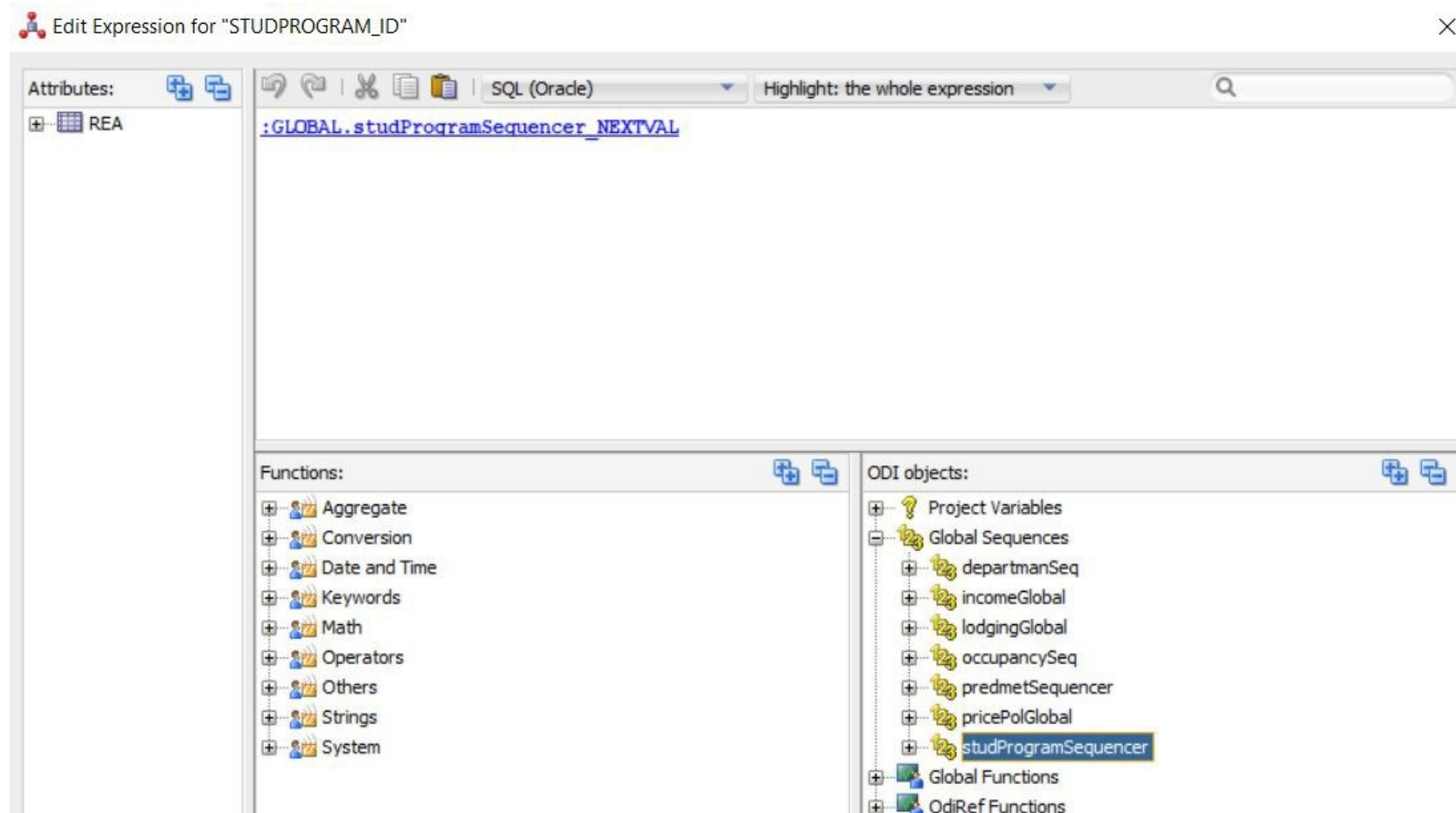
The screenshot shows the 'Edit Expression for "OBV\_ID"' window in Oracle ODI. The main text area contains the expression `#FAKULTET.IdPrisustva`. The 'Attributes' panel on the left lists `JED_PRIS` and `OBA_PRISTUTNOST`. The 'Functions' panel at the bottom left lists various SQL functions like `Aggregate`, `Conversion`, `Date and Time`, `Keywords`, `Math`, and `Operators`. The 'ODI objects' panel at the bottom right shows a tree structure with `Project Variables` containing `IdPrisustva`, and `Global Sequences`, `Global Functions`, and `OdiRef Functions`. A pop-up window for `IdPrisustva` is open, showing the `Refreshing` tab. The 'Schema' is set to `TRG_TestServer_LS` and the 'Select Query' is `select 4 from dual`. Other tabs in the pop-up include `Definition`, `History`, `Markers`, `Memo`, `Version`, and `Privileges`.

- mapiranje realizacija\_plana\_stage



- **mapiranje realizacija\_plana\_stage**

- ova stage tabela će igrati ulogu kod mapiranja dimenzije PREDMETI\_DIM
- preuzima sve kolone tabele REALIZACIJA\_PLANA pri čemu svaka realizacija dobija svoj jedinstveni identifikator umesto kompozitnog ključa



