Univerzitet u Novom Sadu
FAKULTET TEHNIČKIH NAUKA

Departman za industrijsko inženjerstvo i inženjerski menadžment
Inženjerstvo informacionih sistema

PROJEKTOVANJE SKLADIŠTA PODATAKA

EUROPEAN SOCCER LEAGUES

Saša Ilić,IT4/2019

Novi Sad, 2022.

SADRŽAJ

ZADATAK I CILJEVI PROJEKTA

Osnovni zadatak projekta je kreiranje skladišta podataka namenjenog za pohranjivanje podataka koji se tiču odigranih utakmica u evropskim fudbalskim ligama,kao i za kasnije kreiranje raznovrsnih izveštaja na osnovu prikupljenih podataka. Jedan od prvih zadataka,koji u velikoj meri omogućava kasniju lakšu izradu projekta jeste bilo određivanje koje vrednosti iz izvora podataka je potrebno sačuvati i od njih napraviti odgovarajuće dimenzije i tabelu činjenica,a koje je potrebno odbaciti kao nerelevantne,odnosno na taj način potrebno je odrediti šta će biti fokus projekta,te na čemu će se kasnije zasnivati analize.

Cilj projekta je da se na osnovu prikupljenih podataka i kreiranog skladišta podataka, pre svega omogući jednostavnije kreiranje korisnih i interaktivnih izveštaja,koji će imati za cilj da prikažu stanje podataka vezanih za odigrane utakmice u evropskim fudbalskim ligama,te da se omogući lakša analiza relevantnih podataka,koji se tiču kako odigranih utakmica,tako i timova koji su učestvovali u tim utakmicama.

Oblast odabranog projekta obuhvata veoma detaljne podatke vezane za odigrane utakmice izmedju timova u jedanaest po kvalitetu najjačih evropskih fudbalskih liga.Razlog odabira ove teme leži pre svega u zainteresovanosti ovom temom,kao i popularnosti fudbala,koji prema nekim statistikama predstavlja najpopularniji sport na planeti,koji je praćen od ogromnog broja ljudi. Svakodnevno se odigra veliki broj utakmica na kojima se dešavaju različite aktivnosti,poput postizanja golova,dobijanja javnih opomena u vidu žutih ili crvenih kartona,izvodjenja kornera,fauliranja protivničkih igrača. Analiziranje ovih dešavanja po utakmicama mogu biti zanimljive kako i običnim navijačima koji žele da steknu uvid u statistike svojih omiljenih timova,tako i različitim zaposlenim kadrovima u ovoj oblasti,kako bi mogli da analiziraju spomenuta dešavanja i pomoću njih steknu bolji uvid i dublje razumevanje u razloge pojedinih dešavanja. Iz ovoga se može zaključiti da ova oblast ima popriličan potencijal za pre svega izradu skladišta podataka,te kasnije postavljanje raznih zanimljivih upita i pravljenje izveštaja koji mogu u velikoj meri da olakšaju uvid u sve timove,njihove odigrane utakmice kao i ostvarene uspehe na tim utakmica.

U nastavku dokumentacije biće detaljno objašnjen postupak kreiranja skladišta podataka,počevši od samog izbora dataset-a,kreiranja odgovarajuće OLAP šeme na osnovu koje će kasnije i biti implementirano skladište podataka,kao i sam ETL postupak koji za cilj ima transformaciju i pročišćavanje izvornih podataka i njihovo punjenje u kreirano skladište podataka. Na kraju,biće dat grafički prikaz i objašenjenje kreiranih izveštaja,koji predstavljaju odgovore na specificirane korisničke zahteve.

OPIS POSTUPKA KREIRANJA DW SISTEMA

Postupak projektovanja DW sistema započinje pretraživanjem i pronalaženjem odgovarajućeg seta podataka na Internetu. Ovo je iako možda ne toliko vremenski zahtevan,ali jedan od najbitnijih koraka u izradi projekta,jer kvalitet odabranog seta podataka(u smislu pogodnosti pripadajućih podataka za analizu) u velikoj meri utiče na dalji tok izrade projekta,kao i na sam kvalitet kreiranog skladišta podataka. Odabrani set podataka,na osnovu kojeg je vršena izrada projekta je javan i svako može da pristupi podacima koji se nalaze u ovom setu podataka. Izvorni set podataka se nalazi u sqlite formatu,što je predstavljalo određenu sitnu prepreku izradi projekta, u smislu što instalirani SSIS i SSRS paketi u okviru programskog okruženja Visual Studio, nisu predvidjeni za rad sa ovakvim tipom podataka. Treba napomenuti da je sqlite format zapravo pojednostavljeni sql,te su se podaci u izvornom setu podataka nalazili u tabelama sa pripadajućim primarnim i stranim ključevima,što je kasnije poprilično olakšalo izradu projekta. Zbog toga,bilo je potrebno nakon njegovog preuzimanja i razumevanja podataka, koristeći jedno od okruženja koje podržava rad sa sqlite formatom,razvijeno od strane Jet Brainsa - Data Grip, prebaciti u format sa kojim su gore spomenuti paketi pomoću kojih se vrši etl proces i kreiraju izveštaji kompatibilni. Zbog lakoće i jednostavnosti rada sa datim tipom podatka,izabran je CSV format,te su tabele koje su se nalazile u izvornom setu podatka,prebačene u ovaj format. Koristeći spomenuto programsko okruženje - Data Grip, već u startu je izvršena selekcija onih obeležja koja će biti relevantna za izradu projekta i kreiranje DW sistema,te csv fajlovi koji se nalaze u prilogu fajla Csv files,zapravo sadrže u određenoj meri,inicijalno n pročišćene podatke. Dodatno pročišćavanje podataka i njihova transformacija su vršeni kroz ETL proces,što će biti detaljnije objašnjeno u nastavku dokumentacije. Nakon toga usledilo je sagledavanje problema i definisanje zadatataka i ciljeva. Definisanje zadataka i ciljeva podrazumeva specifikaciju korisničkih zahteva,na osnovu kojih se kasnije kreiraju odgovorajući izveštaji koji imaju za cilj da pruže odgovore na postavljene zahteve.

Nakon projektovanja OLAP šeme, gde se definišu tabele dimenzija, činjenična tabela, veze između tih tabela i atributi tabela, kreiraju se DDL naredbe za tabele dimenzija i za činjeničnu tabelu, kao i sekvence za potrebe primarnih ključeva, i kreiraju se veze između tabela. Dimenzije koje su identifikovane na osnovu izvornog seta podataka su:Home team,Away team,Home team attrributes,Away team attributes,Season,League,Country, i naravno vremenska dimenzija,koja predstavlja obaveznu dimenziju u svakom skladištu podataka. Takodje,kreira se skripta za vremensku dimenziju, koja odgovara vremenskoj dimenziji u OLAP šemi. Nakon toga kreira se ETL proces, u kom se vrše određene transformacije nad podacima,te se nakon toga kreirane tabele popunjavaju transformisanim,prečišćenim podacima. Na kraju, kreiraju se izveštaji na osnovu kreiranog skladišta podataka i korisničkih zahteva, čime se dobija uvid u podatke na interaktivan način putem izveštaja.

SPECIFIKACIJA ZAHTEVA KORISNIKA

Zahtevi korisnika predstavljeni su u obliku pitanja, a na te zahteve, odnosno pitanja biće odgovoreno kasnije,nakon detaljne analize, putem izveštaja koji će biti kreirani. Treba napomenuti da izveštaji naravno ne daju isključivo jedan odgovor kojim će biti odgovoreno na konkretna postavljena pitanja,već se korišćenjem kreiranih izveštaja može odgovoriti na mnoga pitanja.

Spisak zahteva:

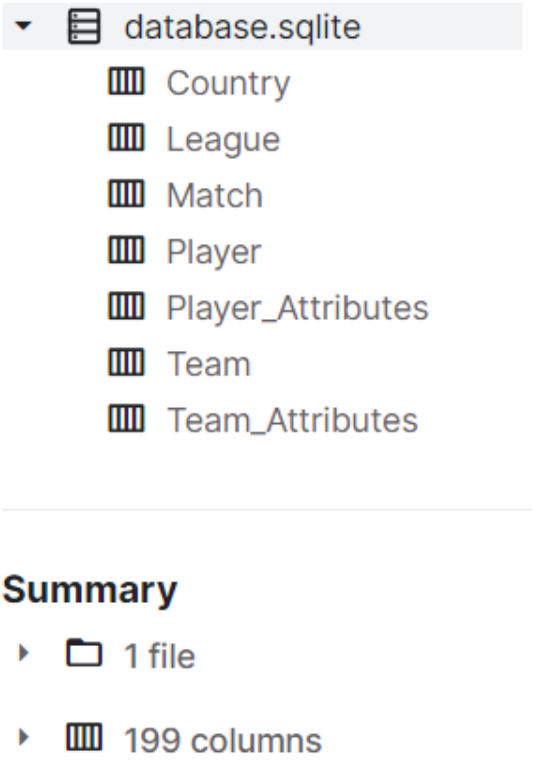
1. Koliko je ukupno golova postignuto u Engleskoj Premijer ligi u 2014. Godini?
2. Koliki je prosečan broj kartona u poslednjem kolu u nemačkoj Bundesligi u sezoni 2015/2016?
3. Koliki procenat šuteva,od svih upućenih šuteva(i u okvir gola,i van njega) završi u golu u italijanskoj Serie A ligi u sezoni 2014/2015?
4. Koji fudbalski tim ima najveći broj postignuthih golova u gostima u Španskoj ligi u sezoni 2012/2013 i koliko je to golova?
5. Koliko se procentualno napravljenih faulova pretvori u javnu opomenu u vidu kartona u Franckuskoj fudbalskoj ligi u sezoni 2015/2016?
6. Koji fudbalski tim je postigao najviše golova u utakmicama koje se igraju na domaćem terenu u periodu od 2008. do 2016. godine, a koji na utakmicama koje se igraju na gostujućem terenu?
7. Koja fudbalska liga je bila najefikasnija,u pogledu postignutih golova u periodu od 2012. do 2016. godine?
8. U kojoj sezoni je fudbalski tim Arsenal iz Engleske Premier lige imao najveće ocene,odnosno opšti rejting posmatrano relevantne atribute?

SPECIFIKACIJA MODELA

1. Specifikacija izvora podataka

Primarni izvor podataka predstavlja set podataka pod nazivom European Soccer Database, koji se nalazi na sajtu Kaggle. Link ka izvoru podataka:
https://www.kaggle.com/datasets/hugomathien/soccer

Kao što je već bilo reči, izvorni dataset sadrži podatke u sqlite formatu,gde su podaci smešteni u tabele. Set podataka sadrži ukupno 7 tabela,koje u sebi imaju realne podatke iz sveta fudbala o preko 25 hiljada odigranih utakmica u periodu od 2008. do 2016.godine u 11 najjačih evropskih fudbalskih liga. Pored toga,set podataka sadrži i podatke o timovima koji su učestvovali u datim utakmicama,njihovim atributima koji su prikupljeni sa sajta FIFA,igračima i njihovim atributima. Kao što se može videti na slici 1, dataset sadrži ogroman broj kolona,pri čemu su neke od njih sadržale podatke u XML formatu,pa je jedan od zadataka pre započinjanja izrade DW sistema,zapravo bio prečišćavanje izvornog dataset-a i izbacivanje problematičnih podataka. Kao prilog, biće postavljen obrađen dataset.



Slika 1: Izvorni set podataka

Podaci koji se nalaze u ovom dataset-u dobijeni su prikupljanjem podataka sa više različitih izvora od strane autora dataset-a:

1. http://football-data.mx-api.enetscores.com
2. http://www.football-data.co.uk
3. http://sofifa.com

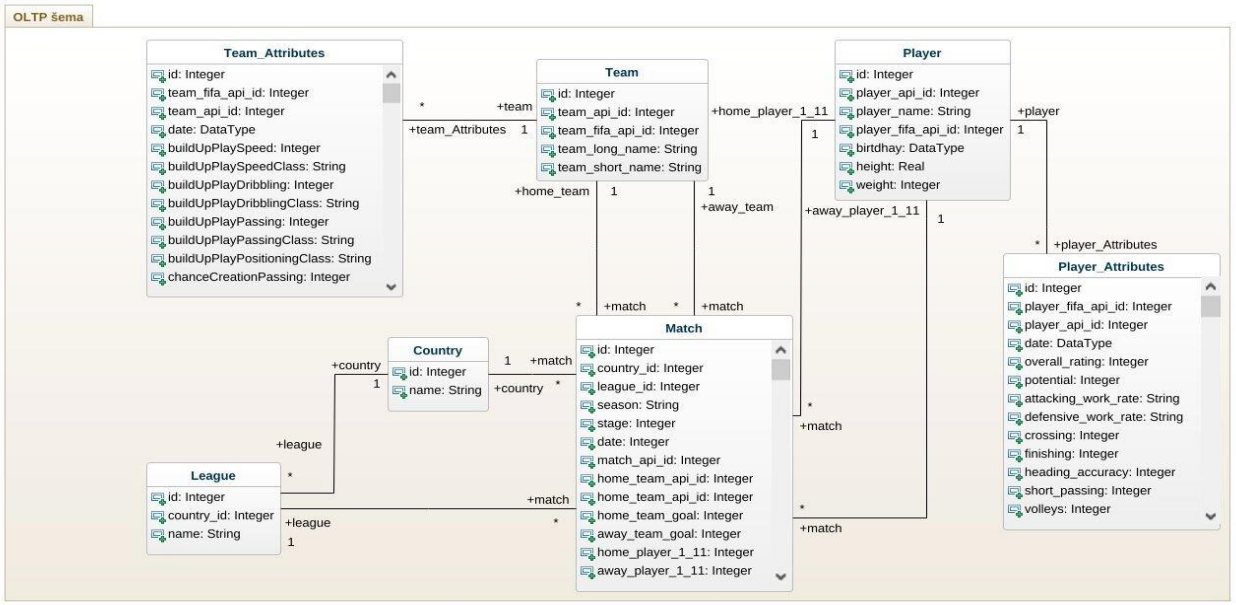
Pored ovog primarnog izvora podataka,korišćena su još dva pomoćna dataset-a. Reč je o sledećim setovima podataka:

1. World Population dataset – https://www.kaggle.com/datasets/iamsouravbanerjee/world-population-dataset
2. European Soccer Database Suplementary - https://www.kaggle.com/datasets/jiezi2004/soccer

Prvi od gore dva navedena izvora podataka,World population dataset,korišćen je kako bi se dodatno proširile informacije o državama koje se nalaze u izvornom setu podataka. Izvorni dataset je došao sa poprilično oskudnim informacija o državama,jer su se u njemu nalazili samo podaci o nazivima država. Postupak integracije izvornog dataset-a sa pomoćnim biće detaljnije prikazan prilikom objašnjavanja ETL procesa. Drugi set podataka zapravo predstavlja dodatak,odnosno proširenje izvornog seta podataka,koje sadrži podatke o dešavanjima na utakmicama koje se nalaze u izvornom dataset-u. Iz ovog seta podataka biće preuzete informacije kao što su broj kartona na utakmici,broj faulova na utakmici broj šuteva na utakmici i broj kornera na utakmici.

