ETL za skladišta podataka - Microsoft SQL Server 2017 Integration Service

Antolić, Sofija

Sveučilište u Rijeci – Odjel za informatiku
Diplomski studij informatike, modul Informacijski i komunikacijski sustavi

Sofija Antolić

ETL ZA SKLADIŠTA PODATAKA - MICROSOFT SQL SERVER 2017 INTEGRATION SERVICES

Diplomski rad

Mentori: prof. dr. sc. Patrizia Poščić
dr. sc. Danijela Jakšić

Rijeka, rujan 2019

Ovom prilikom želim zahvaliti svim profesorima i asistentima, te naročito mentorima svog završnog i diplomskog rada za lijepe godine akademske suradnje.

– Sofija Antolić

Sadržaj

1. Uvodna riječ

Poslovna informatika, odnosno poslovna inteligencija, jedna je od najraširenijih grana u informatičkoj industriji. Unutar ovog polja razvijaju se velike i istaknute tvrtke te se generiraju značajni novčani prihodi dok su zanimanja vezana uz poslovnu inteligenciju u kontinuiranoj potražnji na tržištu rada.

Elektronske baze podataka u raznim varijantama u modernoj poslovnoj klimi idu ruku pod ruku sa svakom uspješnom kompanijom. Naravno, iza njih stoje razne tehnologije i ljudi koji na njima rade. Firme često imaju svoj tim informatičara ili kupuju usluge informatičkih firmi koje rade u području poslovne inteligencije, to može biti i varijanta gdje kompanija ima mali tim svojih informatičara, a kada je potrebno raditi neki značajan razvoj tehnologije onda ugovaraju vanjske suradnike. Uzevši u obzir važnost i profitabilnost ove grane informatike, očekivano je da tehnološki divovi poput Amazona, Microsofta i sličnih prodaju komercijalan softver za rad s bazama podataka i da postoje tvrtke koje se bave specifično poslovnom inteligencijom.

Važna komponenta poslovnih informatičkih sustava je skladište podataka s kojim su povezani procesi koji ga pune ili održavaju (ETL procesi). Tema rada je demonstracija tih procesa kroz izradu praktičnog primjera.

Softver koji će biti proučen i opisan u ovom radu proizvod je tvrtke Microsoft. **Microsoft SQL Server** je poznati sustav koji nudi kompletno okruženje za rad s relacijskom bazom podataka. Specifičnije, fokus rada je na **Microsoft SQL Server 2017 Integration Services** (skraćeno SSIS). **SSIS** je dio MS SQL Server softvera koji nudi brojne mogućnosti za razvoj ETL-a.

SSIS je potrebno staviti u kontekst, u radu će prvo biti opisana nužna teorija za razumijevanje poslovne inteligencije. Poslovna inteligencija općenito, potom skladište podataka te **ETL**. Teoretski dio je kratak jer je fokus rada **SSIS** (odnosno primjena alata).

Nakon teoretskog uvoda slijedi praktični dio rada u kojem će se na primjerima prikazati moguće implementacije ETL-a kroz SSIS. Demonstracija je prikazana na skladištu podataka za fiktivnu franšizu pekari. ETL će biti prikazan u funkciji punjenja tablica činjenica i tablica dimenzija, a nakon toga i u funkciji izrade izvješća odnosno u reportingu. Fokus će biti na komponentama alata koje su po mojoj procijeni najučestalije i najbitnije u praktičnoj primjeni.

2. Teoretska podloga

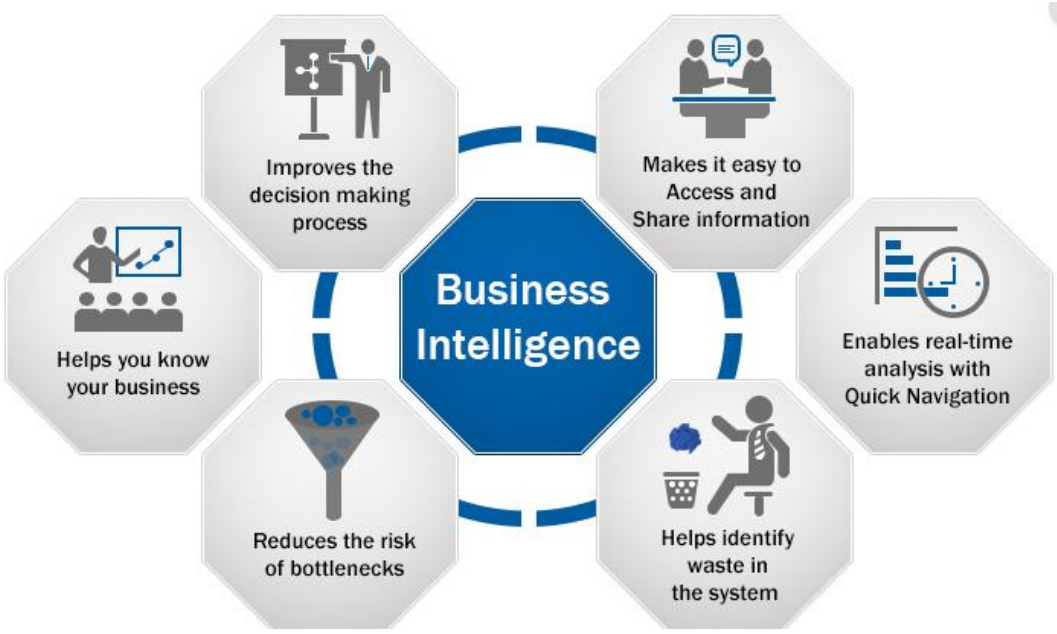
2.1 Poslovna inteligencija

Poslovna inteligencija (eng. **Business intelligence**) koja se često u industriji spominje kraticom **BI** (u kontekstu: BI Tools, BI Developers itd.) je apstraktni pojam koji obuhvaća granu informatičke industrije koja se bavi poslovnim podacima. Tu spadaju razni aspekti djelatnosti: skladišta podataka, razvoj poslovnih aplikacija, obrada podataka.

„Poslovna inteligencija ne govori poslovnim korisnicima što da rade ili što će se dogoditi u slučaju ako odaberu neki put, niti je poslovna inteligencija samo generiranje izvješća. Umjesto toga poslovna inteligencija nudi mogućnost razumijevanja trendova i dublji uvid u poslovanje.“ [1]. Ova rečenica iz literature jako dobro u najkraćoj crti opisuje srž BI-ja.

Osnovna ideja koja stoji iza BI grane je da velika poslovanja zbog svoje veličine moraju imati nekakav informatički sustav koji bilježi sve bitne podatke kroz cijeli životni ciklus poslovanja, na taj način dobiva se povijesni pregled nekih ključnih metrika u koje onda menadžer ima uvid i može donositi bolje poslovne odluke. Brže donošenje strateški povoljnih poslovnih odluka je bitno za jako velike tvrtke i zato danas postoje informatičke firme koje nude isključivo usluge BI-ja tvrtkama koje nisu informatičke (ne isplati im se imati vlastiti tim informatičara, vlastitu infrastrukturu), ali im je BI nužna komponenta poslovanja.

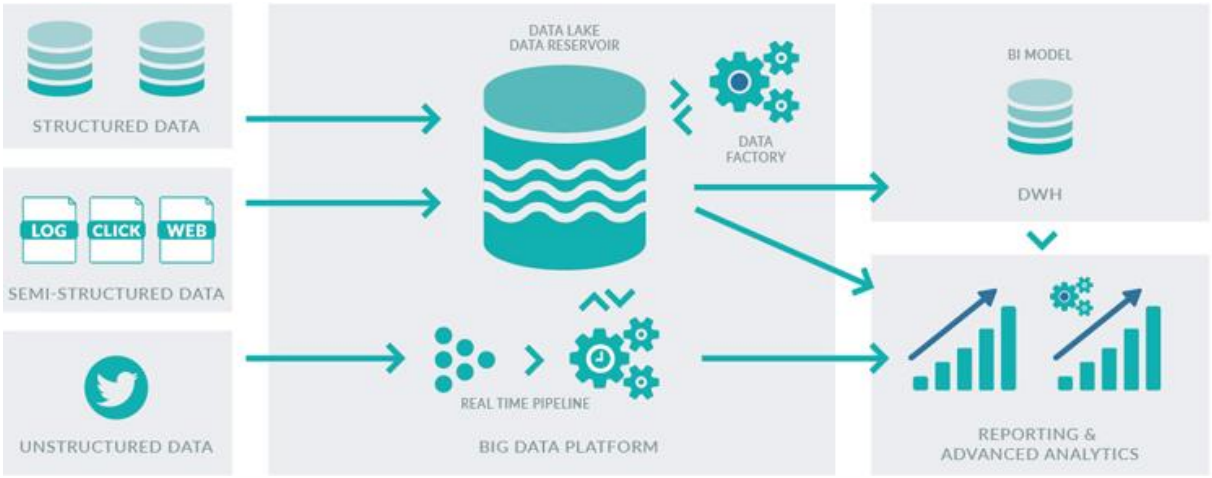
BI možemo promatrati sa stajališta programera/informatičara i sa stajališta menadžera odnosno osobe kojoj je potreban BI za poslovanje, dana ilustracija prikazuje BI kroz kontekst poslovanja.



Slika 1 Poslovna inteligencija u kontekstu poslovanja [7]

Slika 1 prikazuje dobrobiti koje BI donosi poslovanju, bolji uvid u poslovanje, bolje uočavanje potencijalnih problema i ostalo.

S druge strane možemo gledati BI kao programer, sljedeća ilustracija (Slika 2) prikazuje tipičnu arhitekturu nekog BI informacijskog sustava.



Slika 2 Arhitektura poslovne inteligencije[8]

Lijeva strana ilustracije prikazuje moguće izvore podataka, to mogu biti strukturirani podaci (iz neke transakcijske baze ili datoteka), a mogu biti i nestrukturirani (na skici primjer Twittera). Srednji dio prikazuje tehničku komponentu u kojoj se odvija ETL, ovdje vidimo da se podaci mogu procesirati sekvencijalno ili u stvarnom vremenu. Skroz desno skica prikazuje finalni proizvod koji uključuje **DWH** iz kojeg se može izvući izvještaj ili analitika, ali može biti i u stvarnom vremenu. Hibridna varijanta potrebna je gdje se analitika i izvještaj rade istovremeno kombinacijom podataka iz skladišta i podataka koji dolaze u realnom vremenu (u par milisekundi razlike od stvarne transakcije koja je zabilježena) i za takav pristup je potrebna arhitektura koja ima dio za procesiranje u stvarnom vremenu i dio za trajnu pohranu povijesti. Na taj način klijent vidi iscrtavanje grafova momentalno, ali isto tako može u par klikova vidjeti trend unatrag par mjeseci u usporedbi s trenutnim stanjem.

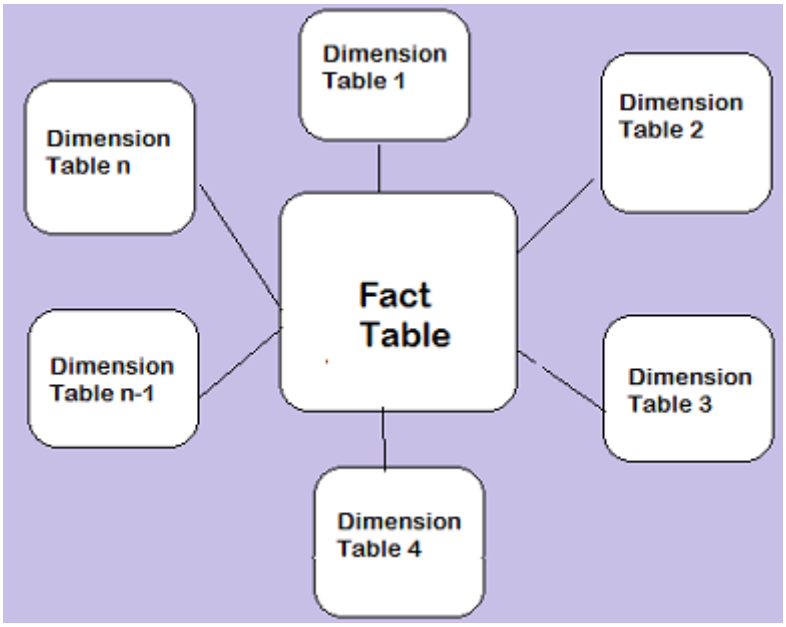
2.2 Skladišta podataka

„Skladište podataka je repozitorij za sve podatke prikupljene od strane nekog poslovnog sustava. Skladištenje podataka fokusirano je na sakupljanje podataka iz raznih izvora za pristup i analizu.“ [2]

Skladište podataka (eng. **Data Warehouse**) je središnja komponenta u BI-ju. Svi relevantni podaci nekog poslovanja spremaju se u elektroničkom obliku u neki dugotrajni oblik pohrane, ideja je zabilježiti sve povijesne podatke poslovanja kako bismo nad njima vršili analize.

Tehnološki gledano implementacije skladišta mogu biti različite, hardverski i softverski, s obzirom na program koji je tema seminara, ovdje se bavimo tradicionalnim (i najučestalijim) oblikom skladištenja podataka.

Dimenzijski model skladišta je u širokoj upotrebi, to je model SQL baze podataka koja se sastoji od dva tipa tablica: dimenzije i činjenice.



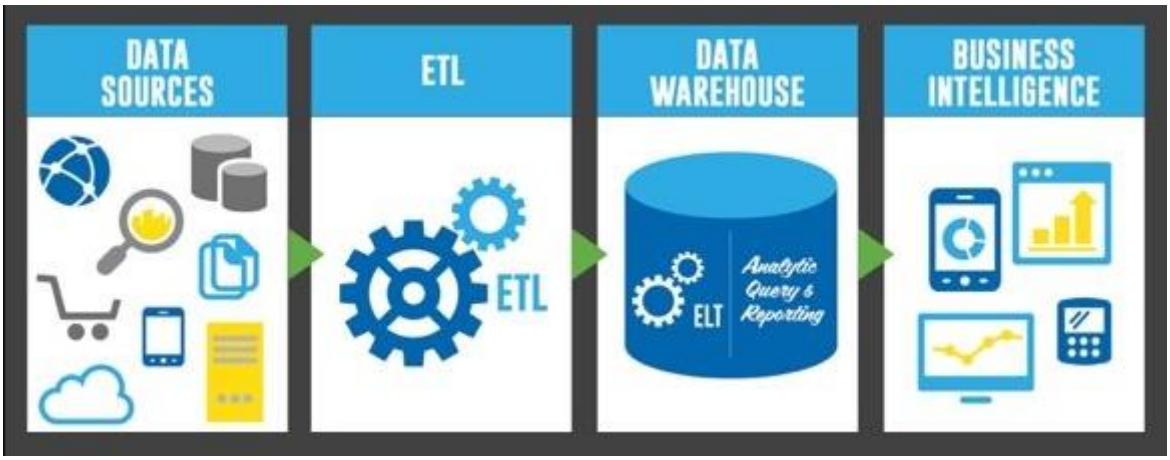
Slika 3 Dimenzijski model [9]

Dimenzije predstavljaju entitete koji postoje u našem poslovanju. Na primjer za neku bankovnu kompaniju dimenzije bi mogle biti: poslovnica, klijent, usluga. Tablice činjenica na apstraktnoj razini možemo gledati kao nekakve interakcije, događaje koji se odvijaju između dimenzija. U činjenicama se nalaze svi numerički podaci nad kojima vršimo analize. Na primjer tablica činjenica mogla bi se zvati fct_klijent_usluga_poslovnica i ona bi sadržavala sve usluge uslužene klijentima banke u svim poslovnicama, uz podatke koji su za uslugu bitni (uplaćen je određeni iznos novca, provizija je kalkulirana po nekoj stopi, transakcija se odvila u neko vrijeme...). Takva tablica mogla bi se koristiti za jednostavne analize, na primjer pogledamo mjesečni profit poslovnica. Ali također mogli bismo i izraditi i nešto kompleksnije, na primjer prema geografskoj distribuciji klijenata i poslovnica, te odlascima u poslovnice vidjeti koja bi nam mogla biti sljedeća potencijalna lokacija za otvaranje nove poslovnice.

2.3 ETL

ETL je složeni akronim od tri početna slova engleskih riječi: extract, transform, load. U kontekstu skladištenja podataka, ETL je termin koji se koristi za proces koji prenosi podatke iz nekog izvora do skladišta.

"ETL je skraćenica za extract, transform, load, tri funkcije kombinirane u jednom alatu za povlačenje podataka iz jedne baze i smještanje istih u drugu bazu**."** [3]



Slika 4 Položaj ETLa u arhitekturi [10]

Slika 4 prikazuje ETL kao tehnički sloj između raznih mogućih izvora podataka (internet, dokumenti…) i u konačnici samog **DW**-a. Ovdje se može vidjeti i kratica ELT, što se više manje odnosi na postupak koji ima isti cilj samo je izveden drugačije (zamjena load i transform koraka).

Korak E, odnosno ekstrakcija prvi je korak u procesu koji se može obaviti na razne načine, ekstrakcija je postupak vađenja podataka iz izvora podataka za naše skladište. Taj izvor može biti strukturiran ili nestrukturiran, najčešće se u kontekstu poslovanja radi o strukturiranim podacima koji dolaze u nekom digitalnom obliku, CSV dokumenti na primjer.

Korak T je transformacija podataka, postupak kojim ekstrahirane podatke prilagođavamo za ulazak u DW, kako bi podaci bili unificirani. U ovom koraku radimo korake poput zamjena vrijednosti, deriviranja novih vrijednosti (npr. sumiranje), transformacije slova, generiranja ključeva i ostalo.

L je „load", učitavanje podataka u finalnu tablicu u skladištu podataka. ETL je fokus praktičnog dijela ovog rada i kroz SSIS biti će prikazana većina funkcionalnosti koje on nudi za migraciju podataka između dvije točke.

3. Praktična demonstracija i opis SSIS alata

3.1 Instalacija

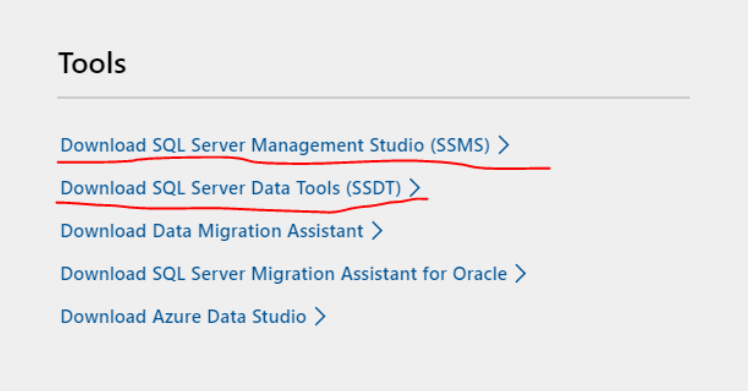
Kako bi ovaj projekt bio moguć za demonstrirati i opisati prvo je potrebno ispuniti neke preduvjete. Potrebno je instalirati sve potrebne komponente, nakon toga postaviti scenarij korištenja, jer izuzev objašnjavanja pukih funkcionalnosti SSIS-a prikazat ćemo i primjer koji bi mogao biti scenarij u stvarnoj upotrebi.

Sve potrebno za instalaciju je moguće pronaći navigacijom od glavne stranice namijenjene za SQL Server softver: https://www.microsoft.com/en-us/sql-server/sql-server-downloads.

Moramo preuzeti i instalirati **SQL Server 2017**, u ovom slučaju on-premises varijantu (Slika 5). To u biti znači da ga instaliramo na neki lokalni stroj, suprotno od "cloud" varijante gdje je sve u virtualnom oblaku.

Slika 5 Preuzimanje probne verzije MSSQL 2017

Sljedeće potrebno od alata se može pronaći u "Tools" sekciji na dnu stranice (Slika 6). Potrebni su **SSMS**, koji je u biti radno okruženje za integraciju svih SQL Server komponenti i **SSDT** koji sadrži BI alate među kojima je i nama najbitniji - **SQL Server Integration Services**.



Slika 6 Alati potrebni za demonstraciju

Instalacija je vođena intuitivnim koracima i dosta jednostavna tako da nema potrebe za dubljim pojašnjavanjem tog dijela.

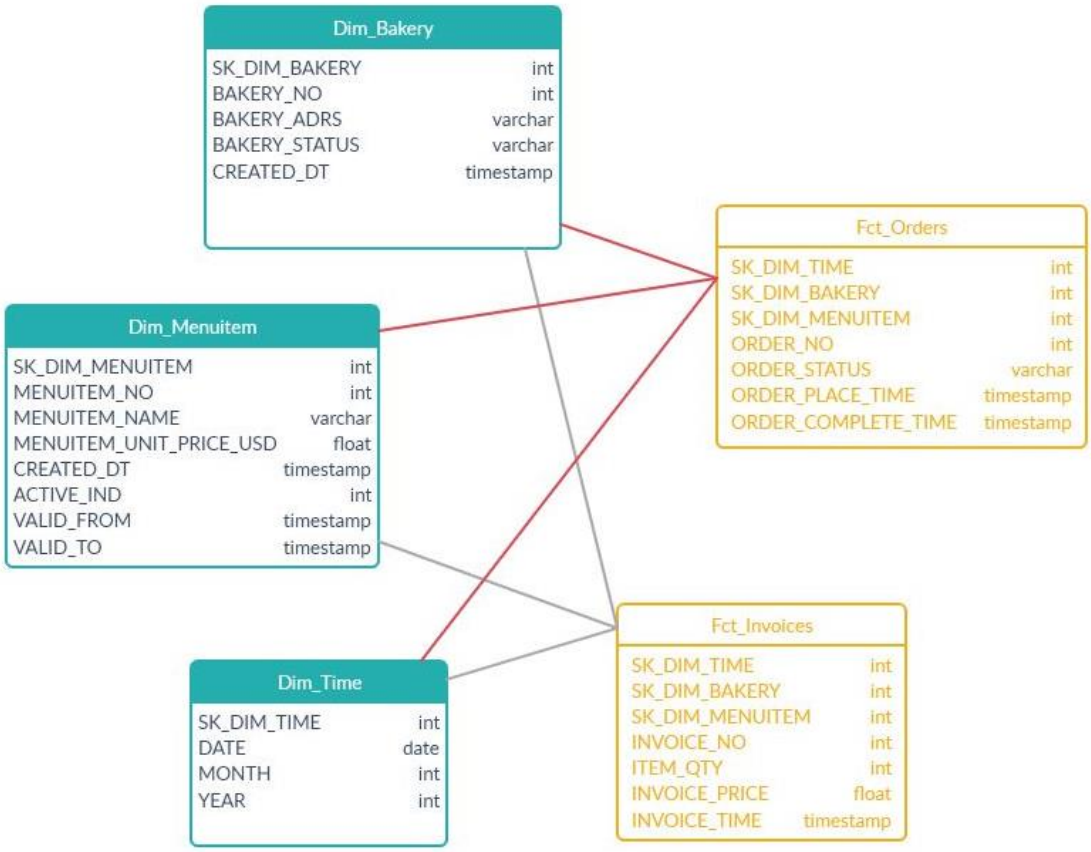
3.2 Postavljanje scenarija

Kako bi korištenje SSIS-a dobilo smisao, potrebno je kreirati nekakav scenarij, odnosno postaviti pretpostavku poslovnog okruženja.