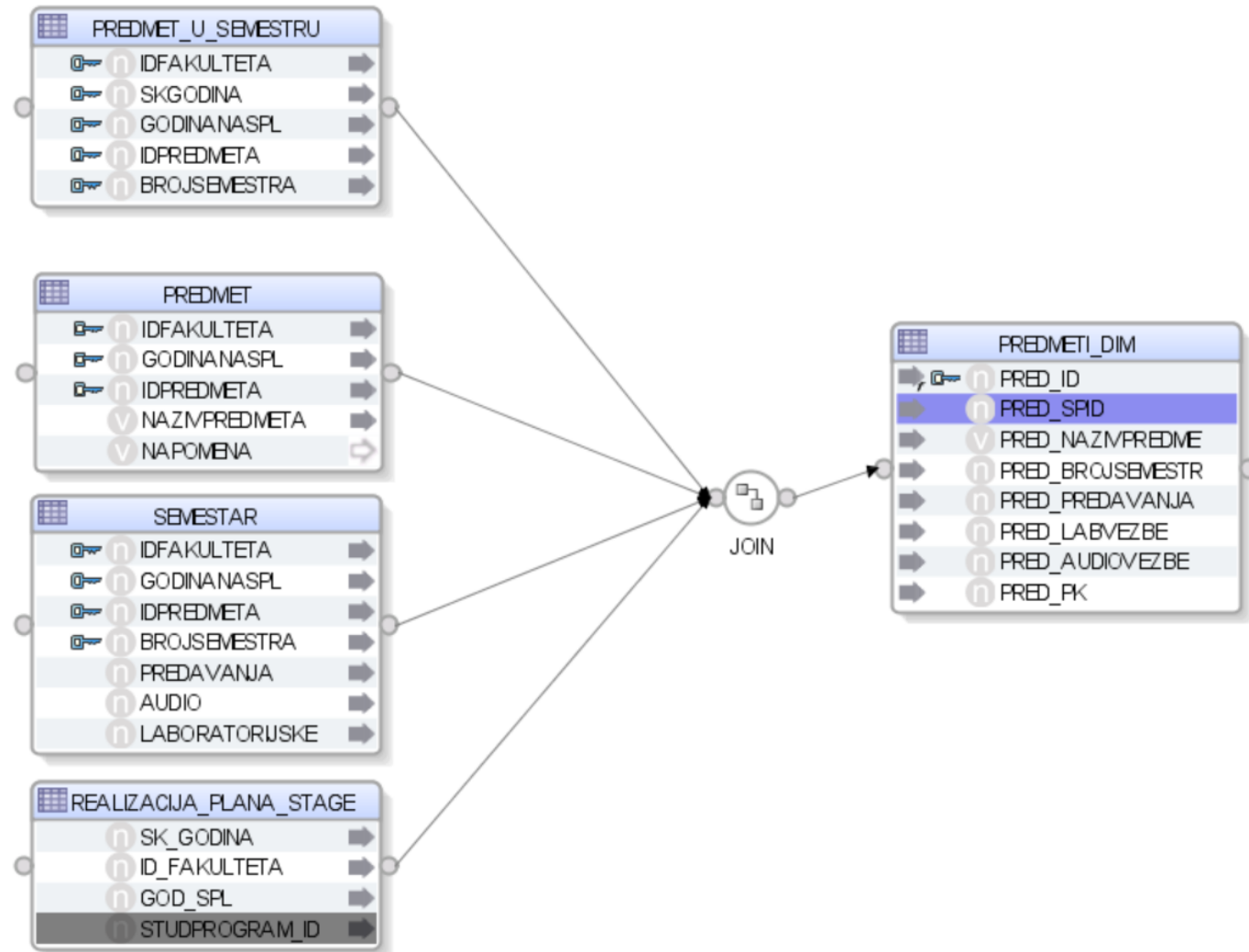
- mapiranje stage\_to\_studprogram



- **mapiranje stage\_to\_studprogram**

- Dimenzija STUD\_PROGRAM\_LEVEL preuzima sve attribute osim identifikatora fakulteta, uključujući i generisanu vrednost ključa iz stage tabele
- Ova vrednost će biti strani ključ u dimenziji PREDMETI\_DIM, pa se na taj način kreira hijerarhija ove dimenzije
- Identifikator fakulteta je neophodan da bi se došlo od predmeta do studijskog programa u izvornoj bazi podataka, a STUD\_PROGRAM\_LEVEL ga sada ne poseduje. Ovo je razlog zašto realizacija plana nije direktno mapirana na STUD\_PROGRAM\_LEVEL, nego na stage tabelu i zašto se u mapiranju dimenzije predmet, koja će biti objašnjena u nastavku, koristi STAGE\_STUDPROGRAM.
- Evidentno je da je redosled izvršavanja ovih mapiranja bitan, pa će ona biti organizovana u okviru paketa, što takođe sledi u nastavku.

- mapiranje predmet\_dim



- mapiranje predmet\_dim

- Za popunjavanje dimenzije predmeta, osim izvorne tabele PREDMET potrebne su i informacije koje se nalaze u tabeli SEMESTAR (broj predavanja, auditornih i laboratorijskih vezbi) kao i toj realizaciji predmeta u semestru. Svaka realizacija predmeta će biti novi red u dimenziji PREDMETI\_DIM sa različitim stranim ključem ka dimenziji studijskog programa, pa je polazna tabela u ovom mapiranju PREDMET\_U\_SEMESTRU.

Join condition:

```
SEM.IDFAKULTETA = PRE.IDFAKULTETA AND  
SEM.GODINANASPL = PRE.GODINANASPL AND  
SEM.IDPREDMETA = PRE.IDPREDMETA AND  
PRE.GODINANASPL = STA.GOD_SPL AND  
PRE.IDFAKULTETA = STA.ID_FAKULTETA AND  
PREU.IDPREDMETA = PRE.IDPREDMETA AND  
PREU.IDFAKULTETA = PRE.IDFAKULTETA AND  
PRE.GODINANASPL = PREU.GODINANASPL AND  
SEM.IDFAKULTETA = STA.ID_FAKULTETA AND  
SEM.GODINANASPL = STA.GOD_SPL AND  
PREU.IDFAKULTETA = SEM.IDFAKULTETA AND  
PREU.GODINANASPL = SEM.GODINANASPL AND  
PREU.IDPREDMETA = SEM.IDPREDMETA AND  
PREU.BROJSEMESTRA = SEM.BROJSEMESTRA AND  
STA.SK_GODINA = PREU.SKGODINA AND  
STA.ID_FAKULTETA = PREU.IDFAKULTETA AND  
PREU.GODINANASPL = STA.GOD_SPL
```

## ● procedura InsertVremeDim

- Ova procedura je zadužena za popunjavanje dimenzije VREME\_DIM.