Na osnovu izvora podataka kreiran je dijagram klasa kako bi se stekao bolji uvid i razumevanje podataka i veza između njih.

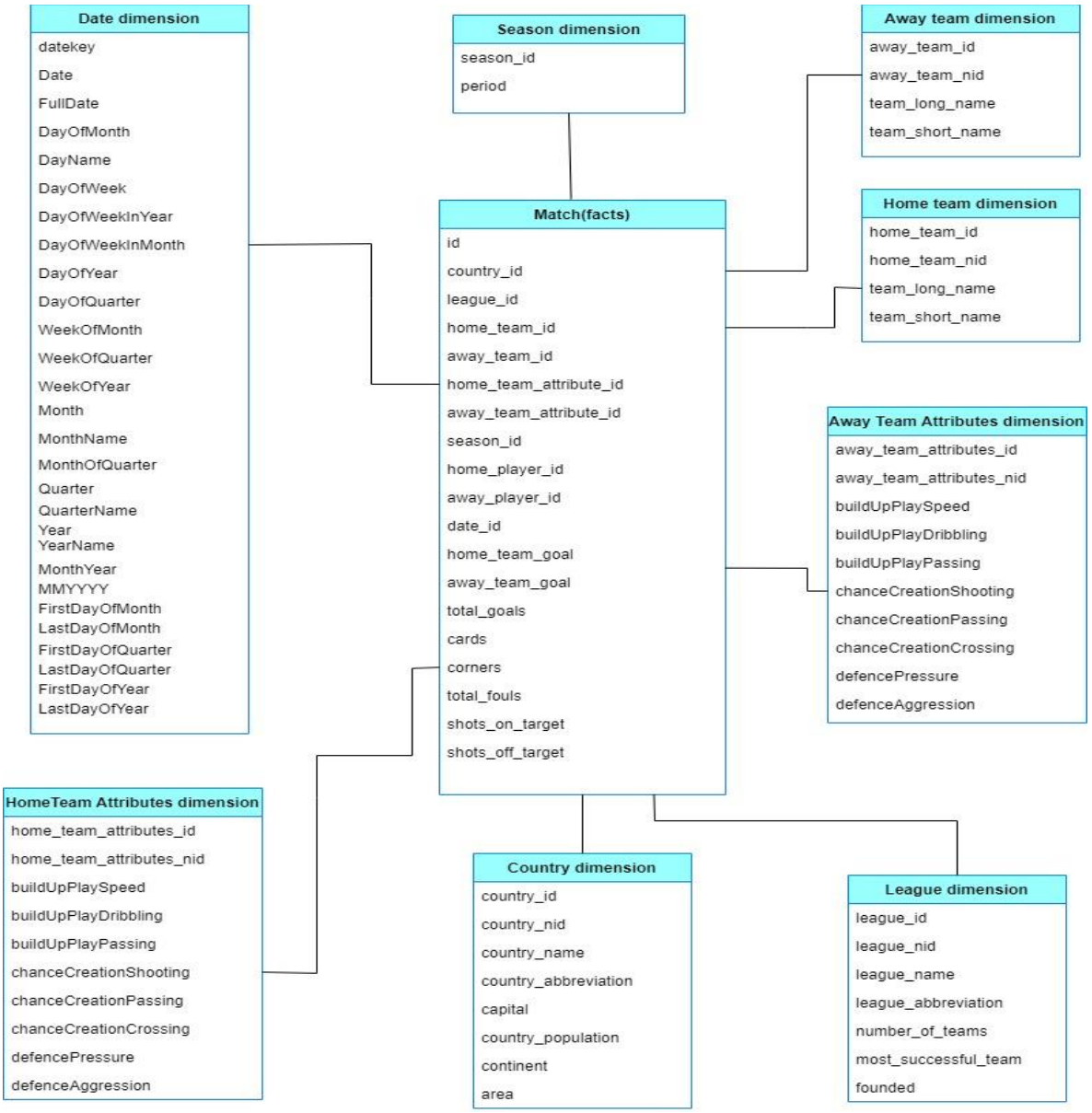
Prikaz dijagrama klasa,odnosno OLTP šeme:



Slika 2:OLTP šema

2. Specifikacija ciljnog DataWarehouse sistema

Na osnovu OLTP šeme podataka, korisničkih zahteva i procesa poslovanja, kreirana je OLAP šema. Atributi vezani za samu odigranu utakmicu definisani su kao mere u okviru tabele činjenica, dok su tabele koje sadrže podatke koji se ponavljaju, i vezani su za veći broj utakmica, izdvojene kao tabele dimenzija. Tabele Player i Player_Attributes su označene kao višak i uklonjene zbog kreiranja zvezde šeme i donesene odluke da se fokus stavi na analizu odigranih utakmica,a ne igrača koji su učestvovali u njoj. Na slici 3 prikazana je zvezda šema koja je kreirana na osnovu identifikovanih tabela dimenzija i tabele činjenice.



Slika 3: OLAP šema

2.a Specifikacija zahtevanih dimenzija

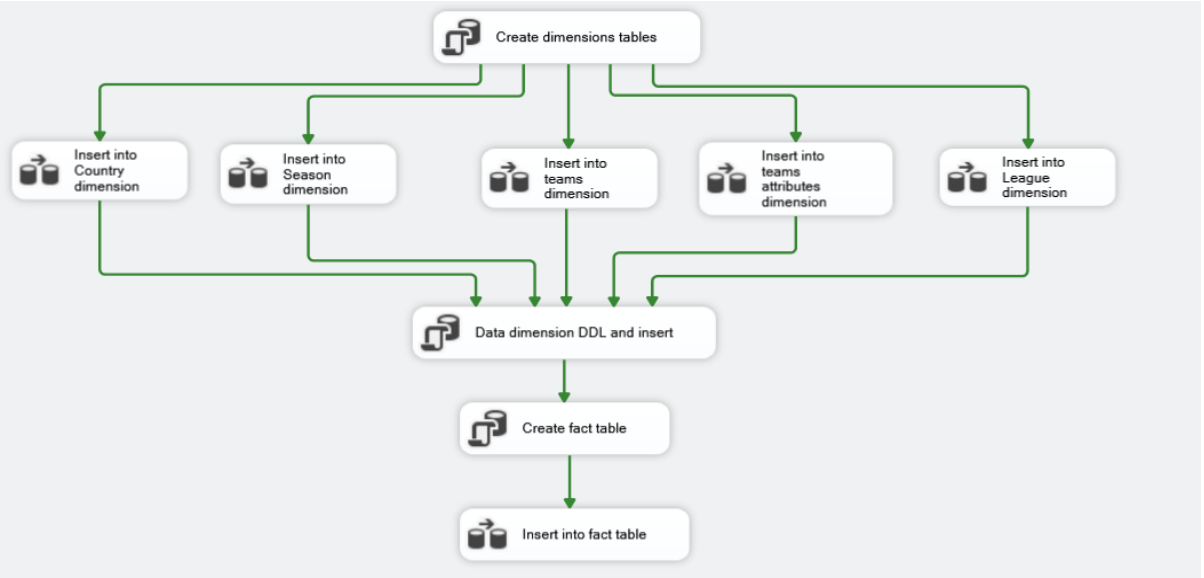
Dimenzije koje se mogu videti na OLAP šemi prikazanoj na slici 3 su kreirane na osnovu izvora podataka. Dimenzije League, Country, Home team,Away team,Home team attributes i Away team attributes su preuzete iz izvora uz modifikaciju i dodavanje određenih obeležja. Takodje,obeležja koja su bila suvišna su izostavljena. Dimenzija season kreirana je na osnovu obeležja season koje se nalazilo u izvornoj tabeli Match. Dimenzija Date je obavezna dimenzija u svakoj zvezda šemi i ona je kreirana na osnovu skripta koji je javno dostupan na internetu,te su obeležja koja se nalaze u ovoj dimenziji zapravo preuzeta iz tog skripta uz očuvanje originalnih naziva tih obeležja. U daljim koracima dokumentacije detaljno će biti objašnjeno kako su ove tabele dimenzija kreirane.

2.b Specifikacija zahtevanih mera

Činjeničnu tabelu predstavlja izvorna tabela Match koja je uz određene modifikacije pretvorena u tabelu činjenica. Činjenična tabela sadrži mere relevantne za praćenje jedne odigrane utakmice,a to su broj postignutih golova domaćina i gosta,ukupan broj postignutih golova,broj katona,broj faulova,broj kornera i broj šuteva u okvir i van na meču. Kao što se može videtina slici 3,činjenična tabela sadrži strane ključeve svih dimenzija koje je okružuju,i u okviru ETL procesa je bilo potrebno uraditi mapiranje prirodnih na veštačke ključeve,te u tabelu činjenica na osnovu prirodnih ključeva u izvoru podataka upisati odgovarajuće veštačke ključeve. Slično kao i kod tabela dimenzija,prilikom kreiranja tabele činjenica iz izvora podataka su uklonjena obeležja koja su bila irelevantna za ovaj projekat ili nejasna za razumevanje.

OPIS ETL PROCESA

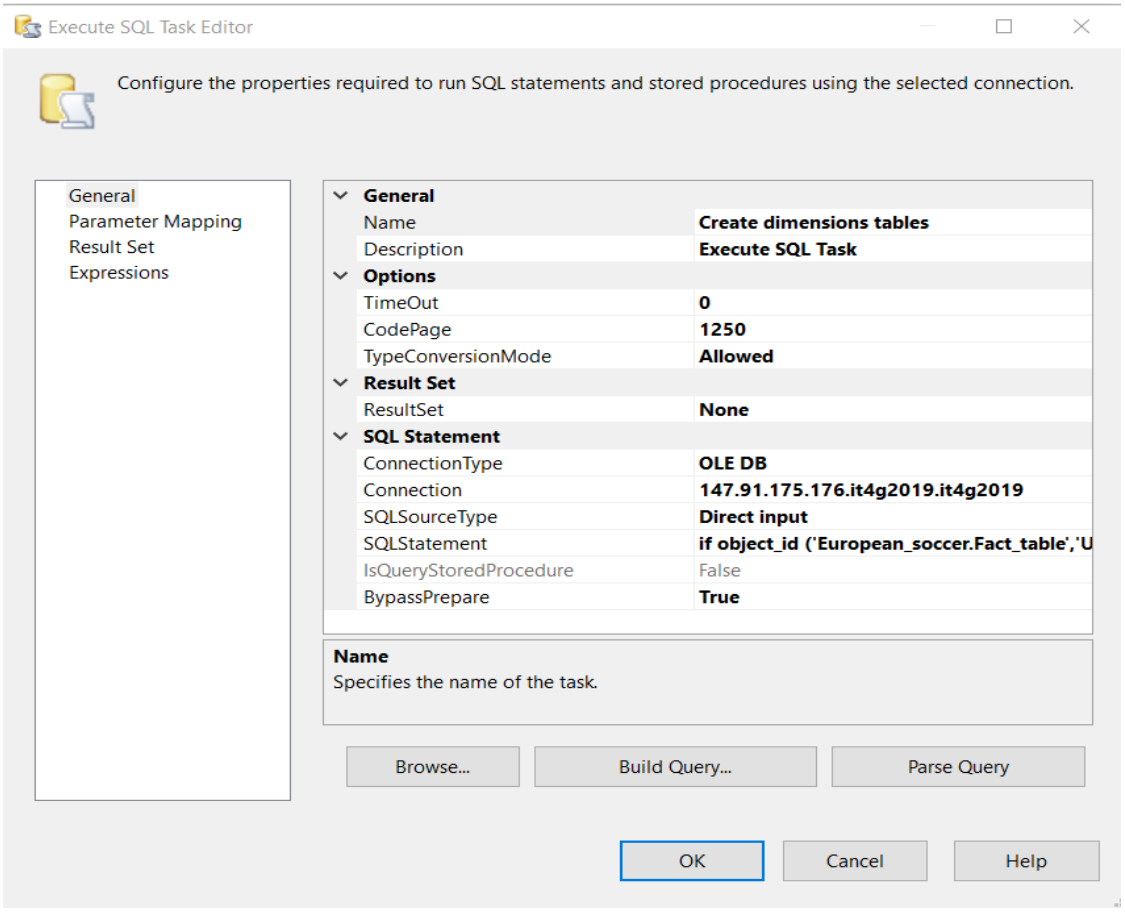
Za kreiranje ETL procesa, korišćene su aplikacije Visual Studio i u okviru njega Integration Services Project, kao i SQL Server Management Studio, i SQL Server za kreiranje samog skladišta podataka. Kako bi bilo omogućeno kreiranje ovog projekta, potrebno je instalirati dodatak SQL Server Data Tools for Visual Studio. Na slici 4 prikazan je izgled celog ETL procesa,odnosno izgled control flow-a,koji predstavlja početnu stranicu prilikom pokretanja SSIS-a, i on sadrži sekvencu taskova,i specificira redosled i uslove njihovih izvršavanja. Najkorišćeniji tipovi taskova su Execute SQL,Execute package,Data flow, Script. Prilikom izrade ovog projekta korišćeni su Execute SQL Task i Data flow,koji sadrži logiku transformacije podataka,o čemu će biti reči malo kasnije.



