



PROCES RAZMENE PODATAKA TOKOM INTEGRACIJE INFORMACIONIH SISTEMA DVE BANKE

DATA EXCHANGE PROCESS DURING DATA INTEGRATION BETWEEN INFORMATION SYSTEMS OF TWO BANKS

Petar Trećakov, *Fakultet tehničkih nauka, Novi Sad*

Oblast: INFORMACIONE TEHNOLOGIJE

Kratak sadržaj – U radu je prikazan proces razmene podataka tokom procesa integracije informacionih sistema Vojvodanske banke i OTP banke. Prikazana su teorijska razmatranja i ključni pojmovi vezani za migraciju podataka i ETL. Opisane su i prikazane metodologije i alati koji su korišćeni pri migraciji podataka – MSSQL, SSIS i Winforms.

Ključne reči: Baza podataka, informacioni sistem, migracija podataka

Abstract – In the paper is presented the process of data exchange between the information systems of Vojvodjanska banka and OTP banka, during the data migration process. The definition of terms and literature review of data migration and ETL are described in the paper. Methodology and tools used during the data migration process are presented and described – MSSQL, SSIS and Winforms.

Keywords: Database, information system, data migration

1. UVOD

Smisao odabranog rada je da se kroz primer u praksi prikaže proces razmene podataka između informacionih sistema dve različite banke. U radu će biti prikazan proces razmene podataka, odabrane tehnologije i struktura baza podataka sa prikazanim tabelama koje su se koristile prilikom razmene podataka. Takođe će biti prikazano kako se vrši razmena podataka od izvorišne do odredišne baze korišćenjem SSIS paketa.

U radu je opisan proces razmene podataka tokom spajanja informacionih sistema Vojvodanske banke i OTP banke. U poglavlju teorijske osnove prikazani su osnovni pojmovi vezani za migraciju podataka, faze migracije, ETL i razvoj ETL-a. U poglavlju Metod istraživanja šablonski je prikazano kako izgleda migracija u slučaju Vojvodanske i OTP banke, zatim su prikazane tabele u bazi podataka, kao i tok razmene podataka korišćenjem SSIS paketa. U poglavlju Rezultati istraživanja na konkretnom primeru je prikazano kako izgleda razmena podataka iz baze podataka Vojvodanske banke u bazu OTP banke.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Darko Stefanović, vanred. prof.

2. TEORIJSKE OSNOVE

Poglavlje teorijske osnove predstavlja sva teorijska razmatranja u vezi rada. Ovde će biti opisani pojmovi vezani za migraciju podataka, ETL i alati koji se koriste pri migraciji podataka.

2.1. Migracija podataka

Migracija podataka je proces selekcije, pripreme, ekstrakcije i transformisanja podataka i trajne razmene podataka iz jednog kompjuterskog skladišnog sistema u drugi. Takođe, ocenjivanje kompletnosti migriranih podataka i uklanjanje zastarelih podataka smatraju se delom procesa migracije podataka. Migracija podataka je jedno od ključnih razmatranja pri implementaciji, poboljšanju ili konsolidaciji bilo kog sistema, i uglavnom se sprovodi na način da se postigne najviši mogući nivo automatizacije, čime se oslobađaju ljudski resursi od repetitivnih zadataka. Migracija podataka nastaje zbog niza razloga, uključujući zamenu serverske opreme, poboljšanja ili održavanja, migraciju aplikacija, konsolidaciju web sajtova, oporavak od štete, i relokacija data centra [1] [2].

2.2 ETL – Extraction, Transformation, Loading

Extraction, Transformation i Loading procesi odgovorni su za operacije koje se održavaju u pozadini data warehouse arhitekture. U okviru ETL procesa, prvo se podaci ekstrahuju iz njihovih izvora koji mogu biti OLTP ili nasleđeni sistemi, fajlovi bilo kog formata, web stranice ili razni tipovi dokumenata. Uglavnom se samo podaci koji se razlikuju od podataka prethodnog izvršavanja ETL procesa (nove, modifikovane ili obrisane informacije) ekstrahuju iz izvora. Nakon ove faze, ekstrahovani podaci propagiraju se u specijalni deo data warehouse-a, koji se naziva Data Staging Area (DSA), gde se obavlja transformacija, homogenizacija i čišćenje podataka. Najčešće korišćene transformacije jesu filteri i provere koje osiguravaju da podaci poštuju poslovna pravila i ograničenja integriteta, kao i transformacije shema koje osiguravaju da se podaci uklapaju u odredišnu shemu data warehouse-a. Na kraju, podaci se učitavaju u centralni data warehouse. U tradicionalnom data warehouse okruženju, ETL proces periodično osvežava data warehouse tokom perioda niskog opterećenja (npr svake noći) i postoji određen vremenski okvir tokom kojeg se proces mora izvršiti. U sadašnje vreme, poslovne potrebe zahtevaju skoro pa real-time osvežavanje data warehouse-a i značajna pažnja usmerena je ka ovoj vrsti tehnološkog napretka [3].