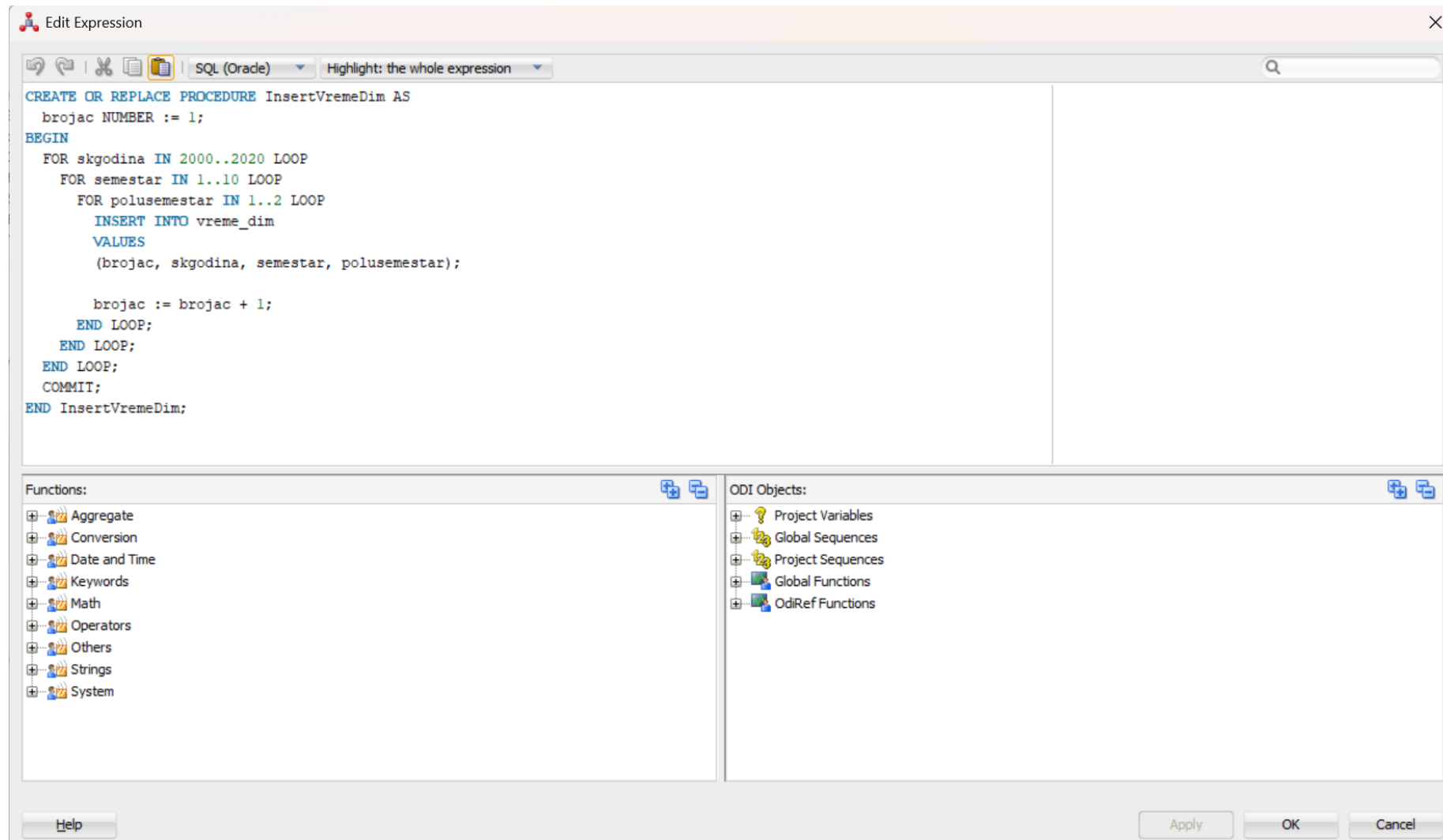
```
CREATE OR REPLACE PROCEDURE InsertVremeDim AS
    brojac NUMBER := 1;
BEGIN
    FOR skgodina IN 2000..2020 LOOP
        FOR semestar IN 1..10 LOOP
            FOR polusemestar IN 1..2 LOOP

                INSERT INTO vreme_dim
                VALUES (brojac, skgodina, semestar, polusemestar);

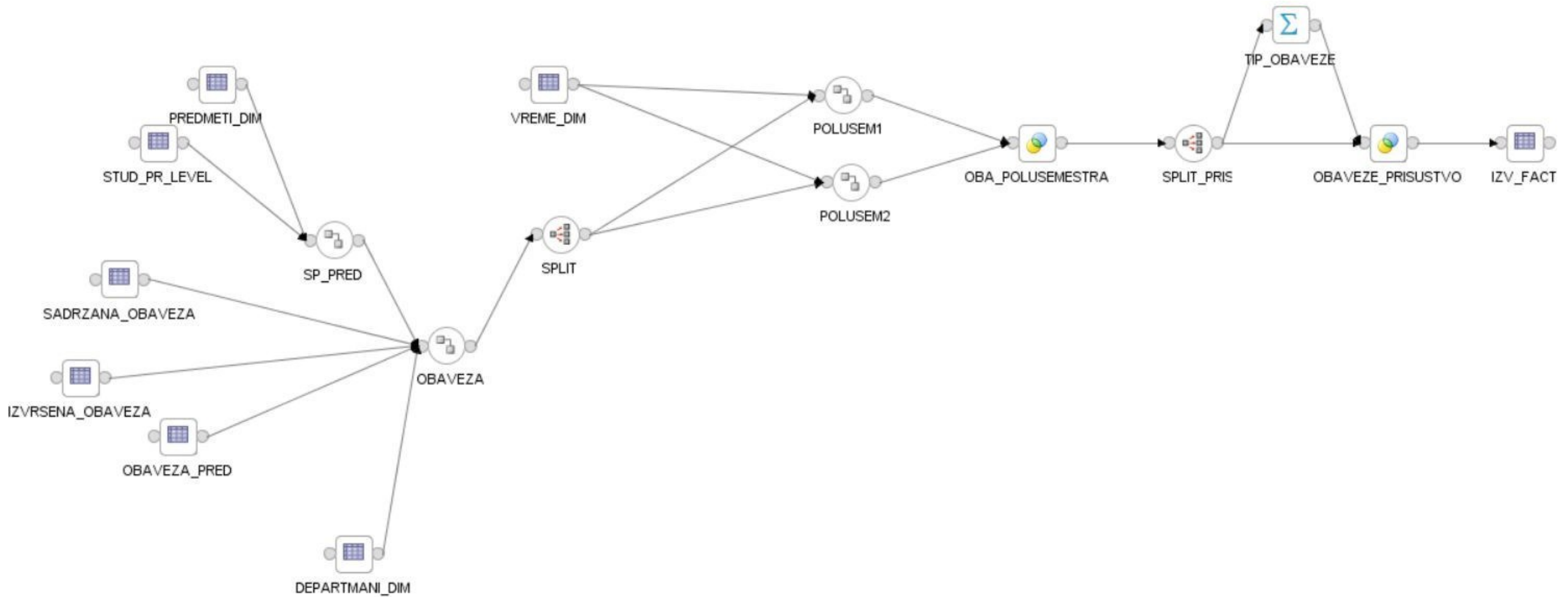
                brojac := brojac + 1;
            END LOOP;
        END LOOP;
    END LOOP;

    COMMIT;
END InsertVremeDim;
/
```