Slika 4: Etl proces

Prvi korak u kreiranju ETL procesa jeste kreiranje tabela dimenzija koristeći DDL naredbe, konkretno pomoću create table naredbe, i kreiranje sekvenci za svaku od kreiranih tabela korišćenjem create sequence naredbe. Prilikom izrade ovog projekta kreirana je šema European_soccer čiji je naziv u skladu sa izabranim projektom. Svi kreirani objekti bili su kreirani isključivo u okviru ove šeme. Zatim su te naredbe ubačene u okviru SSIS Execute SQL Task-a,kako bi se obezbedilo da se svaki put prilikom pokretanja ETL-a vrši provera i brisanje u slučaju da zahtevane tabele i sekvence postoje,te da se nakon toga vrši ponovno kreiranje datih tabela i sekvenci sa ciljem obezbedjenja konzistentnosti upisa podatka u odgovarajuće tabele dimenzija prilikom svakog ponovnog pokretanja ETL-a. Pored toga u okviru SSIS Execute SQL Task-a definisana je i konekcija ka bazi podataka gde će biti kreirane tabele,te nakon toga u njih smešteni odgovarajući podaci. Na slici 5 dat je prikaz Execute SQL Task-a koji je korišćen za kreiranje tabela dimenzija. Kao što se može uočiti,u okviru polja Connection,potrebno je zadati već spomenuti konekcioni string ka već postojećoj studentskoj bazi podataka u kojoj je potrebno izvršiti DDL skript unet u polje SQLStatement. Ostali

parametri ovog task-a su po defaultu podešeni i nije ih potrebno naknadno menjati.



Slika 5: Execute SQL Task

Zbog obimnosti DDL skripte za kreiranje tabela dimenzija i pripadajućih sekvencu,na narednim slikama biće dat samo deo prikaza korišćenih SQL naredbi.

```
if object_id ('European_soccer.Away_Team_Attributes','U') is not null
    drop table European_soccer.Away_Team_Attributes;
go
if object_id('European_soccer.SEQ_away_team_attributes_id','SO') is not null
    drop sequence European_soccer.SEQ_away_team_attributes_id;
go
if object_id ('European_soccer.Season','U') is not null
    drop table European_soccer.Season;
go
if object_id('European_soccer.SEQ_season_id','SO') is not null
    drop sequence European_soccer.SEQ_season_id;
go
if object_id ('European_soccer.Date_dimension','U') is not null
drop table European_soccer.Date_dimension;
go
if schema_id ('European_soccer') is not null
    drop schema European_soccer;
```

Slika 6: Uslovno brisanje objekata

```
create table European_soccer.League(
    league_id int not null constraint DFT_League_league_id default(next value for European_soccer.SEQ_league_id),
    league_nid int,
    league_name varchar(100) not null,
    league_abbreviation varchar(4) not null,
    number_of_teams int not null,
    most_successful_team varchar(50),
    founded int,


    constraint PK_League primary key (league_id),
    constraint UQ_League_league_nid unique (league_nid),
    constraint UQ_League_league_name unique (league_nid)
)
create table European_soccer.Country(
    country_id int not null constraint DFT_Country_country_id default(next value for European_soccer.SEQ_country_id),
    country_nid int,
    country_name varchar(50) not null,
    country_abbreviation varchar(5) not null,
    capital varchar(50) not null,
    country_population int,
    continent varchar(30),
    area decimal(20,2)

    constraint PK_Country primary key (country_id),
    constraint UQ_League_country_nid unique (country_nid),
    constraint UQ_League_country_name unique (country_name)
)
create table European_soccer.Season(
    season_id int not null constraint DFT_Season_season_id default(next value for European_soccer.SEQ_season_id),
    period varchar(10) not null,

    constraint PK_Season primary key (season_id)
)
```

Slika 7: Kreiranje tabela dimenzija

```
create schema European_soccer;
go
create sequence European_soccer.SEQ_fact_id as int
start with 1
increment by 1
no cycle
go
create sequence European_soccer.SEQ_league_id as int
start with 1
increment by 1
no cycle
go
create sequence European_soccer.SEQ_country_id as int
start with 1
increment by 1
no cycle
go
create sequence European_soccer.SEQ_home_team_id as int
start with 1
increment by 1
no cycle
go
create sequence European_soccer.SEQ_away_team_id as int
start with 1
increment by 1
no cycle
go
create sequence European_soccer.SEQ_home_team_attributes_id as int
start with 1
increment by 1
no cycle
```

Slika 8: Kreiranje sekvenci

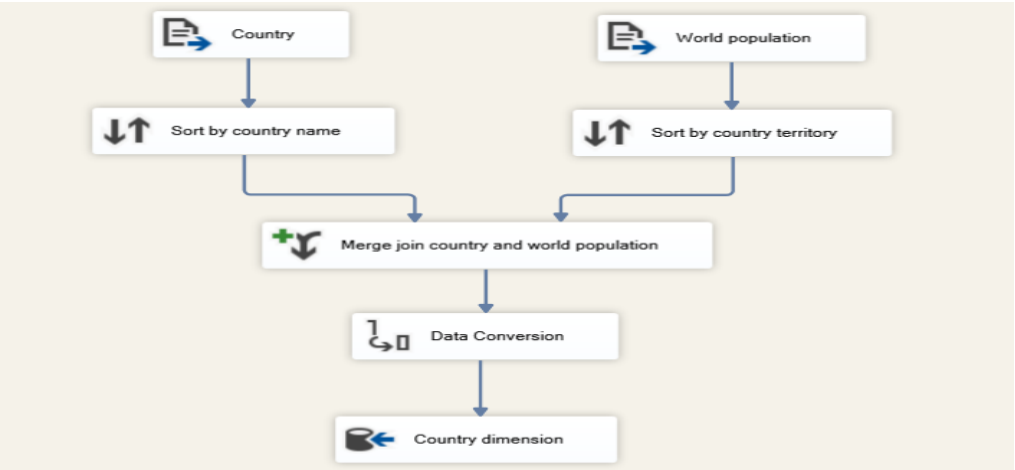
Na slici 6 prikazane su komande za uslovno brisanje tabela dimenzija,sekvenci i na kraju šeme ukoliko već postoje. Na slici 7 dat je prikaz postupka kreiranja nekih od kreiranih tabela

dimenzija tokom projekta,a reč je o tabelama Season,Country i League. Analogno ovim tabelema,bile su kreirane i sve druge tabele dimenzija u okviru ovog projekta. Ono što je potrebno već sada primetiti,jeste da se za generisanje vrednosti veštačkog ključa koristi sekvenca,te da se pored vrednosti veštačkog ključa,koji predstavlja primarni ključ u tabeli,pamti i vrednost primarnog ključa u izvornom setu podataka,tzv. prirodnog ključa(*Natural key*). Očuvanje prirodnog ključa je od izuzetnog značaja za kasnije adekvatno povezivanje tabela dimenzija sa tabelom činjenica. Takođe,na slici 8 dat je prikaz kreiranja sekvenci koje se koriste za automatsko generisanje vrednosti primarnog ključa prilikom upisa podataka u tabele.

Sledeći korak u okviru ETL procesa jesu posebni Data flow-ovi,koji se paralelno izvršavaju, za svaku od tabela dimenzija u OLAP šemi. U ovom koraku potrebno je učitati podatke iz CSV fajlova,koji su napravljeni na osnovu izvornog i dopunskih setova podataka,na već objašnjen način. Ovaj zadatak je sproveden koristeći SSIS-ov Data Flow Task. Izvori podataka korišćeni za izradu ovog zadatka su,a koji će biti priloženi u prilogu u okviru foldera Csv files su:

1. Country
2. Match
3. Team
4. Team_Attributes
5. League
6. card_detail
7. corner_detail
8. foulcommit_detail
9. shotoff_detail
10. shoton_detail
11. world_population

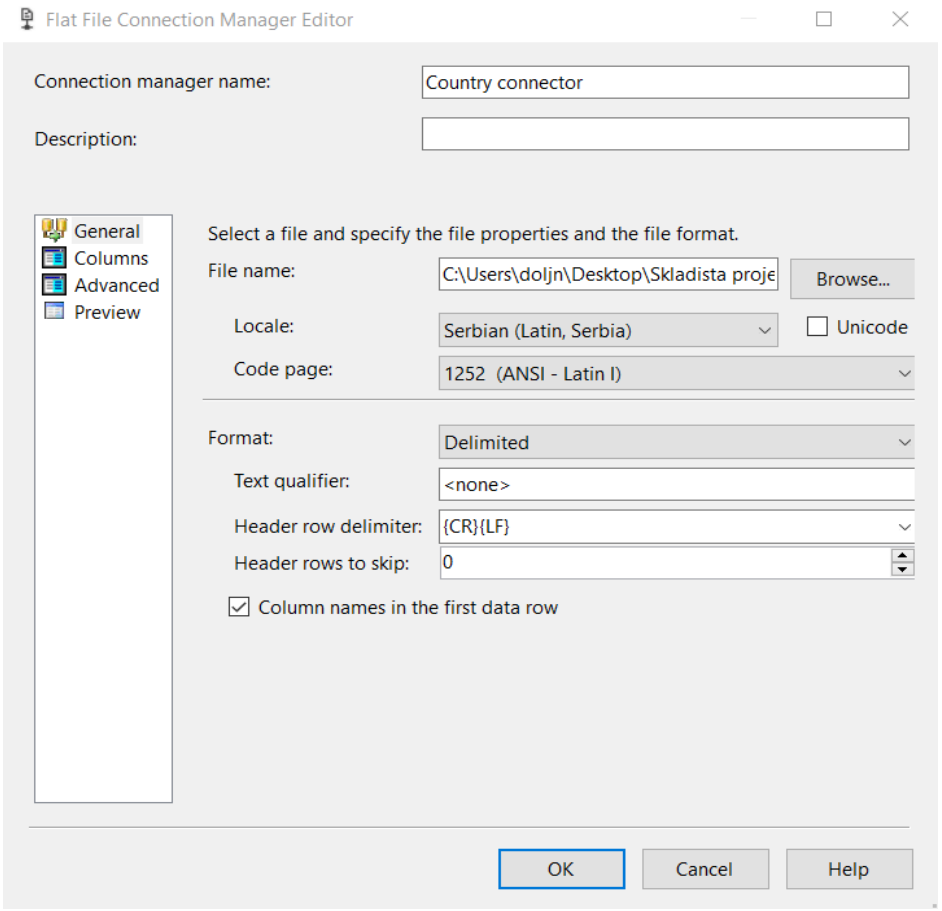
Svaki Data flow se sastoji od izvora,destinacije i niza transformacija pomoću kojih se izvorišni podaci transformišu na željeni način. Za učitavanje CSV fajlova u Integration Service Project koristi se Flat File Source komponenta. U nastavku će biti prikazani i opisani postupci za neke od tabela a proces učitavanja podataka će biti najpre pokazan kroz primer dimenzije Country.Na slici 9 dat je prikaz izgleda data flow-a za učitavanje podataka u već kreiranu tabelu Country.



Slika 9: Data flow za tabelu Country

Kao što se može videti na slici 9, data flow za kreiranje dimenzije Country se sastoji iz 7

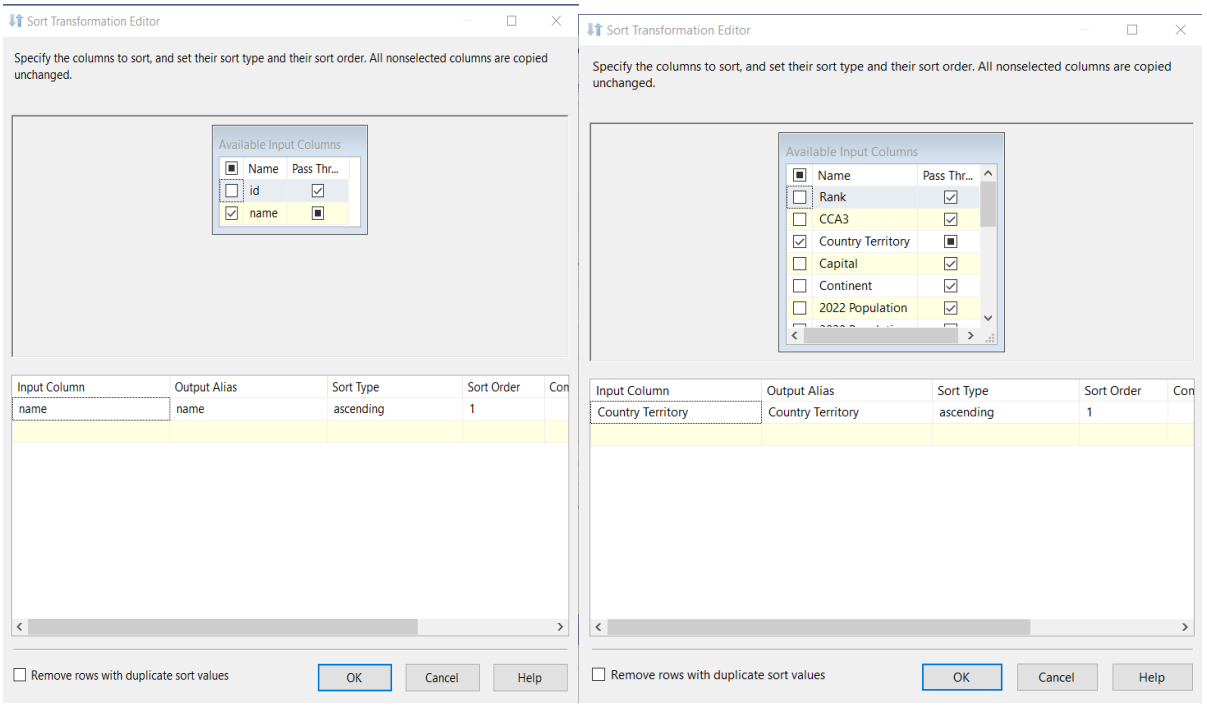
komponenti. Prva komponenta jeste Flat File Source, u kom je kreirana veza ka gore pomenutom CSV fajlu u kom se nalaze izvorni podaci o državi. Komponenta Flat File Source omogućava učitavanje izvornog fajla uz pomoć Conection Manager koji je prikazan na slici 10. Takođe,u opciji Columns koja se vidi sa leve strane na slici 10 prikazane su kolone koje se nalaze u izvornom fajlu i dat je njihov kratak preview a u opciji Advanced moguće je promeniti tip podatka za svaku kolonu koja se učitava, iz razloga što CSV fajlovi tretiraju sve podatke kao stringove. Međutim, da bi ovaj postupak bio transparentniji,isto to je moguće uraditi i pomoću komponente Data conversion,kao što je to urađeno u okviru data flow-a za kreiranje dimenzije Country.