3. METOD ISTRAŽIVANJA

Poglavlje Metod istraživanja predstavlja šematske prikaze i opise procesa migracije u Vojvođanskoj banci. Prikazan je tok procesa migracije podataka, od izvorne baze do određene baze. Takođe su prikazane tabele u koje se smeštaju podaci koji će biti migrirani.

Pored tabela prikazani su SSIS paketi koji predstavljaju alat putem kojeg se podaci transformišu i učitavaju u tabele. Nakon SSIS paketa prikazan je način na koji radi Winforms aplikacija koja služi za automatizovanje pokretanja SSIS paketa.

3.1. Migracija podataka u Vojvodanskoj banci

Migracija podataka počinje identifikovanjem izvora podataka. U ovom slučaju, izvor podataka jeste Oracle baza. Podaci iz Oracle baze transformišu se putem SSIS paketa. Nakon transformacije, podaci se perzistiraju u određenu MSSQL bazu podataka.

3.2. Prikaz procesa migracije

Nakon prvog koraka identifikovanja izvorne Oracle baze, potrebno je kreirati Stage tabele koje su veza između izvornih tabela i određinih tabela.

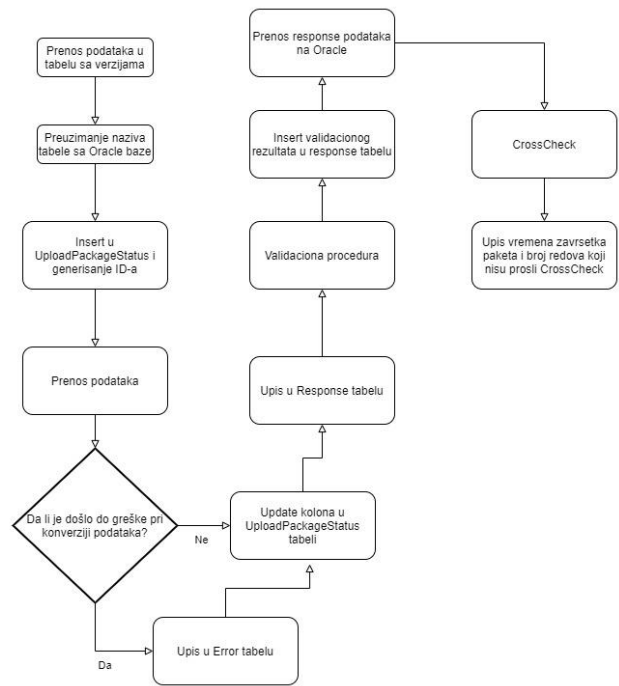
Tabele koje se koriste su sledeće:

1. Stage tabela.
2. Error event tabela.
3. Tabela sa verzijama.
4. Response tabela.
5. Oracle environment tabela.
6. UploadPackageStatus tabela.

Najpre se u Oracle environment tabelu upisuju podaci o nazivu tabele, nazivu izvorne tabele i na kojoj oracle shemi se nalazi tabela. Podaci iz Oracle environment tabele se kasnije koriste u aplikaciji radi lakog pokretanja paketa. UploadPackageStatus tabela koristi se za perzistiranje podataka o izvršenim paketima. Ovdje se nalaze podaci o nazivu paketa, nazivu izvorne i određene tabele, vreme pokretanja i vreme završetka paketa, kao i da li je bilo grešaka pri izvršavanju paketa.