Za ovaj slučaj zamislit ćemo da radimo SSIS pakete za jednu lokalnu franšizu pekari. Pekara ima svoj manji informacijski sustav koji uključuje nama nevidljive transakcijske sustave koji dnevno dostavljaju izvještaje svih transakcija neke poslovnice, koje onda treba dalje obrađivati. Na našoj strani nalazi se skladište podataka i zahtjevi za procesiranjem transakcijskih podataka.

3.3 Model pekare

Kreirat ćemo manju bazu u SQL Serveru prema ilustriranom modelu (Slika 7).



Slika 7 Dimenzijski model DW pekare

Imamo tri dimenzijske tablice, Dim_Bakery će biti tablica u kojoj su evidentirane sve poslovnice pekare, to je također mala dimenzija koja će biti punjena metodom truncate/reload, odnosno nakon svake promjene u stanju poslovnica jednostavno će se pobrisati i napuniti ponovno bez praćenja povijesti.

Druga važna dimenzija je Dim_Menuitem, u njoj su evidentirani svi pekarski proizvodi unutar franšize i ovo je takozvana "tip 2" dimenzija, što znači da ćemo takvu tablicu puniti na način da pratimo i povijesne zapise proizvoda.

Dimenzija Dim_Time je ovdje jer je dimenzija vremena potrebna u dimenzijskom modelu i zbog konvencionalnih i praktičnih razloga.

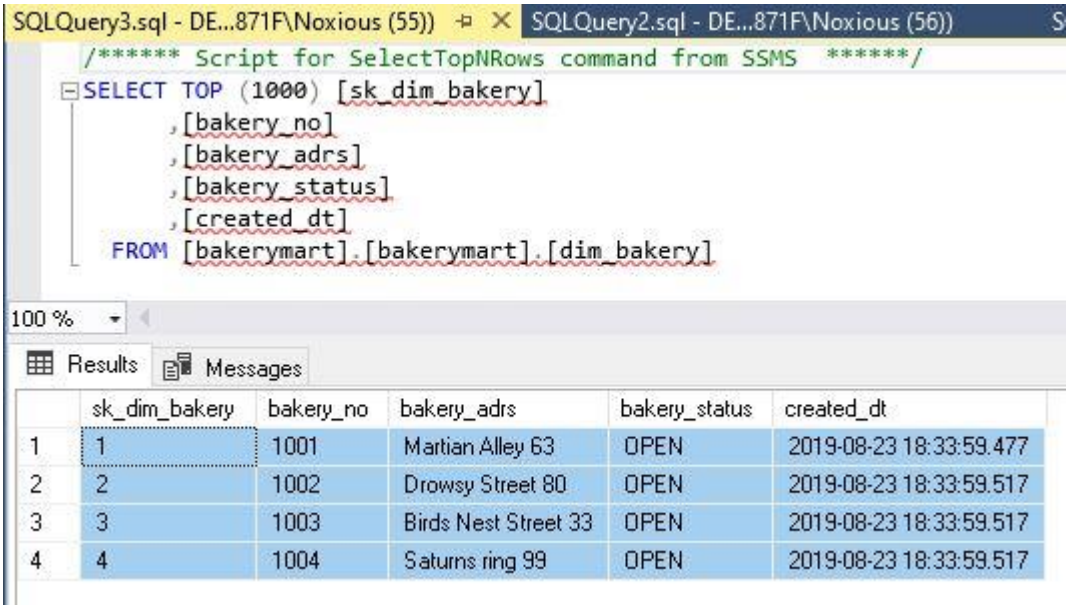
Sljedeće tablice u modelu su tablice činjenica, imamo dvije tablice činjenica. Zamislimo da pekarska franšiza ima opciju brzog naručivanja putem aplikacije za posebnu skupinu ugovorene klijentele (kafići, restorani, dućani), u tablicu Fct_Orders spremat ćemo podatke o tim narudžbama. Fct_Invoices nam je ključna tablica činjenica jer ćemo u nju spremati podatke o svim izdanim računima, što znači da je ključna za praćenje toka financija u našem slučaju.

Model je dosta pojednostavljen za svrhu demonstracije, u stvarnoj situaciji bi sigurno bilo potrebno skupiti još podataka poslovanju da bi sustav davao punu sliku. (stanje skladišta, nabava materijala i namirnica, tko je specijalna klijentela, koliki su rashodi na osoblje itd...).

Za postavljanje baze treba kreirati bazu, njene tablice i napraviti inicijalne zapise u dimenzijske tablice. To je moguće pronaći u skripti priloženoj kao praktični dio rada pod nazivom "diplomski_db_ddl.sql". Dovoljno ju je izvršiti u **SSMS** query editoru.

3.4 Pregled baze

Kao što je već rečeno, upoznavanje s SSIS-om nema smisla ako nema kontekst, iz tog razloga imamo bazu podataka koju ćemo puniti, a nakon tih manjih primjera koji prikazuju najbitnije funkcionalnosti biti će prikazane i ostale naprednije funkcije. Pogledajmo prvo naše podatke u dimenzijama.



	sk_dim_bakery	bakery_no	bakery_adrs	bakery_status	created_dt
1	1	1001	Martian Alley 63	OPEN	2019-08-23 18:33:59.477
2	2	1002	Drowsy Street 80	OPEN	2019-08-23 18:33:59.517
3	3	1003	Birds Nest Street 33	OPEN	2019-08-23 18:33:59.517
4	4	1004	Saturns ring 99	OPEN	2019-08-23 18:33:59.517

Slika 8 dim_bakery

Results Messages

	sk_dim_menuitem	menitem_no	menuitem_name	menuitem_unit_price_usd	active_ind	valid_from	valid_to	created_dt
1	1	101	Puffy Flake	0.5	1	2019-08-23 18:33:59.517	NULL	2019-08-23 18:33:59.517
2	2	102	Salty Pretzel	1.5	1	2019-08-23 18:33:59.540	NULL	2019-08-23 18:33:59.540
3	3	103	Doughnut	1	1	2019-08-23 18:33:59.540	NULL	2019-08-23 18:33:59.540
4	4	104	Hot-Dog	2	1	2019-08-23 18:33:59.540	NULL	2019-08-23 18:33:59.540
5	5	105	Milky Crust	0.8	1	2019-08-23 18:33:59.540	NULL	2019-08-23 18:33:59.540
6	6	106	Pizza Cut	2.5	1	2019-08-23 18:33:59.540	NULL	2019-08-23 18:33:59.540
7	7	107	Crisp Muffin	1	1	2019-08-23 18:33:59.540	NULL	2019-08-23 18:33:59.540

Slika 9 dim_menuitem

Results Messages

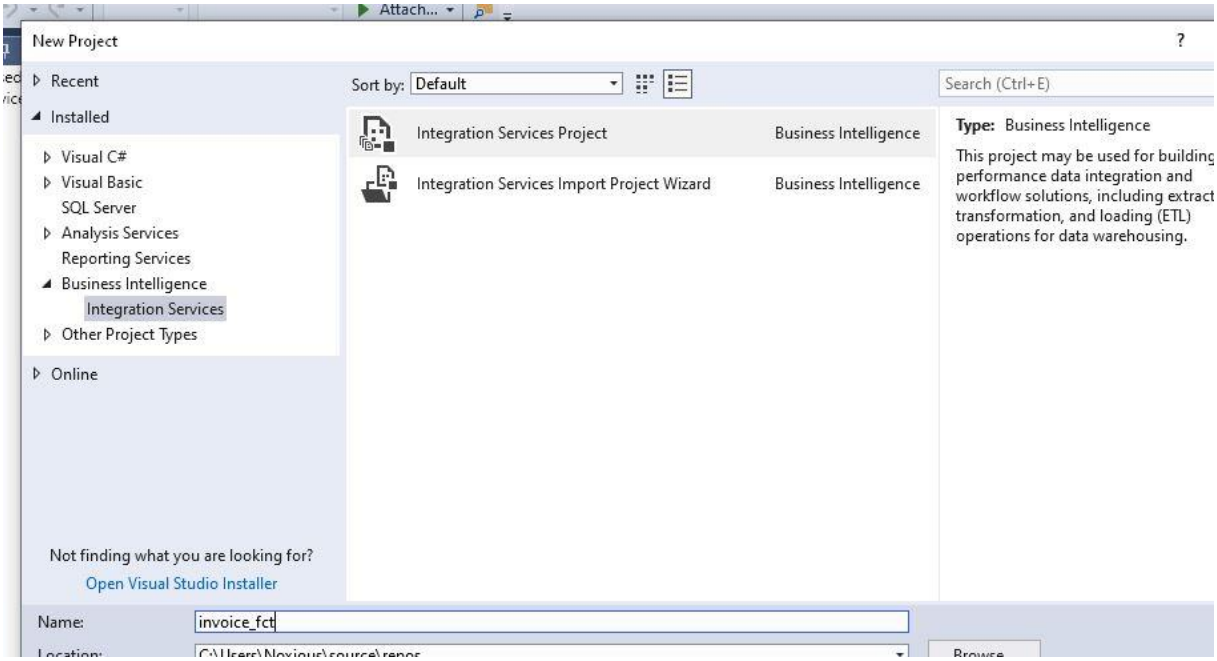
	sk_dim_time	date	month	year
1	20190822	2019-08-22	8	2019
2	20190823	2019-08-23	8	2019
3	20190824	2019-08-24	8	2019
4	20190825	2019-08-25	8	2019

Slika 10 dim_time

Tablice dimenzija (Slika 8, 9, 10) se nalaze u bazi uz trenutno prazne tablice činjenica. Prilikom punjenja tablica činjenica, dimenzije će nam služiti kao “lookup” tablice – na taj način si omogućujemo validaciju tijekom samog unosa u tablice činjenica kako bi potvrdili da se radi o stvarnim podacima za postojeće pekare i proizvode.

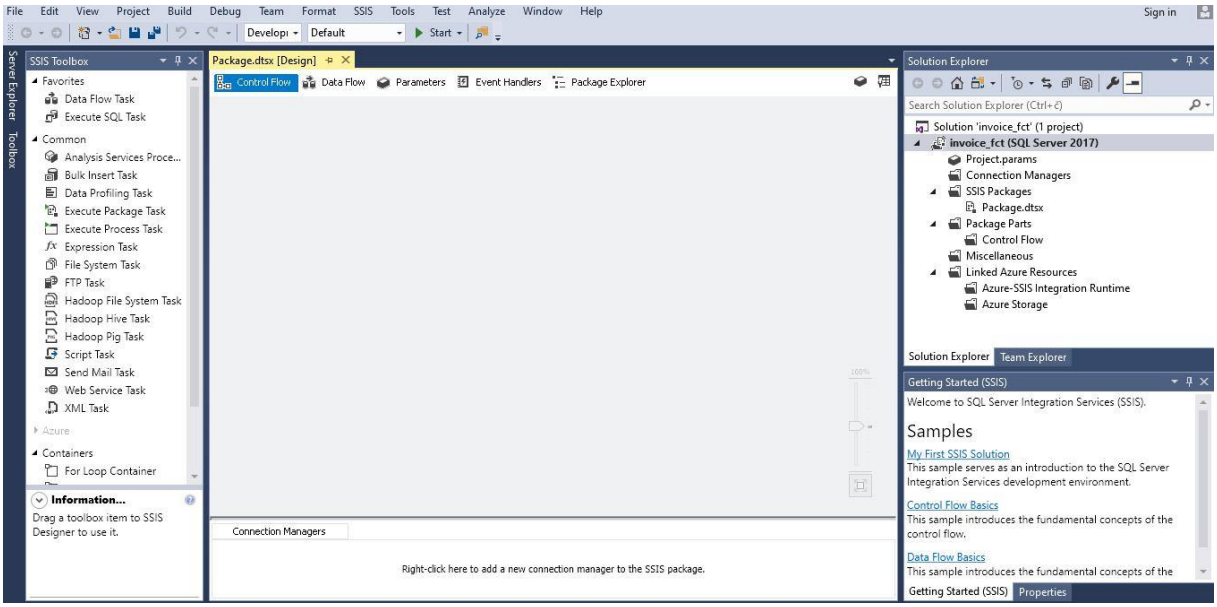
3.5 Upoznavanje sa SSIS sučeljem

Kada otvorimo Visual Studio 2017 (SSDT) prvo je potrebno izraditi novi **Integration Services** projekt odlaskom na **New Project** -> **Integration Services Project** (Slika 11).



Slika 11 Izrada SSIS Projekta

Nakon ovog koraka dobivamo pogled na radno sučelje SSIS-a (Slika 12). Skroz lijevo na slici vidimo SSIS **Toolbox**, tu se nalaze sve funkcije i alati koje možemo uključiti u dizajn našeg SSIS paketa. Sve to uz pomoć drag& drop ubacujemo na središnji gornji dio, to je površina na kojoj se odvija dizajn i još poneka funkcija ali o ostalim tabovima nešto kasnije. Na sredini na dnu nalazi se **Connection manager**, u tom okviru definiramo veze na naše izvore i odredišta. Skroz desno nalazi se hijerarhijska struktura datoteka vezanih uz projekt.

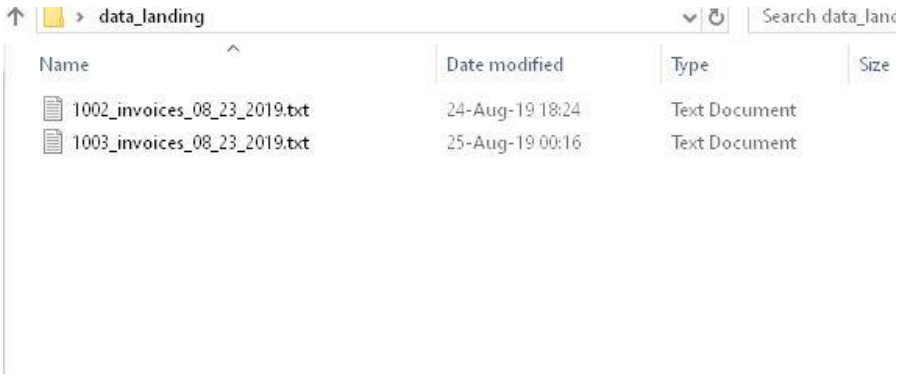


Slika 12 Pogled na SSIS sučelje

3.6 Scenarij 1: Punjenje tablica činjenica

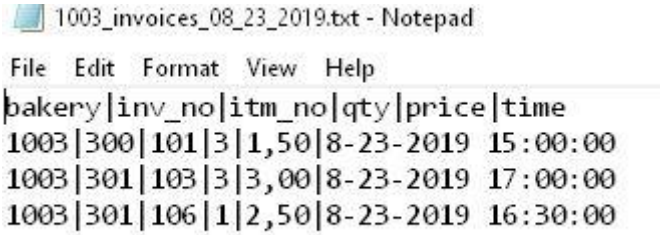
3.6.1 Fct_Invoices (osnovne funkcije, foreach loop container)

Kao prvi primjer za uvod u osnovne koncepte napravit ćemo SSIS paket koji puni tablicu činjenica Fct_Invoices. Zamislimo scenarij u kojemu nam svaki dan na kraju radnog vremena svaka naša pekara (nebitno kojom tehnologijom) pošalje na naš glavni server gdje se nalazi SSIS i DWH; tekstualnu datoteku sa ispisom svih obavljenih prodaja na taj dan, što rezultira time da u nekom našem **data_landing** direktoriju imamo tekstualne datoteke nazvane po formatu **{bakery_no}_invoice_{datum}.txt** .



Slika 13 data_landing direktorij

Neka je situacija idealna pa su sve datoteke identičnog formata (iako u praksi ponekad može biti problema), tako da u svakoj tablici imamo zapis prikazan na slici (Slika 14). Svaki zapis označava jedan prodani proizvod na nekom računu, broj računa (inv_no) se ponavlja jer jedan račun može imati više proizvoda. Stupci su odvojeni znakom "|". S obzirom da će nam trebati staging tablica u procesu, potrebno je izraditi i nju prema kolumnama u izvoru, iskoristit ćemo dani **DDL** (Slika 15) za kreaciju tablice. U SSIS-u takav korak nije nužan i mogli smo primjer izvesti i na drugi način, držanjem podataka u memoriji i obradom u memoriji do konačne destinacije ali ovaj primjer ilustrirati će tradicionalniji pristup ETL-u.

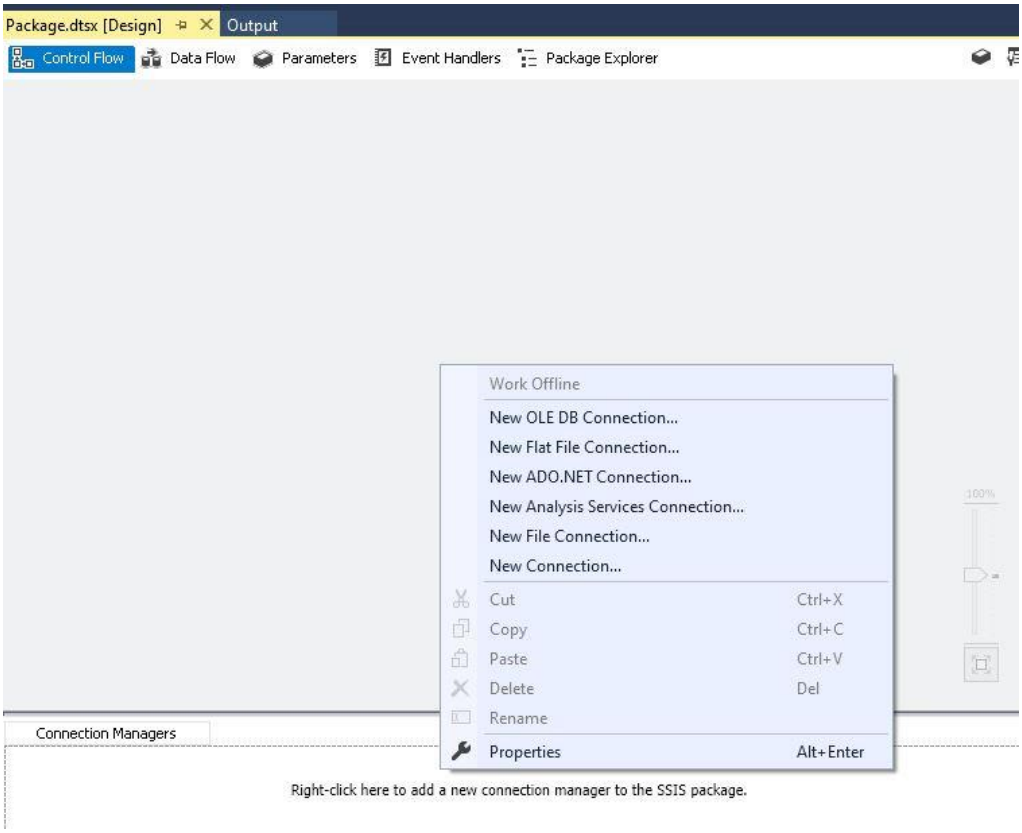


Slika 14 Uzorak podataka u izvoru

```
CREATE TABLE bakerymart.stg_fct_invoices(
bakery_no int,
invoice_no int,
item_no int,
item_qty int,
invoice_price float,
invoice_time datetime
);
```

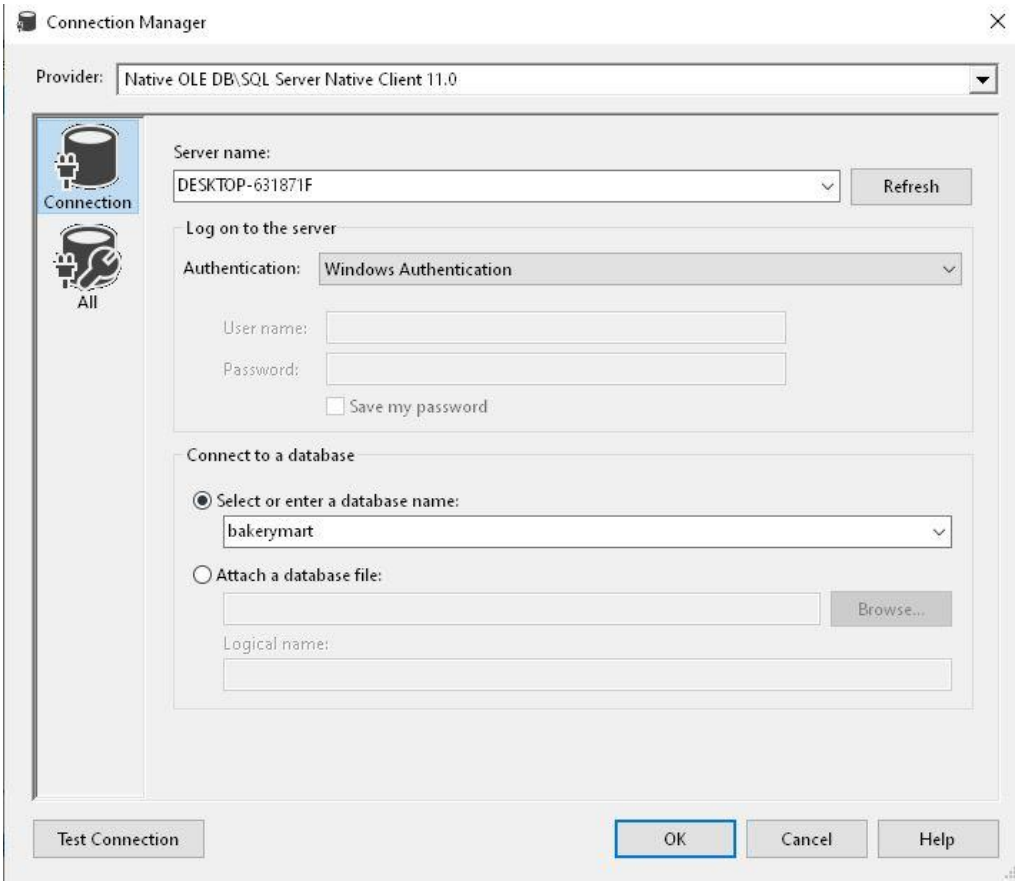
Slika 15 Invoices- Staging DDL

Nakon što smo izradili tablicu za staging možemo početi izrađivati SSIS paket, u prethodnom poglavlju izradili smo projekt i došli do sučelja. Prvi korak koji je dobro napraviti je isplanirati konekcije koje su nam potrebne, u našem slučaju trebat će nam konekcija na prethodno kreiranu bazu **bakerymart** i konekcija na datoteku, za prvu demonstraciju napraviti ćemo konekciju na samo jednu datoteku. Desnim klikom na **Connection Manager** otvara se meni sa opcijama za konekcije (Slika 16). Prvo odabiremo **New OLE DB Connection**.