Slika 10: Podešavanje Flat File Connection Manager-a

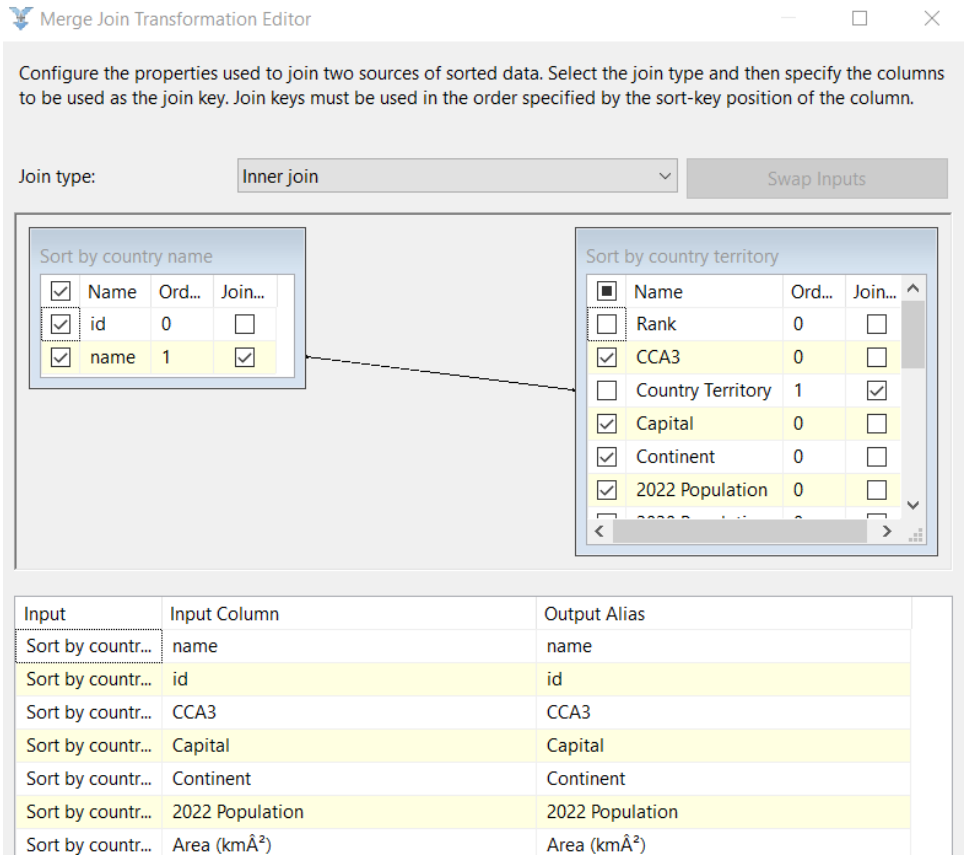
Analogno ovome,pomoću Flat File Source komponente učitani su i podaci iz world_population CSV fajla koji sadrži opširnije podatke o svakoj državi. Da bi bilo moguće spojiti podatke iz dva različita fajla, oni moraju prethodno biti sortirani po onim obeležjima po kojima se vrši spajanje,pri čemu ta obeležja moraju biti istog tipa. Zbog toga, nakon učitavanja podataka iz izvornih fajlova, sledi Sort komponenta koja omogućava sortiranje po željenom obeležju. Takođe,treba spomenuti još jednu jako korisnu funkcionalnost koja postoji u Sort komponenti a koja je korišćena u okviru odredjenih data flow-ova,a to je brisanje duplikata po sortiranoj vrednosti. Na slici 11 dat je prikaz izgleda Sort komponenti za sortiranje obeležja name iz izvornog fajla County i obeležja Country Territory iz izvornog fajla world_population. Iako se na prvi pogled može pomisliti da se radi o potpuno raziličitim obeležjima, i jedno i drugo

obeležje zapravo predstavlja naziv države,a s obzirom da je naziv države jedinstven te da svaka država ima sopstveni, jedinstveni naziv,ovo obeležje je idealno za spajanje fajlova Country i world_population. Spajanjem ova dva fajla u velikoj meri se povećava informativnost Country dimenzije različitim podacima koji se tiču države,kao što su površina,broj stanovnika,jedinstveni kod države,glavni grad.



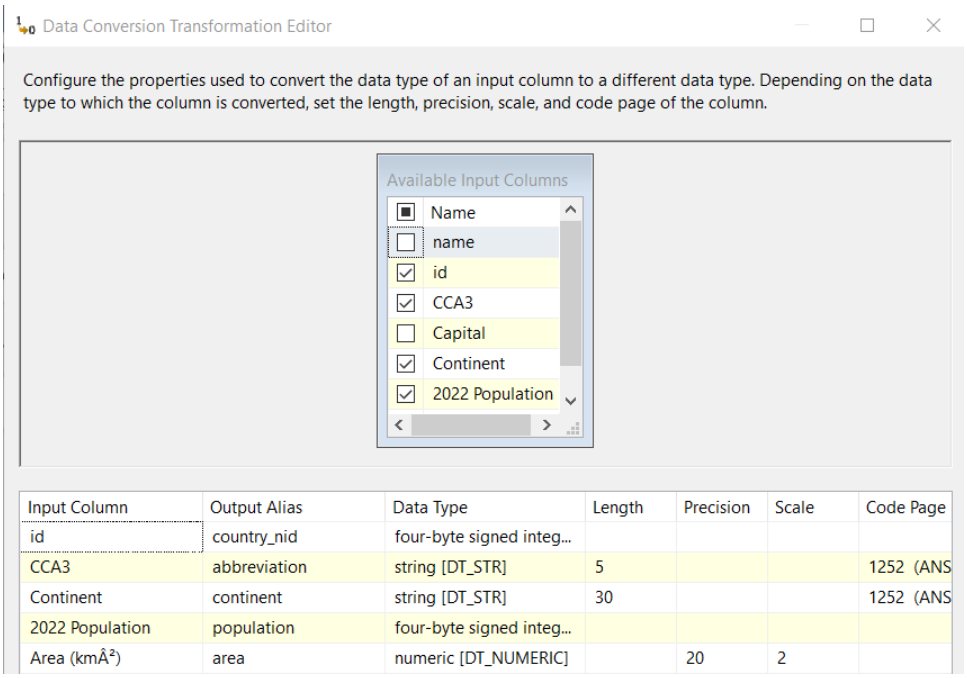
Slika 11: Podešavanje Sort Transformation komponente

Nakon Sort komponente sledi komponenta Merge Join,koja omogućava spajanje fajlova i biranje kolona koje želimo da se nalaze u destinaciji. Ova komponenta automatski prepoznaje koja su obeležja sortirana,te po njima vrši spajanje fajlova. Takođe,potrebno je obratiti pažnju koja vrsta spajanja je odabrana. Po default-u,odabran je inner join,međutim ovo je moguće promeniti na left outer ili full outer join u zavisnosti od toga šta želimo postići spajanjem i na koji način spojiti podatke. Prikaz funkcionisanja Merge Join komponente koja spaja dva sortirana ulaza u jedan izlaz dat je na slici 12.



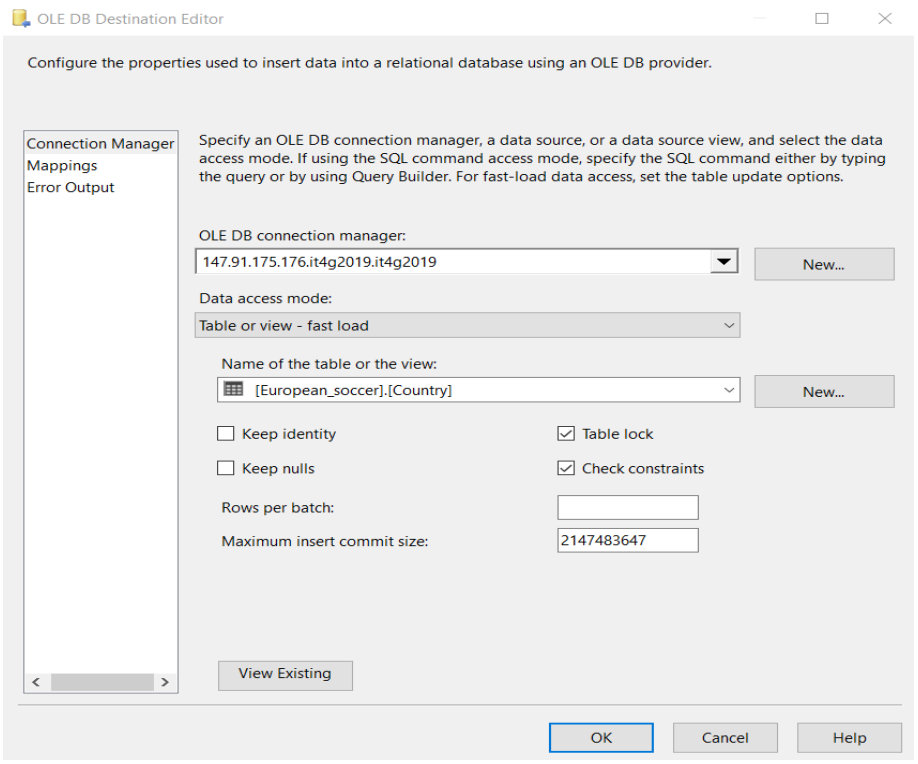
Slika 12: Podešavanje Merge Join komponente

Nakon Merge Join komponente sledi komponenta Data conversion,koja je korišćena pre svakog upisa u odredišnu tabelu u svim data flow-ovima. Data conversion komponenta služi za promenu tipa podatka,i korišćena je sa ciljem kako bi se podaci transformisali u željeni format,koji odgovara specificiranom formatu prilikom kreiranja tabela. Ova komponenta funkcioniše tako što se iz spiska obeležja izaberu ona čiji format želimo promeniti,te je nakon toga potrebno promeniti format,i eventualno code page,iz razloga što se u pojedinim situacijama code page CSV fajla i code page odredišne tabele u bazi podataka mogu razlikovati. U tom slučaju je potrebno transformisati podatke u onaj code page koji odgovara bazi podataka,a to je 1252 – ANSI Latin I. Takođe, obeležjima se automatski dodeljuje izlazni naziv “Copy of [naziv kolone]”, što je poželjno promeniti radi kasnijeg lakšeg mapiranja sa nazivima kolona u bazi podataka. Na slici 13 dat je prikaz podešavanja Data conversion komponente pre upisa podataka u tabelu Country dimenzije.

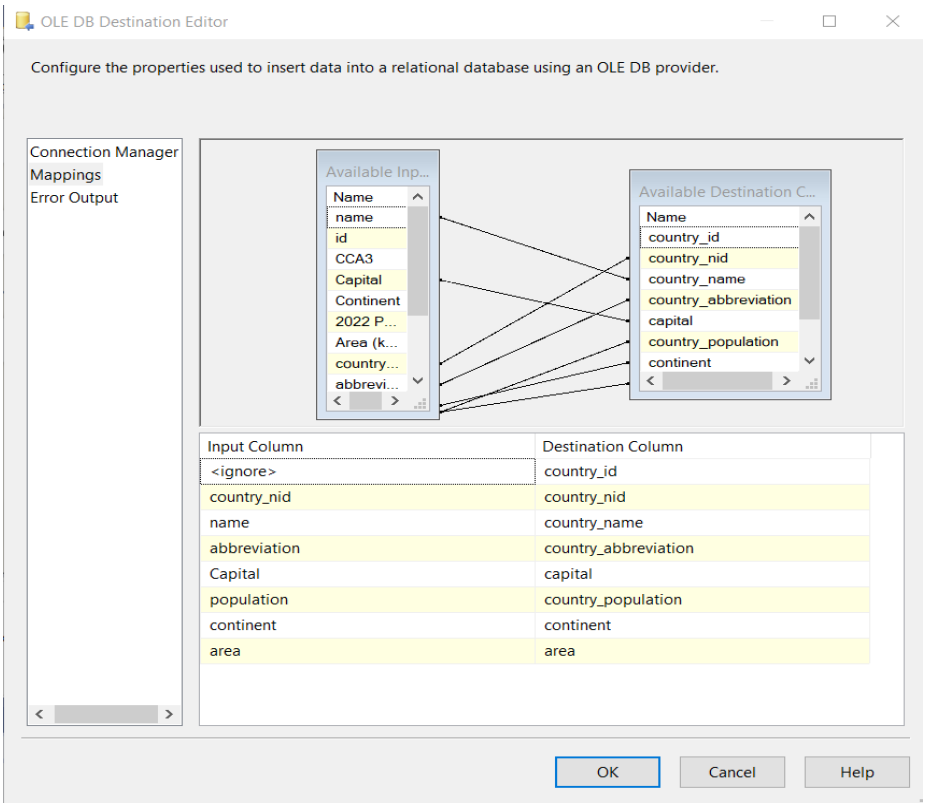


Slika 13: Podešavanje Data conversion komponente

Sledeći korak i poslednji korak u okviru svakog data flow-a jeste punjenje prethodno kreirane tabele u bazi podataka. Za realizaciju ovog zadatka potrebno je koristiti OLE DB Destination komponentu. Bitno je izabrati pravu tabelu u bazi podataka i voditi računa da navedeni tipovi podataka prilikom učitavanja iz CSV fajla i tipovi podataka navedeni u DDL-u budu apsolutno isti,što je prethodno urađeno na opisan način putem data conversion komponente. Takođe bitno je navesti i ispravnu konekciju i podesiti mapiranje podataka, što znači da za svaku kolonu u destinaciji moramo označiti koja kolona iz izvora će je mapirati. Ukoliko postoje neka polja unutar tabele koja se ne mapiraju,kao što je na primer veštački ključ koji se automatski generiše pomoću kreirane sekvence,onda se to polje ostavlja prazno, to jest, ostavlja se vrednost <ignore>. Na narednim slikama prikazan je postupak upisivanja podataka u odredišnu tabelu. Najpre je na slici 14 prikazano podešavanje connection manager-a,pomoću kog se podešava konekcija ka bazi podataka,a nakon toga na slici 15 je prikazan postupak mapiranja. Na potpuno identičan način se vrši upisivanje i u sve ostale tabele,kako tabele dimenzija,tako i u tabelu činjenica,te je samim tim primer upisivanja u tabelu dimenzije Country reprezentativan primer korišćenja ove komponente i sprovodjena postupka upisa prečišćenih podataka u skladište podataka.