- **procedura InsertVremeDim**
- Kreiranje unutar ODI-ja



- mapiranje izvrsena\_obaveza\_fact



## ● mapiranje izvršena\_obaveza\_fact

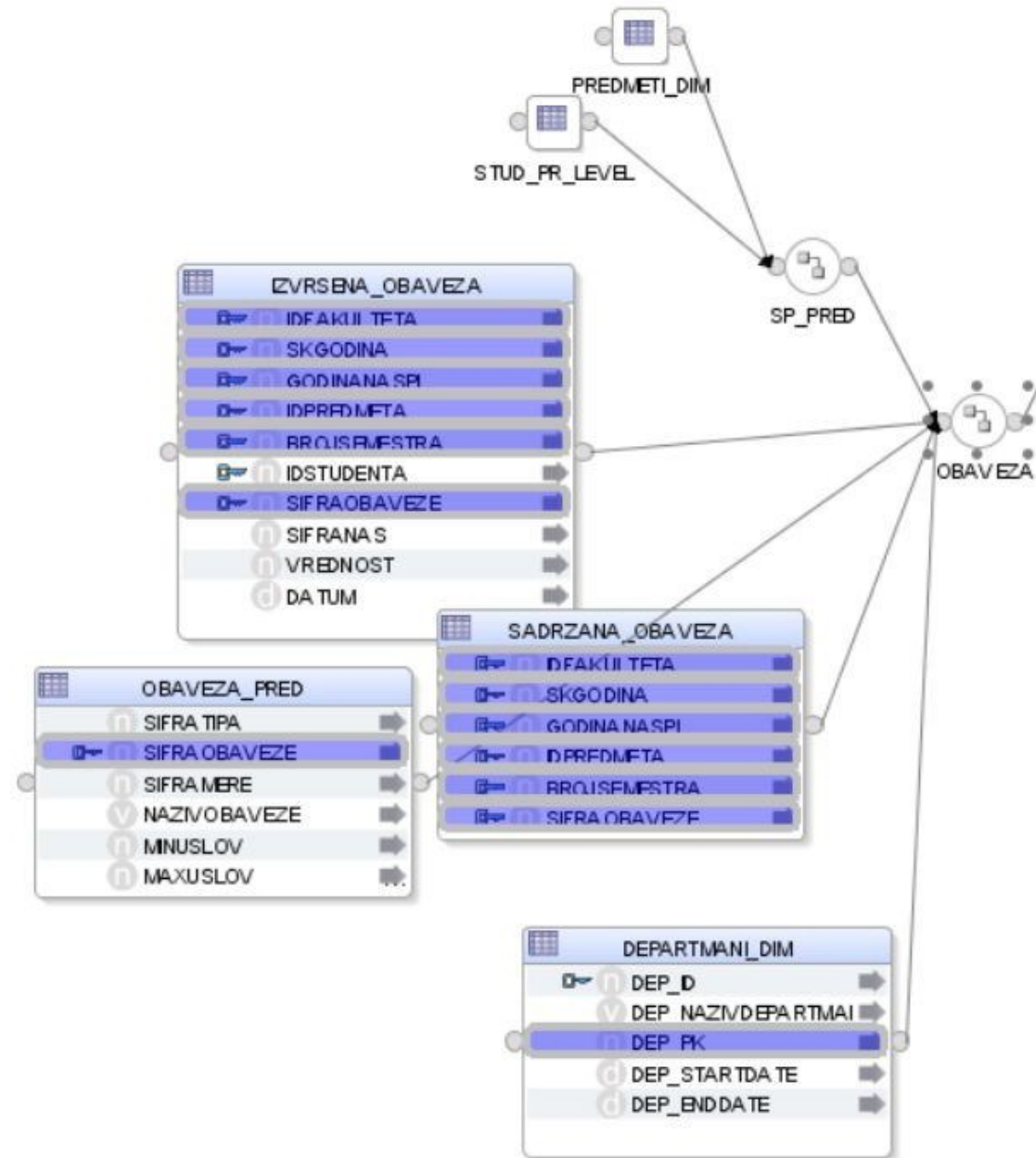
- Osnovna tabela od koje se polazi u ovom mapiranju je IZVRSENA OBAVEZA u izvornoj bazi podataka. Osim nje, potrebne su i informacije o samoj vrsti obaveze (naziv vrste obaveze i sifra tipa obaveze), pa je potrebno uraditi spoj i sa tabelama OBAVEZA\_PREDMETA i SADRZANA\_OBAVEZA
- Prva stvar koju treba imati na umu je granularnost ciljne tabele činjenica i granularnost izvorne tabele. U izvornoj tabeli svaki red se odnosi na jednu vrstu obaveze, dok se u tabeli činjenica red odnosi na tip obaveze sa izuzetkom vrste obaveze *prisustvo* koje treba da postane tip obaveze u Data Warehouse sistemu. Za postizanje željene granularnosti iskorišćena je AGGREGATE komponenta
- Tabela činjenica sadrži strane ključeve ka dimenzijama: VREME\_DIM, TIPOBAVEZE\_DIM, NASTAVNICI\_DIM, PREDMETI\_DIM, STUDENTI\_DIM i DEPARTMANI\_DIM
- Primarni ključ dimenzije NASTAVNICI\_DIM je isti u Data Warehouse sistemu i izvornoj bazi podataka pa se ova tabela dimenzije uopšte ne koristi (Potreban je samo identifikator nastavnika koji se preuzme iz tabele IZVRSENA OBAVEZA). Isto važi i za dimenziju STUDENTI\_DIM
- Za dimenziju PREDMETI\_DIM potrebno je uraditi spoj sa odgovarajućim studijskim programom, pošto studijski program sadrži školsku godinu i semestar, što je neophodno za spoj sa torkama iz izvornih tabela
- Dimenzija DEPARTMANI\_DIM osim svog novog ključa, koji dobija prilikom inserta u Data Warehouse sistem, čuva informaciju o ključu departmana iz izvorne baze podataka. Zbog već objašnjene istorije izmena koja se čuva u ovoj dimenziji, više redova može imati istu vrednost tog ključa što može dovesti do neželjenog spoja sa više redova ove tabele dimenzije
- Pošto traženje odgovarajućeg reda u dimenziji vremena nije moguće samo uz pomoć semestra i školske godine (za svaki semestar školske godine postoje dva polusemestra), potrebno je posmatrati datum kada je izvršena obaveza pa tu postoje dva slučaja