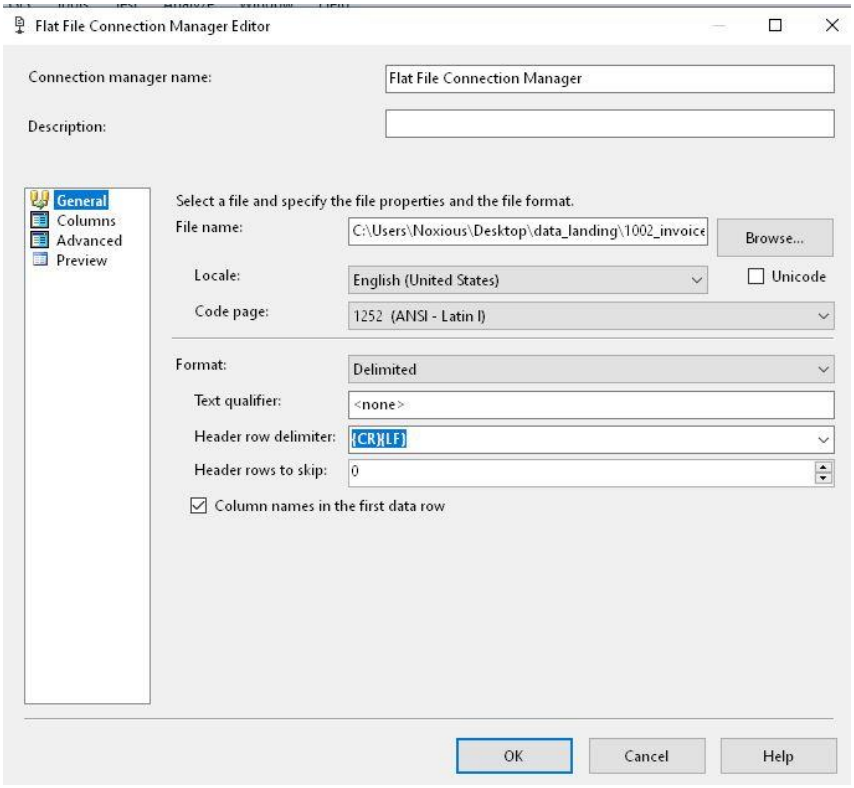
Slika 16 Desni klik Connection Manager

Sljedeći prozor (Slika 17) je konfiguriranje veze na DWH, s obzirom da je lokalna baza, potrebno je samo dodati **Server name** i odabrati našu bazu sa liste (ako je veza uspješna, lista s bazama bi se trebala sama učitati tako da možemo odabrati). U slučaju problema može se koristiti **Test connection** za testiranje veze na bazu.

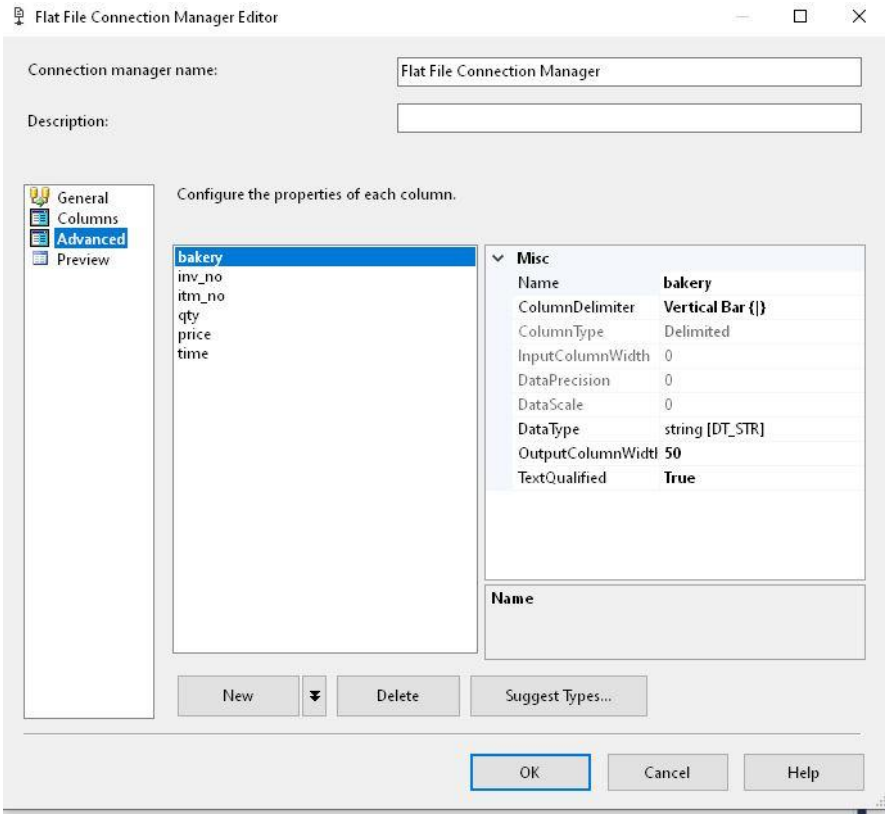


Slika 17 OLE DB konfiguracija konekcije

Nakon ove konekcije konfigurirat ćemo sljedeću odabirom opcije **New Flat File connection**, u ovom koraku pod **File name** definiramo putanju do datoteke koju čitamo (Slika 18), tu postoje još neke druge postavke (encoding, delimiteri) no ukoliko odemo na **Preview** opciju možemo vidjeti da je najčešće SSIS sam već prepoznao strukturu dokumenta. Ono što on ne prepoznaje sam je tip podatka, po općoj postavki sve bude učitano kao tekst, ali mi možemo otići na **Advanced** (Slika 19) i tamo odabrati ili **Suggest types** ili ručno odabrati kakav tip podatka želimo. Podatke je moguće i u kasnijoj obradi prebaciti između tipova, ali ovdje smo dodali konverziju direktno na izvor da odmah odgovara tipu podatka u staging tablici.

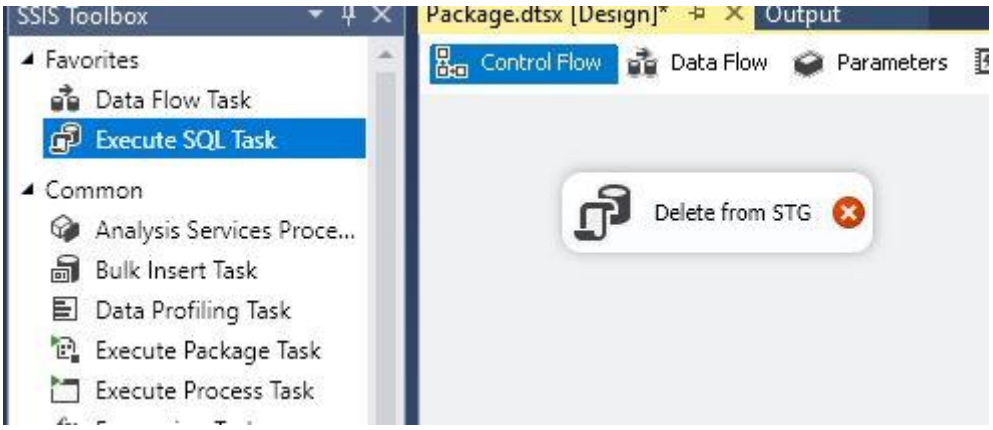


Slika 18 Flat File Connection – general



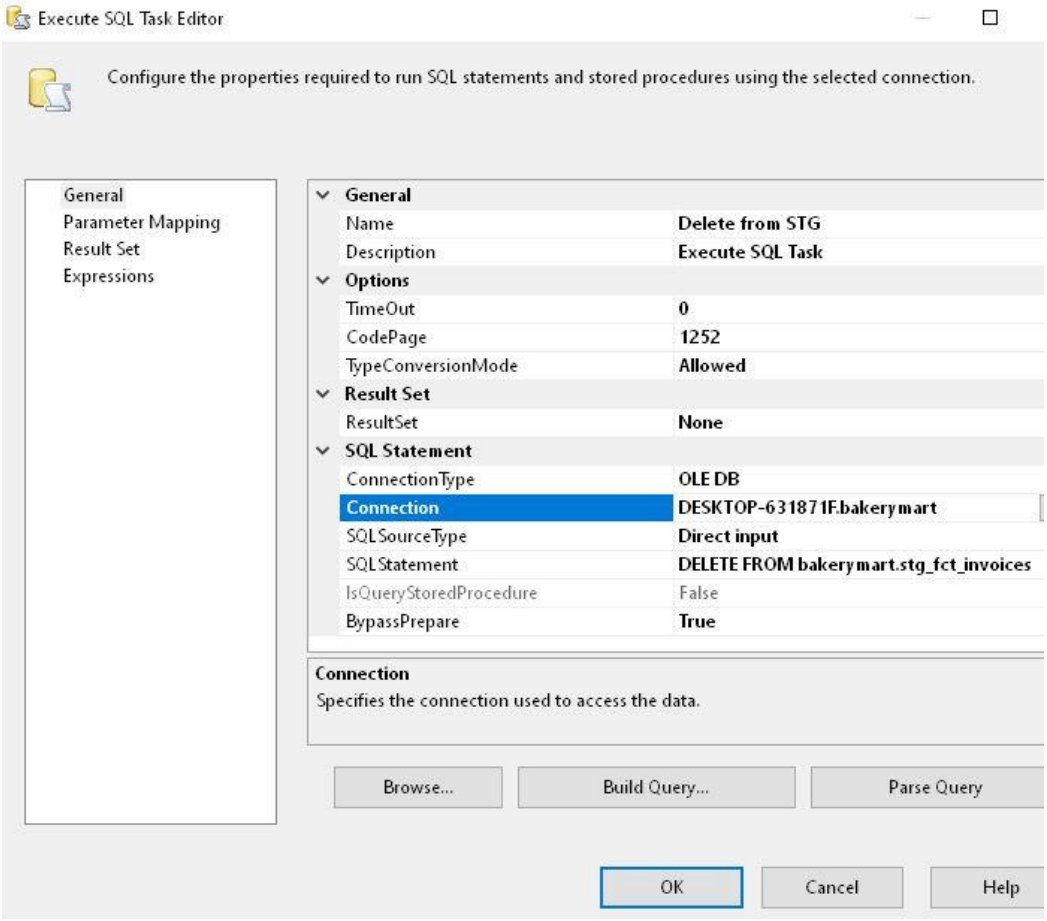
Slika 19 Flat File Connection – advanced

Sada s postavljenim konekcijama, možemo krenuti na dizajniranje SSIS paketa. U **Control Flow** ćemo sa **SSIS Toolbox** sekcije kliknuti i dovući **Execute SQL Task** komponentu te ju prikladno preimenovati (Slika 20).



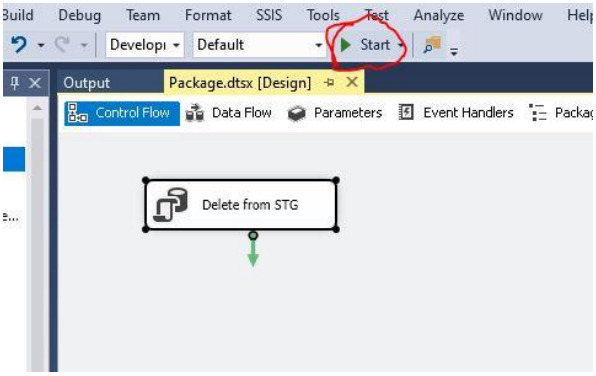
Slika 20 Exec SQL Task - Prva komponenta

Ova komponenta služit će nam na početku izvršavanja paketa da bismo ispraznili staging tablicu prije unosa podataka. Lijevi dvoklik dovesti će nas do konfiguracije komponente (to vrijedi za sve komponente). Ovdje ćemo odabrati konekciju na bazu u polju **Connection** te dodati potrebnu **DELETE FROM** klauzulu u **SQL Statement** (Slika 21).

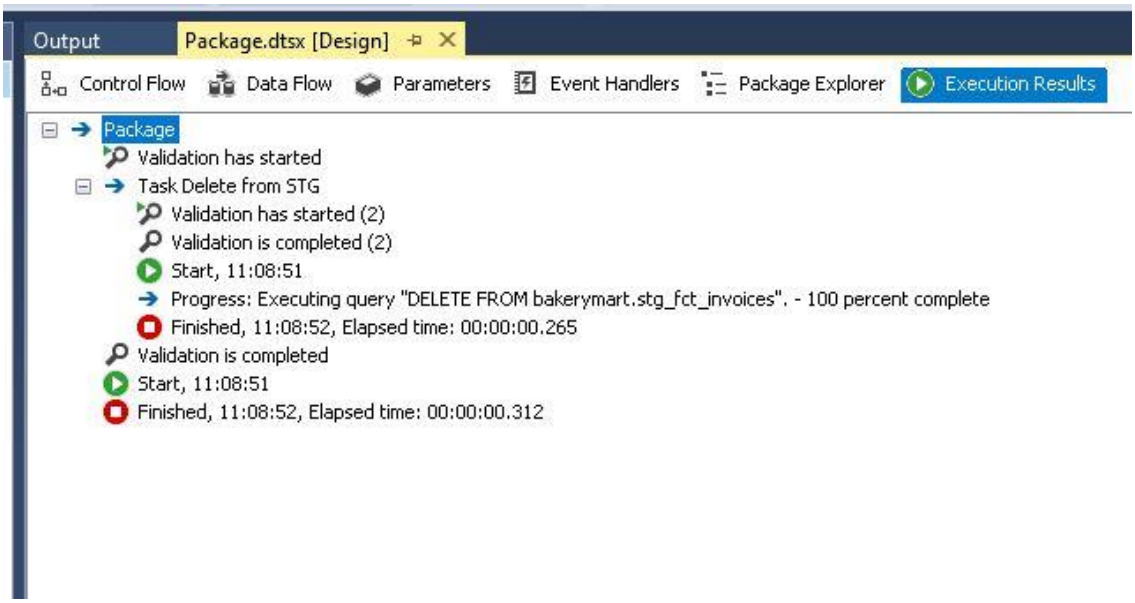


Slika 21 Exec SQL Task - konfiguracija

Sada već možemo pokušati izvršiti naš SSIS paket sa ovom jednom komponentom kako bismo testirali radi li klikom na **Start** (Slika 22). Rezultate izvršavanja možemo vidjeti u **Execution results** tabu (Slika 23) koji će nam biti jako koristan za ispravljanje pogrešaka.

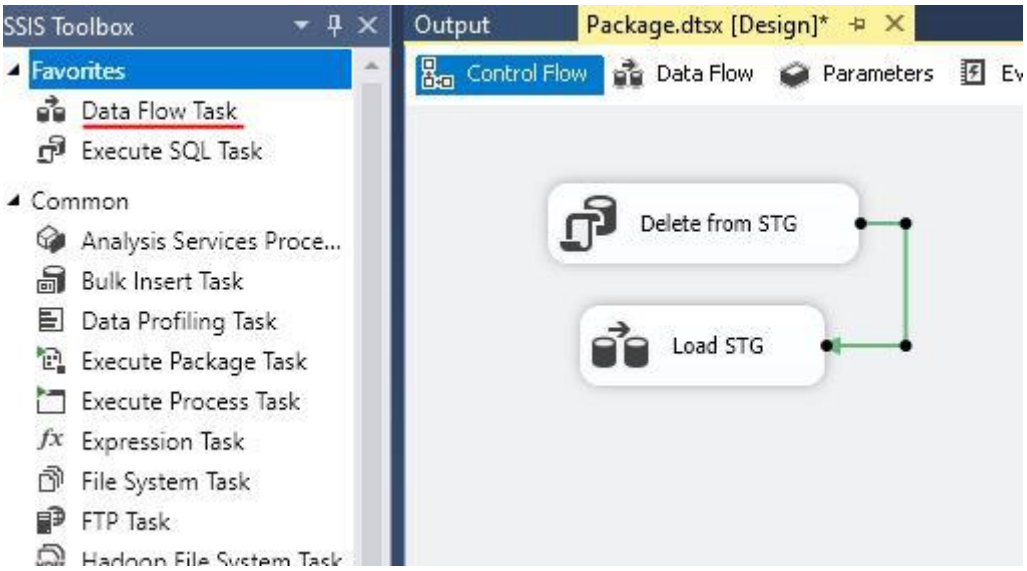


Slika 22 Start gumb



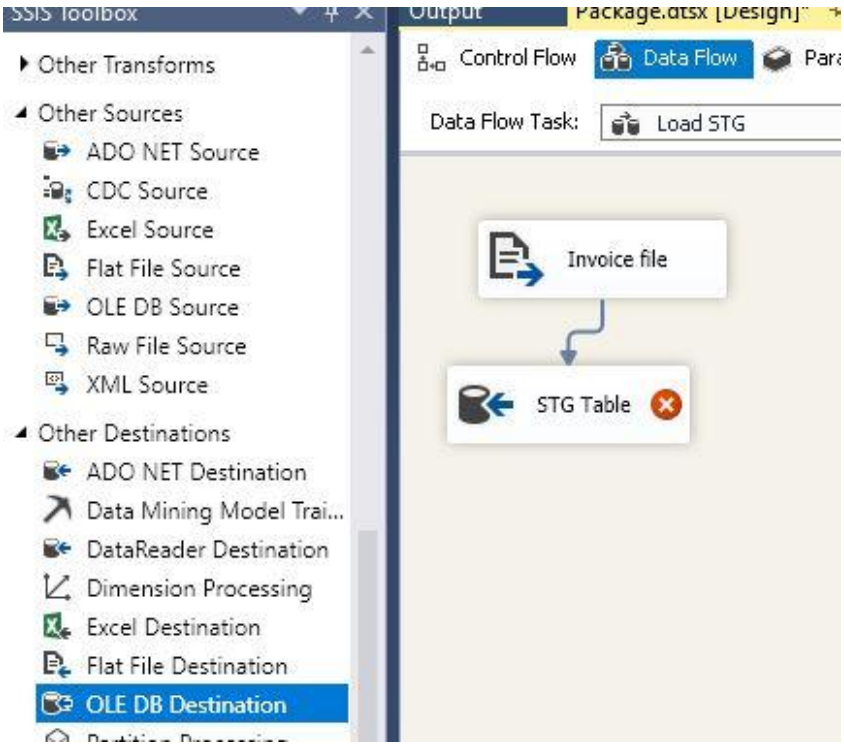
Slika 23 Execution results tab

Sljedeći korak će biti komponenta **Data Flow Task** koji povlačenjem dodajemo u **Control Flow**. Na novi task povezat ćemo prethodnu komponentu povlačenjem strelice te tako definirati da se nakon prve komponente izvršava druga. **Data Flow Task** preimenovali smo u **Load Stg** jer će se u njemu odviti učitavanje podataka u staging tablicu (Slika 24).