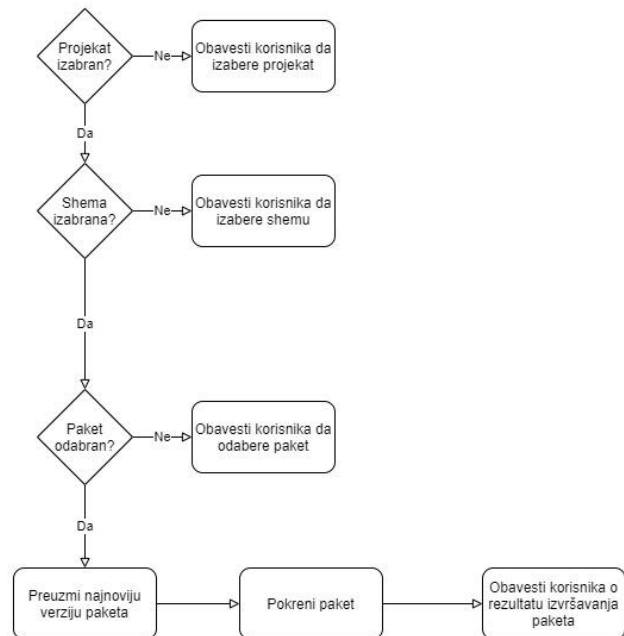
U konkretnu stage tabelu prenose se podaci iz izvorne tabele. U error event tabelu smeštaju se podaci o greškama prilikom prenosa podataka. U tabeli sa verzijama se čuvaju svi prethodni prenosi podataka koji su se izvršili. U response tabeli se čuvaju podaci o uspešno ili neuspešno prenetim podacima. Stage tabela kreira se na osnovu tabele logičkog mapiranja.

Nakon kreiranja tabele, sledeći korak jeste kreiranje SSIS paketa. U okviru SSIS paketa vrše se sve neophodne transformacije podataka, prenos podataka i prenos u Error tabelu. Radi automatizacije izvršavanja paketa, kreirana je Windows forms aplikacija, koja omogućava odabir više paketa i izvršavanje svih odabranih paketa. Nakon transformacija i čišćenja podataka, podaci se iz stage tabele učitavaju u produkcionu tabelu, što je ujedno i poslednja faza prenosa podataka.



Slika 1. Šematski prikaz SSIS paketa

Winforms aplikacija služi za automatizovano pokretanje SSIS paketa. Aplikacija najpre vrši proveru da li je izabran projekat u kom se nalazi paket. Ukoliko nije, projekat je potrebno izabrati. Dalje se vrši provera da li je izabrana oracle shema na kojoj se nalazi tabela. Ukoliko nije, korisnik je potrebno da izabere shemu. Na kraju je potrebno da se odabere sam paket koji će se izvršiti. Pre pokretanja paketa, aplikacija preuzima najnoviju verziju paketa. Nakon izvršenja paketa, korisnik dobija informaciju o uspešnom ili neuspešnom izvršavanju paketa.



Slika 2. Šematski prikaz Winforms aplikacije

4. REZULTATI ISTRAŽIVANJA

Poglavlje rezultati istraživanja prikazuje konkretan primer gde su opisane sve korišćene tabele u bazi podataka, SSIS paketi koji se koriste za transformaciju podataka, i prikazana je Winforms aplikacija koja služi za automatizovanje pokretanja SSIS paketa.

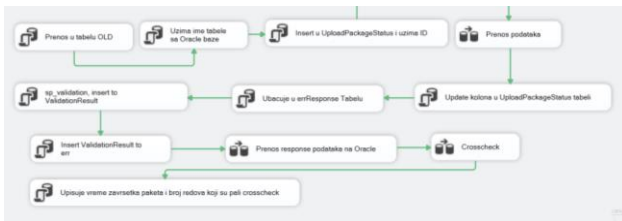
4.1. Baze podataka

Tabele koje su korišćene u prenosu podataka su sledeće:

1. Employees tabela: služi za perzistiranje najnovijih podataka.
2. EmployeesOld tabela: služi za smeštanje prethodnih verzija podataka iz Employees tabele. Pri pokretanju paketa prvo se prenose podaci iz Employees u EmployeesOld tabelu, a zatim se Employees tabela prazni.
3. EmployeesError tabela: služi za perzistiranje podataka o greškama koje su se desile tokom izvršavanja paketa. Greške se mogu desiti tokom transformacije podataka, validacije podataka ili cross-checka.
4. ErrEmployees tabela: služi za perzistiranje rezultata izvršavanja paketa, bilo da je došlo do grešaka ili do uspešnog transformisanja podataka.
5. UploadPackageStatus tabela: služi za perzistiranje podataka o paketima, kao što su početak i kraj izvršavanja paketa, broj redova koji su preneti i nazivi tabela.
6. ChangeOracleEnvironment tabela: pomoćna tabela koja služi za smeštanje podataka o projektu kom pripadaju paketi, kao i shema na kojoj se nalaze tabele.
7. SqlOracleTableNameMapping pomoćna tabela u kojoj se nalaze nazivi tabela sa MSSQL kao i Oracle baze. Tabela služi za proveru da li su nazivi tabela isti.
8. ValidationResult tabela: služi za perzistiranje podataka o rezultatima validacione procedure.
9. ValidationErrorCode tabela: služi za perzistiranje podataka o tipovima grešaka koje se mogu javiti u validacionoj proceduri.

4.2. SSIS paketi

Prikazan je SSIS paket koji služi za transformisanje i prenos podataka iz Oracle baze u MSSQL bazu. U nastavku je prikazan radni tok paketa Employees.



Slika 3. Employees SSIS paket

Prvi korak u prenosu podataka jeste prenos u tabelu OLD. U tabeli OLD sadržane su sve prethodne verzije podataka, i tabela služi kao istorijski pregled podataka. U ovom slučaju tabela se zove EmployeesOLD. Tabela Employees se prazni da ne bi došlo do nagomilavanja podataka, jer se u njoj perzistiraju samo aktuelni podaci. Naredni korak jeste preuzimanje naziva izvorne tabele sa Oracle baze.

Zatim se vrši insertovanje reda u UploadPackageStatus tabelu i generisanje jedinstvenog id broja. UploadPackageStatus tabela sadrži informacije o izvršavanju paketa, koji paket se izvršio u koje vreme, da li je bilo grešaka pri izvršavanju i broj redova input i output tabela. Naredni korak jeste prenos podataka odnosno Data flow komponenta. Oracle source komponenta omogućava pristup podacima izvorne tabele. Izvorni podaci zatim prolaze kroz konverziju korišćenjem Data Conversion komponente.

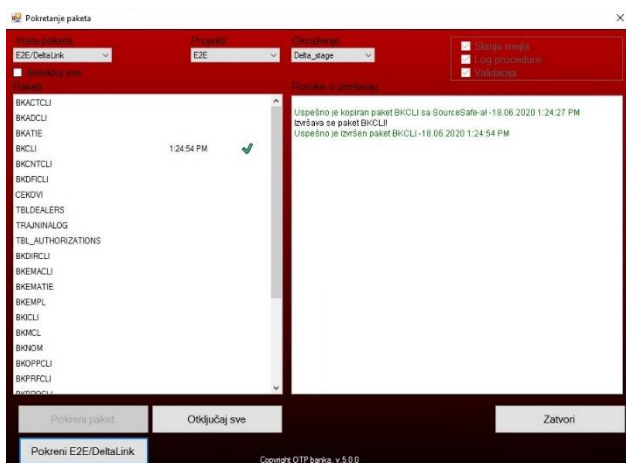
Script Component komponenta smešta informacije o greškama u kolone Description Error koja predstavlja opis greške i Column Error koja predstavlja naziv kolone kod koje je došlo do greške. Naredni korak jeste prosleđivanje informacije o grešci i smeštanje u tabelu. Prosleđivanje informacija o greškama radi se putem Derived Column komponente. U ovoj komponenti prave se dve nove kolone kojima se prosleđuje Description Error i Column Error podaci iz prethodne komponente. U narednom koraku svi podaci zajedno sa opisima grešaka i kolona sa greškama (ukoliko je došlo do greške) smeštaju se u tabelu EmployeesError korišćenjem OLE DB Destination komponente. Ukoliko nije došlo do greške prilikom konverzije, koristi se Derived Column komponenta, u kojoj se nalazi idUploadPackageStatus kolona i DataMigrationStatus kolona. DataMigrationStatus kolona sadrži informaciju o statusu migriranog podatka i može imati sledeće vrednosti:

- -1: Greška pri validaciji/transformisanju podataka
- 1: Podatak prošao validaciju
- -2: Greška pri učitavanju podataka
- 2: Podatak učitao

Nakon Derived Column komponente koristi se Script Component komponenta ukoliko je došlo do greške pri insertovanju podataka u odredišnu bazu. Zatim se greške pri insertovanju slično kao i greške pri konverziji smeštaju u Error tabelu, sa podacima o tome koja je greška u pitanju i koja kolona. Naredni korak jeste Execute SQL Task komponenta, u kojoj se vrši update kolona u UploadPackageStatus tabeli. U UploadPackageStatus tabelu vrši se upis podataka o broju redova input tabele, broju redova output tabele, broju redova error tabele ukoliko je došlo do greške, i vreme završetka paketa.