- mapiranje izvrsena\_obaveza\_fact

- SP\_PRED: Prirodni spoj za ovu hijerarhijsku dimenziju
- OBAVEZA: svi uslovi spoja su prikazani na slici. Problem sa istorijom kod dimenzije za departmane je rešen uslovom DEPARTMANI\_DIM.DEP\_ENDDATE is null.

```

IZVRSENA_OBAVEZA.IDFAKULTETA = SADRZANA_OBAVEZA.IDFAKULTETA AND
IZVRSENA_OBAVEZA.SKGODINA = SADRZANA_OBAVEZA.SKGODINA AND
IZVRSENA_OBAVEZA.GODINANASPL = SADRZANA_OBAVEZA.GODINANASPL AND
IZVRSENA_OBAVEZA.IDPREDMETA = SADRZANA_OBAVEZA.IDPREDMETA AND
IZVRSENA_OBAVEZA.BROJSEMESTRA = SADRZANA_OBAVEZA.BROJSEMESTRA AND
IZVRSENA_OBAVEZA.SIFRAOBAVEZE = SADRZANA_OBAVEZA.SIFRAOBAVEZE AND
SADRZANA_OBAVEZA.SIFRAOBAVEZE = OBAVEZA_PRED.SIFRAOBAVEZE AND
DEPARTMANI_DIM.DEP_PK = IZVRSENA_OBAVEZA.IDFAKULTETA AND
DEPARTMANI_DIM.DEP_ENDDATE is null AND
STUD_PR_LEVEL.SP_SKGODPRED = SADRZANA_OBAVEZA.GODINANASPL AND
STUD_PR_LEVEL.SP_GODSTUDPROGRAMA = SADRZANA_OBAVEZA.SKGODINA AND
PREDMETI_DIM.PRED_PK = IZVRSENA_OBAVEZA.IDPREDMETA
    
```



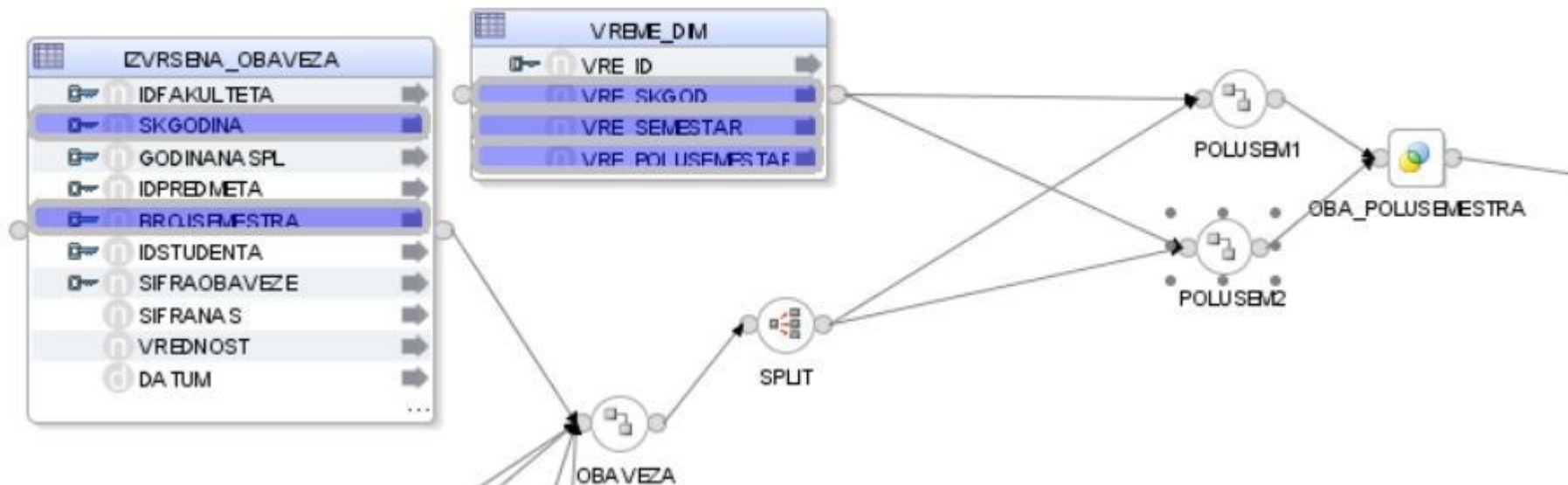
- **mapiranje izvrsena\_obaveza\_fact**

- SPLIT: Sve torke OBAVEZA su podeljene na one koje su se desile u prvom i one koje su se desile u drugom polusemestru. Primjer za drugi polusemestar:

```
(EXTRACT (MONTH FROM IZVRSENA_OBAVEZA.DATUM) IN (12, 1) OR  
(EXTRACT (MONTH FROM IZVRSENA_OBAVEZA.DATUM) = 11 AND EXTRACT (DAY FROM IZVRSENA_OBAVEZA.DATUM) >= 16))  
OR  
(EXTRACT (MONTH FROM IZVRSENA_OBAVEZA.DATUM) BETWEEN 5 AND 9 OR  
(EXTRACT (MONTH FROM IZVRSENA_OBAVEZA.DATUM) = 4 AND EXTRACT (DAY FROM IZVRSENA_OBAVEZA.DATUM) >= 16))
```