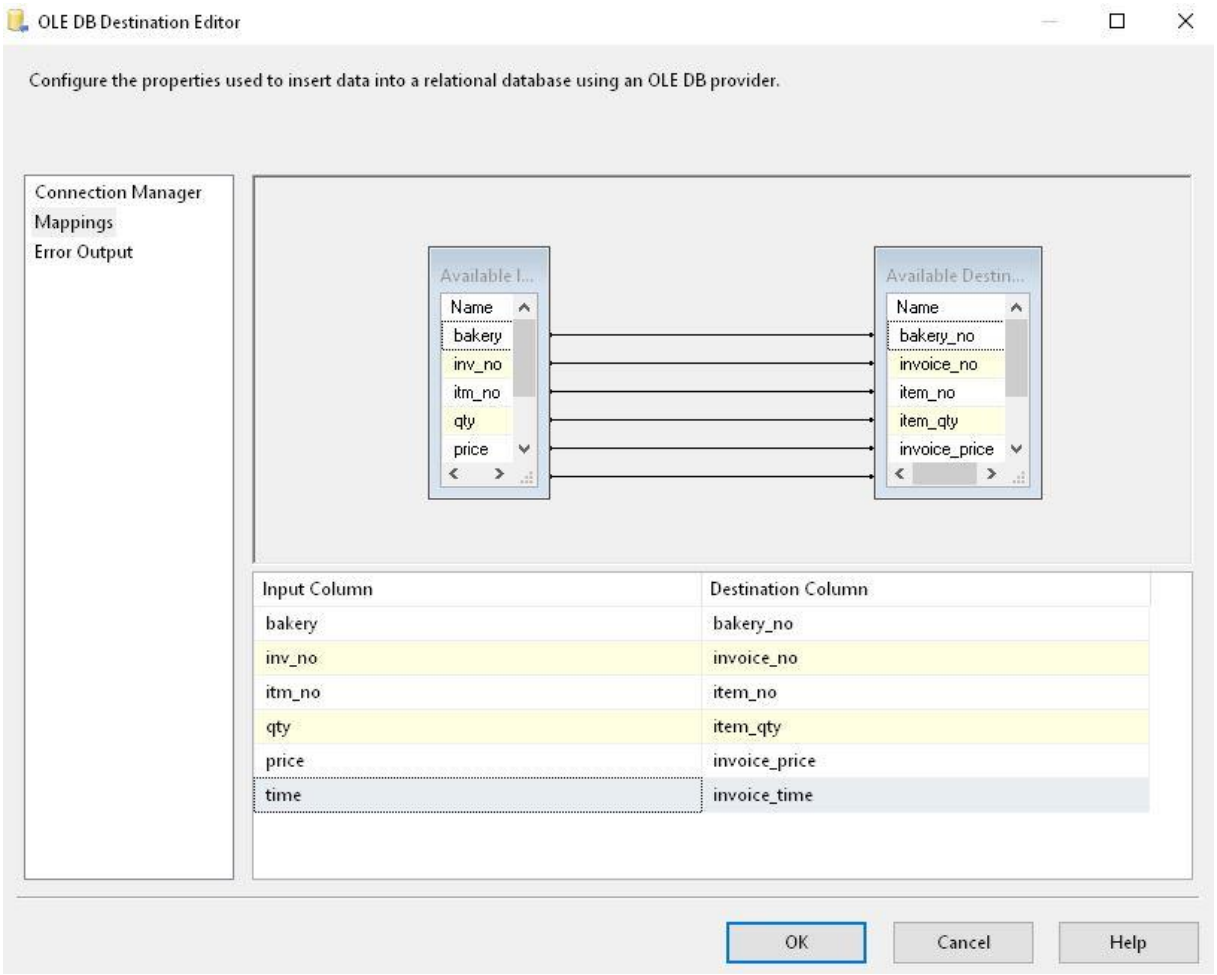
Slika 24 Data Flow Task - Druga komponenta

Lijevim dvoklikom na novi **Data Flow Task** ulazimo u novi okvir naziva **Data flow** (desno od Control Flow), razlog je što **Data Flow Task** ima svoj vlastiti tok i puno novih komponenti koje u njega možemo dodavati. U ovaj task dodat ćemo komponentu **Flat File Source** i povezati ju na komponentu **OLE DB Destination**. S obzirom da smo prethodno definirali konekcije, u konfiguraciji ovih komponenti je potrebno samo odabrati te konekcije, to je jednostavan korak. Rezultat bi trebao biti kao na slici (Slika 25).



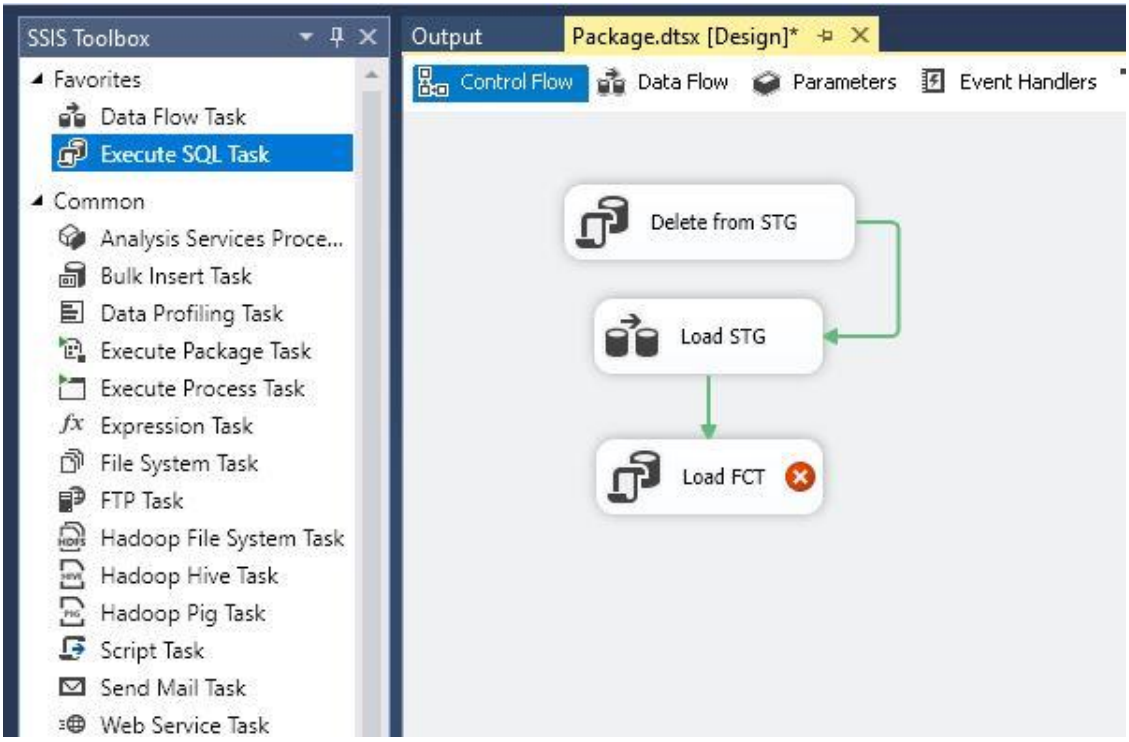
Slika 25 Data Flow - Druga komponenta

U **OLE DB Destination** konfiguraciji bitno je obratiti pažnju na sekciju **Mappings** (Slika 26). Ovdje definiramo mapiranje između izvornih kolumni i destinacije odnosno mapiramo koja kolumna iz izvorne datoteke ulazi u koju kolumnu u staging tablici.



Slika 26 Mapiranje od izvora do destinacije

Ukoliko sada izvršimo paket bez greške, podaci iz datoteke bi trebali biti zapisani u staging tablicu u **bakerymart** bazi. Sljedeći korak je punjenje podataka iz staging tablice u finalnu **Fct_Invocies** tablicu. Ovdje ćemo koristiti tradicionalni pristup u izvršiti to putem **Execut SQL Task** komponente, potrebno je dodati i povezati novu komponentu (Slika 26).



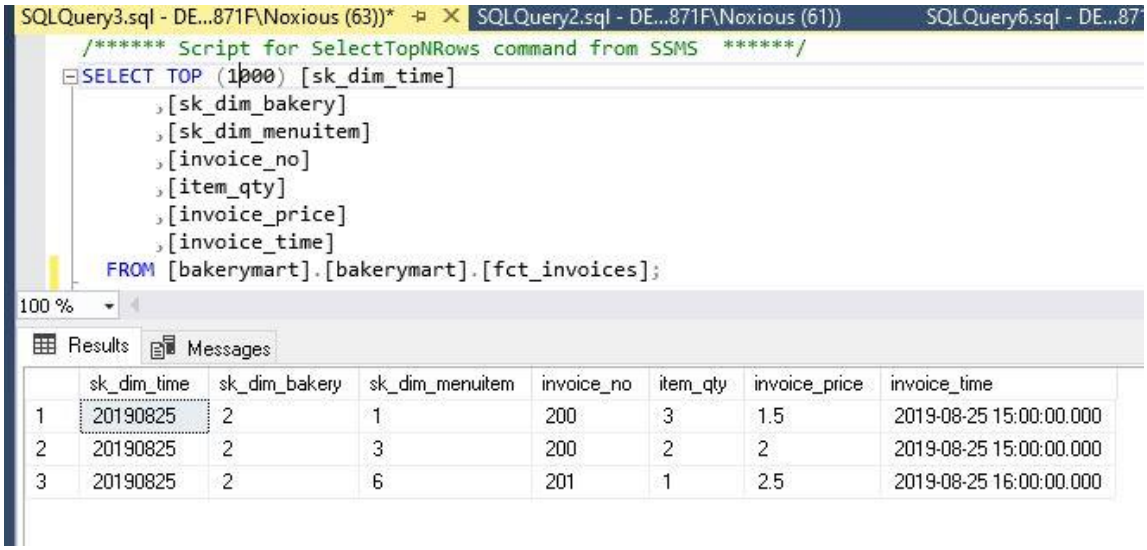
Slika 27 Exec SQL Task - Treća komponenta

U konfiguraciji komponente dodajemo upit prikazan na slici (Slika 28). Ovo je takozvana **INSERT INTO SELECT** klauzula s kojom možemo raditi **INSERT** u tablicu tako da uzimamo direktno rezultat nekog SELECT-a, u SELECT-u radimo **JOIN** staging tablice sa tablicama dimenzija kako bismo dobili ključeve iz dimenzja. **Execute SQL Task** komponenta može izvršavati razne kompleksne upite koji mogu, a i ne moraju sadržavati rezultate.

```
INSERT INTO bakerymart.fct_invoices (sk_dim_time, sk_dim_bakery, sk_dim_menuitem, invoice_no, item_qty, invoice_price, invoice_time)
  SELECT sk_dim_time, sk_dim_bakery, sk_dim_menuitem, invoice_no, item_qty, invoice_price, invoice_time
  FROM bakerymart.stg_fct_invoices
  join bakerymart.dim_time on CAST(stg_fct_invoices.invoice_time AS DATE) = dim_time.date
  join bakerymart.dim_bakery on stg_fct_invoices.bakery_no = dim_bakery.bakery_no
  join bakerymart.dim_menuitem on stg_fct_invoices.item_no = dim_menuitem.menitem_no
  ;
```

Slika 28 Exec SQL Task - kompleksni upit

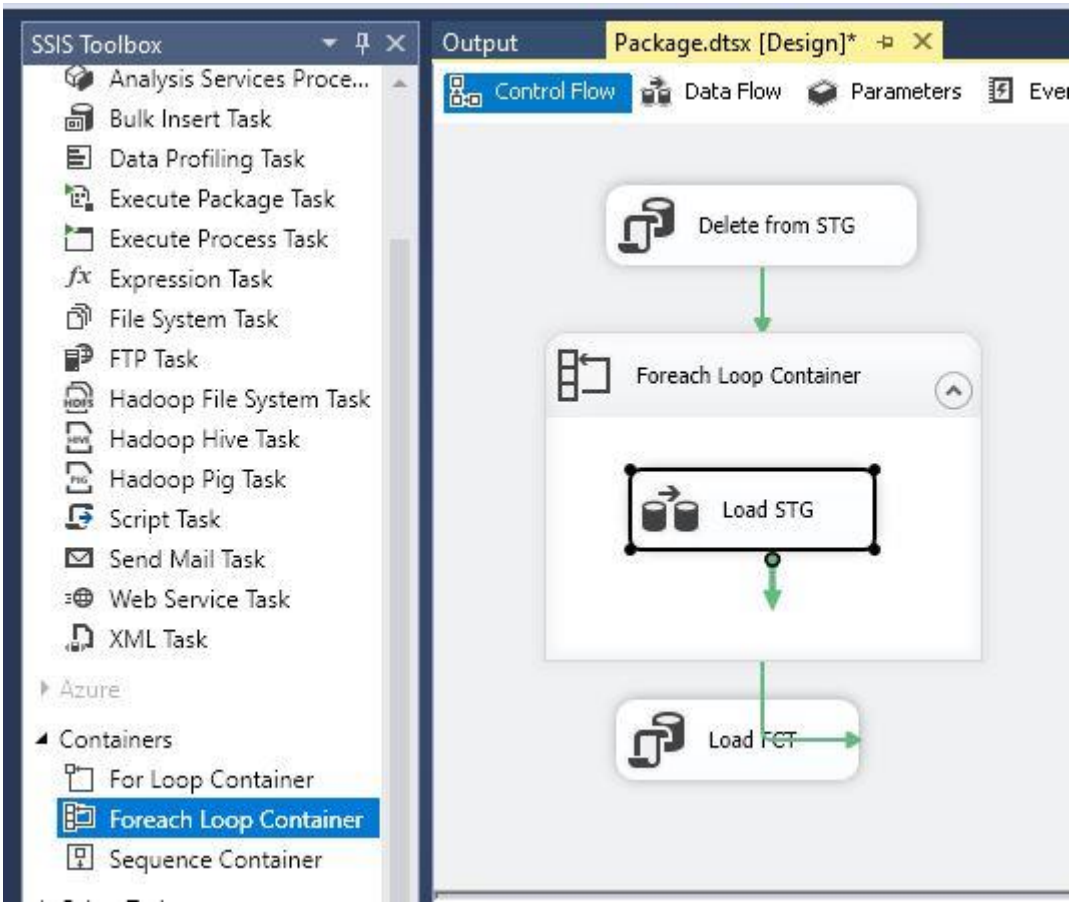
Izvršavanjem paketa trebali bismo imati uspješan upis u tablicu **Fct_Invoices** (Slika 29).



	sk_dim_time	sk_dim_bakery	sk_dim_menuitem	invoice_no	item_qty	invoice_price	invoice_time
1	20190825	2	1	200	3	1.5	2019-08-25 15:00:00.000
2	20190825	2	3	200	2	2	2019-08-25 15:00:00.000
3	20190825	2	6	201	1	2.5	2019-08-25 16:00:00.000

Slika 29 Podaci u tablici Fct_Invoices

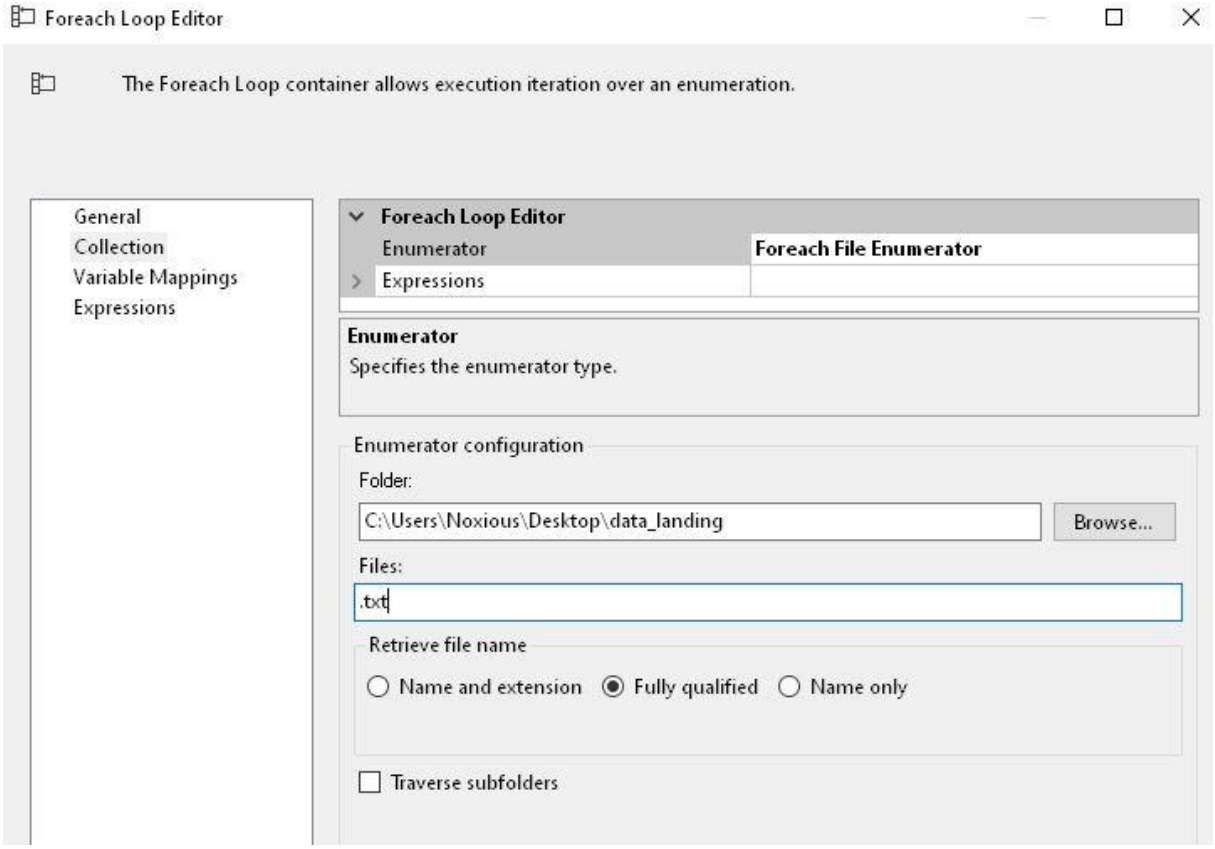
Iako je ovo funkcionalno rješenje, potrebno je učitati više datoteka koje dinamički dolaze u sustav. Prvi korak je dodavanje **Foreach Loop Container** komponente i umetanje **Data Flow** komponente unutra (Slika 30).



Slika 30 Foreach Loop Container

U **Foreach Loop Container** konfiguraciji (Slika 31) kao **Enumerator** postaviti ćemo **Foreach File Enumerator**. U **Folder** ćemo dodati putanju do direktorija, te u **Files** izraz

"*.txt". Takva konfiguracija znači da će petlja prolaziti kroz sve datoteke u direktoriju data_landing koje završavaju ekstenzijom ".txt". Osim toga potrebno je i u **Data Flow Task** komponenti u **Flat File Source** promijeniti **ConnectionString** da dinamički uzima putanju do datoteke u svakoj iteraciji umjesto fiksne vrijednosti koju smo ranije zadali.



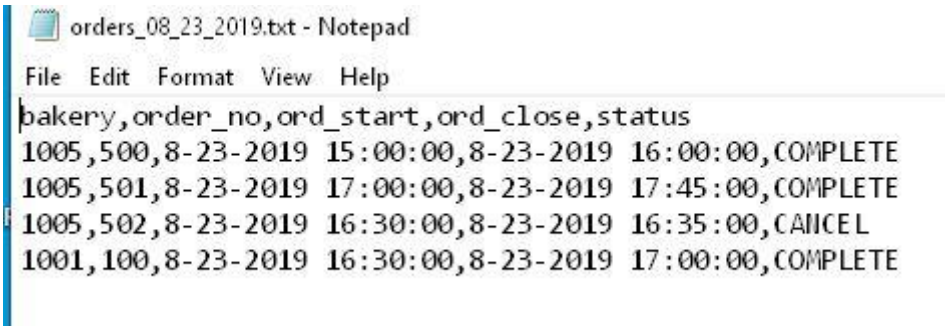
Slika 31 Foreach Loop – konfiguracija

Izvršavanje ovako napravljenog paketa rezultira u tome da se dinamički učitava svaka datoteka u direktoriju i njeni podaci.

3.6.2 Fct_Orders (data conversion, derived column, lookup)

Iako prethodni primjer elegantno pokazuje osnovnu funkciju SSISa-a, ima bitan dio logike napisan ručno kao SQL, a to nije baš u duhu gotovih alata. S obzirom da SSIS ima **Lookup** komponentu koja je u biti **JOIN**, ovaj primjer biti će izveden bez ručnog pisanja SQL-a i bez staging tablice.

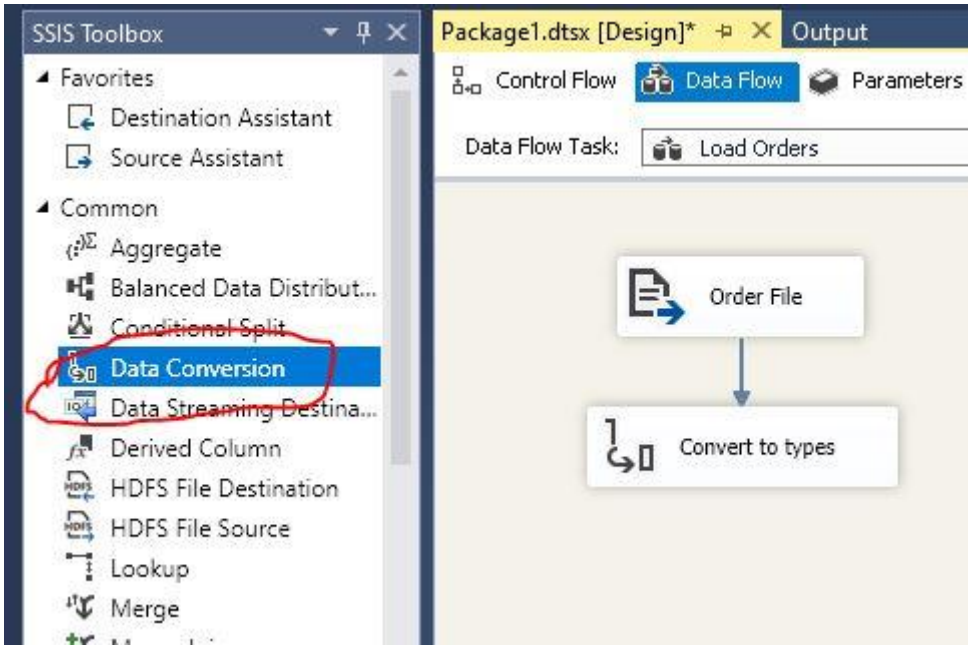
Zamislimo da imamo slučaj u kojem nam aplikativni server za narudžbe dnevno dostavlja listu svih kompletiranih narudžbi za taj datum u neki direktorij na server u prikazanom formatu (Slika 32).