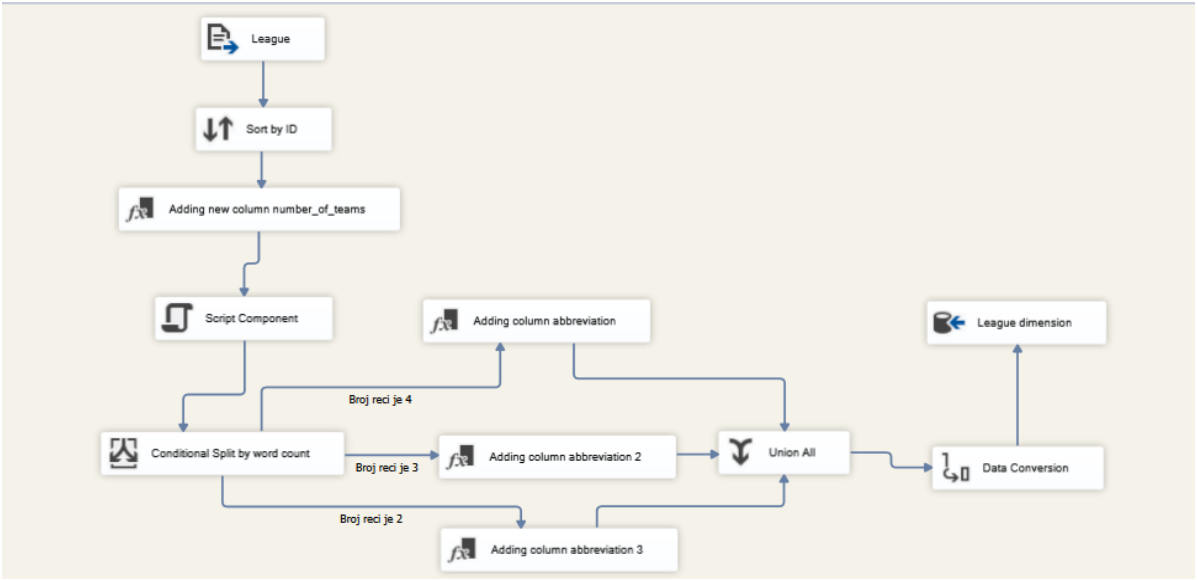


Slika 14: Podešavanje konekcije u OLE DB destination komponenti



Slika 15: Podešavanje mapiranja u OLE DB destination komponenti

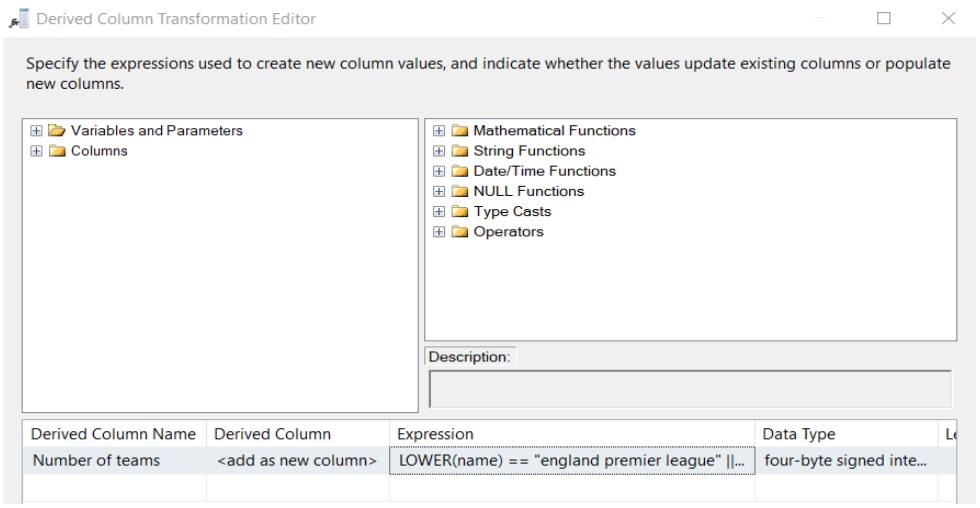
Sledeći će biti prikazan data flow za učitavanje podataka u tabelu League dimenzije, jer se u njemu nalaze neke drugačije stavke u odnosu na data flow za dimenziju Country,kao i u odnosu na data flow-ove za sve ostale tabele dimenzija. Za komponente koje se ponavljaju kao što su Flat File Source,Sort,OLE DB Destination,Data conversion i koje su već objašnjene prilikom razmatranja data flow-a za učitavanje podataka u tabelu Country dimenzije,podrazumevaće se da se i u okviru ovog flow-a sprovode na indentičan način,te će veći fokus biti posvećen novim komponentama,koje se do sada nisu pojavljivale. Na slici 16 dat je prikaz izgleda data flow-a za učitavanje podataka u tabelu dimenzije League.



Slika 16: Data flow za tabelu League

Nakon učitavanja podataka i sortiranja i brisanja duplikata putem Sort komponente, dodata je komponenta Derived column. Pomoću ove komponente moguće je dodati novu kolonu,koja se dobija primenom definisanog expression-a na ulazne podatke u komponentu. Konkretno,u ovom slučaju je bio cilj da se pomoću Derived column komponente napravi nova kolona number of teams koja zapravo predstavlja broj fudbalskih ekipa u ligi. Korišćenjem ugnježdenih if uslova,odnosno ternarnih uslovnih operatora jer u SSIS ne postoji klasičan if uslov, na osnovu naziva lige se koloni dodeljuje odgovarajuća vrednost koja se kreće od 10 do 20,u zavisnosti od toga koliko timova se nalazi u datoj ligi. Sam izgled Derived column komponente dat je na slici 17,a korišćeni uslov unutar njega ima sledeći oblik:

LOWER(name) == "england premier league" || LOWER(name) == "spain liga bbva" || LOWER(name) == "italy serie a" || LOWER(name) == "france ligue 1" ? 20 : (LOWER(name) == "scotland premier league" ? 12 : (LOWER(name) == "switzerland super league" ? 10 : 18))



Slika 17: Prikaz derived column komponente

Sledeća komponenta koja se do sada nije pojavljivala a sreće se u ovom data flow-u jeste Script komponenta. Script je komponenta koja omogućava pisanje proizvoljnog koda u C# jeziku,i koja se izmedju ostalog može koristiti za sprovođenje transformacija nad podacima,slično kao što se to radi kod Derived column komponente. Ono što je razlika,jeste da Script komponenta pruža mnogo komfortnije okruženje za pisanje koda,te omogućava pisanje značajno kompleksnije logike jer dozvoljava korišćenje svih mogućih funkcija koje postoje u .NET okruženju. Za potrebe izrade ovog projekta script komponenta je korišćena sa sličnim ciljem kao gore opisana Derived column komponenta. S obzirom da bi stalno pisanje obimnih uslova u okviru expression-a dovelo do nepreglednosti napisanog koda,odlučeno je da se za kreiranje kolona Most successful team i Founded,koje se kreiraju na osnovu naziva fudbalske lige koristi script komponenta. Prikaz dela specificiranih uslova u script komponenti dat je na slici 18.

```
switch (Row.name) {
    case "England Premier League":
        Row.Founded = 1992;
        Row.Mostsuccesfulteam = "Manchester United";
        break;
    case "Belgium Jupiler League":
        Row.Founded = 1895;
        Row.Mostsuccesfulteam = "Anderlecht";
        break;
    case "France Ligue 1":
        Row.Founded = 1832;
        Row.Mostsuccesfulteam = "PSG";
        break;
    case "Germany 1. Bundesliga":
        Row.Founded = 1962;
        Row.Mostsuccesfulteam = "Bayern Munich";
        break;
    case "Italy Serie A":
        Row.Founded = 1929;
        Row.Mostsuccesfulteam = "Juventus";
        break;
    case "Netherlands Eredivisie":
        Row.Founded = 1956;
        Row.Mostsuccesfulteam = "Ajax";
        break;
    case "Poland Ekstraklasa":
        Row.Founded = 1926;
        Row.Mostsuccesfulteam = "Legia Warszaw";
        break;
```

Slika 18: Korišćenje switch-a u Script komponenti

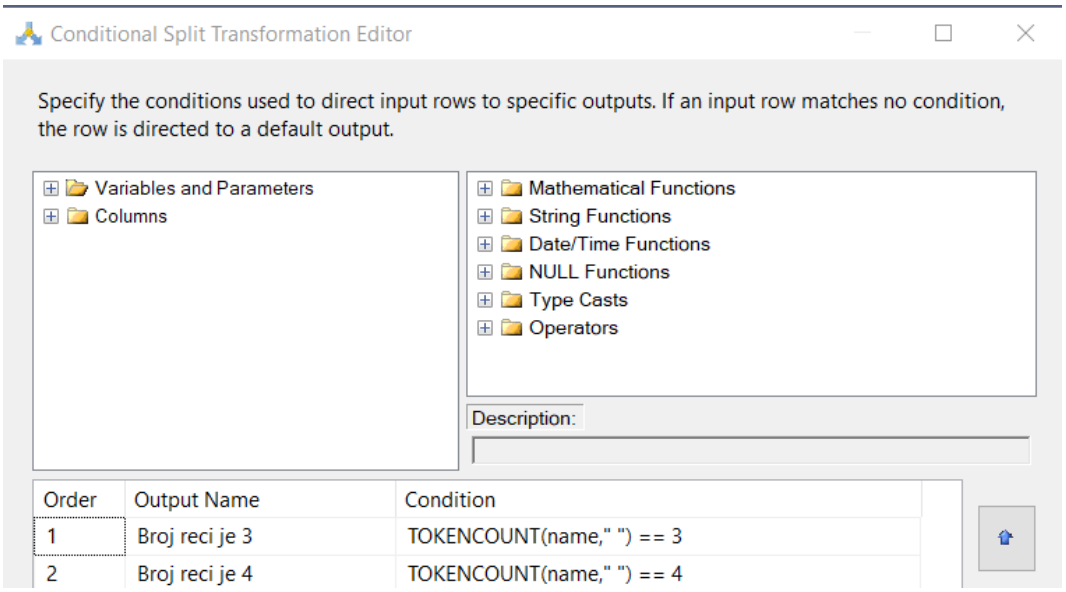
Nakon toga, dodata je komponenta Conditional Split,koja omogućava grananje toka u zavisnosti

od ispunjenosti zadatog uslova, s ciljem kako bi se u zavisnosti od toga koliko reči se nalazi u nazivu lige,tok podataka razdvojio na tri grane. Kao što se može videti na slici 16,na kojoj je prikazan data flow moguća su tri scenarija,a to je da naziv lige sadrži 2 reči,3 reči ili pak 4 reči. Nakon toga se primenom derived column komponenti na već opisan način kreira nova kolona,koja u sva tri slučaja ima isti naziv – abbreviation i ona se dobija tako što se od svake reči uzme prvo slovo,te se nakon toga sva tri toka spajaju pomoću Union all komponente. Izgled Conditional split komponente dat je na slici 19,a expression specificiran u okviru derived column komponenti u zavisnosti od broja reči je sledeći:

1. Broj reči je 2:
   LEFT(name,1) + SUBSTRING(name,FINDSTRING(name," ",1) + 1,1)

2. Broj reči je 3:
   LEFT(name,1) + SUBSTRING(name,FINDSTRING(name," ",1) + 1,1) + SUBSTRING(name,FINDSTRING(name," ",2) + 1,1)

3. Broj reči je 4:
   LEFT(name,1) + SUBSTRING(name,FINDSTRING(name," ",1) + 1,1) + SUBSTRING(name,FINDSTRING(name," ",2) + 1,1) + SUBSTRING(name,FINDSTRING(name," ",3) + 1,1)