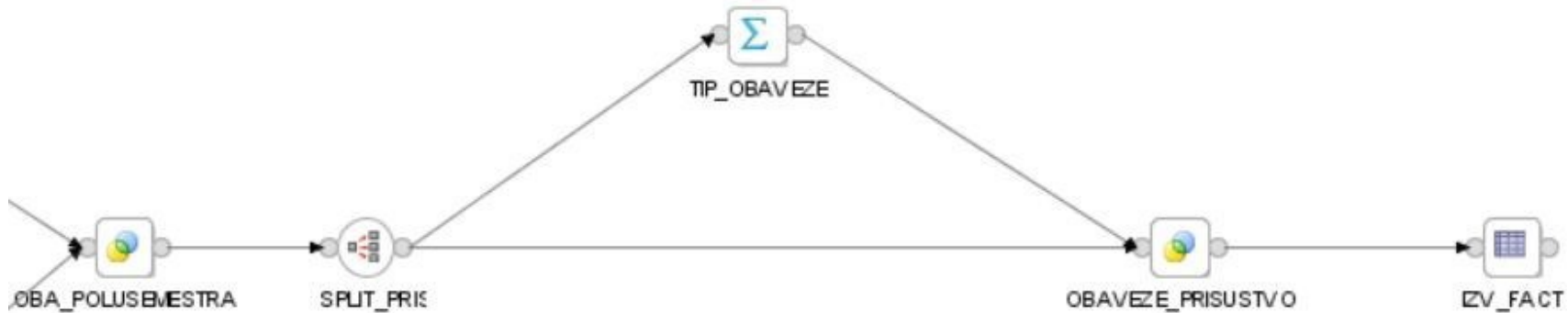
- POLUSEM2: Sada je moguće dobiti odgovarajući red iz VREME\_DIM proverom semestra i godine:

```
VREME_DIM.VRE_SEMESTAR = IZVRSENA_OBAVEZA.BROJSEMESTRA AND  
VREME_DIM.VRE_SKGOD = IZVRSENA_OBAVEZA.SKGODINA AND VREME_DIM.VRE_POLUSEMESTAR = 2
```



- **mapiranje izvršena\_obaveza\_fact**

- Sve torke su ponovo unirane sa tom razlikom što je svaka torka spojena sa odgovarajućim redom iz dimenzije vremena (imamo potrebni VRE\_ID).



- SPLIT\_PRISUTNOST: Sada je potrebno ponovo podeliti torke, ako ovaj put dobijeni skupovi nisu disjunktni. Sve torke idu na agregaciju po tipu obaveze pa je potrebno postaviti uslov koji je tačan za svaku torku (naziv obaveze nije nedostajuća vrednost). Ovaj skup predstavlja ulaz za komponentu agregacije TIP\_OBAVEZE. Drugi skup ide direktno na komponentu unije OBAVEZE\_PRISUSTVO kao jedan od ulaza.

General

**Split Conditions**

Input Connector Points    Output Connector Points

Output Connector Points:

Name	De...	Bo...	Splitter Condition	Is ...	Con...
OU...			OBA_POLUSEMESTRA.NAZIVOBAVEZE = 'prisutnost'	<input type="checkbox"/>	...
OU...			OBA_POLUSEMESTRA.NAZIVOBAVEZE IS NOT NULL	<input type="checkbox"/>	...

- **mapiranje izvrsena\_obaveza\_fact**

- Opcije za komponentu agregacije su na slici. Atributi po kojima se grupiše su svi osim same vrednosti ocene nad kojom je primenjena skupovna funkcija SUM. Ovim je postignuta željena granularnost.

**Attributes**

General

Connector Points

Attributes:

Name	Data Type	Length	Scale	Data for...	Expression	Execute ...	Fixed Ex...	Is Group...
VRE_ID	NUMERIC				OBA_POLUSEMESTRA.VRE_ID	No Hint		Yes
STD_ID	NUMERIC	6	0		OBA_POLUSEMESTRA.IDSTUDENTA	No Hint		Yes
DEP_ID	NUMERIC				OBA_POLUSEMESTRA.DEP_ID	No Hint		Yes
SIFRANAS	NUMERIC	3	0		OBA_POLUSEMESTRA.SIFRANAS	No Hint		Yes
PRED_ID	NUMERIC				OBA_POLUSEMESTRA.PRED_ID	No Hint		Yes
SIFRATIPA	NUMERIC	2	0		OBA_POLUSEMESTRA.SIFRATIPA	No Hint		Yes
VREDN...	NUMERIC	2	0		SUM(OBA_POLUSEMESTRA.VREDNO...	No Hint		No

- Posto je zahtev da se prisutnost agregira kao deo predispitne obaveze ali i da se posmatra kao samostalan tip, potrebno je izabrati opciju UNION ALL, a ne UNION prilikom uniranja agregiranih torki i torki koje se odnose na prisustvo. Može se desiti da je ta prisutnost jedina predispitna obaveza za nekog studenta pa se dva ista reda na ovaj način ne ponište.

**Attributes**

General

**Operators**

Input Connector Points    Output Connector Points

Input Connector Points:

Name	Description	Bound Object	Connected From	Operator
INPUT1			TIP_OBAVEZE	
INPUT2			SPLIT_PRISUTNOST.OUTP...	UNION ALL