Slika 32 Uzorak Orders podataka

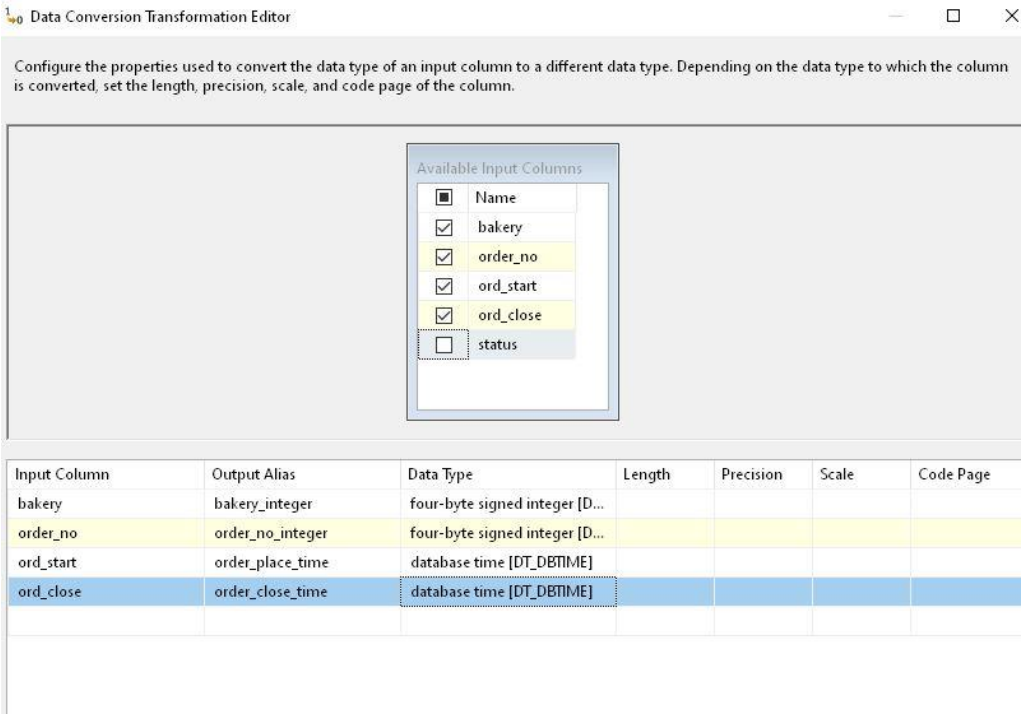
Ovu vježbu započet ćemo slično kao i prethodnu, potrebno je definirati konekcije i zatim jedan **Data Flow Task**. Unutar Data Flow-a dodati ćemo **Flat File Source**, ali ovog puta nećemo podešavati nikakvu konverziju kao u prethodnom primjeru, na taj način komponenta će učitati podatke kao **STRING**.

Nakon toga potrebno je dovući i spojiti **Data Conversion** komponentu, kada je to obavljeno projekt bi trebao izgledati kao na slici (Slika 33).



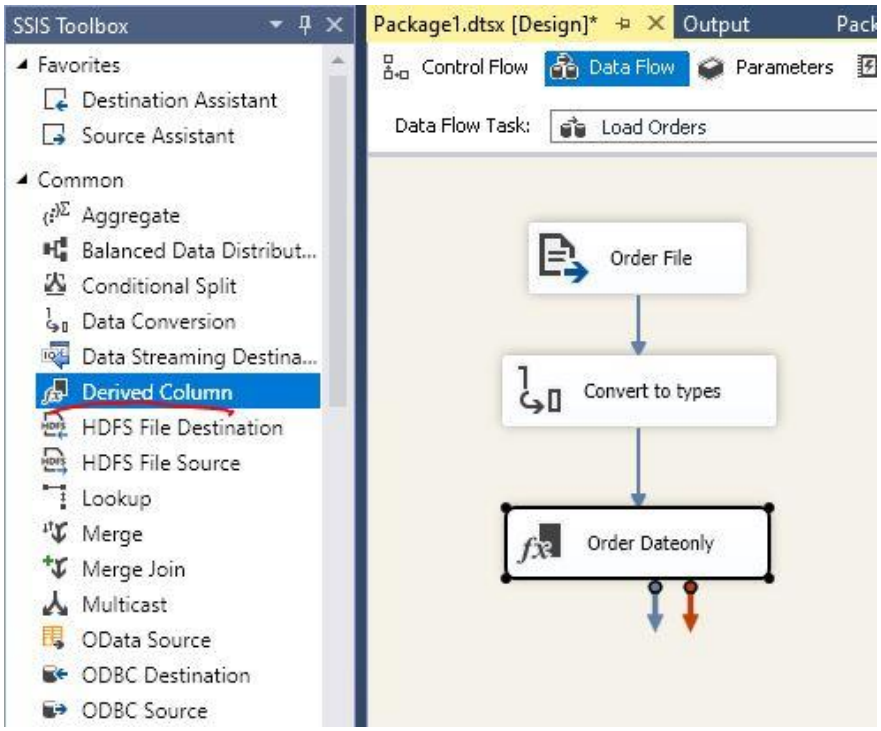
Slika 33 Data Conversion komponenta

Lijevim dvoklikom ulazimo u konfiguraciju **Data Conversion** komponente (Slika 34), tu možemo definirati u koje tipove želimo konvertirati ulazne podatke. Konvertiramo sve da odgovara tipovima u finalnoj **Fct_Orders** tablici. Kolumnu "Status" nije potrebno konvertirati jer već je tekstualan tip.

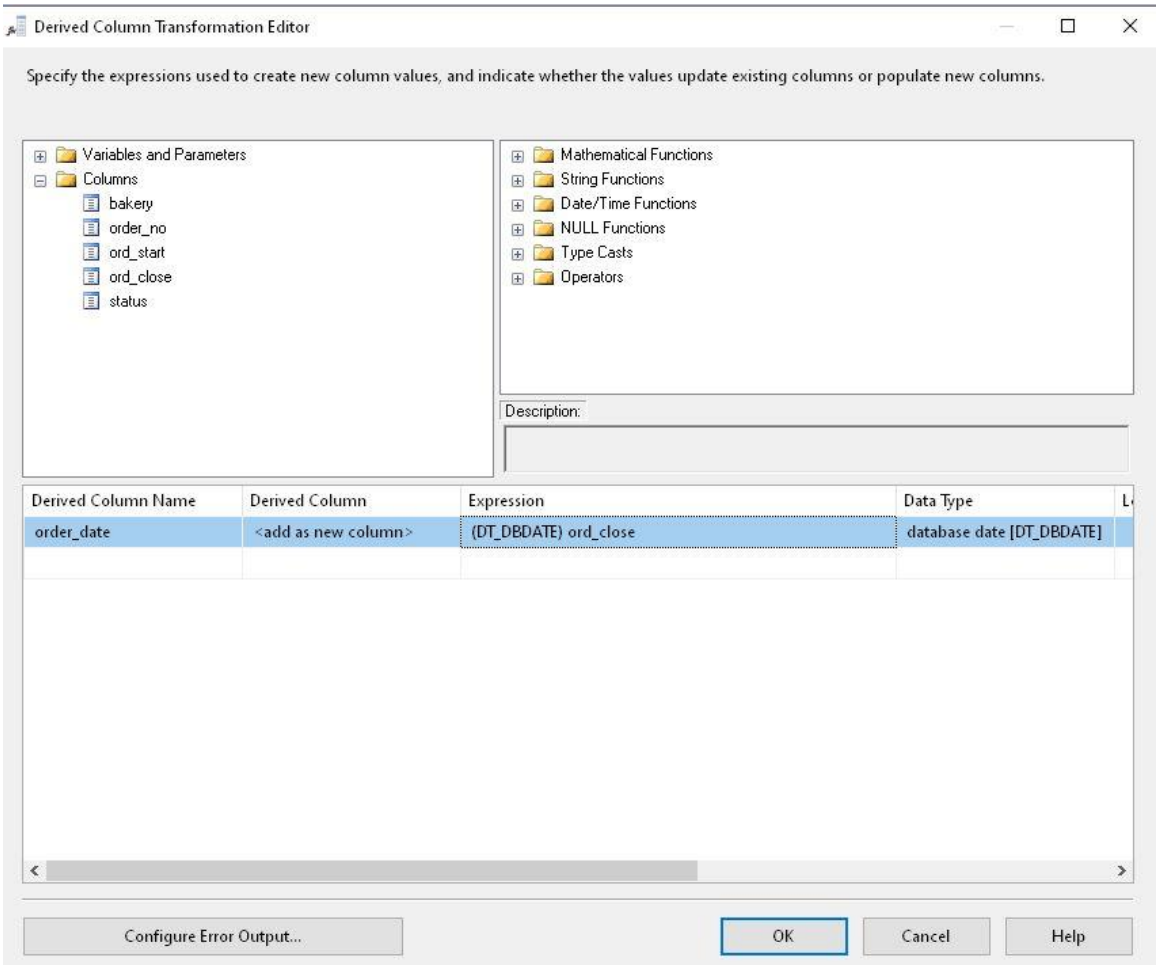


Slika 34 Data Conversion konfiguracija

Sljedeća komponenta koju dodajemo je **Derived Column** (Slika 35), nju dodajemo kako bismo iz vremena kada je narudžba kompletna (ord_close) dobili samo vrijednost datuma. U konfiguraciji (Slika 36) dodajemo novu kolumnu order_date koja je u biti samo ord_close kolumna konvertirana u datum, na taj način stvaramo novi podatak koji ćemo kasnije koristiti za lookup na dimenziju vremena.

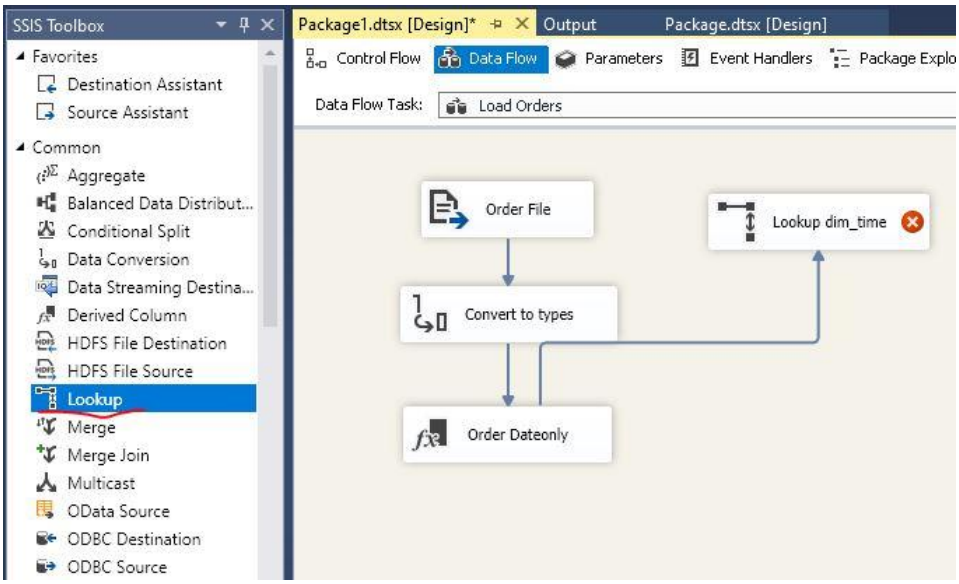


Slika 35 Derived Column komponenta



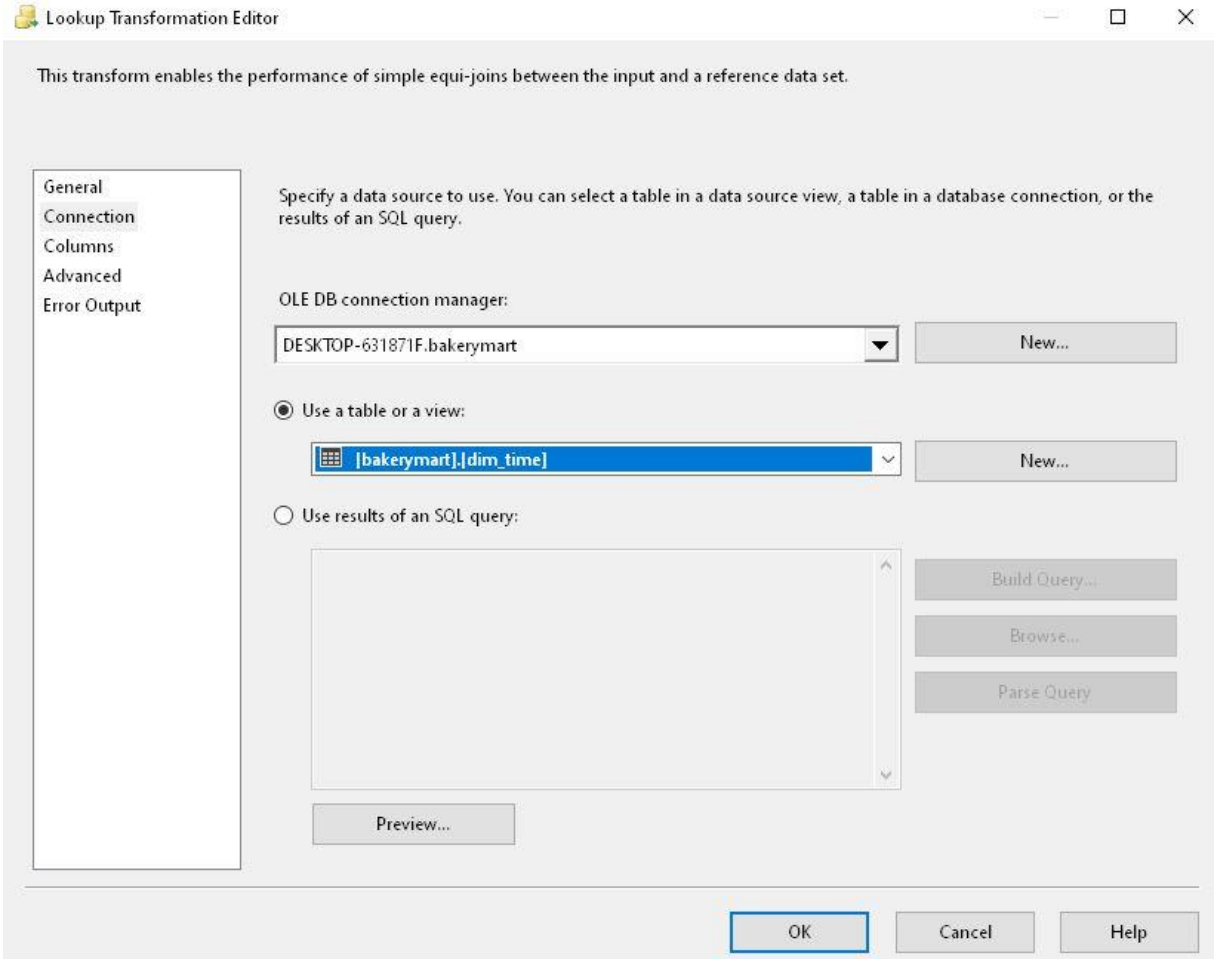
Slika 36 Derived Column konfiguracija

Slijedi dodavanje **Lookup** komponente u **Data Flow** (Slika 37).

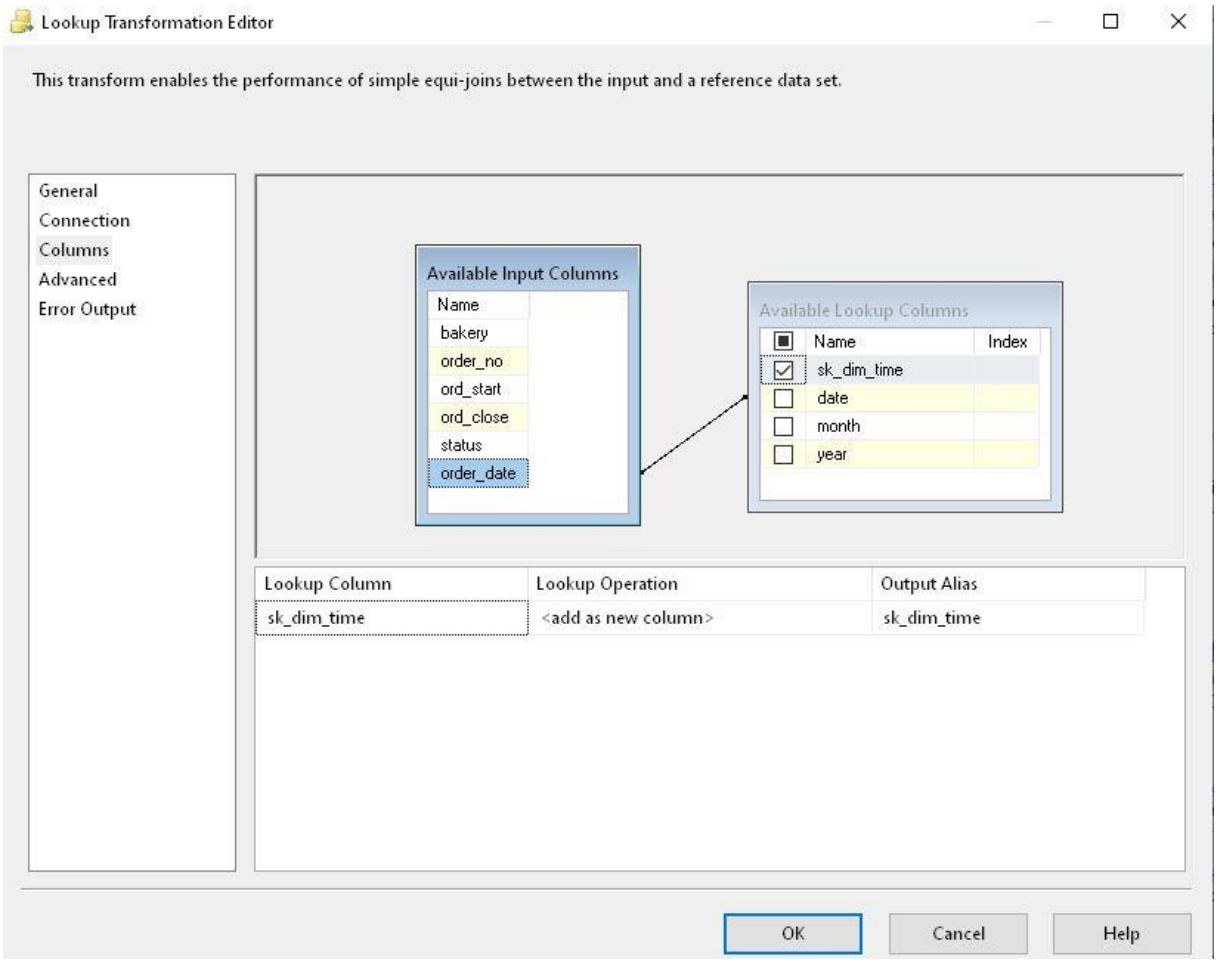


Slika 37 Lookup komponenta

Nakon dodavanja **Lookup** komponente potrebno je odrediti konfiguraciju. U **Connection** opcijama (Slika 38) odabrat ćemo konekciju na **bakerymart** i dimenziju **dim_time**. Nakon toga u **Columns** opcijama (Slika 39) označit ćemo u lijevoj tablici kolumnu **sk_dim_time**, to je kolumna čiju vrijednost želimo dohvatiti za daljnju obradu. Povlačenjem od lijeve do desne tablice mapirati ćemo order_date kolumnu sa date kolumnom u dimenziji vremena (što u biti znači da na toj kolumni radimo **JOIN**). Ovom akcijom na naše postojeće učitane kolumne dodat ćemo i kolumnu iz dimenzije vremena (sk_dim_time) tamo gdje je pronađena ista vrijednost datuma (match).