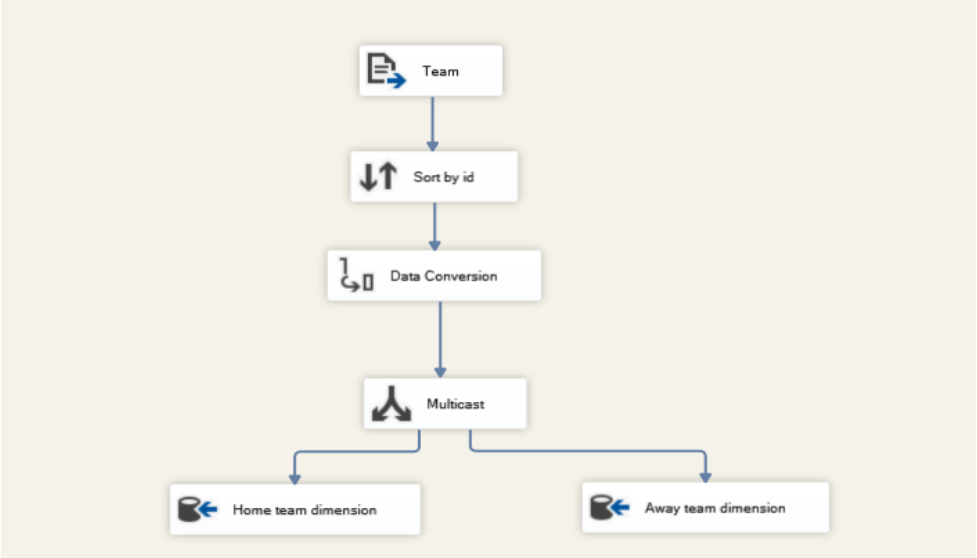


Slika 19: Prikaz Conditional Split komponente

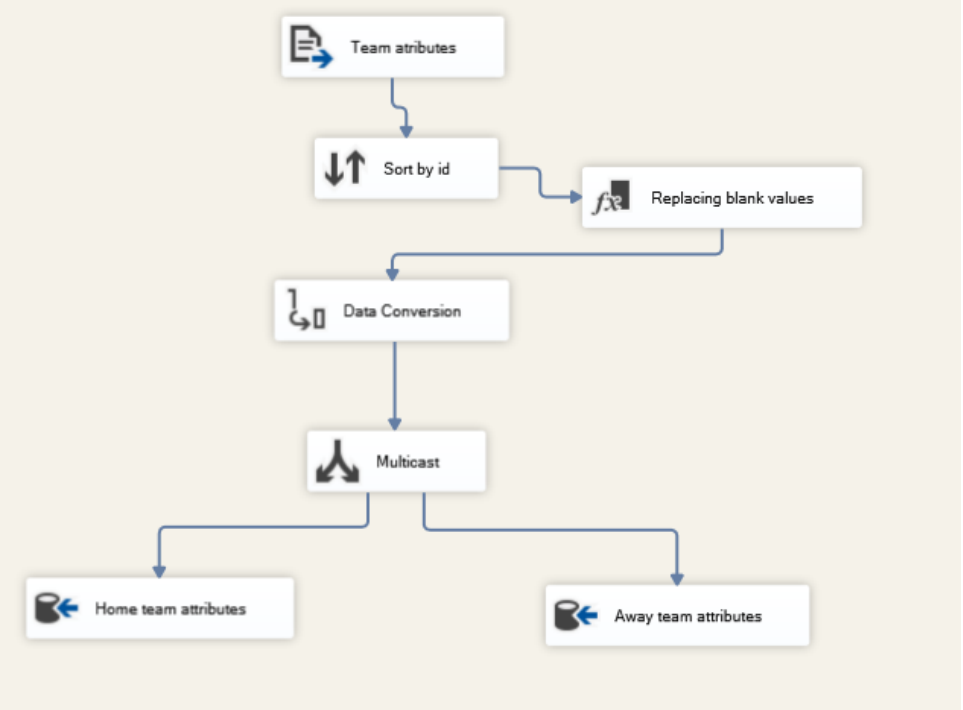
Na potpuno identičan način malopre prikazanom načinu učitavanja transformisanih podataka u kreirane tabele dimenzija,putem komponenti koje su objašnjene, kreirane su i ostale dimenzije. Bez detaljnijeg objašnjavanja,na narednim slikama dat je pregled data flow-ova za kreiranje preostalih dimenzija. Ono što je još jedino preostalo spomenuti,a ne sreće se u prethodno objašnjenim data flow-ovima,jeste korišćenje Multicast komponente,koja je korišćena u data

flow-u za kreiranje Home i Away team dimenzije,kao i u data flow-u za kreiranje Home i Away team attributes dimenzije. Ova komponenta ima za cilj da razdvoji tok podataka na dva identična toka,odnosno bolje reći,ima za cilj da napravi kopiju postojećeg toka podataka. Ovo je korišćeno iz razloga što se u tabele Home team i Away team dimenzije upisuju potpuno identični podaci,a s obzirom da se radi o skladištu podataka,redundantnost memorisanja podataka je skroz u redu i prihvatljiva je.

Slika 20: Data flow za tabelu Season



Slika 21: Data flow za tabele Home team i Away team



Slika 22:Data flow za tabele Home team attributes i Away team attributes

Nakon kreiranja svih dimenzija, ubačena je komponenta SSIS Execute SQL Task za kreiranje i popunjavanje vremenske dimenzije,koja predstavlja obaveznu dimenziju u svakom skladištu podataka. Vremenska dimenzija se kreira pomoću skripte za kreiranje vremenske dimenzije,koja je preuzeta sa interneta i koja sadrži veliki broj obeležja. Zbog preobimnosti obeležja u skripti,pojedina obeležja koja se ponavljaju bilo je potrebno izbaciti. Vremenska dimenzija se popunjava svim datumima između osamnaestog jula 2008. godine i dvadesetšestog maja 2016. godine,jer je uvidom u podatke,utvrdjeno da su sve utakmice odigrane između tih datuma. Zbog spomenute obimnosti date skripte za kreiranje i popunjavaje vremenske dimenzije,ona neće biti prikazana putem slika u okviru ove dokumentacije,već će biti data kao prilog u vidu sql fajla dateDimension. Nakon kreiranja svih tabela dimenzija i vremenske dimenzije, sledi kreiranje tabele činjenica i popunjavanje iste podacima,što ujedno predstavlja i završni deo ETL procesa. Razlog zašto se tabela činjenica kreira i popunjava poslednja su upravo ograničenja koja skladište podataka implementirano na osnovu zvezda šeme mora da ispoštuje, a to je da tabela činjenica treba da sadrži strane ključeve svih tabela dimenzija,koje prethodno moraju biti kreirane i popunjene podacima.

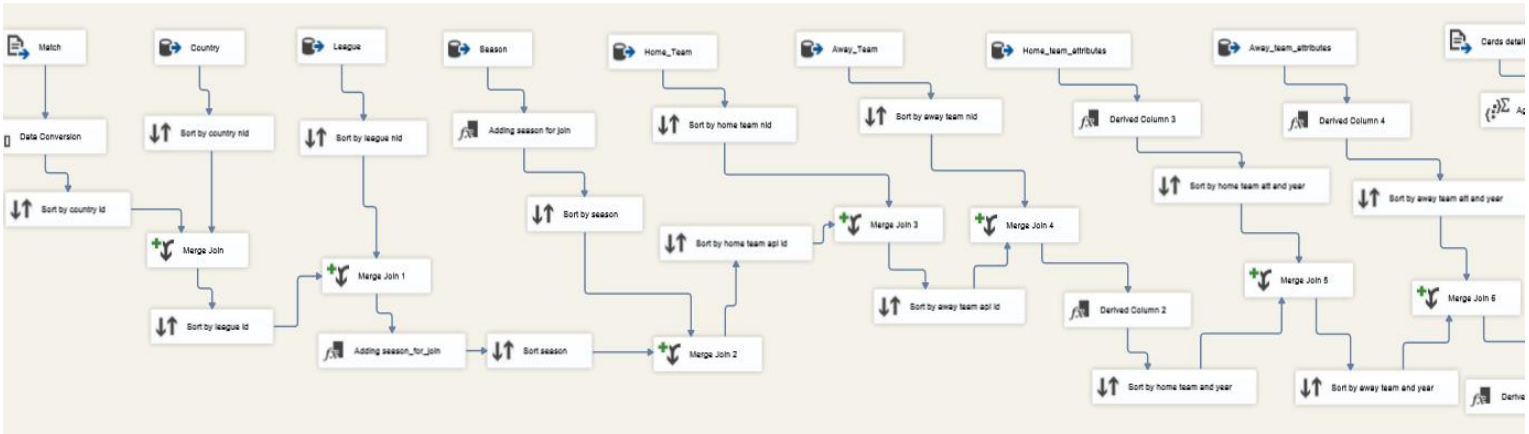
Slično kao i prilikom kreiranja tabela dimenzija,bilo je potrebno pre svega napraviti DDL skript za kreiranje tabele činjenica,kao i odgovarajuću sekvencu koja će služiti za automatsko generisanje vrednosti primarnog ključa tabele činjenica,te nakon toga te naredbe ubaciti u okviru SSIS Execute SQL Task-a. U DDL-u su dodata ograničenja za strane ključeve i na taj način je činjenična tabela povezana sa ostalim tabelama dimenzije.Takodje,dodate su i mere vezane za same odigrane utakmice.Na slici 23 dat je prikaz korišćenih DDL naredbi za kreiranje tabele činjenica.

```
create table European_soccer.Fact_table(
    fact_id int not null constraint DFT_Fact_table_fact_id default(next value for European_soccer.SEQ_fact_id),
    country_id int,
    league_id int,
    home_team_id int,
    away_team_id int,
    home_team_attributtes_id int,
    away_team_attributtes_id int,
    season_id int,
    date_id int,
    stage int,
    home_team_gol int,
    away_team_gol int,
    total_goals int,
    cards int,
    corners int,
    total_fouls int,
    shots_on_target int,
    shots_off_target int

    constraint PK_Fact_table primary key (fact_id),
    constraint FK_Fact_table_country_id foreign key (country_id) references European_soccer.Country (country_id),
    constraint FK_Fact_table_league_id foreign key (league_id) references European_soccer.League(league_id),
    constraint FK_Fact_table_home_team_id foreign key (home_team_id) references European_soccer.Home_Team(home_team_id),
    constraint FK_Fact_table_away_team_id foreign key (away_team_id) references European_soccer.Away_Team(away_team_id),
    constraint FK_Fact_table_season_id foreign key (season_id) references European_soccer.Season(season_id),
    constraint FK_Fact_table_home_team_attributes_id foreign key (home_team_attributtes_id) references European_soccer.Home_Team_Attributes(home_team_attributes_id)
    constraint FK_Fact_table_away_team_attributes_id foreign key (away_team_attributtes_id) references European_soccer.Away_Team_Attributes(away_team_attributes_id)
    constraint FK_Fact_table_date_id foreign key (date_id) references European_soccer.Date_dimension (DateKey),
```

Slika 23: Kreiranje tabele činjenica

Zbog velikog broja komponenti nije moguće prikazati u celosti sliku data flow-a za učitavanje podataka u tabelu činjenica,tako da će biti prikazani neki njegovi delovi i biće dato objašnjenje sprovedenog postupka u okviru data flow-a. U okviru data flow-a tabele činjenica, prvo su učitani svi podaci potrebni za kreiranje iste, u koje spadaju informacije preuzete iz izvornog CSV fajla Match koji zapravo u izvornom setu podataka predstavlja tabelu na osnovu koje će se uz odredjene modifikacije i dodavanje dodatnih mera biti napravljena tabela činjenica,kao i vrednosti tabela dimenzija koje će služiti za spajanje podataka na osnovu vrednosti prirodnog ključa,te prebacivanje veštačkih ključeva tabela dimenzija u tabelu činjenica. Pored toga,bilo je potrebno učitati i fajlove u kojima su sadržane buduće mere koje predstavljaju dešavanja na utakmici,a reč je o broju kartona,broju faulova,broju šuteva u okvir gola,broju šuteva van okvira gola i broju kornera na utakmici. Dakle, prvi deo punjenja tabele činjenica treba da se odvija tako što se izvorni csv fajl spaja sa svakom od dimenzija i iz svake dimenzije se uzimaju veštački primarni ključevi tabela i zatim upisuju u tabelu Činjenica. Deo data flow-a koji obavlja tu funkciju dat je na slici 24.

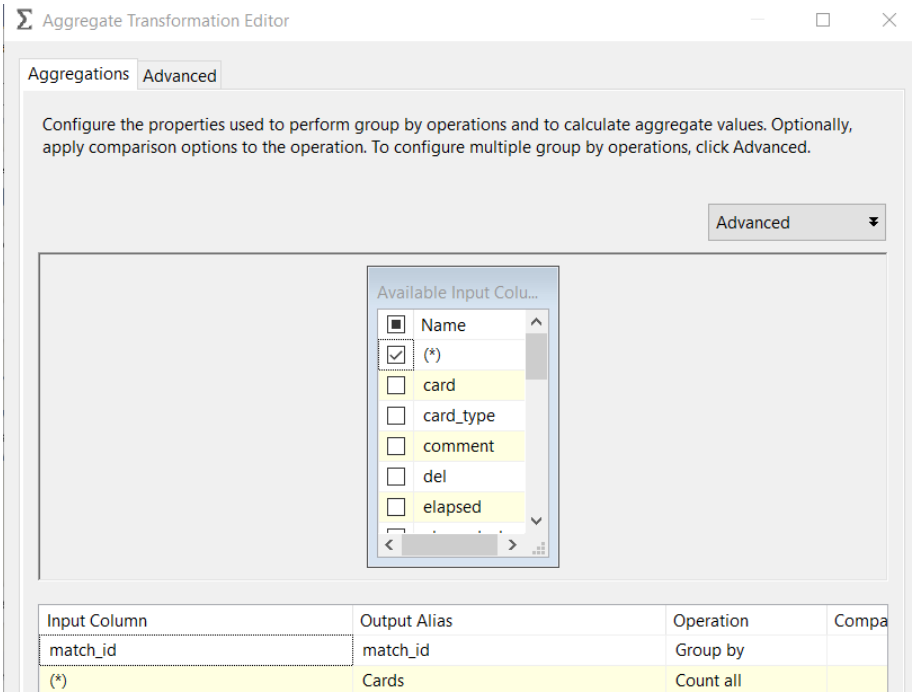


Slika 24:Deo data flow-a za mapiranje ključeva

Ono što je interesantno spomenuti a razlikuje se od dosadašnje procedure spajanja različitih

izvora,opisanog u prethodnom delu dokumentacije,jeste spajanje sa Home team attributes i Away team attributes dimenzijom. U tabelama ovih dimenzija se nalazi više atributa za svaki tim, odnosno njihovi atributi po godinama jer svaki tim može napredovati ili nazadovati iz godine u godinu,pa je saglasno tome,a s ciljem da bi podaci bili usaglašeni i konzistentni potrebno uzeti atribute tima koji su iz godine kada je meč odigran. To je učinjeno tako što su podaci sortirani ne samo po obeležju po kom bi se inače spajali,a to su u ovom slučaju u primeru Home Team attributes dimenzije home_team_api_id iz izvornog csv fajla i home_team_attributes_nid iz popunjene tabele dimenzije,već i po godini koja je uzeta pomoću derived column komponenti koje prethode sortiranju,kao što se može i videti na slici 24. Na ovaj način se postiže odredjem vid improvizacije spajanja po složenom ključu koji se sastoji iz dva obeležja. Na taj način se obezbedjuje spajanje sa isključivo onim atributom koji je tim imao te godine.