4.3. Winforms aplikacija

Radi jednostavnijeg pokretanja SSIS paketa, napravljena je aplikacija u Windows Forms tehnologiji. Za izradu aplikacije korišćen je Visual Studio alat i C# programski jezik. Projekat se sastoji iz nekoliko klasa: klasa GetDataFromDB pribavlja podatke iz baze o paketima i tabelama u koje se prenose podaci, SetDataToDB služi za update-ovanje podataka u bazi o paketima i tabelama, i glavna forma jeste ExecPackage. Klasa GetDataFromDB sadrži Dataset GetProject. GetProject dataset pribavlja sve projekte iz TabeleChangeOracleEnvironment u kojoj su smešteni projekti. Pre izvršavanja paketa kroz aplikaciju, vrše se provere da li su odabrani svi neophodni parametri: da li je odabrana vrsta projekta, da li je odabrana shema sa koje se prenose podaci, da li je odabran paket i okruženje. Na narednoj slici prikazan je interfejs aplikacije za pokretanje paketa.



Slika 4. Winforms aplikacija

5. ZAKLJUČAK

U radu je opisan proces razmene podataka između dve banke. Kroz poglavlje teorijske osnove opisani su osnovni pojmovi u vezi razmene podataka, migracije podataka i SSIS paketa.

Prikazan je proces razmene iz Oracle baze jedne banke, u MSSQL bazu druge banke.

Dalje su prikazane različite tabele i njihove uloge u procesu razmene podataka. Nakon tabela prikazan je SSIS paket koji služi za razmenu podataka između izvora (Oracle baza) i odredišta (MSSQL baza). Sledeći deo koji je prikazan jeste šematski prikaz Winforms aplikacije i provere koje aplikacija izvršava da bi se mogao izvršiti SSIS paket.

U narednom poglavlju je kroz konkretan primer prikazano sve što je bilo opisano u prethodnom poglavlju. Prvo su prikazane tabele gde će biti smešteni transformisani podaci. Zatim je prikazan SSIS paket koji vrši razmenu od izvorišne baze, transformiše podatke, i upisuje podatke u odredišnu bazu. Na kraju je prikazana Winforms aplikacija koja služi za automatizovano izvršavanje SSIS paketa.

SSIS paketi imaju nekoliko prednosti:

1. Grafički interfejs omogućava da se na brz način povežu komponente toka podataka, i vrlo brzo se dolazi do željenog rezultata.
2. Ukoliko u osnovnom paketu komponenti ne postoji komponenta koja je potrebna za dalji rad, moguće je preuzeti sa interneta u obliku third party plugina.
3. SSIS omogućava izmenu i dodavanje C# koda unutar komponenti, što pomaže pri prilagođavanju potrebama projekta.

Neki od nedostataka SSIS paketa:

1. Ukoliko tabela ima veliki broj kolona (200+), mapiranje kolona postaje otežan zadatak koji dugo traje, i onda korišćenje paketa kao brzo rešenje gubi svoju svrhu.
2. Pri instalaciji SSIS paketa i plugina treba biti oprezan, jer se mora obratiti pažnja na verziju SQL Servera, verziju Visual Studia, da li je sistem 32-bitni ili 64-bitni i slično.

6. LITERATURA

- [1] J. Morris, Practical Data Migration, 2012.
- [2] A. Warmuth, B. Dufrasne, J. Appel, W. Bauer, S. Douglass, P. Klee, M. Pura, M. Wells and B. Wesselbaum, DS8870 Data Migration Techniques, 2017.
- [3] P. Vassiliadis and A. Simitsis, "Extraction, Transformation, and Loading".

Kratka biografija:



Petar Trećakov rođen je u Novom Sadu 1992. godine. Diplomirao je na Ekonomskom fakultetu u Subotici 2016. godine, smer Poslovni informacioni sistemi. Kontakt: petar.trecakov@gmail.com