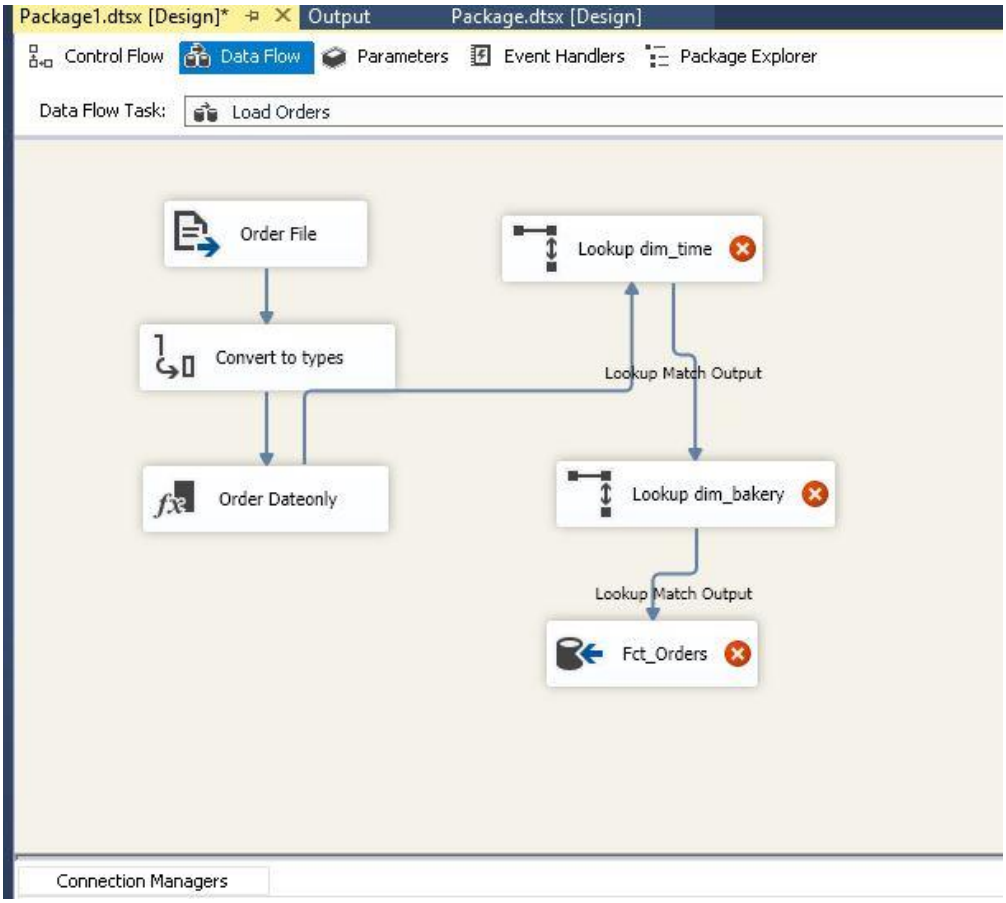


Slika 38 Lookup konfiguracija 1



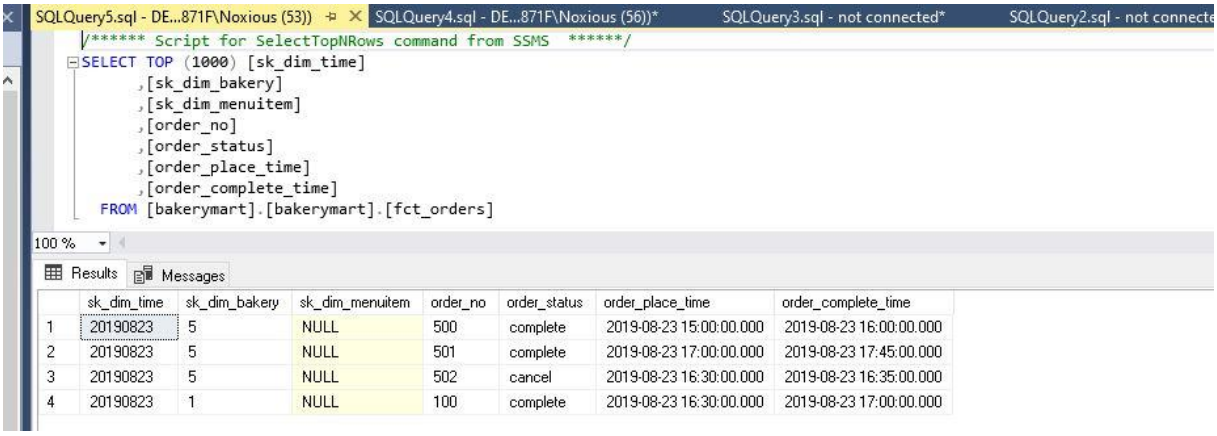
Slika 39 Lookup konfiguracija 2

Koristeći istu logiku, na ovaj **Lookup** možemo dodati još jednu **Lookup** komponentu za dimenziju **Dim_Bakery**. Nakon toga set podataka bi trebao biti potpun, sve je spremljeno u memoriji i nema staginga. To znači da na kraj dodajemo **OLE DB Destination** komponentu koja je povezana na finalnu tablicu **Fct_Orders**, paket je prikazan na slici (Slika 40).



Slika 40 Data Flow dovršeni

Izvršimo paket i pošaljimo upit na tablicu **Fct_Orders**, podaci bi trebali biti učitani (Slika 41). Može se primijetiti da je vrijednost sk_dim_menuitem kolumne NULL, to je jednostavno zbog promjene odluka u dizajnu i iako smo očekivali da će narudžbe imati u sebi proizvode, zapravo uopće nisu poslani podaci o proizvodima nego je samo bitno o kojoj se pekari radi i koji je vremenski okvir u kojemu je narudžba kompletirana.



	sk_dim_time	sk_dim_bakery	sk_dim_menuitem	order_no	order_status	order_place_time	order_complete_time
1	20190823	5	NULL	500	complete	2019-08-23 15:00:00.000	2019-08-23 16:00:00.000
2	20190823	5	NULL	501	complete	2019-08-23 17:00:00.000	2019-08-23 17:45:00.000
3	20190823	5	NULL	502	cancel	2019-08-23 16:30:00.000	2019-08-23 16:35:00.000
4	20190823	1	NULL	100	complete	2019-08-23 16:30:00.000	2019-08-23 17:00:00.000

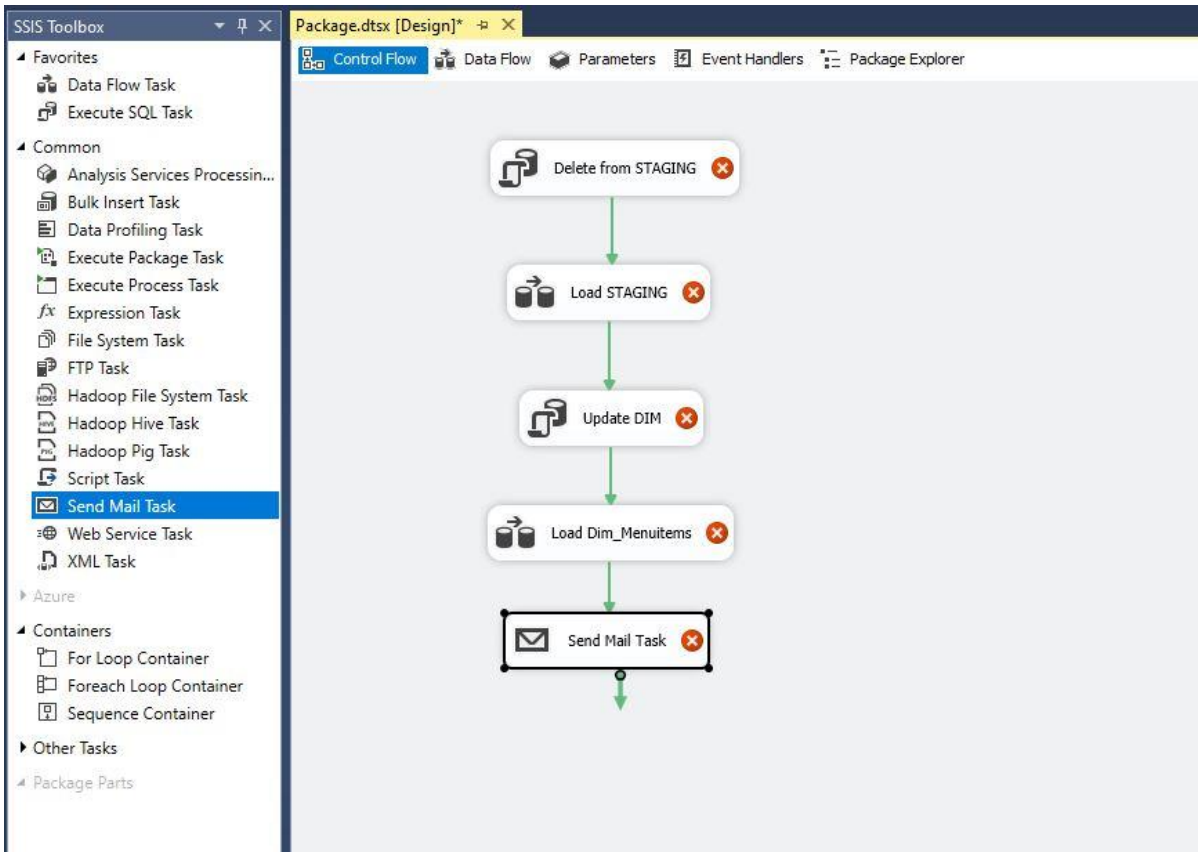
Slika 41 Podaci u Fct_Orders

3.7 Scenarij 2: Punjenje dimenzija

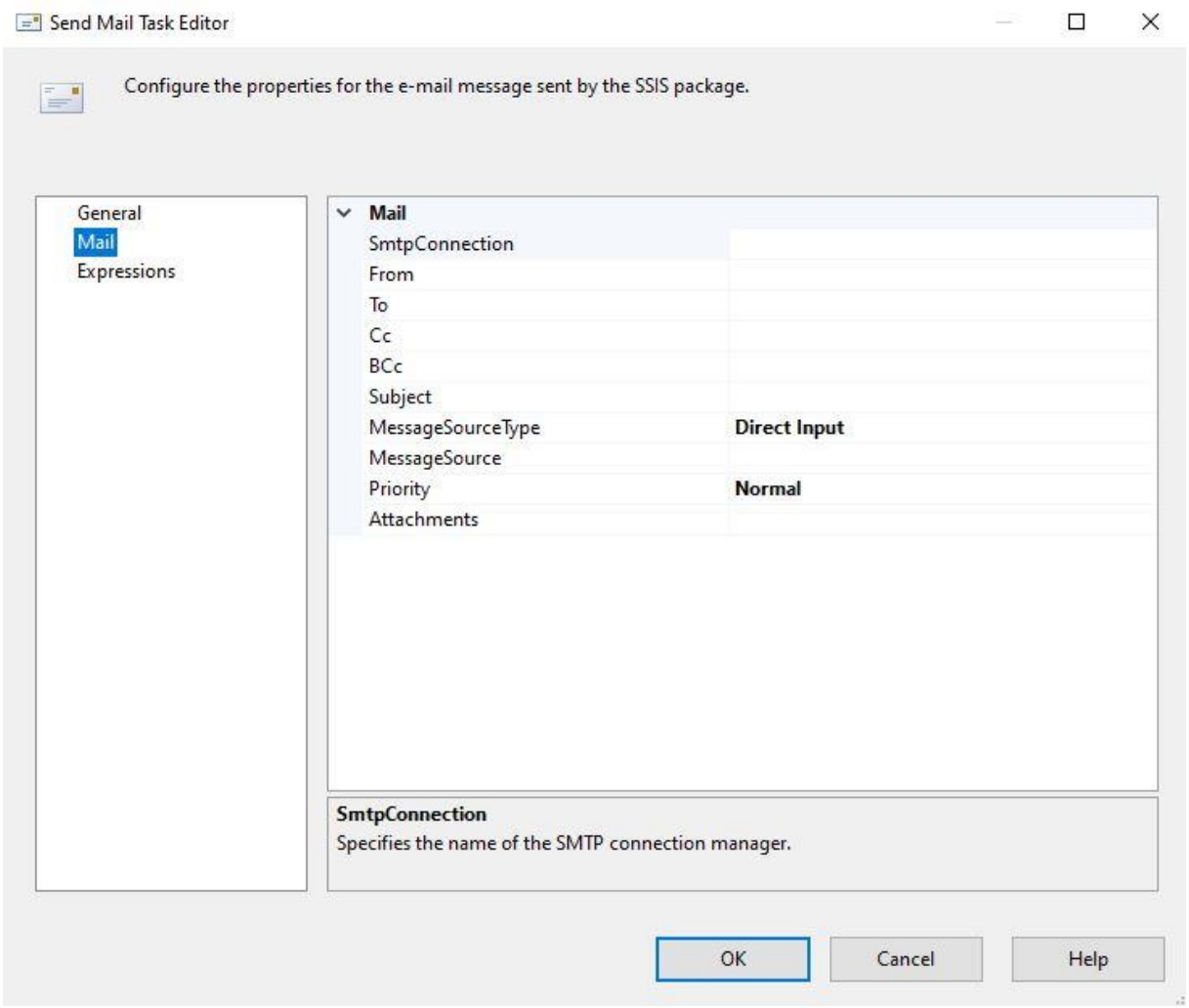
3.7.1 Dim_Menuitems (union all, lookup grananje, send email task)

S obzirom da je Dim_Bakery tip 1 dimenzija, učitavanje podataka u nju je trivijalno. Međutim, tip 2 može biti kompleksan pa ćemo proći kroz konceptualni primjer koji ilustrira kako bi se kroz SSIS moglo izvesti punjenje tip 2 dimenzija za slučaj Dim_Menuitems.

Control flow dijagram (Slika 42) sadrži dvije **Execute SQL Task** komponente, dvije **Data Flow Task** komponente i jedan **Send E-mail Task**. Send Email Task komponenta (Slika 43) ima jednostavnu konfiguraciju, sve je poput standardnog slanja elektronske pošte osim što je potrebno specificirati **SMTP** server. Ova komponenta može se koristiti za notifikacije osobama zaduženima za nadzor procesa, ukoliko su procesi automatizirani da nakon izvršavanja u predviđeno vrijeme pošalju status izvršenja.



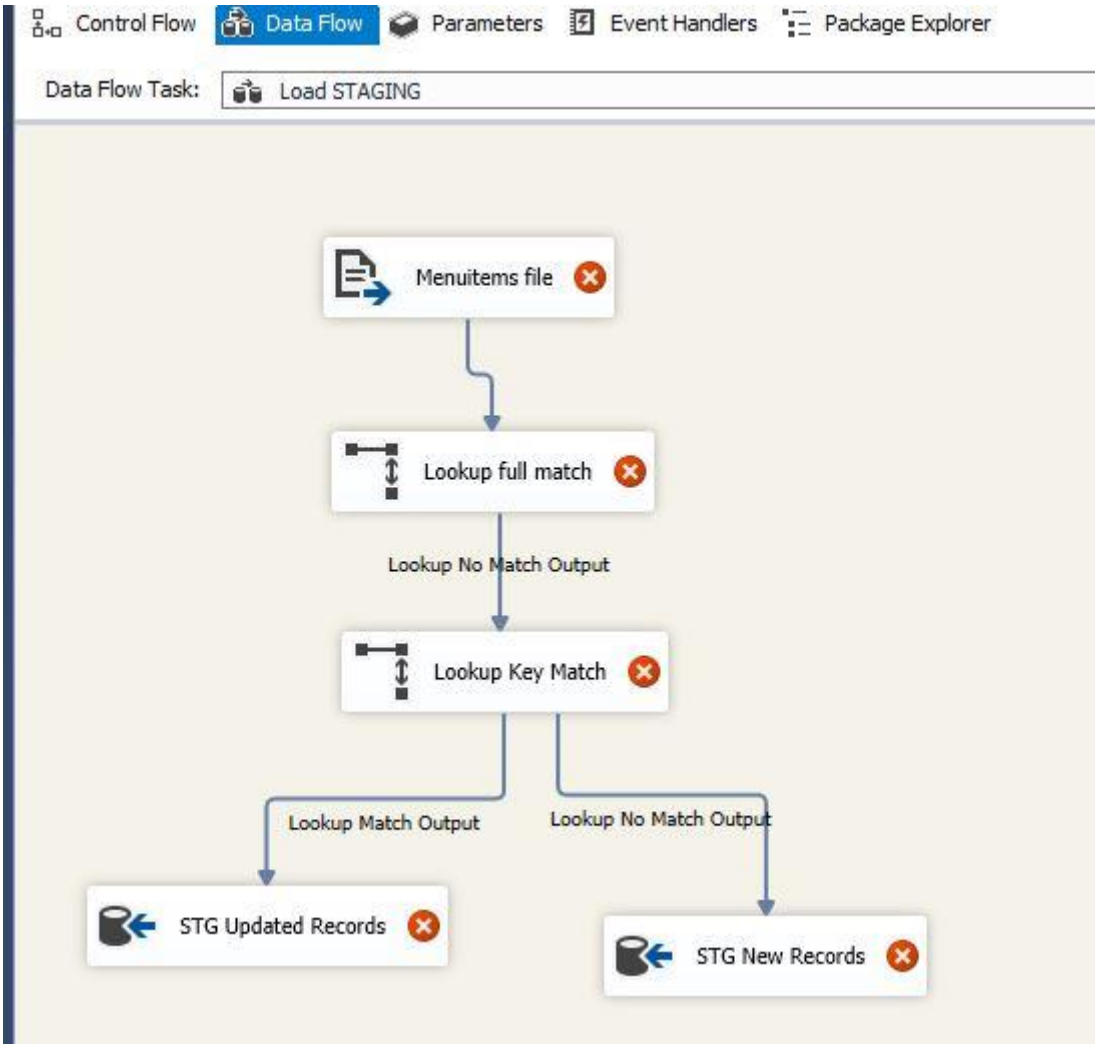
Slika 42 Control Flow - tip 2 dimenzija



Slika 43 Send E-mail Task

Za ovaj slučaj ćemo prvo pripremiti dvije staging tablice, jednu za nove zapise u dimenziji i jednu za promijene u postojećim zapisima u dimenziji. Prvi korak paketa je napraviti **DELETE FROM** iz tih staging tablica (prva Exec SQL komponenta). Druga komponenta (**Load STAGING**) vrši ekstrakciju podataka iz datoteke, lookup i punjenje u staging tablice (Slika 44).

Zamislimo da nam netko na kraju tjedna šalje listu proizvoda koja bilježi novo stanje proizvoda za sljedeći radni tjedan (ukoliko je došlo do novih proizvoda, promjene cijena proizvoda). Prvi korak nakon ekstrakcije podataka je **Lookup** na postojeću tablicu dimenzija. Lookup izvodimo na način da odaberemo sve kolumne izvora, tako ćemo pronaći sve zapise koji u potpunosti odgovaraju onome što već imamo u bazi. Umjesto da sačuvamo zapise koji su **Match**, zapravo ćemo uzeti sve one koji su **No Match** i poslati ih sljedećem Lookup koraku. U sljedećem Lookup koraku (**Lookup Key Match**) napravit ćemo match po kolumni koja je primarni ključ izvornih podataka (jedinstveno identificira proizvod), te ćemo granati rezultat. Ukoliko **match** postoji, to znači da imamo promjene u podacima za već postojeće proizvode i njih ćemo upisati u staging za promijenjene zapise (**STG Updated Records**). Zapise koji nemaju **match**, tretiramo kao posve nove proizvode u bazi i njih ćemo poslati u **STG New Records**.



Slika 44 Data Flow - Load STAGING - tip 2 dimenzija

Sada su podaci spremljeni u staging, prije nego ih pošaljemo u finalnu Dim_Menuitems tablicu, potrebno je označiti da zapisi koji dobivaju svoju novu verziju nisu više važeći, to se može postići dodavanjem upita (Slika 45) u **Execute SQL Task (Update DIM)** komponentu.

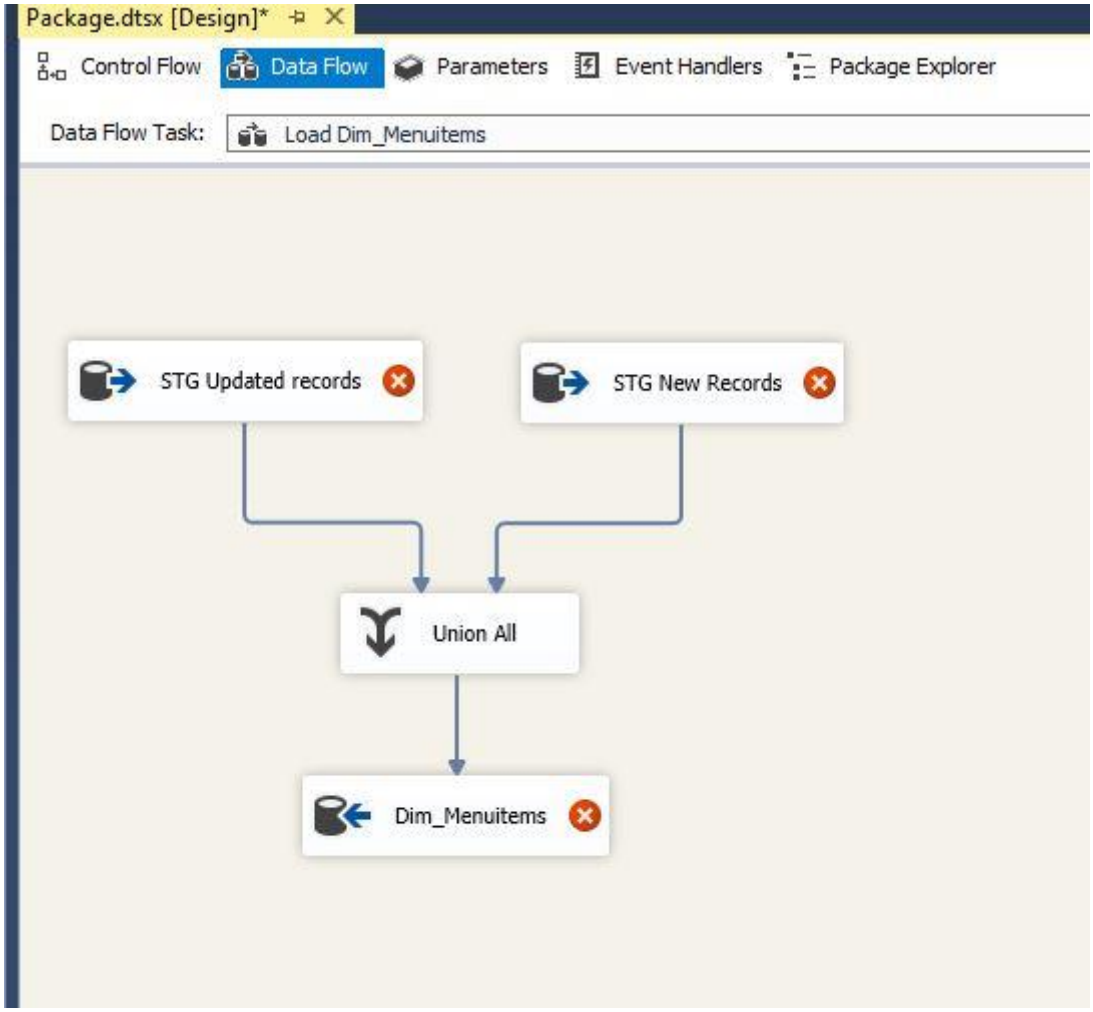
```
UPDATE bakerymart.dim_menuitems
SET
dim menuitems.valid_to = stg_updated_menuitems.valid_from
dim_menuitems.active_ind = 0
FROM bakerymart.dim_menuitems INNER JOIN bakerymart.stg_updated_menuitems ON
dim_menuitems.menuitem_no = stg_updated_menuitems.menuitem_no
```

Slika 45 UPDATE query

Ovakav upit radi **UPDATE** operaciju na postojećim zapisima u bazi tako da radi **JOIN** između dim_meniutems i stg_updated_menuitems. Konačni rezultat je da će za svaki zapis koji ima novu verziju, postojeći zapis dobiti indikator aktivnosti 0 i valid_to na valid from vrijeme novog zapisa. To je način na koji stare zapise činimo nevažećima.