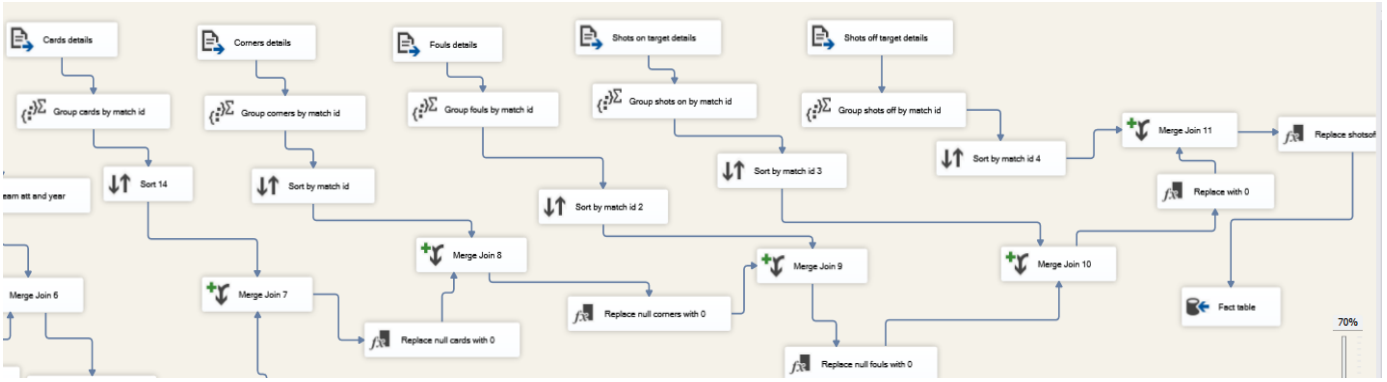
Drugi deo data flow-a odnosi se na spajanje sa csv fajlovima koji u sebi sadrže spomenute informacije o dešavanjima na meču,s ciljem popunjavanja tabele činjenica merama. Treba napomenuti da su dve od ukupno osam mera,već nalazile u izvornom csv fajlu Match,te da su one sačuvane i na osnovu njih je pomoću derived column komponente izvedena treća mera,a reč je o meri total goals,koja predstavlja ukupan broj postignutih golova na meču. Ostalih 5 mera prikupljeno je na osnovu dodatnih csv fajlova,o kojima sada ide reč. S obzirom da ovi fajlovi sadrže po jedan red za svaki dogadjaj na utakmici(što će reći da ako naprimer posmatramo fajl koji sadrži informacije o kartonima,on će sadržati 5 redova sa istim match id,ako je na utakmici sa tim match id bilo dodeljeno 5 kartona),nakon njihovog učitavanja bilo je neophodno uraditi grupisanje u odnosu na match id,kako bismo dobili informaciju o broju posmatranih dešavanja na utakmici sa datim match id-jem. To je učinjeno pomoću Aggregate komponente,čiji je prikaz za sprovedeno grupisanje broja kartona po utakmici dat na slici 25. Na identičan način je uradjeno grupisanje i za sve ostale mere.



Slika 25:Prikaz Aggregate komponente

Deo data flow-a koji obavlja gore opisanu funkciju pribavljanja mera,te njhovo spajanje sa tokom podataka i konačno upisivanje podataka u tabelu činjenica dat je na slici 26. Takođe,na ovoj slici možemo primetiti i da je u slučaju da imamo null vrednosti za određena dešavanja na

utakmici,podrazumevano da se nisu ni zbila na toj utakmici,te su takve vrednosti zamenjene sa nulom pomoću derived column komponenti.



Slika 26: Deo data flow-a za pribavljanje mera

PRIKAZ IZVEŠTAJA KAO ODGOVOR NA ZAHTEVE KOJI SU POSTAVLJENI OD STRANE KORISNIKA

Sledeći izveštaji kreiraju se u okviru Visual Studio-a korišćenjem tipa projekta koji se naziva Report Server Project. Izveštaji se kreiraju tako što se nakon kreiranja projekta, kreira Shared Data Source u kom se definiše konekcija ka malopre kreiranom skladištu podataka. Na taj način, svi izveštaji će koristiti te podatke. Nakon kreiranja Data source-a, kreiraju se izveštaji koji imaju zadatak da daju odgovore na svaki korisnički zahtev. Koristeći Query dizajner, biraju se tabele koje su potrebne za kreiranje izveštaja i ukoliko je potrebno vrše se filtracija i sortiranje. Nakon učitavanja podataka, bira se određeni način prikaza izveštaja, i u nastavku će biti prikazane tabele, kao i neki chart-ovi poput pie chart-a,data bar-a,smooth line with markers. Kako bi izveštaji bili interaktivni, omogućeno je za neke izveštaje interaktivno sortiranje,row visibility,unošenje različitih parametara od strane korisnika,padajuće liste. Za određene izveštaje dodato je grupisanje, dodatno filtriranje/sortiranje, agregatne funkcije i određeni indikatori radi bolje preglednosti podataka, poput zvezdice za rejting fudbalskog tima. U nastavku biće prikazano 5 kreiranih izveštaja,te korisnički zahtevi na koje ti izveštaji daju odgovor.

1.Prvi izveštaj

Prvi kreirani izveštaj ima za cilj da odgovori na sledeći korisnički zahtev:

- Koji fudbalski tim je postigao najviše golova u utakmicama koje se igraju na domaćem terenu u periodu od 2008. do 2016. godine, a koji na utakmicama koje se igraju na gostujućem terenu?

Izveštaj omogućava korisniku da sam izabere koliko najboljih timova u odnosu na date golove u u gostima i kod kuće će mu se prikazati,a po defaultu se prikazuje top 10 timova. Radi preglednosti,na slici 27 prikazano je top 5 timova. U zavisnosti od broja golova,data bar je obojen različitim bojama,što se najbolje može videti prosleđivanjem vrednosti 100 za parametar top.

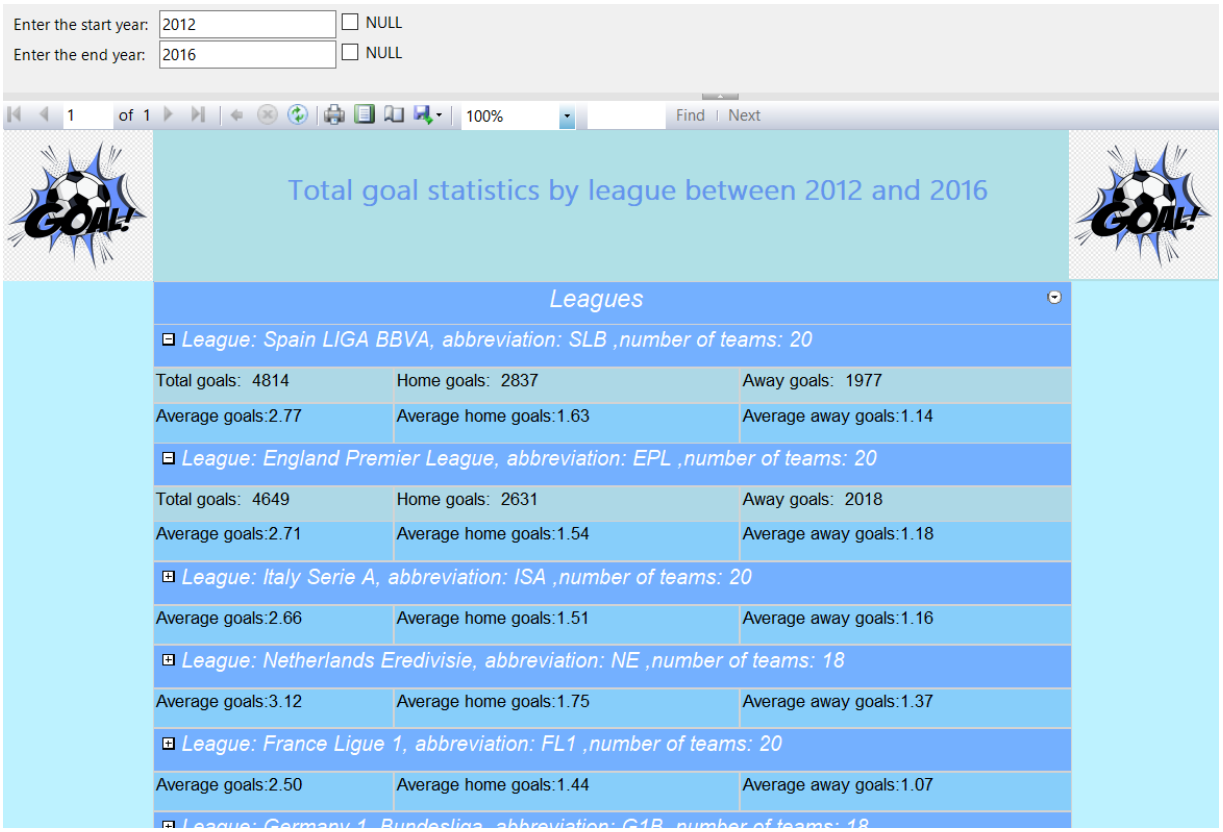
Slika 27: Prvi izveštaj

2. Drugi izveštaj

Drugi kreirani izveštaj ima za cilj da odgovori na sledeće korisničke zahteve:

- Koja fudbalska liga je bila najefikasnija,u pogledu postignutih golova u periodu od 2012. do 2016. Godine?
- Koliko je ukupno golova postignuto u Engleskoj Premijer ligi u 2014. godini?

Izveštaj omogućava korisniku da unese početnu i krajnju godinu,te se u zavinosti od prosledjenih paramatera od strane korisnika i sam izveštaj menja,odnosno prikazuje podatke za različite periode. Ovaj izveštaj prikazuje statistike vezane za golove u jedanaest najjačih evropskih liga. Na slici 28 vidi se samo deo tih liga,a primenom interaktivnog sortiranja možemo sortirati izveštaj tako da se na vrhu nalaze one lige koje su efikasnije,odnosno u kojima je postignuto više golova. Na vrhu možemo da primetimo da je najefikasnija fudbalska liga u periodu od 2012. do 2016. godine bila španska BBVA liga sa 4814 postignutih golova u datom periodu.

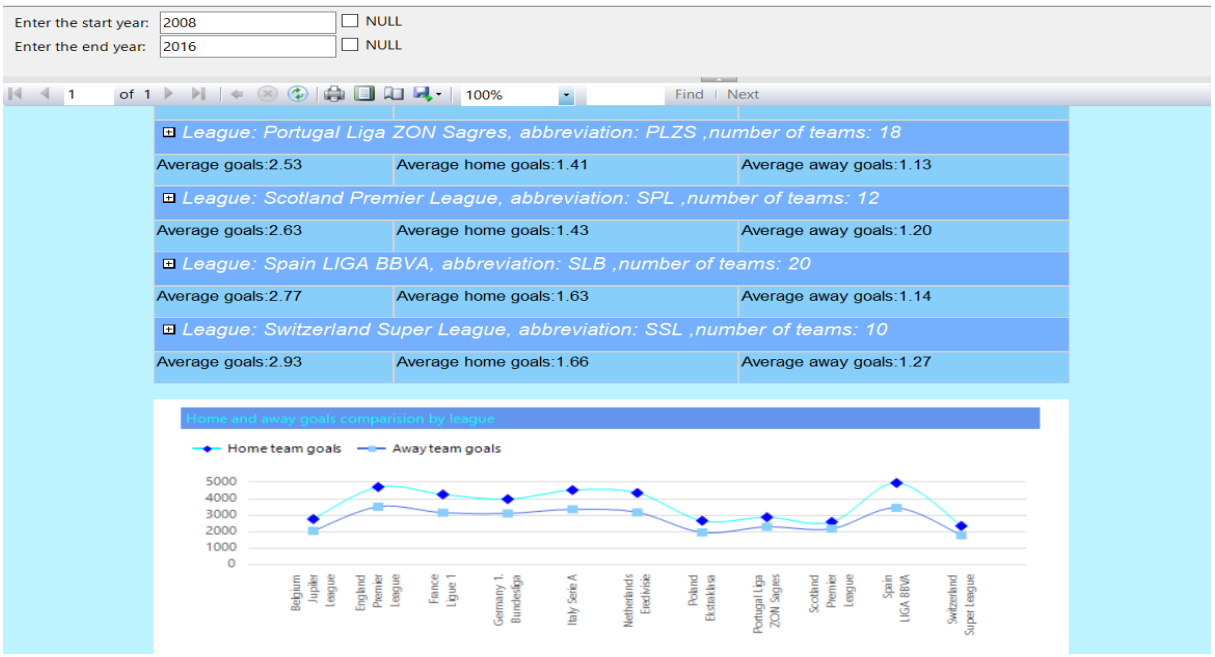


Slika 28: Drugi izveštaj

Da bismo odgovorili na drugi korisnički zahtev,koji za cilj ima utvrdjivanje koliko je postignuto golova u Engleskoj Premier ligi u toku 2014 godine,dovoljno je da za početnu i krajnju godinu prosledimo istu godinu,a to je u ovom slučaju - 2014. Godina,kao što je to uradjeno na slici 29. Uvidom u izveštaj,vidimo da je u toku 2014.godine u Engleskoj Premijer ligi postignuto 1035 golova,kao i još neke zanimljive informacije poput prosečnog broja postignutih golova u toj godini.