- **mapiranje izvršena\_obaveza\_fact**

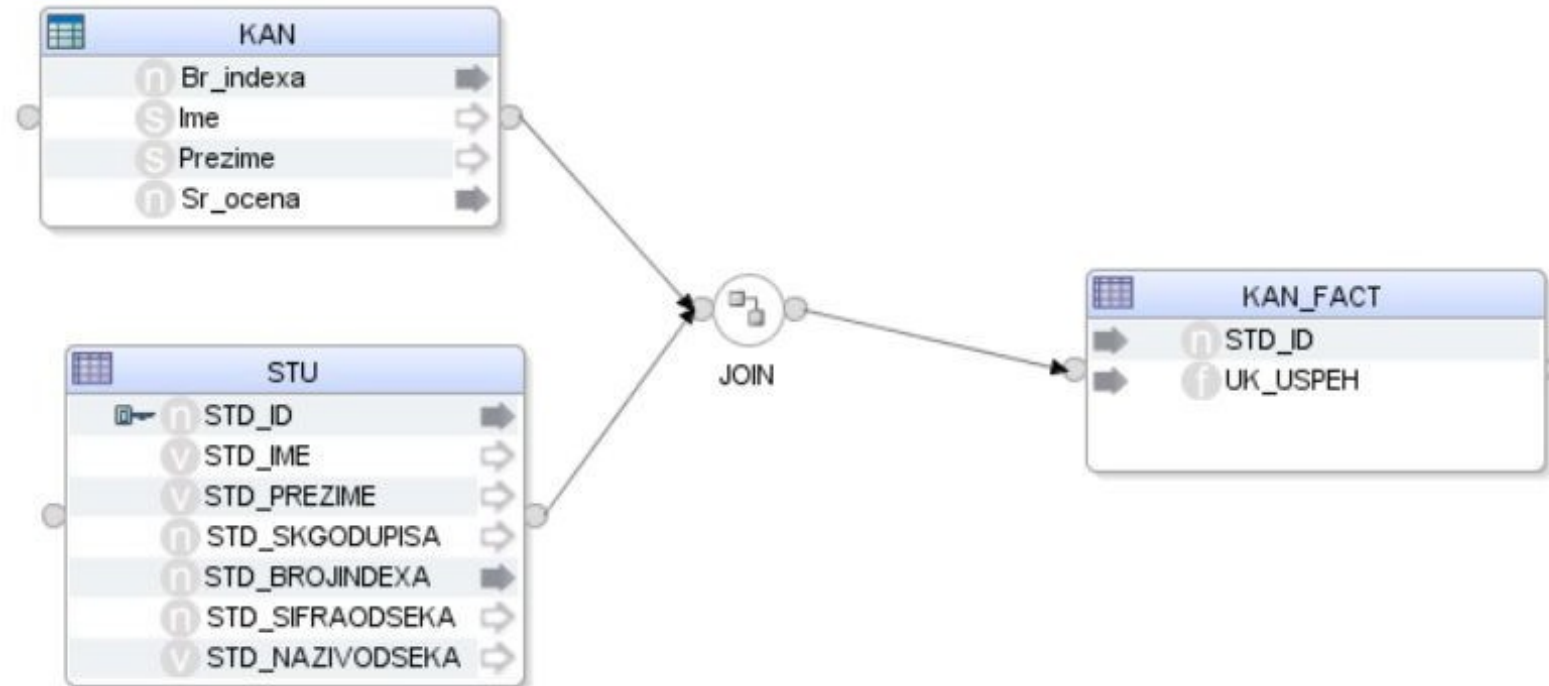
- OBAVEZE\_PRISUSTVO: svi atributi iz ove komponente idu direktno na ciljnu tabelu činjenica. Kolona šifra tipa obaveze se unira sa vrednosti varijable na nivou projekta *IdPrisustva*.

Attributes		
General	Attributes:	
Operators	Target	Expression for : INPUT1 (TIP_OBAVEZE)
	VRE_ID	TIP_OBAVEZE.VRE_ID
	DEP_ID	TIP_OBAVEZE.DEP_ID
	SIFRANAS	TIP_OBAVEZE.SIFRANAS
	PRED_ID	TIP_OBAVEZE.PRED_ID
	SIFRATIPA	TIP_OBAVEZE.SIFRATIPA
	VREDNOST	TIP_OBAVEZE.VREDNOST
	STD_ID	TIP_OBAVEZE.STD_ID

- Konačne vrednost tablele činjenica:

Attributes												
General	Attributes:											
Target	Name	Data T...	Length	Scale	Da...	Expression	Execut...	Fixed ...	Key	Insert	Update	Check ...
Journalizing	OBV_ID	NUMBER				OBAVEZE_PRISUSTVO.SIFRATIPA	No Hint		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Constraints	NAS_ID	NUMBER				OBAVEZE_PRISUSTVO.SIFRANAS	No Hint		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Connector Points	PRED_ID	NUMBER				OBAVEZE_PRISUSTVO.PRED_ID	No Hint		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	VRE_ID	NUMBER				OBAVEZE_PRISUSTVO.VRE_ID	No Hint		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	STD_ID	NUMBER				OBAVEZE_PRISUSTVO.STD_ID	No Hint		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	DEP_ID	NUMBER				OBAVEZE_PRISUSTVO.DEP_ID	No Hint		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	IO_VR...	NUMBER				OBAVEZE_PRISUSTVO.VREDNOST	No Hint		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

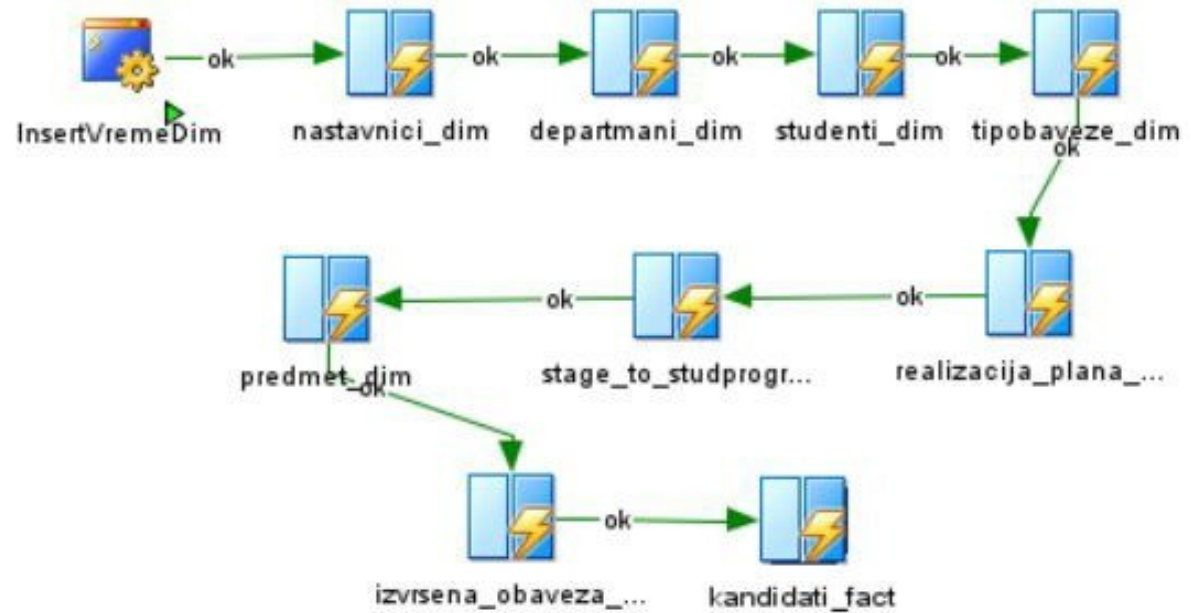
- mapiranje kandidati\_fact



- **mapiranje kandidati\_fact**

- Spoj između dimenzije STUDENTI\_DIM i fajla kandidati.txt odrađen je po broju indeksa studenta.
- Id studenta za tabelu činjenica je preuzet iz tabele dimenzije, a srednja ocena iz fajla.