Zadnji korak je napraviti punjenje finalne tablice iz staging tablica. U **Load Dim_Menuitems** komponenti (Slika 46) koristimo kao izvor staging tablice i dodajemo

komponentu **Union All**, koja u biti funkcionira poput unije u SQL-u. Spajamo set podataka iz dva izvora u jedan i naposlijetku sve upisujemo u Dim_Menuitems tablicu.

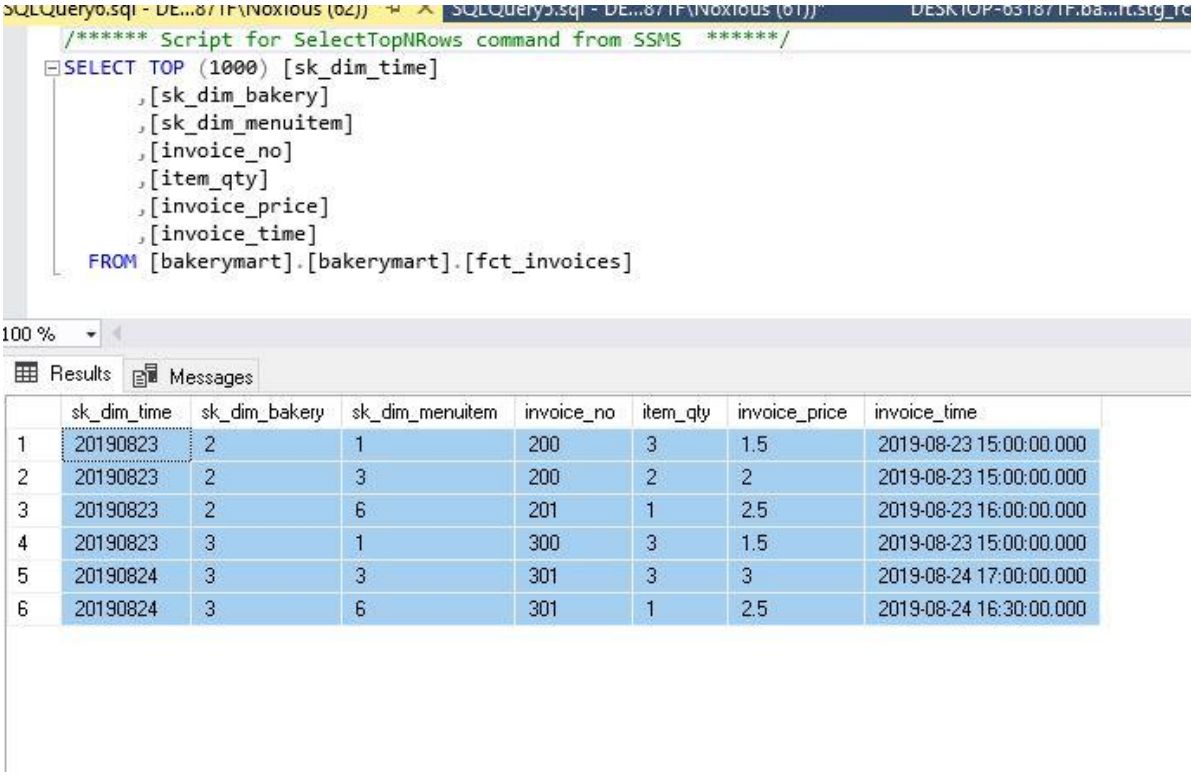


Slika 46 Data Flow - Load Dim - tip 2 dimenzija

3.8 Scenarij 3: Posebni slučajevi (aggregate, sort)

Zamislimo da smo dobili novi zahtjev od tima koji se bavi vizualizacijom podataka. Razvijena je „lightweight“ mobilna aplikacija za menadžere i zbog performansa je potrebno izraditi specijalne tablice koje su zapravo agregirane činjenice kako bi se operacije izračuna obavljale na našem **bakerymart** serveru, a ne u memoriji korisničke aplikacije.

Izvorni podaci za obradu nalaze se u **Fct_Invoices** tablici koju smo prethodno napunili (Slika 47). Sada je potrebno kreirati nove posebne tablice koje ćemo imenovati sa prefiksom **rpt_**, DDL se nalazi na slici (Slika 47).



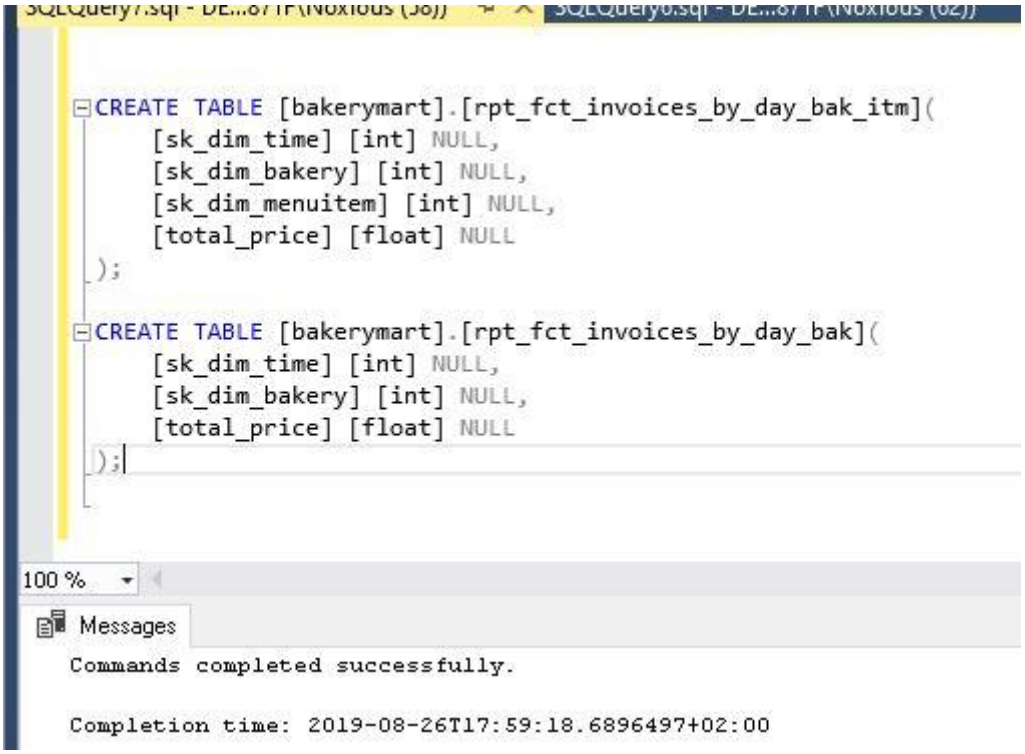
```
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP (1000) [sk_dim_time]
      ,[sk_dim_bakery]
      ,[sk_dim_menuitem]
      ,[invoice_no]
      ,[item_qty]
      ,[invoice_price]
      ,[invoice_time]
  FROM [bakerymart].[bakerymart].[fct_invoices]
```

100 %

Results | Messages

	sk_dim_time	sk_dim_bakery	sk_dim_menuitem	invoice_no	item_qty	invoice_price	invoice_time
1	20190823	2	1	200	3	1.5	2019-08-23 15:00:00.000
2	20190823	2	3	200	2	2	2019-08-23 15:00:00.000
3	20190823	2	6	201	1	2.5	2019-08-23 16:00:00.000
4	20190823	3	1	300	3	1.5	2019-08-23 15:00:00.000
5	20190824	3	3	301	3	3	2019-08-24 17:00:00.000
6	20190824	3	6	301	1	2.5	2019-08-24 16:30:00.000

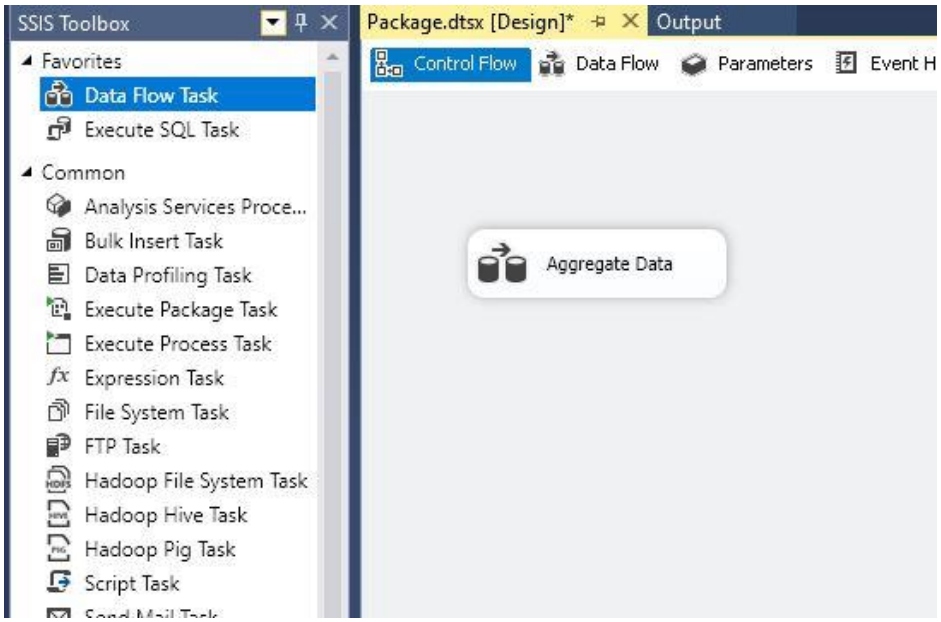
Slika 47 Fct_Invoices - podaci



Slika 48 Rpt DDL

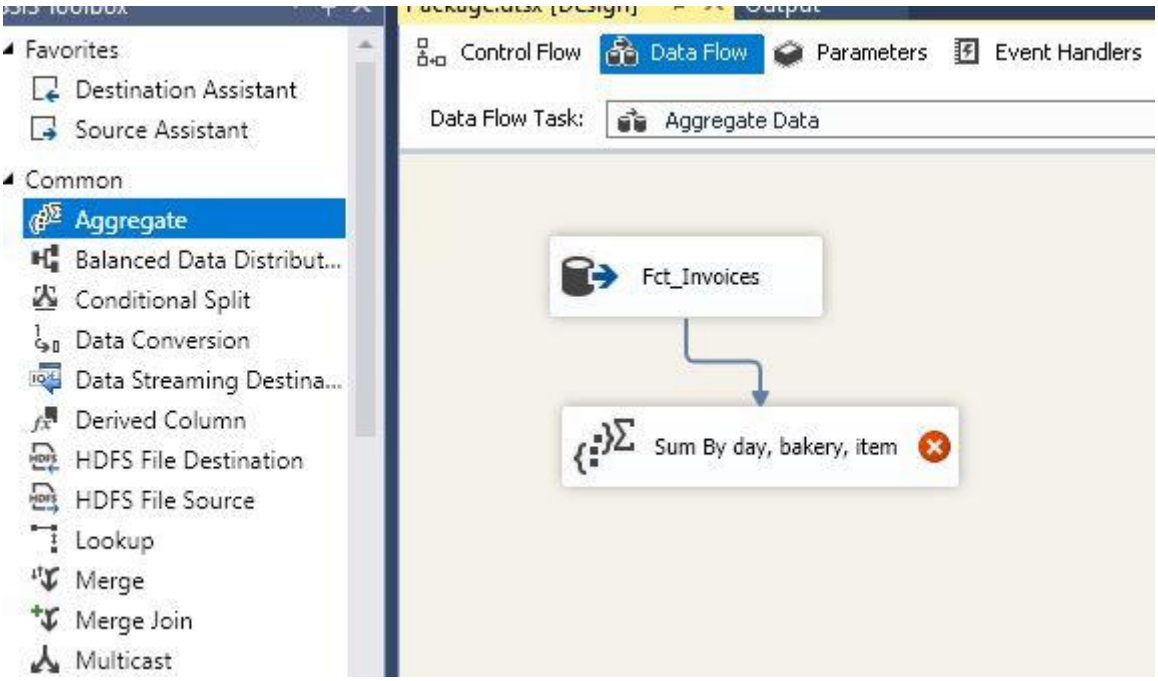
Imamo dvije nove tablice, **rpt_fct_invoices_by_day_bak_itm** je prva tablica, ona će prikazivati sumu prodaje grupiranu po danu za svaki proizvod u pekari (daje uvid u trend prodaje proizvoda u pojedinoj poslovnici). U drugoj tablici **rpt_fct_invoices_by_day_bak** grupiramo prodaju po danu i pekari, odnosno ona daje uvid u ukupan prihod koji je generirala poslovnica u jednom danu.

Prvi korak je postavljanje konekcija, u ovom slučaju trebat će nam samo konekcija na **bakerymart** bazu u **Connection Manager**-u. Potom dodajemo **Data Flow Task** nazvan Aggregate Data (Slika 49).



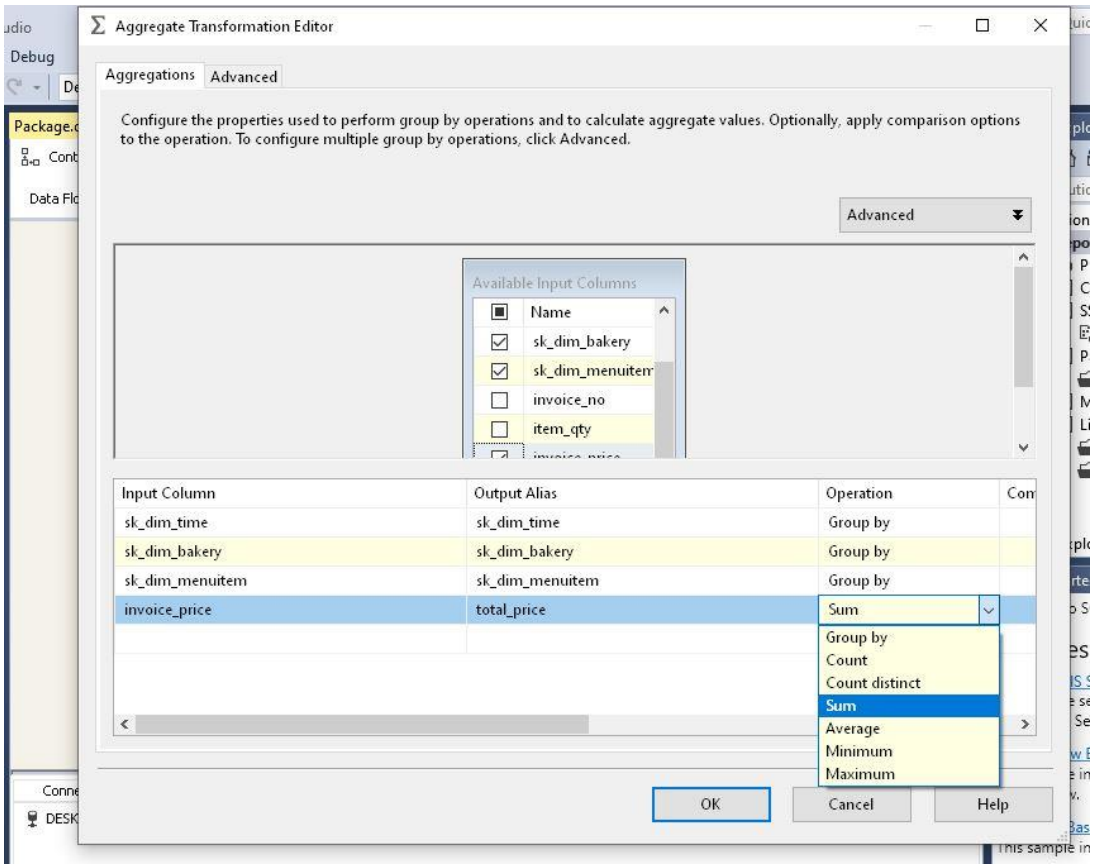
Slika 49 Data Flow Task - Aggregate Data

Unutar **Data Flow** komponente dodajemo **OLE DB Source**, odnosno tablicu Fct_Invoices. Iz SSIS Toolbox-a povlačimo i komponentu **Aggregate** (Sum by day, bakery, item) i na nju povezujemo izvor podataka (Slika 50).



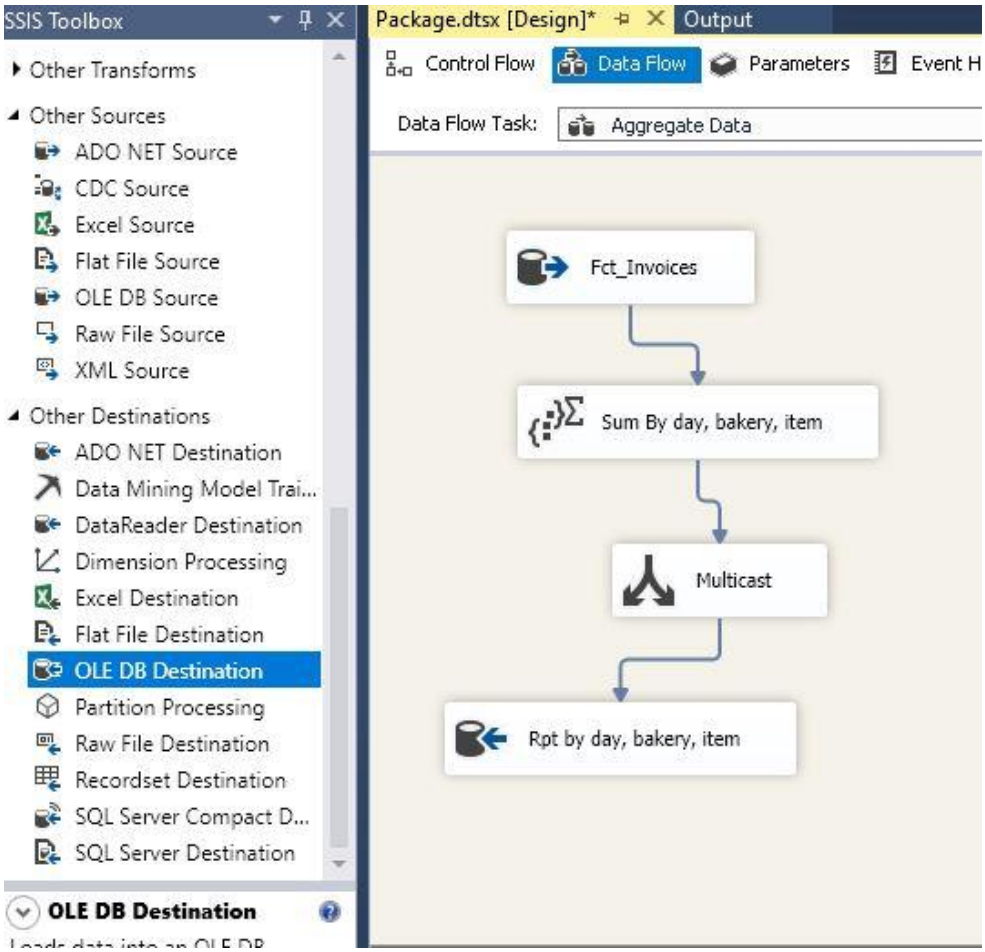
Slika 50 Data Flow - prvi Aggregate

U konfiguraciji **Aggregate** komponente (Slika 51) odabrat ćemo ključeve po kojima želimo grupirati (agregirati podatke) i u ovom slučaju za kolumnu cijene odabrat ćemo sumu. Rezultat je suma cijene grupirane po tri odabrana ključa. To su podaci za prvu **rpt_** tablicu.



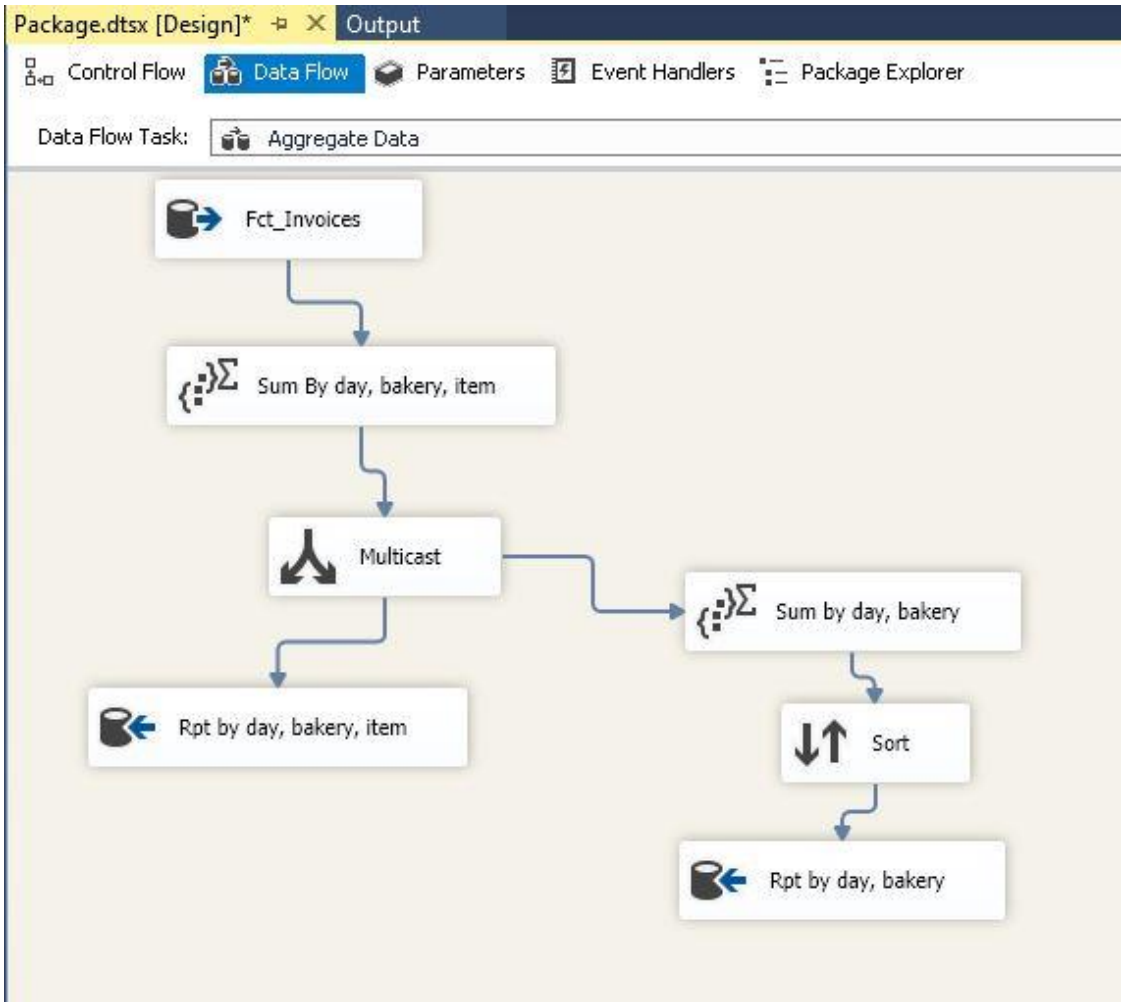
Slika 51 Aggregate - konfiguracija

Podatke dobivene iz **Aggregate** komponente usmjerit ćemo u prvu rpt_ tablicu, ali ćemo prije toga dodati i komponentu **Multicast**. Multicast omogućuje da jednostavno razgranamo dobivene podatke na dva smjera, u ovom slučaju podatke spremamo u finalnu tablicu ali iste šaljemo i na daljnje procesiranje (Slika 52).