Slika 29: ,Drugi izveštaj

Ono što još treba istaći,jeste da ovaj izveštaj sadrži i drugi deo u vidu linijskog dijagrama sa markerima,koji se prikazuje odmah ispod tabele. Ovaj dijagram ima za cilj da prikaže odnos između postignutih golova u gostima i kod kuće po različitim ligama. Analiziranjem ovog drugog dela izveštaja može se steći uvid u to gde domaći teren ima veliku prednost pa su razlike izmedju broja postignutih golova u gostima i kod kuće značajne,odnosno gde domaći teren i nema toliku prednost,pa su razlike izmedju broja postignutih golova u gostima i kod kuće manje. Prikaz drugog dela izveštaja dat je na slici 30.



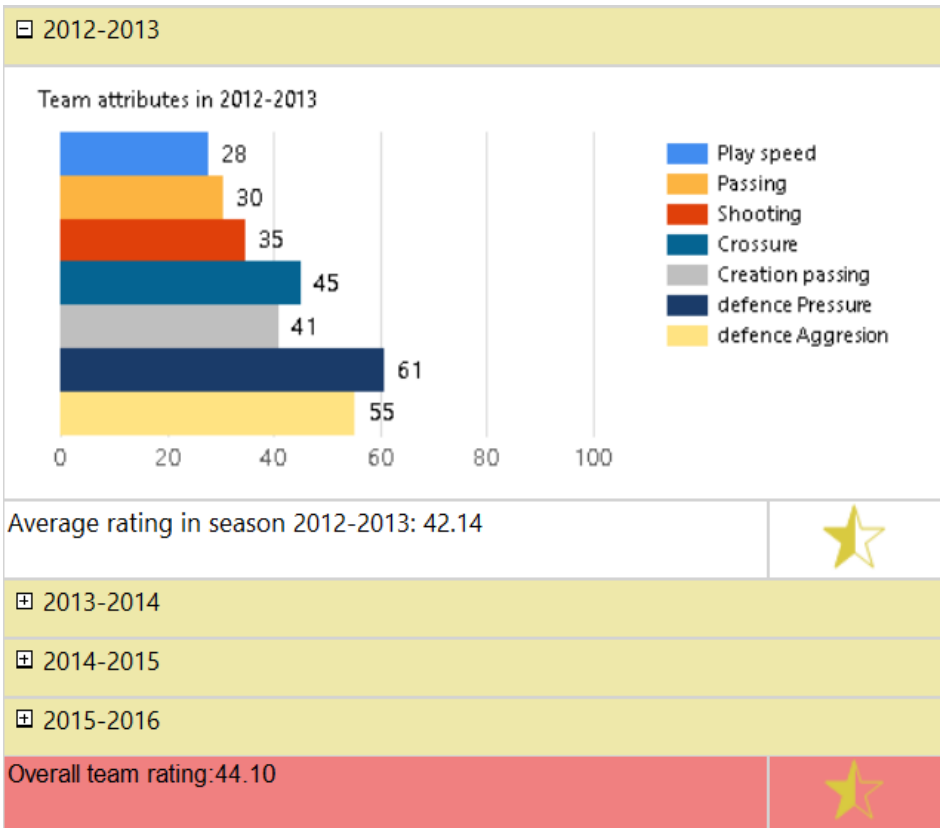
Slika 30: Drugi izveštaj,drugi deo

3. Treći izveštaj

Treći kreirani izveštaj ima za cilj da odgovori na sledeći korisnički zahtev:

- U kojoj sezoni je fudbalski tim Arsenal iz Engleske Premier lige imao najveće ocene,odnosno opšti rejting posmatrano relevantne atribute?

Slika 31: Treći izveštaj



Slika 32: Treći izveštaj,nastavak

Kao što se može videti na slikama 31 i 32,ovaj izveštaj prikazuje atribute timova po različitim sezonama. Podaci koji se tiču atributa tima su najpre grupisani po sezonama,zatim po pripadajućim timova na koje se ti atributi odnose,i na kraju je izvšeno grupisanje po ligama kojim ti timovi pripadaju. Ukoliko se otvore podaci za svaku ligu, vidljivi su pojedinačno svi timovi u okviru te lige,i slično tome,ukoliko se otvore podaci za svaki tim,vidljive su pojedinačne sezone. Proširivanjem sezone,stiče se uvid u atribute tima u toj sezon,koji su prikazani pomoću data bar-a.Takodje dodat je indikator zvezdice za prikaz prosečnog rejtinga tima,kako na kraju svake pojedinačne sezone,tako i na kraju svih sezona. Uvidom u izveštaj možemo videti da je Arsenal imao najveće ocene u sezoni 2011-2012.

4.Četvrti izveštaj

Četvrti kreirani izveštaj ima za cilj da odgovori na sledeće korisničke zahteve:

- Koliko se procentualno napravljenih faulova pretvori u javnu opomenu u vidu kartona u Francuskoj fudbalskoj ligi u sezoni 2015/2016?
- Koliki procenat šuteva,od svih upućenih šuteva(i u okvir gola,i van njega) završi u golu u italijanskoj Serie A ligi u sezoni 2014/2015?
- Koliki je prosečan broj kartona u poslednjem kolu u Bundesligi u sezoni 2015/2016?

Izveštaj omogućava korisniku da pomoću padajuće liste izabere željenu fudbalsku ligu i sezonu za koju želi da se prikaže izveštaj. Da bi se izveštaj pokrenuo oba parametra moraju biti odabrana. Izveštaj ima za cilj da prikaže odigrane utakmice po kolima u odabranoj ligi za odabranu sezonu. Pored toga,izveštaj prikazuje i opširnu statistiku koja se tiče dešavanja na svakoj pojedinačnoj utakmici. Za svako kolo prikazuje se prosečan broj golova,kartona,faulova,kornera,šuteva. Takođe,dodati su i procenti koji pokazuju koliko prosečno šuteva završi u golu,odnosno koliko prosečno napravljenih faulova bude kažnjeno javnom opomenom u vidu kartona. Takodje,prosek svih ovih podataka se pokazuje i na kraju sezone,nakon svih odigranih kola. Treba napomenuti da su podaci u okviru grupe sortirani po datumu odigravanja utakmice.

Matchs statistics per round in France Ligue 1 in season 2015-2016

Round 1

Date of match	Teams	Result	Fouls	Cards	Cards %	Corners	Shots on target	Shots off target	Goal %
Friday, 07. 08 2015.	LOSC Lille-Paris Saint-Germain	0:1	37	7	18.92%	5	12	5	5.88%
Saturday, 08. 08 2015.	SC Bastia-Stade Rennais FC	2:1	32	2	6.25%	8	5	8	23.08%
Saturday, 08. 08 2015.	Olympique de Marseille-SM Caen	0:1	20	1	5.00%	16	14	15	3.45%
Saturday, 08. 08 2015.	Montpellier Herault SC-Angers SCO	0:2	27	5	18.52%	14	9	14	8.70%
Saturday, 08. 08 2015.	FC Nantes-En Avant de Guingamp	1:0	29	5	17.24%	8	13	4	5.88%
Saturday, 08. 08 2015.	OGC Nice-AS Monaco	1:2	35	6	17.14%	12	7	10	17.65%
Saturday, 08. 08 2015.	ES Troyes AC-GFC Ajaccio	0:0	15	4	26.67%	13	5	9	0.00%
Sunday, 09. 08 2015.	Girondins de Bordeaux-Stade de Reims	1:2	26	5	19.23%	7	14	12	11.54%
Sunday, 09. 08 2015.	Olympique Lyonnais-FC Lorient	0:0	23	3	13.04%	9	14	15	0.00%
Sunday, 09. 08 2015.	Toulouse FC-AS Saint-Etienne	2:1	19	2	10.53%	7	7	12	15.79%
	Average stats after 1. round	Avg. goals: 1.70	Avg. fouls: 26.30	Avg. cards: 4.00	Avg %: 0.15	Avg. corners: 9.90	Avg. shot on: 10.00	Avg. shot off: 10.40	Avg %: 0.09

Round 2

	Average stats after 2. round	Avg. goals: 1.60	Avg. fouls: 26.80	Avg. cards: 5.30	Avg %: 0.20	Avg. corners: 8.90	Avg. shot on: 10.30	Avg. shot off: 9.70	Avg %: 0.08

Round 3

Slika 33: Četvrti izveštaj

Pick the league: England Premier League
Pick the season: 2015-2016
View Report

Matchs statistics per round in England Premier League in season 2015-2016

Round 1

	Average stats after 1. round	Avg. goals: 3.00	Avg. fouls: 25.70	Avg. cards: 4.70	Avg %: 0.19	Avg. corners: 9.00	Avg. shot on: 13.50	Avg. shot off: 8.60	Avg %: 0.13

Round 2

Date of match	Teams	Result	Fouls	Cards	Cards %	Corners	Shots on target	Shots off target	Goal %
Friday, 14. 08 2015.	Aston Villa-Manchester United	0:1	24	4	16.67%	8	5	8	7.69%
Saturday, 15. 08 2015.	Tottenham Hotspur-Stoke City	2:2	29	4	13.79%	7	17	9	15.38%
Saturday, 15. 08 2015.	Swansea City-Newcastle United	2:0	20	5	25.00%	8	11	9	10.00%
Saturday, 15. 08 2015.	West Ham United-Leicester City	1:2	23	5	21.74%	12	11	8	15.79%
Saturday, 15. 08 2015.	Watford-West Bromwich Albion	0:0	23	3	13.04%	6	11	11	0.00%
Saturday, 15. 08 2015.	Southampton-Everton	0:3	25	6	24.00%	18	14	8	13.64%
Saturday, 15. 08 2015.	Sunderland-Norwich City	1:3	14	3	21.43%	12	11	7	22.22%
Sunday, 16. 08 2015.	Manchester City-Chelsea	3:0	32	6	18.75%	6	16	10	11.54%
Sunday, 16. 08 2015.	Crystal Palace-Arsenal	1:2	26	2	7.69%	12	20	7	11.11%
Monday, 17. 08 2015.	Liverpool-Bournemouth	1:0	30	5	16.67%	14	9	20	3.45%
	Average stats after 2. round	Avg. goals: 2.40	Avg. fouls: 24.60	Avg. cards: 4.30	Avg %: 0.18	Avg. corners: 10.30	Avg. shot on: 12.50	Avg. shot off: 9.70	Avg %: 0.11

Slika 34: Četvrti izveštaj

Na slikama 33 i 34 dat je samo uopšeni prikaz delova četvrtog izveštaja kako bi se stekao uvid u njegov izgled. Odgovore na postavljene korisničke zahteve nažalost nije moguće prikazati putem slika zbog obimnosti izveštaja,te je za otkrivanje odgovora potrebno pokrenuti izveštaj. Uvidom u pokrenuti izveštaj,dolazi se do sledećih odgovora na postavljena pitanja,respektivno:

- U Francuskoj fudbalskoj ligi u sezoni 2015/2016 se u karton procentualno pretvori 16% napravljenih faulova,odnosno otprilike svaki 6 faul se sakcioniše kartonom,
- U italijanskoj Serie A ligi u sezoni 2014/2015 u golu procentualno završi 13% upućenih šuteva
- Prosečan broj kartona u poslednjem kolu u Bundesligi u sezoni 2015/2016 je 3.78

5. Peti izveštaj

Peti kreirani izveštaj ima cilj da odgovori na sledeći korisnički zahtev:

- Koji fudbalski tim ima najveći broj postignuthih golova u gostima u Španskoj ligi u sezoni 2012/2013 i koliko je to golova?

Ovaj izveštaj pomoću pie chart-a prikazuje top 5 domaćih ili gostujućih fudbalskih ekipa(u zavisnosti od toga za šta se korisnik opredeli,a podrazumevano je da se po defaultu prikazuje za domaće ekipe), u odabranoj ligi,koja se bira pomoću padajuće liste u odnosu na odabranu kategoriju koja se takođe bira iz padajuće liste. Moguće kategorije jesu golovi,kartoni,faulovi i korneri. Takođe,korisnik ima opcionu mogućnost da unese i sezonu koja ga interesuje. U slučaju da ne unese vrednost za ovaj parametar,biće mu prikazani podaci za sve sezone. Na slici 35 dat je izgled izveštaja na kom su parametri podešeni tako da se dobije odgovor na postavljeni korisnički zahtev. Kao što se može videti, najveći broj postignutih golova u gostima u Španskoj ligi u sezoni 2012/2013 ima fudbalski tim Barcelona i to je 52 postignuta gola.

Slika 35: Peti izveštaj

Kombinacijom parametara u ovom izveštaju moguće je kreirati poprilično velik broj različitih prikaza izveštaja,a na narednoj slici 36,biće dat još jedan prikaz izveštaja,u slučaju kada korisnik ne unese vrednost za parametar sezona.

Slika 36: Peti izveštaj

ZAKLJUČAK

U okviru ove dokumentacije prikazan je i objašnjen proces kreiranja DW sistema na osnovu izvornog seta podatala koji sadrži podatke o odigranim utakmicama u 11 fudbalskih liga. Postupak je podrazumevao razumevanje podataka, kreiranje OLAP šeme, implementaciju iste, zatim kreiranje ETL procesa i popunjavanje skladišta podacima, i na kraju odgovaranje na korisničke zahteve putem izveštaja.

Dalji razvoj ovog projekta može podrazumevati nekoliko stvari. Prva stvar može biti prikupljanje najnovijih podataka o odigranim utakmicama,s obzirom da je prošlo već 6 godina od poslednjeg ažuriranja dataset-a i po mogućnosti svakodnevno ažuriranje skladišta kako bi podaci u svakom trenutku bili verodostojni stvarnim podacima. Sledeća stavka može biti korišćenje podataka kako bi se vršile određene predikcije,kao što su predvidjanja ishoda utakmica ili broja postignutih golova.

Takođe,dalji razvoj bi mogao obuhvatiti i uključivanje one podšeme transakcione baze podataka koja u ovom projektu nije obrađena,a reč je o igračima i njihovim atributima. Uključivanjem igrača, bila bi kreirana šema pahuljica, i u velikoj meri bi budući kreirani izveštaji nad takvim skladištem podataka bili značajno informativniji.