- paket za izvršavanje celokupnog etl procesa



# Sadržaj

---

Šema OLTP baze podataka

Šema DW baze podataka

Specifikacija ETL procesa

Implementacija ETL procesa

Materijalizovani pogledi

# Materijalizovani pogledi

---

- Projektovani Data Warehouse sistem sadrži dva materijalizovana pogleda MVNASPRED i MVPREDVREME.
  - MVNASPRED je materijalizovani pogled nad dimenzijama NASTAVNICIDIM i PREDMETIDIM. U njemu se čuvaju podaci vezani za analize izvršenih obaveza grupisani po predmetima i nastavnom osoblju koje drži te predmete. Podaci za ovaj materijalizovani pogled se dobijaju iz tabele činjenica IZVRSENA\_OBAVEZA. Pored stranih ključeva iz dimenzija sa kojima je povezan, on trenutno sadrži četiri kolone. Za svaku torku čuva se: procenat desetki i šestica, srednju ocenu i odstupanje od srednje ocene. Do ovih vrednosti se dolazi putem matematičkih operacija, a detaljan postupak će biti objašnjen u nastavku. Ukoliko je korisniku jednog ovakvog sistema potrebna još neka analiza nad ove dve dimenzije, to se rešava dodavanjem nove kolone.

# Materijalizovani pogledi

---

- Projektovani Data Warehouse sistem sadrži dva materijalizovana pogleda MVNASPRED i MVPREDVREME.
  - MVPREDVREME je materijalizovani pogled nad dimenzijama PREDMETIDIM i VREMEDIM. Čuva podatke koji se mogu koristiti za analizu izvršenih obaveza na predmetima u vremenu. Podaci neophodni, da bi ova tabela mogla biti popunjena preuzimaju se iz tabele činjenica IZVRSENA\_OBAVEZA. U njemu se pored stranih ključeva iz dimenzija nad kojima je definisan, trenutno nalazi jedna kolona u kojoj se čuva procenat prisutnosti na predmetima. Kao i u prethodnom slučaju i ovaj materijalizovani pogled je moguće proširiti kolonom, koja korisniku daje uvid u neke složenije analize.

# Materijalizovani pogledi

---

- **Popunjavanje materijalizovanih pogleda**

- Za materijalizovani pogled MVNASPRED, svi neophodni podaci, nalaze se u tabeli činjenica IZVRSENE\_OBAVEZE\_FACT. Kako su sve analize koje se vrše vezane za krajnje ocene studenata, najpre treba izdvojiti torke čiji je tip obaveze 'zavrсна'. Kako je ovaj materijalizovani pogled nad dimenzijama NASTAVNICIDIM i PREDMETIDIM, potrebno je grupisati selektovane torke po šifri nastavnika i šifri predmeta. Iz ovako dobijenog skupa podataka mogu se izračunati srednje vrednosti ocena, odstupanje od srednje ocene i ukupan broj krajnjih ocena po nastavniku i predmetu. Prosečna ocena se računa kao količnik ukupnog zbira ocena i broja ocena. Za odstupanje se može iskoristiti funkcija *stddev()*.

# Materijalizovani pogledi

---

```
CREATE MATERIALIZED VIEW MVNASPRED AS
SELECT obv.nas_id, obv.pred_id,
       ROUND((SELECT COUNT(*)
              FROM izvrsena_obaveza_fact
              WHERE nas_id = obv.nas_id AND pred_id = obv.pred_id AND
                    io_vrednost = 10) / COUNT(*), 2) AS procenat10,
       ROUND((SELECT COUNT(*)
              FROM izvrsena_obaveza_fact
              WHERE nas_id = obv.nas_id AND pred_id = obv.pred_id AND
                    io_vrednost = 6) / COUNT(*), 2) AS procenat6,
       ROUND(AVG(obv.io_vrednost), 2) AS srednjaOcena,
       ROUND(STDDEV(obv.io_vrednost), 2) AS odstupanje
FROM tipobaveze_dim NATURAL JOIN izvrsena_obaveza_fact obv
WHERE obv.nazivtipaobaveze = 'zavrsna'
GROUP BY obv.nas_id, obv.pred_id;
```

# Materijalizovani pogledi

---

- **Popunjavanje materijalizovanih pogleda**
  - Materijalizovani pogled MVPREDVREME kao izvor podataka takođe koristi tabelu činjenica IZVRSENE\_OBAVEZE\_FACT. Kako su podaci u tabeli činjenica izraženi u poenima, potrebno ih je prevesti u procenite. Kako je potrebno izvršiti analizu prisutnosti po predmetima u vremenu, selektuju se torke iz tabele činjenica IZVRSENE\_OBAVEZE\_FACT, koje imaju tip obaveze 'prisutnost'. Da bi se dobio maksimalan broj poena koje studenti mogu dobiti za ovu vrstu obaveze, potrebna je kolona *obv\_summax* dimenzije *tipobaveze\_dim*.

# Materijalizovani pogledi

---

```
CREATE MATERIALIZED VIEW MVPREDVREME AS
SELECT
    vre_id,
    pred_id,
    SUM(io_vrednost) / COUNT(*) *
        (SELECT obv_summax FROM tipobaveze_dim
         WHERE obv_nazivtipaobaveze = 'prisutnost')
        AS ProcenatPrisutnosti
FROM izvrsena_obaveza_fact NATURAL JOIN tipobaveze_dim
WHERE obv_nazivtipaobaveze = 'prisutnost'
GROUP BY
    vre_id,
    pred_id;
```

# DW sistem za obračunje nastavnog procesa

---

SISTEMI SKLADIŠTA PODATAKA

*Kraj prezentacije*