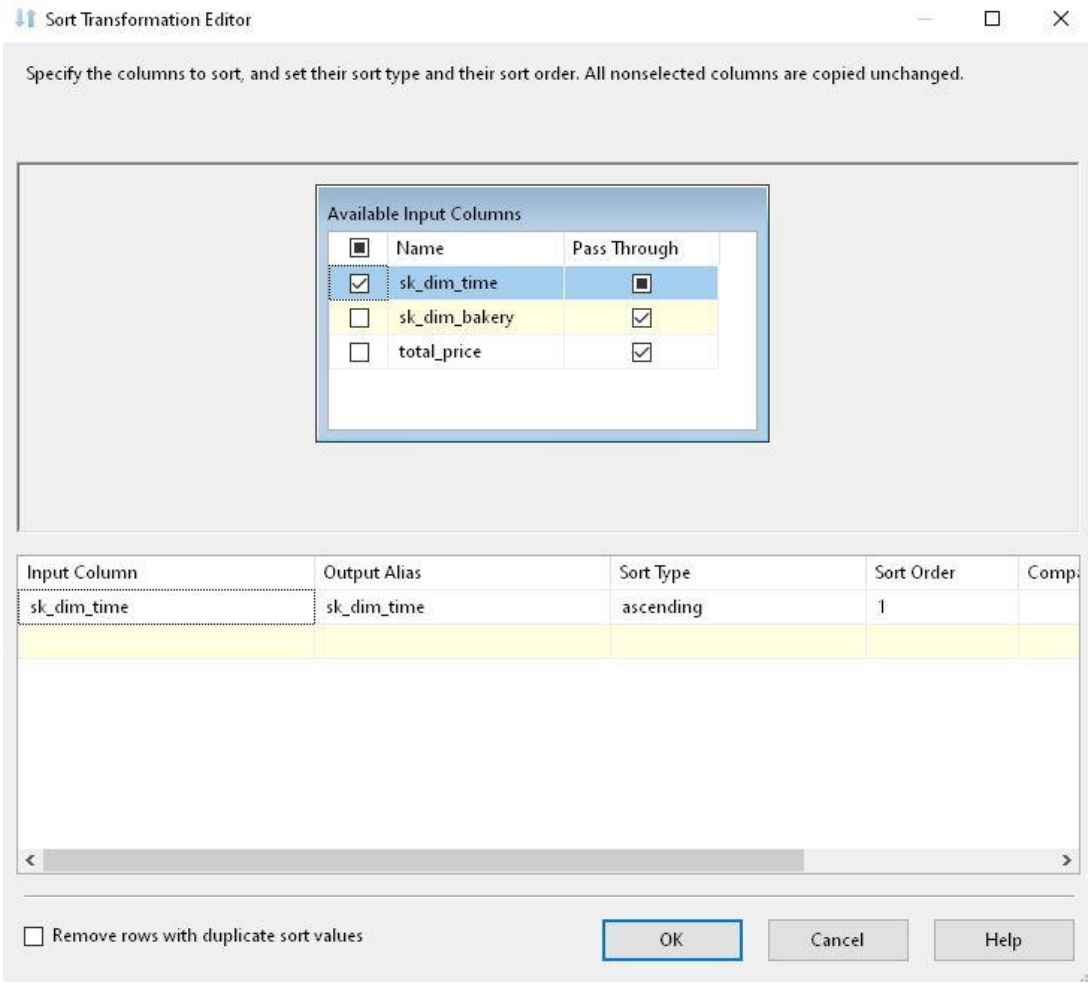
Slika 52 Data Flow – Multicast

Koristeći isti princip kao u prethodnim koracima, na **Multicast** nadovezat ćemo još jednu **Aggregate** komponentu koja će iz podataka isključiti ključ proizvoda i dobit ćemo samo sumu po datumu i pekari. Također, prije upisa podataka u drugu **rpt_** tablicu dodat ćemo i komponentu **Sort** (Slika 53).



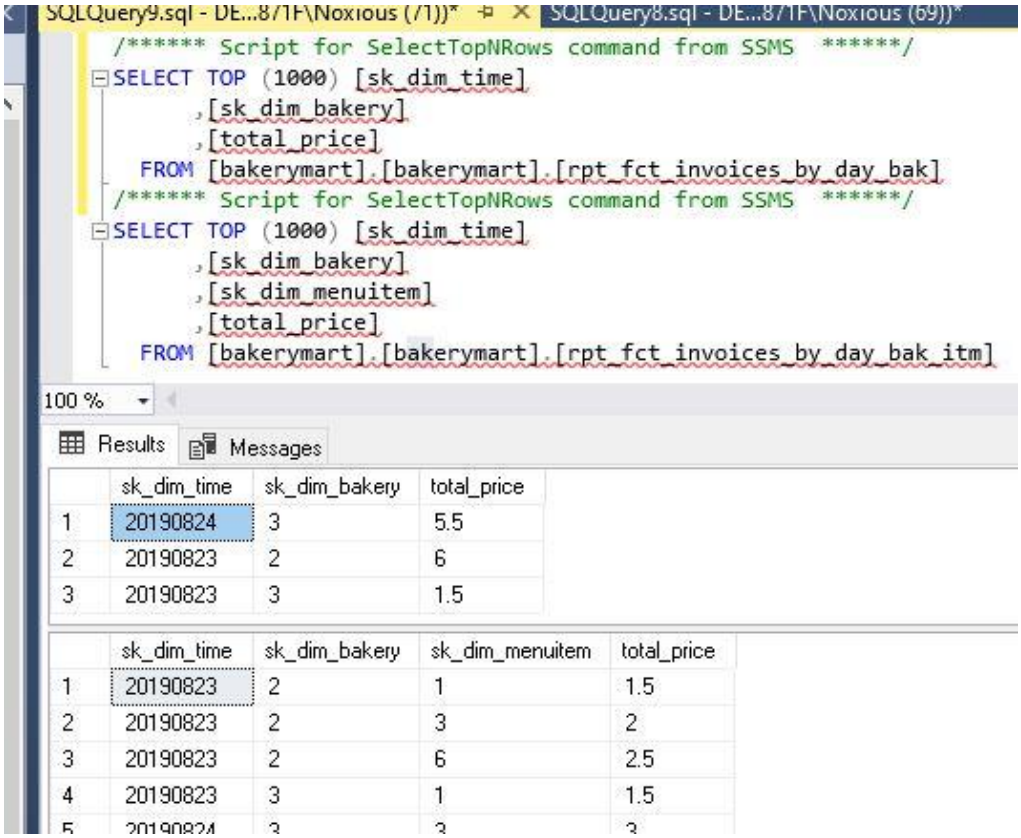
Slika 53 Data Flow - Sort

Funkcija **Sort** vrši sortiranje podataka po ključu, u konfiguraciji (Slika 54) biramo ključ po kojem želimo sortirati podatke (ne mora biti jedan) i tok (uzlazni ili silazni). Sortiranje može biti korisno za performans kod slanja upita na tablice, upiti rade brže kada se filtrira po sortirnom ključu.



Slika 54 Sort - konfiguracija

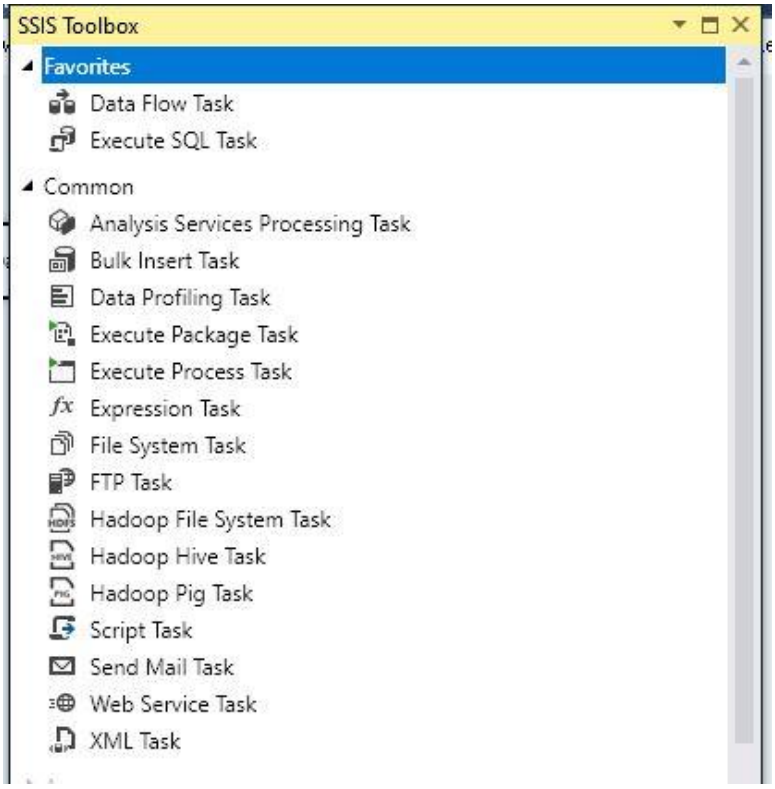
Nakon uspješnog izvođenja ovakvog paketa, tablice koje je zatražio tim za izvješća (reporting tim) bi trebale imati u sebi agregirane podatke (Slika 55), to će uvelike pomoći performansu aplikacije za vizualizaciju.



Slika 55 Agregirani podaci

3.9 Dodatne funkcionalnosti

S obzirom da primjeri nisu pokrili sve funkcionalnosti SSIS-a, valja spomenuti još nekoliko njih. Za početak iz SSIS Toolbox-a u Control Flow prozoru (Slika 56).

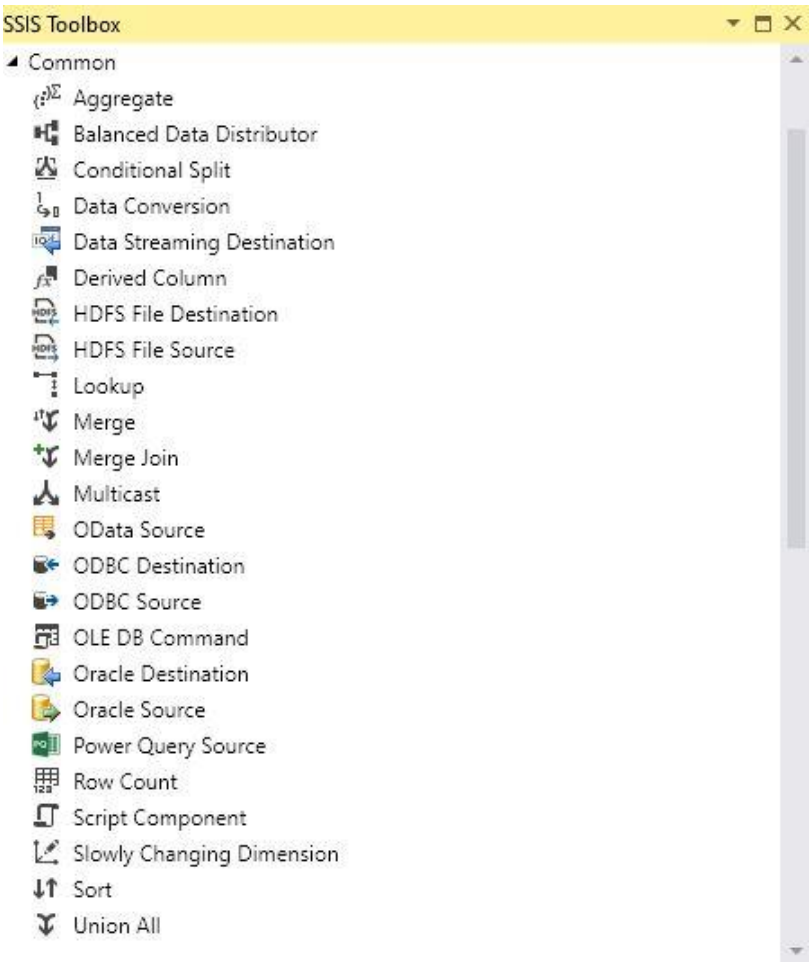


Slika 56 Toolbox - Control Flow

Execute Package Task može se koristiti kada želimo izvršiti unutar nekog SSIS paketa drugi SSIS paket. Može služiti za orkestraciju izvođenja više paketa, pa možemo izraditi jedan paket koji samo poziva druge pakete potrebnim redoslijedom i šalje obavijesti ovisno o uspjehu ili neuspjehu izvršavanja. **Execute Process Task** je dobar za izvršavanje nekih CMD naredbi, batch skripti ili pozivanje nekih aplikacija od drugog proizvođača.

FTP Task i File System Task su također bitne komponente kada radimo s čitanjem podataka iz datoteka ukoliko je potrebno raditi nekakve manipulacije s datotekama tokom procesa, na primjer nakon što smo obradili datoteku želimo ju premjestiti negdje u arhivu u trajnu pohranu, a ne ostaviti u direktoriju za procesiranje.

Drugi set alata nalazi se u Toolbox-u na Data Flow okviru (Slika 57).



Slika 57 Toolbox - Data Flow

Conditional Split je korisna komponenta kada želimo razgranati podatke u dva smjera po nekom uvjetu. Na primjer možemo odlučiti počistiti tablicu činjenica od zastarjelih podataka (eng. cold data) pa ćemo sve starije od godinu dana preusmjeriti na upis u datoteke i pohraniti negdje u trajnu pohranu koja (eng. cold storage).

Row Count je komponenta koju možemo koristiti za praćenje broja zapisa u toku SSIS izvršavanja, te možemo dodati neku logiku koja nam već tokom samog procesa radi neki oblik validacije podataka (sve se upisao u bazu prema očekivanju).

3.10 ETL i reporting

U prethodnom primjeru prikazali smo bitnu ulogu ETL-a za reporting (izvješća). ETL je naravno nužan za pročišćavanje podataka, unificiranje vrijednosti, minimizaciju loših podataka, dobre tablice činjenica i dimenzija su nužne da bi reporting bio kvalitetan.

Osim što je ETL bitan za upisivanje valjanih i dobrih podataka u bazu, on je potreban i za olakšavanje analitike i prikaza podataka. Osim tablica dimenzija i činjenica u skladištu podataka dobro je imati i agregirane verzije tablica, tablice koje sadrže razne kalkulacije i metrike potrebne za reporting alat. Iako postoje moćni alati koji sami računaju takve kalkulacije u memoriji nad podacima koje dobiju iz tablica u skladištu, bolje je rasteretiti alat zbog performansa i odraditi agregacije unutar skladišta.

S obzirom da reporting najčešće koriste krajnji poslovni korisnici koji ponekad nemaju informatička znanja, trend je napraviti takve aplikacije maksimalno jednostavnima i komfornima za korištenje. Prikaz podataka je često realiziran kroz neke web aplikacije jer je korisniku to najjednostavnije, u tom slučaju performans je jako bitan jer korisnik ne želi čekati dugo da dobije svoj izvještaj.

4. Zaključak

Poslovna inteligencija je moćna grana informatičke industrije koja raste skupa sa svim velikim firmama drugih industrija kojima je potreban informacijski sustav. DWH, analize podataka, vizualizacije i izvješća nužna su komponenta da bi veliko poslovanje moglo biti uspješno. Poslovni korisnici koji prosperiraju od BI rješenja su najčešće menadžeri bez dubokog informatičkog znanja, iz tog razloga bitno im je servirati podatke koje trebaju na jednostavan i brz način. Krajnji korisnik jednostavno nema vremena prilagođavati se loše razvijenim aplikacijama i skladištima.

Kako bi se moglo raditi sa skladištem podataka neke industrije potrebno je imati i barem neko osnovno razumijevanje prirode podataka i poslovnih procesa koji iza tih podataka stoje. Bitna je dobra komunikacija s klijentom i ponekad i dodatna edukacija o grani poslovanja kojom se bavi klijent da bismo za njihove podatke mogli raditi kvalitetan ETL i reporting.

Zadnjih godina poslovna inteligencija se razvija u smjeru umjetne inteligencije, više nisu bitna samo skladišta podataka i izviješća, nego se rade i prediktivne analize (koristi se strojno učenje). Također, kompanije postaju prisutne na socijalnim mrežama, koje generiraju mišljenja, recenzije. Tako nastaje veliki set podataka koji nisu strukturirani (eng. „big data“). Otkrivanje važnih metrika u takvom setu postaje dio poslovne inteligencije (eng. „data discovery“). Trend je i spremanje podataka u „oblak“ (eng. „in cloud“) što za sobom povlači pitanje sigurnosti.

Međutim, kroz rad je opisan BI u kontekstu DWH-a, ETL-a, reportinga. Postoje razni gotovi alati koji informatičarima olakšavaju rad sa skladištima podataka i jedan od njih je SSIS, alat za lakšu implementaciju jednostavnih i kompleksnih ETL rješenja koji dobro radi i stabilan je.

U razvoju ETL procesa trend je koristiti gotove alate jer su puno jednostavniji za uporabu i razvoj procesa je brži, pa time i jeftiniji. Sučelja u alatima su pretežno vizualna, što omogućuje bolji pregled kompleksnijih procesa. Također, lakše je prebaciti zadatke među developerima i timovima ako oboje poznaju isti alat, nego kada se radi o ručno pisanim skriptama.

S druge strane, ETL alati nisu savršeni i ne mogu uvijek u potpunosti ispuniti očekivanja za neki scenarij, iz tog razloga često je ipak potrebno uz gotove dijelove dodati i neke ručno rađene skripte kojima zaobilazimo blokade koje ETL alat ima.

Osobno ne volim baš gotove alate, pogotovo ako se radi o besplatnim alatima otvorenog koda ili alatima manjih kompanija jer njihove funkcionalnosti znaju imati problema i biti ograničavajuće. Također znaju imati probleme s bug-ovima i rušenjem. Mislim da SSIS ima lijep balans između gotovih funkcija za jednostavniju upotrebu ali i naprednih funkcija za ručnu implementaciju tamo gdje je nužno, to je i demonstrirano u primjerima. Također, nije bilo problema sa stabilnosti aplikacije ili značajnih grešaka. Iako upotrebljivost alata uvelike ovisi o scenariju na koji ga se primjenjuje, mislim da se iz primjera da zaključiti da je SSIS solidan alat za rad s najučestalijim praktičnim primjerima.

5. Literatura

1. Pratt, Mary: What is BI? Business intelligence strategies and solutions, s Interneta, https://www.cio.com/article/2439504/business-intelligence-definition-and-solutions.html, 23 kolovoza 2019
2. Rose, Margaret: Data warehouse, s Interneta, https://searchdatamanagement.techtarget.com/definition/data-warehouse, 23 kolovoza 2019
3. Beal, Vangie: ETL – Extract, Transform, Load, s Interneta, https://www.webopedia.com/TERM/E/ETL.html, 24 kolovoza 2019
4. Microsoft: Lesson 1: Create a project and basic package with SSIS, s Interneta, https://docs.microsoft.com/en-us/sql/integration-services/lesson-1-create-a-project-and-basic-package-with-ssis?view=sql-server-2017, 20 kolovoza 2019
5. Tutorial Gateway: SSIS Slowly Changing Dimension Type 2, s Interneta, https://www.tutorialgateway.org/ssis-slowly-changing-dimension-type-2/, 26 kolovoza 2019
6. Sheldon, Robert: Implementing Foreach Looping Logic in SSIS, s Interneta, https://www.red-gate.com/simple-talk/sql/ssis/implementing-foreach-looping-logic-in-ssis/, 25 kolovoza 2019
7. Roberson, Nathan: How Business Intelligence Helps You Understand Your Consumer, s Interneta, https://www.business2community.com/business-intelligence/business-intelligence-helps-understand-consumer-0770764, 23 kolovoza 2019
8. M., Oscar: What can Big Data do for BI?, s Interneta, https://www.clearpeaks.com/what-can-big-data-do-for-bi/ , 23 kolovoza 2019
9. KnowledgeHills: Dimensional Modeling tutorial – OLAP, data warehouse design, s Interneta, http://knowledgehills.com/dimensional-modeling/dimensional-modeling-tutorial.htm, 23 kolovoza 2019
10. Syncsort: ETL (Extract, Transform and Load), s Interneta, https://www.syncsort.com/en/glossary/etl, 23 kolovoza 2019

6. Popis Kratica

BI	eng. business intelligence
DDL	eng. Data Definition Language
DW	eng. data warehouse
DWH	eng. data warehouse
ETL	eng. Extract Transform Load
MSSQL	eng. Microsoft SQL Server
SQL	eng. Structured Query Language
SSDT	eng. SQL Server Data Tools
SSIS	eng. SQL Server Integration Services
SSMS	eng. SQL Server Managment Studio

7. Popis Slika

8. Popis priloga

1. CD sa praktičnim projektom